ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Μαθηματική ανάλυση αλγορίθμων κρυπτογράφησης

Του φοιτητή
Αναστασιάδη Απόστολου
Αρ. Μητρώου: 123921

Επιβλέπων
Τζέκης Παναγιώτης
Καθηγητής

Ημερομηνία 09/02/2024

Τίτλος Δ.Ε.: Μαθηματική ανάλυση αλγορίθμων κρυπτογράφησης

Κωδικός Δ.Ε.: 23250

Ονοματεπώνυμο φοιτητή: Αναστασιάδης Απόστολος

Ονοματεπώνυμο εισηγητή: Τζέκης Παναγιώτης

Ημερομηνία ανάληψης Δ.Ε.: 16-09-2023

Ημερομηνία περάτωσης Δ.Ε.: 09-02-2024

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Αναστασιάδη Απόστολου που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Πρόλογος

Η εξήγηση, στο μέτρο του δυνατού, εννοιών όπως αυτές των αλγορίθμων, της κρυπτογραφίας και της κρυπτανάλυσης κρίθηκε απαραίτητη όταν αρχίσαμε να ασχολούμαστε με αυτή την εργασία. Οι αλγόριθμοι αποτελούν τον θεμέλιο λίθο της μαθηματικής σκέψης καθώς και της κρυπτογραφίας η οποία παίζει πρωτεύοντα ρόλο στην σύγχρονη πραγματικότητα της επικοινωνίας μεταξύ μερών που επιθυμούν τη διασφάλιση του απόρρητου των συνδιαλλαγών τους οποιαδήποτε μορφή και αν έχουν αυτές.

Η τεχνολογική εξέλιξη και ένας κόσμος που αλλάζει ταχύτατα έφεραν την ψηφιακή επικοινωνία στο προσκήνιο και την κατέστησαν ζωτικό κομμάτι της καθημερινότητάς μας αφού και στις διαδικτυακές μας συναλλαγές, χωρίς εμείς να το καταλαβαίνουμε φυσικά, χρησιμοποιούνται οι κρυπτογραφικοί αλγόριθμοι.

Σημαντικό είναι να τονίσουμε ότι όσο εξελίσσεται η κρυπτογραφία άλλο τόσο εξελίσσεται η κρυπτανάλυση δηλαδή η δυνατότητα παραβίασης των αλγορίθμων κρυπτογραφίας από κάποιους που επιθυμούν να υποκλέψουν στοιχεία χρηστών.

Ζούμε σε μία συναρπαστική και εξόχως ανασφαλή εποχή σε όλα τα επίπεδα και η τεχνολογία της ψηφιακής επικοινωνίας με χρήση των αλγορίθμων δεν θα μπορούσε να μην είναι τμήμα όλης αυτής της συγκυρίας την οποία καλείται να αντιμετωπίσει καθημερινά και αποτελεσματικά.

Περίληψη

Στα πλαίσια αυτής της πτυχιακής εργασίας έπρεπε να αναλύσουμε την σημασία των όρων αλγόριθμοι συμμετρική και ασύμμετρη κρυπτογραφία. Με την συνδρομή μίας μικρής ιστορικής ανασκόπησης θελήσαμε να εξηγήσουμε τον δεσμό που ενώνει τους αλγόριθμους και τα μαθηματικά.

Επιπλέον διασαφηνίσαμε τον όρο πολυπλοκότητα χώρου και χρόνου και εξηγήσαμε την αναγκαιότητα του υπολογισμού της. Τέλος αναλύσαμε μαθηματικά τρεις αλγόριθμους ασύμμετρης κρυπτογραφίας σε μία προσπάθεια εστίασης της σπουδαιότητας της μαθηματικής επιστήμης στο επιστημονικό πεδίο της κρυπτογραφίας.

ΛΕΞΕΙΣ-ΚΛΕΙΔΙΑ

Αλγόριθμοι, Κρυπτογραφία, Περιπλοκότητα, Συμμετρική Κρυπτογραφία, Ασύμμετρη Κρυπτογραφία Μαθηματικά

Abstract

In the context of this thesis, we tried to analyze the meaning of the terms algorithm, symmetric and asymmetric cryptography. With the aid of a short historical review, we wanted to explain the bond that connects algorithms with mathematics. Furthermore, we clarify the meaning of time and space complexity and why we should calculate it. In addition, we analyzed mathematically three asymmetric cryptographic algorithms in an effort to focus on the importance of mathematics in the cryptography field.

KEYWORDS

Algorithms, Cryptography, Complexity, Symmetric Cryptography, Asymmetric Cryptography Mathemartics

Πίνακας Περιεχομένων

Ευρετήριο πινάκων και εικόνων

Ευρετήριο πινάκων και εικόνων

Κεφάλαιο 1

Πίνακες

Εικόνες

Κεφάλαιο 4
Εικόνες

Εισαγωγή πτυχιακής εργασίας

Στο πλαίσιο αυτής της πτυχιακής εργασίας προσπαθήσαμε να αναλύσουμε την έννοια του αλγόριθμου, της κρυπτογραφίας και να αναλύσουμε μαθηματικά τους αλγόριθμους ασύμμετρης κρυπτογράφησης Dsa, Rsa και Ecdsa. Αρχικά, στο πρώτο κεφάλαιο εξηγούμε τι σημαίνει αλγόριθμος και στην ιστορική αναδρομή προσπαθούμε να περιγράψουμε την εξελικτική του πορεία στο χρόνο. Στη συνέχεια αναφερόμαστε στους τρόπους περιγραφής των αλγορίθμων, στα χαρακτηριστικά τους, στις εφαρμογές που μπορεί να έχουν, στους τύπους αλγορίθμων, στον τρόπο επαλήθευσης της ορθότητάς τους και στην ανάλυσή τους.
Κομβικό σημείο είναι η ανάλυση χρονικής και χωρικής πολυπλοκότητας ενός αλγόριθμου.
Στο δεύτερο κεφάλαιο διασαφηνίζουμε τους όρους κρυπτολογία, κρυπτογραφία και κρυπτανάλυση και φυσικά κάνουμε μία ιστορική αναδρομή για να κατανοήσουμε την διαδρομή της κρυπτογραφίας μέχρι τις μέρες μας διανθισμένη με κάποια παραδείγματα. Παραθέτουμε τις αρχές της μοντέρνας κρυπτογραφίας, εξηγούμε την σημασία των ορισμών, αναφερόμαστε στους στόχους της και εξηγούμε τις εφαρμογές της. Αμέσως μετά αναφερόμαστε στα είδη των κρυπτογραφικών αλγορίθμων και κάνουμε μία πρώτη προσπάθεια εξήγησης της κρυπτογραφίας δημόσιου και ιδιωτικού κλειδιού.
Στο τρίτο κεφάλαιο γίνεται μία αναφορά στον όρο συμμετρική κρυπτογραφία ή κρυπτογραφία ιδιωτικού κλειδιού καθώς και στις χρήσεις της. Επισημαίνουμε τις κατηγορίες των συμμετρικών αλγορίθμων κρυπτογράφησης οι οποίες είναι οι αλγόριθμοι ροής και οι αλγόριθμοι τμήματος.
Τέλος, αναλύουμε κάποιους αλγόριθμους συμμετρικής κρυπτογραφίας.
Στο τέταρτο κεφάλαιο εστιάζουμε στην εξήγηση και στη σημασία της ασύμμετρης κρυπτογραφίας ή αλλιώς της κρυπτογραφίας δημόσιου κλειδιού παραθέτουμε τα χαρακτηριστικά της, τονίζουμε την σημασία της για το Διαδίκτυο. Δεν παραλείπουμε να αναφερθούμε στις χρήσεις της στα πλεονεκτήματα και τα μειονεκτήματά της. Στη συνέχεια αναλύουμε μαθηματικά τους αλγόριθμους Rsa,Dsa και Ecdsa.
Το πέμπτο κεφάλαιο περιέχει τις κρίσεις και τα συμπεράσματά μας.

Κεφάλαιο 1

Εισαγωγή

Στο πρώτο κεφάλαιο αρχικά ορίζουμε την έννοια της λέξης “αλγόριθμος” πρώτα γενικά και στη συνέχεια ειδικά εξηγώντας τα συστατικά στοιχεία του και τονίζοντας τη χρησιμότητά του μέσω ενός παραδείγματος.
Αμέσως μετά κάνουμε μία ιστορική αναδρομή στα πλαίσια της οποίας αναφερόμαστε αρχικά στην εννοιολογική ρίζα της λέξης αλγόριθμος αφού στην αρχή αλγόριθμος σήμαινε πολύ απλά ένα σύνολο οδηγιών για την ολοκλήρωση μίας εργασίας.
Ειδικότερα αναφερόμαστε στον Βαβυλωνιακό πολιτισμό, στον αιγυπτιακό πολιτισμό, στον κινέζικο πολιτισμό, στον πολιτισμό των Μάγια, στον πολιτισμό των Ελλήνων, τον ινδικό πολιτισμό, στους αλγόριθμους στην Ιταλία. Επισημαίνουμε ότι ο όρος αλγόριθμος υιοθετήθηκε από Ευρωπαίους λόγιους. Στη συνέχεια ασχολούμαστε με την αυτοματοποίηση των αλγορίθμων, δηλαδή στο πέρασμα από τους υπολογισμούς στο χέρι στην μηχανική υλοποίησή τους. Θα ήταν μεγάλη παράλειψη να μην συμπεριλάβουμε σε αυτό το κεφάλαιο τους τρόπους περιγραφής των αλγορίθμων δηλαδή με την χρήση ψευδοκώδικα ή την χρήση φυσικής γλώσσας. Επιπλέον παραθέτουμε τα χαρακτηριστικά των αλγορίθμων, τους τομείς εφαρμογής τους και τους τύπους τους. Είναι απαραίτητη η αναφορά στους τρόπους ελέγχου της ορθότητας των αλγορίθμων αλλά και η εξήγηση του τρόπου ανάλυσής τους.
Ακόμα παρουσιάζουμε τις δύο προσεγγίσεις με τις οποίες μπορούμε να υπολογίσουμε τον χρόνο τρεξίματος ενός αλγόριθμου. Τέλος αναλύουμε τον όρο “ασυμπωτικό ρυθμό αύξησης”.

1. Αλγόριθμοι

1.1 Ορισμός

Ένας αλγόριθμος, άτυπα, μπορεί να χαρακτηριστεί ως μία καλώς ορισμένη υπολογιστική διαδικασία η οποία πρέπει να επιλύσει ένα συγκεκριμένο πρόβλημα εντός πεπερασμένου χρόνου.[1]

Ορίζουμε έναν αλγόριθμο ως μία ακολουθία πεπερασμένων κανόνων ή οδηγιών που θα πρέπει να ακολουθηθούν κατά γράμμα σε περίπτωση υπολογισμών ή κατά τη διάρκεια άλλων λειτουργιών που έχουν ως σκοπό την επίλυση προβλημάτων.[2]

Ειδικότερα, μπορούμε επίσης να πούμε ότι ένας αλγόριθμος είναι μία διαδικασία επίλυσης μαθηματικών προβλημάτων με ένα πεπερασμένο σύνολο βημάτων στα οποία συχνά συμπεριλαμβάνονται αναδρομικές συναρτήσεις.

Η ακολουθία υπολογιστικών βημάτων η οποία συνιστά έναν αλγόριθμο απεικονίζει την είσοδο (input) του προβλήματός μας δηλαδή τα δεδομένα τα οποία είναι απαραίτητα για να δουλέψει στην έξοδο (output) η οποία αποτελεί την λύση του προβλήματος.[2]

Κάθε είσοδος η οποία συνάδει με τις προδιαγραφές καλείται νόμιμη και λέμε ότι ορίζει ένα συγκεκριμένο στιγμιότυπο (instance) του προβλήματος.

Ένας αλγόριθμος επιλύει ένα πρόβλημα όταν για κάθε στιγμιότυπο του προβλήματος τερματίζει μετά από πεπερασμένο χρόνο με σωστή έξοδο.

Δεν πρέπει σε καμία περίπτωση να παραλείψουμε ότι ένας αλγόριθμος δέχεται ως είσοδο είτε μία τιμή είτε ένα σύνολο τιμών και στην έξοδό του παράγεται είτε μία τιμή είτε ένα σύνολο τιμών.

Η διατύπωση του προβλήματος με γενικούς όρους περιγράφει την επιθυμητή είσοδο καθώς και την επιθυμητή έξοδό του.

Ας δούμε ένα παράδειγμα προς κατανόηση των παραπάνω. Υποθέτουμε ότι επιθυμούμε να ταξινομήσουμε μία ακολουθία αριθμών κατά αύξουσα σειρά. Αυτού του είδους τα προβλήματα προκύπτουν αρκετά συχνά στην καθημερινή πρακτική και παρέχει γόνιμο έδαφος εισαγωγής πολλών τεχνικών ταξινόμησης καθώς και εργαλείων ανάλυσης.

Το πρόβλημα της ταξινόμησης ορίζεται ως εξής :

- Είσοδος :μία ακολουθία αριθμών πλήθους n $<a_1, a_2 \ldots\ldots a_n>$
- Έξοδος :μία ταξινομημένη ακολουθία της ακολουθίας εισόδου

Δηλαδή, αν υποθέσουμε ότι η ακολουθία εισόδου ήταν η <13,33,5555,5>, η εφαρμογή ενός ορθού αλγόριθμου ταξινόμησης θα είχε ως έξοδο <5,13,33,5555>.

Μία ακολουθία εισόδου τέτοιας μορφής ονομάζεται στιγμιότυπο του προβλήματος ταξινόμησης (instance of the sorting problem).

Η ταξινόμηση χρησιμοποιείται από πολλά προγράμματα ως ενδιάμεσο στάδιο και γι' αυτό το λόγο θεωρείται θεμελιώδης λειτουργία στην επιστήμη των υπολογιστών και ως συνέπεια υπάρχουν πολλά είδη αλγόριθμων ταξινόμησης. [1][3][4]

1.2 Ιστορική αναδρομή

Οι αλγόριθμοι υπήρχαν ανέκαθεν πριν καν να επινοηθεί η ίδια η λέξη που τους περιγράφει.

Οι αλγόριθμοι γενικά είναι οδηγίες οι οποίες πρέπει να ακολουθούνται προκειμένου να ολοκληρωθεί μία συγκεκριμένη εργασία. Οι Βαβυλώνιοι τους χρησιμοποιούσαν στις αποφάσεις τους για διάφορα νομικά θέματα, οι καθηγητές των Λατινικών για να διδάσκουν γραμματική με τον σωστό τρόπο και έχουν χρησιμοποιηθεί από όλους τους πολιτισμούς για πρόβλεψη του μέλλοντος, για να επιλεγεί η καλύτερη ιατρική θεραπεία ή ακόμα για την προετοιμασία φαγητού.

Η ιστορία των αριθμών και των αλγορίθμων είναι στενά συνδεδεμένη με μία ανεπαίσθητη σχέση και η πρακτική της χρήσης των αλγορίθμων ίσως να είναι πιο παλιά ακόμα και από τα ίδια τα μαθηματικά. [5]

1.2.1 Ο Βαβυλωνιακός πολιτισμός

Οι Βαβυλώνιοι ζούσαν στη Μεσοποταμία στην οποία τώρα βρίσκεται η Αίγυπτος και η Συρία. Ήταν μία εύφορη πεδιάδα μεταξύ δύο ποταμών του Τίγρη και του Ευφράτη.

Ο πολιτισμός τους διήρκεσε από το 2000 π.Χ. μέχρι το 600 π.Χ. Αργότερα κατακτήθηκαν από τους Σουμέριους και τους Ακκάδιους.

Ο Βαβυλωνιακός πολιτισμός παρουσίασε σπουδαία επιστημονικά επιτεύγματα ιδιαίτερα στον τομέα των Μαθηματικών. Είναι διάσημοι για τον τρόπο γραφής τους ο οποίος ονομάζεται σφηνοειδής διότι τα γράμματα και τα ψηφία αυτού του τρόπου γραφής μοιάζουν με καρφιά δηλαδή έχουν σφηνοειδές σχήμα.

Οι Βαβυλώνιοι έπαιρναν μια υγρή επιφάνεια φτιαγμένη από πηλό και με την χρήση ενός κοφτερού αντικειμένου χάραζαν αριθμούς και γράμματα και στην συνέχεια την άφηναν να στεγνώσει και έτσι αποθήκευαν πληροφορίες.

Σε αυτές τις πλάκες από πηλό έγραφαν κείμενα και αλλά μαθηματικούς υπολογισμούς. Οι μαθηματικές γνώσεις των Βαβυλώνιων ήταν εξαιρετικές: μπορούσαν για παράδειγμα να υπολογίσουν την τετραγωνική ρίζα ενός αριθμού, ήξεραν τον αριθμό pi (3,14) και υπάρχει μεγάλη πιθανότητα να είχαν γνώση και της εκθετικής συνάρτησης e επίσης είχαν γνώση της Πυθαγόρειου τριάδας 1200 χρόνια πριν να διατυπωθεί από τον Πυθαγόρα. Επιπροσθέτως μπορούσαν να λύσουν εξισώσεις δευτέρου βαθμού (quadratics) ακόμα και πολυώνυμα όγδοου βαθμού, έλυναν γραμμικές εξισώσεις και πολλά άλλα.

Κατά τη διάρκεια της αρχαιότητας η Πυθαγόρειος Τριάδα ήταν αντικείμενο ενασχόλησης πολλών πολιτισμών.

Η Πυθαγόρειος τριάδα είναι απλά τρεις φυσικοί αριθμοί a,b,c οι οποίοι σχετίζονται με τον παρακάτω τύπο:

$$a^n + b^n = c^n \tag{1.1}$$

Το n είναι φυσικός αριθμός και είναι μεγαλύτερος από την μονάδα. Σε περίπτωση που το n ισούται με δύο, τότε μιλάμε για το Πυθαγόρειο Θεώρημα. Οι Βαβυλώνιοι έγιναν διάσημοι από τον Πυθαγόρα.

Είχαν ένα εξηνταδικό αριθμητικό σύστημα στοιχεία του οποίου υπάρχουν στην καθημερινή μας ζωή για να μας το θυμίζουν.

Για παράδειγμα, ένα λεπτό έχει εξήντα δευτερόλεπτα, μία ώρα εξήντα λεπτά, υπάρχουν τριακόσιες εξήντα μοίρες σε ένα κύκλο, αριθμός ο οποίος είναι πολλαπλάσιος του εξήντα. Σε αυτό το αριθμητικό σύστημα δεν υπάρχει το ψηφίο μηδέν και πολλοί άλλοι αρχαίοι πολιτισμοί όπως ο ελληνικός πολιτισμός, ο πολιτισμός των Αράβων των Ινδών και των Ρωμαίων δεν το χρησιμοποιούσαν.

Το ψηφίο μηδέν δεν έχει λογική γιατί κάποιος επιθυμεί να εκφράσει ότι έχει έναν αριθμό αντικειμένων στην κατοχή του και δεν υπάρχει νόημα στο να μιλάει κάποιος για αντικείμενα που δεν έχει.

Στην πραγματικότητα οι Ινδοί εφηύραν το εν λόγω ψηφίο περίπου στα 800 μ.Χ. και ο Fibonacci (Leonardo Bonacci) με στο βιβλίο του “Liber Abaci” το ανέφερε για πρώτη φορά.

Πέρασε μεγάλο χρονικό διάστημα μέχρι να χρησιμοποιηθεί το μηδέν από τον υπόλοιπο κόσμο και πρέπει να σημειώσουμε ότι η πρώτη φορά που αυτό το ψηφίο εμφανίστηκε σε συμβόλαια ήταν γύρω στο 1700.[6]

1.2. 2 Ο Αιγυπτιακός πολιτισμός

Ενώ οι Βαβυλώνιοι εστίαζαν στην αριθμητική φύση των μαθηματικών πράγμα που σημαίνει ότι χρησιμοποιούσαν κατά κύριο λόγο αριθμούς, οι Αιγύπτιοι είχαν μία γεωμετρική θεώρηση των μαθηματικών και αυτό ίσχυε διότι ζούσαν στις όχθες του Νείλου και υπήρχε μία περίοδος ξηρασίας και μία περίοδος υγρασίας κατά τη διάρκεια της οποίας πλημμύριζε όλα τα χωράφια και αφού περνούσε η πλημμύρα οι άνθρωποι ήταν αναγκασμένοι να μετρήσουν ξανά την έκταση των χωραφιών τους για να καθορίσουν εκ νέου των ιδιοκτήτη του κάθε χωραφιού.

Για μία τέτοια εργασία ήταν απαραίτητες γεωμετρικές γνώσεις όπως για παράδειγμα ο υπολογισμός της περιμέτρου ενός συγκεκριμένου κομματιού γης και ο υπολογισμός του εμβαδού του. Επίσης οι Αιγύπτιοι ήθελαν να μετράνε κύκλους, να υπολογίζουν αναλογίες για την δημιουργία πυραμίδων κλπ. Επίσης θα πρέπει να τονιστεί ότι στο αριθμητικό τους σύστημα δεν υπήρχε το ψηφίο μηδέν. Είχαν ένα δεκαδικό αριθμητικό σύστημα το οποίο ήταν βασισμένο σε ψηφία τα οποία απεικονίζονταν με ράβδους. Για αριθμούς μεγαλύτερους από το 1000 χρησιμοποιούσαν άλλα σύμβολα. [6]

1.2. 3 Ο Κινέζικος πολιτισμός

Ο κινέζικος πολιτισμός ήταν ένας απομονωμένος πολιτισμός και παρέμεινε έτσι μέχρι και το 1800 περίπου.

Αυτό σημαίνει ότι η επιστημονική ανάπτυξη της Κίνας ήταν παράλληλη με αυτή της Ευρώπης.

Πολλές επιστημονικές έννοιες συναντήθηκαν στην κινέζικη επιστήμη και στην ευρωπαϊκή με διαφορετικά ονόματα.

Οι Κινέζοι γνώριζαν τους πυθαγόρειους αριθμούς, όπως επίσης και το Πυθαγόρειο Θεώρημα το οποίο ονόμασαν Θεώρημα Gougu (Gougu Theorem).

Για πολύ καιρό τα μαθηματικά στη Κίνα ήταν άγνωστα στους υπόλοιπους πολιτισμούς. Η μαθηματική επιστήμη στην Κίνα είχε μία πρακτική προσέγγιση και όχι τόσο αξιωματική και σε αυτό το σημείο διέφερε από την ευρωπαϊκή προσέγγιση.

Το πιο διάσημο βιβλίο της αρχαίας ιστορίας των κινέζικων μαθηματικών τιτλοφορείται “Εννέα κεφάλαια της μαθηματικής τέχνης” (“Nine chapters of the mathematical art”, “Jiuzhang suanshu”) το οποίο χρονολογείται περίπου στα 100 μ.Χ. και περιέχει περίπου διακόσια σαράντα έξι πρακτικά προβλήματα από ποικίλα μαθηματικά πεδία:

- Το πρώτο κεφάλαιο είχε να κάνει με την επίβλεψη της γης, με χωρικά προβλήματα, με τις τέσσερις κύριες μαθηματικές πράξεις κλασμάτων, με τις προσεγγίσεις του αριθμού PI.
- Το δεύτερο κεφάλαιο σχετίζεται με την ανταλλαγή αγαθών και με αναλογίες
- Το τρίτο κεφάλαιο αναφέρεται σε αναλογίες καθώς και σε ευθείες, αντίστροφες, σύνθετες, αριθμητικές και γεωμετρικές ακολουθίες.
- Στο τέταρτο κεφάλαιο έχουμε αναφορά σε περιοχές και κλάσματα με την μονάδα ως αριθμητή
- Στο πέμπτο κεφάλαιο μπορούμε να διαβάσουμε για την κατασκευή καναλιών, χαντακιών και αναχωμάτων.
- Το έκτο κεφάλαιο αναφέρεται σε αναλογίες καθώς και στη δίκαιη κατανομή των αγαθών.
- Το έβδομο κεφάλαιο σχετίζεται με γραμμικές εξισώσεις.
- Στο όγδοο κεφάλαιο έχουμε την λύση συστημάτων γραμμικών εξισώσεων.
- Το ένατο κεφάλαιο αναφέρεται σε τρίγωνα, στο Πυθαγόρειο Θεώρημα και στις εξισώσεις δευτέρου βαθμού.[6]

1.2.4 Ο πολιτισμός των Μάγια

Ο πολιτισμός των Μάγια άνθισε περίπου από το 250 μ.Χ. έως το 900 μ.Χ. στη χερσόνησο Γιουκατάν (Yucatan Peninsular) στο Μεξικό.

Από αυτούς έχουμε τρεις κώδικες οι οποίοι περιγράφουν τον πολιτισμό τους και είναι χαραγμένοι σε φλοιό δέντρου. Ο πιο καλοδιατηρημένος είναι ο κώδικας Dresden.

Το αριθμητικό τους σύστημα ήταν εικοσαδικό και υπήρχε σύμβολο για το ψηφίο μηδέν αλλά δεν το χρησιμοποιούσαν πολύ.[6]

1.2.5 Ο πολιτισμός των Ελλήνων

Από το 600 π.Χ. έως το 500 μ.Χ. οι Έλληνες έκαναν βήματα προόδου στον τομέα των μαθηματικών. Η αλγοριθμική σκέψη αποτυπώθηκε από τον αλγόριθμο του Ευκλείδη για τον Μέγιστο Κοινό Διαιρέτη και τον αλγόριθμο εύρεσης πρώτων αριθμών του Ερατοσθένη. [6]

Θαλής ο Μιλήσιος

Έζησε από το 634 π.Χ. έως το 546 π.Χ. στο νησί της Μήλου. Διατύπωσε το θεώρημα των αναλογιών σε όμοια τρίγωνα. Επίσης συμπέρανε ότι οι γωνίες βάσης ενός ισοσκελούς τριγώνου είναι ίσες. Πίστευε ότι η γη είναι ένας επίπεδος δίσκος ο οποίος αιωρείται σε ένα ατελείωτο ωκεανό [6]

Πυθαγόρας ο Σάμιος

Έζησε από το 560 π.Χ. έως το 480 π.Χ. στη Σάμο και είναι γνωστός για το Πυθαγόρειο Θεώρημα.[6]

Ευκλείδης, πατέρας της αξιωματικής γεωμετρίας

Ο Ευκλείδης έζησε από το 325 π.Χ. έως το 270 π.Χ. στην Αλεξάνδρεια της Αιγύπτου και θεωρείται ο θεμελιωτής των μαθηματικών και ειδικότερα της γεωμετρίας. Συγκεκριμένα το βιβλίο του που τιτλοφορείται “Στοιχεία” το οποίο ουσιαστικά αποτελεί μία μαθηματική πραγματεία δεκατριών βιβλίων έθεσε τα θεμέλια της αξιωματικής γεωμετρίας. Επίσης είναι εμπνευστής του αλγόριθμου εύρεσης του μέγιστου κοινού διαιρέτη.[6]

Ερατοσθένης

Ο Ερατοσθένης έζησε από το 284 π.Χ. έως το 192 π.Χ. και είναι εμπνευστής του αλγόριθμου ο οποίος **ονομάζεται** “Το κόσκινο του Ερατοσθένη” ο οποίος είναι σχετικά απλός και έχει σχέση με την εύρεση των πρώτων αριθμών μέχρι έναν ορισμένο αριθμό Ν ο οποίος φυσικά αποτελεί το υπολογιστικό όριο. [6]

Ηρών ο Αλεξανδρεύς

Ο Ήρων έζησε στην Αλεξάνδρεια της Αιγύπτου από το 10 μ.Χ. έως το 70 μ.Χ. και είναι διάσημος για τον τύπο υπολογισμού του εμβαδού του τριγώνου. Επίσης δημιούργησε έναν αλγόριθμο με την βοήθεια του οποίου μπορούσε κάποιος να υπολογίσει την τετραγωνική ρίζα ενός αριθμού. [6]

$$E=\sqrt{\tau(\tau-\alpha)(\tau-\beta)(\tau-\gamma)} \tag{1.2}$$

Κλαύδιος Πτολεμαίος

Ο Κλαύδιο έζησε από το 85 μ.Χ. έως το 165 μ.Χ. στην Αλεξάνδρεια της Αιγύπτου. Υπολόγισε κατά προσέγγιση τον αριθμό PI. [6]

Διόφαντος ο Αλεξανδρινός

Ο Διόφαντος έζησε στην Αλεξάνδρεια κατά προσέγγιση από το 201/215 μ.Χ. έως το 285/299 μ.Χ.
Είναι γνωστός ως ο πατέρας της άλγεβρας διότι έγραψε μία συλλογή δεκατριών βιβλίων των οποίων ο τίτλος ήταν "Arithmetica" στα οποία κατεγράφησαν εκατό τριάντα αλγεβρικά προβλήματα.
Τα περισσότερα στοιχεία του βιβλίου ήταν καρποί έρευνας στη μαθηματική βιβλιογραφία που υπήρχε μέχρι φυσικά την εποχή του. Αυτή η συλλογή βιβλίων έγινε διάσημη διότι μπορούσε να χρησιμοποιηθεί ως πηγή μαθηματικής γνώσης.

1.2.6 Ο Ινδικός πολιτισμός

Τα μαθηματικά στην Ινδία επηρεάστηκαν από τους Έλληνες, τους Αιγύπτιους και τους Βαβυλώνιους.
Η ευρωπαϊκή γνώση μεταφέρθηκε στην Ινδία από τον Μέγα Αλέξανδρο όταν την κατέκτησε.
Περίπου τον πέμπτο αιώνα μ.Χ. οι Ινδοί άρχισαν να χρησιμοποιούν το Βαβυλωνιακό σύστημα αρίθμησης το οποίο ήταν με βάση το δέκα. Επίσης εφηύραν το ψηφίο μηδέν. [6]

1.2.7 Οι αλγόριθμοι στην Ιταλία

Στα μεσαιωνικά λατινικά ο όρος algorismus αναφερόταν στους υπολογισμούς με το χέρι με την χρήση των Ινδοαραβικών ψηφίων αρίθμησης θέσης (Indo-Arabic positional numbers) οι οποίοι προήλθαν από την Μέση Ανατολή και σταδιακά αντικαθιστούσαν το ρωμαϊκό τρόπο γραφής αριθμών.
Μιλάμε φυσικά για το σύστημα που χρησιμοποιούμε σήμερα, δηλαδή ως βάση έχουμε τους αριθμούς 1, 2,3,4,5,6,7,8,9,0 με τον συνδυασμό των οποίων μπορούμε να δηλώσουμε ένα αριθμητικό ποσό.
Ο Ιταλός έμπορος και μαθηματικός Λεονάρντο Φιμπονάτσι (Leonardo Fibonacci) εισήγαγε το Ινδοαραβικό σύστημα με το έργο του "Το βιβλίο των υπολογισμών" (Liber Abaci) το 1202 αλλά δεν χρησιμοποίησε τον όρο "αλγόριθμος" (algorismus) αλλά τη φράση "τρόπος των Ινδών" (modus indorum). Ο όρος αλγόριθμος (algorismus) εμφανίστηκε στο ποίημα Carmen de Algorismo του Alexandre de Villedieu στα 1240 το οποίο ήταν στην ουσία ένα εγχειρίδιο απομνημόνευσης των νέων μεθόδων υπολογισμού.
Ο Λεονάρντο Φιμπονάτσι στο βιβλίο που μόλις αναφέραμε ανέπτυξε τον Αλγόριθμο του Πλέγματος (Lattice Algorithm) ο οποίος αποτελεί μία μέθοδο πολλαπλασιασμού μεγάλων αριθμών.
Σε αυτό τον αλγόριθμο θεωρείται ότι οι αριθμοί οι οποίοι αποτελούν τα δεδομένα εισόδου σαφείς σειρές ψηφίων. Ας υποθέσουμε ότι η είσοδός μας αποτελείται από ένα ζευγάρι πινάκων οι οποίοι είναι οι ακόλουθοι:

- X [0 .. m − 1]
- Y [0 .. n − 1]

Οι παραπάνω πίνακες αντιπροσωπεύουν τους εξής αριθμούς εισόδου :

$$x=\sum_{i=0}^{m-1} X[i]*10 \tag{1.3}$$

$$y=\sum_{j=0}^{n-1} Y[j]*10^{j} \tag{1.4}$$

Συνεπώς η έξοδος αποτελείται από έναν πίνακα ο οποίος αντιπροσωπεύει το γινόμενο και αυτός είναι ο Z [0..m + n − 1] δηλαδή ο εξής :

$$z=x*y=\sum_{k=0}^{m*n-1} Z[k]*10^{k} \tag{1.5}$$

Στον αλγόριθμο εφαρμόζονται οι πράξεις της πρόσθεσης και του πολλαπλασιασμού ανά ψηφίο.
Η πρόσθεση μπορεί να υλοποιηθεί με την χρήση ενός βρόχου for. Ο ολοκληρωμένος αλγόριθμος πλέγματος εκφράζεται από τον ακόλουθο τύπο:

$$x*y=\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(X[i]*Y[j]*10^{i*j}) \tag{1.6}$$

Ένα βιβλίο το οποίο τυπώθηκε στη Βενετία το 1501 το οποίο αποδίδεται στον μοναχό Johannes de Sacrobosco ο οποίος έζησε τον δέκατο τρίτο αιώνα τιτλοφορείται “Algorismus Domini” και ο συγγραφέας του χρησιμοποιεί διαγράμματα για να εξηγήσει την νέα μέθοδο υπολογισμού με τους αριθμούς θέσεως.
Πρόσφατα επιβεβαιώθηκε ότι ο όρος algorismus είναι μία λατινοποίηση του ονόματος του πέρση (Ιρανού) λόγιου Muhammad ibn Musa al-Khwarizmi έναν βιβλιοθηκάριο από τον Οίκο της Σοφίας (House of Wisdom) στη Βαγδάτη (Baghdad). Ο ίδιος ήταν συγγραφέας ενός βιβλίου για υπολογισμούς με

Ινδοαραβικούς αριθμούς το οποίο γράφτηκε το 825 μ.Χ. Το αρχικό χειρόγραφο του συγγραφέα το οποίο ήταν γραμμένο σε αραβική γλώσσα χάθηκε αλλά από τον δωδέκατο αιώνα υπήρχαν τουλάχιστον τέσσερις μεταφράσεις στα λατινικά με διαφορετικούς τίτλους. Το πέρασμα από τον ρωμαϊκό τρόπο γραφής αριθμών στον Ινδοαραβικό τρόπο γραφής δεν ήταν μόνο ζήτημα διευκόλυνσης της μεθόδου γραφής αλλά είχε επίσης οικονομικό και κοινωνικό αντίκτυπο λόγω της ραγδαίας ανάπτυξης των εμπορικών συναλλαγών στην Ευρώπη και στην Μεσόγειο. Ο Ινδοαραβικός τρόπος γραφής αριθμών έδωσε στους ανθρώπους τη δυνατότητα να γράφουν αριθμούς με ακρίβεια και έτσι οι υπολογισμοί γινόταν με ταχύτητα. Στην Ιταλία, οι έμποροι από την Φλωρεντία και την Βενετία ήταν οι πρώτοι που ασπάστηκαν αυτό τον τρόπο γραφής.[6]

1.2.8 Η υιοθέτηση του όρου αλγόριθμος

Αργότερα ο όρος "αλγόριθμος" (algorithm) υιοθετήθηκε από ευρωπαίους λόγιους όπως για παράδειγμα ο Gottfried Wilhelm Leibniz ο οποίος και τον χρησιμοποίησε για να ορίσει την μέθοδό του διαφορικού λογισμού (differential calculus).

Στην Encyclopédie του D'Alembert ο όρος "αλγόριθμος" ορίζεται ως εξής :

"..αποτελεί έναν αραβικό όρο ο οποίος χρησιμοποιείται από πολλούς συγγραφείς ιδιαίτερα ισπανικής καταγωγής και έχει την έννοια της αλγεβρικής πρακτικής. "

Όμως, σε κάποιες περιπτώσεις, σημαίνει και αριθμητική με ψηφία. Η γενική χρήση του όρου είναι να δηλώσει την σημειογραφία (τον τρόπο γραφής) και τη μέθοδο (τα βήματα) κάθε τύπου υπολογισμών.
Με αυτή την έννοια χρησιμοποιούμε τους όρους ο αλγόριθμος του ολοκληρωτικού λογισμού, ο αλγόριθμος του εκθετικού λογισμού, ο αλγόριθμος των ημιτόνων κλπ.
Φυσικά, οι αριθμητικές μέθοδοι υπολογισμού με το χέρι που διδάσκονται στα σύγχρονα σχολεία δεν είναι παρά Ινδοαραβικοί αλγόριθμοι (alogirsms) για την διαχείριση των αριθμητικών συμβόλων.
Αυτές οι μέθοδοι έχουν αναδρομική δομή και έτσι είναι δυνατή η διαχείριση τιμών οι οποίες είτε τείνουν προς το άπειρο είτε ορίζονται κατά προσέγγιση.

Ας δούμε ένα παράδειγμα. Στη διαίρεση 2/3 το αποτέλεσμα είναι 0.666666666. Η απλή συνεχής μορφή αυτού του κλάσματος δείχνει καθαρά ότι κάποιοι αριθμοί δεν μπορούν να υπολογιστούν και να παρουσιαστούν χωρίς την βοήθεια μίας αλγοριθμικής μεθόδου.
Σε αυτό το σημείο μπορούμε να πούμε ότι ο τρόπος με τον οποίο γράφονται οι αριθμοί σε ένα αριθμητικό σύστημα αποτελεί έναν αλγόριθμο. Για παράδειγμα ο αριθμός 101 στο Ινδοαραβικό σύστημα αρίθμησης μπορεί να ερμηνευτεί ως ακολούθως :
Ας θεωρήσουμε μία γραμμική ακολουθία θέσεων οι οποίες πρόκειται να καταληφθούν από ποσοτικά σύμβολα γραμμένα από τα δεξιά προς τα αριστερά. Κάθε θέση διαδοχικά αντιπροσωπεύει μία δύναμη του 10 και μπορεί να συμπληρωθεί με τους ακόλουθους αριθμούς :0,1,3,4,5,6,7,8,9. Η πρώτη θέση αντιπροσωπεύει μία κανονική αριθμητική μονάδα, η δεύτερη θέση το δέκα εις την μονάδα, η τρίτη θέση το δέκα εις το τετράγωνο κ.λπ.

Η τιμή ενός αριθμού που παρουσιάζεται με αυτόν τον τρόπο παράγεται με την πρόσθεση κάθε αριθμητικής μονάδας αφού αυτή έχει πολλαπλασιαστεί με την δύναμη του 10.

Ας δούμε τι ισχύει με τον αριθμό 101:

Ο αριθμός101

1 0 1

Οι δυνάμεις με βάση το 10

10^2 10^1 1

Έτσι, ο αριθμός 101 υπολογίζεται ως εξής :

$$(1*100)+(0*10)+(1)=101 \qquad (1.7)$$

Αυτή η εξήγηση του δεκαδικού συστήματος μπορεί εύκολα να προσαρμοστεί για να αντιπροσωπεύσει το δυαδικό σύστημα απλά αντικαθιστώντας την δύναμη του δέκα με την δύναμη του δύο. Στο δυαδικό σύστημα ο αριθμός 101 δηλώνει μία διαφορετική τιμή. Ας δούμε λοιπόν το ανάλογο παράδειγμα:
Ας θεωρήσουμε μία γραμμική ακολουθία θέσεων οι οποίες πρόκειται να καταληφθούν από ποσοτικά σύμβολα γραμμένα από τα δεξιά προς τα αριστερά. Κάθε θέση διαδοχικά αντιπροσωπεύει μία δύναμη του 2 και μπορεί να συμπληρωθεί με δύο μονάδες το 0 ή το 1.

Η πρώτη θέση αντιπροσωπεύει το δύο εις το μηδέν, η δεύτερη θέση το 2 εις την πρώτη, η τρίτη θέση το 2 εις την δευτέρα κλπ. Η τιμή σε αυτή την περίπτωση υπολογίζεται με την πρόσθεση κάθε αριθμητικής μονάδας αφού αυτή έχει πολλαπλασιαστεί με την δύναμη του 2.

Τα αριθμητικά σύμβολα 1,0,1

1 0 1

Οι δυνάμεις με βάση το 2

2^2 2^1 2^0

Έτσι, ο αριθμός 101 υπολογίζεται ως εξής :

$$(1*4)+(0*2)+(1)=5 \quad (1.8)$$

Και στις δύο αυτές περιπτώσεις η περιγραφή ωθεί τον διδασκόμενο να φανταστεί τους πίνακες και αναλόγως να προχωρήσει στην λύση του προβλήματος άσχετα που εμείς για να βοηθηθούμε τους δημιουργήσαμε. Επίσης χωρίς να μπούμε στην ουσία των δύο συστημάτων τα κωδικοποιούμε, με αφαιρετικό τρόπο φυσικά βήμα-βήμα.

Θα μπορούσαμε λοιπόν να πούμε ότι όλα τα αριθμητικά συστήματα είναι αλγοριθμικά με την έννοια του ότι αντιπροσωπεύουν ποσότητες οι οποίες υπολογίζονται με την χρήση διαδικασιών.

Επιπροσθέτως θα ήταν ορθό να επισημάνουμε ότι όλοι οι αριθμοί είναι αλγοριθμικοί αφού παράγονται από αριθμητικά συστήματα τα οποία στην ουσία είναι και τα ίδια αλγόριθμοι.

Τελικά θα ήταν σωστό να τονίσουμε ότι οι ίδιοι οι αριθμοί δεν μετράνε κάτι αλλά αποτελούν σύμβολα σε ακολουθίες ποσοτήτων και διαδικασιών.

Ο αλγόριθμος, προφανώς, αποτελεί ένα οικοδόμημα της ανθρώπινης σκέψης και πρακτικής. [6]

1.2.9 Οι αλγόριθμοι και οι αυτοματοποιημένοι υπολογισμοί

Οι αλγόριθμοι για υπολογισμούς με το χέρι αυτοματοποιήθηκαν σε διαφορετικές χρονικές περιόδους και με διαφορετικούς τρόπους. Τον δέκατο έβδομο αιώνα στην Ευρώπη οι φιλόσοφοι όπως ο Blaise Pascal και ο Gottfried Wilhelm Leibniz ήδη σχεδίαζαν χειροκίνητες αριθμομηχανές (calculators) για τις τέσσερις βασικές αριθμητικές πράξεις δηλαδή την πρόσθεση, την αφαίρεση, τον πολλαπλασιασμό και τη διαίρεση με τη χρήση του δεκαδικού συστήματος.

Στους σύγχρονους καιρούς η αφηρημένη μαθηματική σκέψη αναπτυσσόταν παράλληλα με την μηχανική. Για παράδειγμα η περίφημη φιλοσοφική “Method” του René Descartes ήταν και “μηχανική” και “αλγοριθμική” με έμφαση στην αποδόμηση ενός προβλήματος στα απλούστερα στοιχεία του.

Ο πολωνός ιστορικός Henryk Grossmann υποστήριξε ότι δεν ήταν καθόλου τυχαίο που ο René Descartes ανακάλυψε την λογική του μέθοδο ενώ ο ίδιος σχεδίαζε μηχανικά εργαλεία.

Ειδικότερα ο Grossmann εκφράζει τη γνώμη ότι κάθε μαθηματικός κανόνας έχει τον μηχανικό του χαρακτήρα ο οποίος απαιτεί πνευματική εργασία και υπολογισμό.

Αυτή ακριβώς η επισήμανση του Grossmann μας βοήθησε να καταλάβουμε πως η εξέλιξη της αφηρημένης σκέψης είτε αυτή είχε να κάνει με τα μαθηματικά είτε με τη λογική είτε με τη αλγοριθμική είτε με τη φιλοσοφία είναι πάντα συνδεδεμένη με τα τεχνολογικά επιτεύγματα.

Στη βιομηχανική εποχή ο πρώτος αλγόριθμος που αυτοματοποιήθηκε ήταν ο η Μέθοδος των Πεπερασμένων Διαφορών του Prony (Prony’s method of differences).

Ο Charles Babbage ενσωμάτωσε τον συγκεκριμένο αλγόριθμο στον σχεδιασμό της Διαφορικής του Μηχανής (Difference Engine) με την βοήθεια της οποίας ήθελε να αυτοματοποιήσει τον υπολογισμό του λογαριθμικών πινάκων (εκείνη την χρονική περίοδο, όταν οι θαλάσσιοι δρόμοι χρησιμοποιούνταν για αποικιακή επέκταση, αυτοί οι πίνακες, μαζί φυσικά και με άλλα εργαλεία, βοηθούσαν στον ορισμό του γεωγραφικού μήκους στην πλοήγηση).

Οι "υπολογιστικές μηχανές" του Charles Babbage δεν πήραν σάρκα και οστά διότι ήταν δύσκολη η υλοποίηση του δεκαδικού συστήματος σε μία συσκευή αλλά αυτή η προσπάθεια θεωρείται ως το πρώτο σημείο σύγκλισης μεταξύ των μαθηματικών αλγορίθμων και του βιομηχανικού αυτοματισμού.

Συνεπώς στην Εποχή της Πληροφορίας οι αλγόριθμοι για υπολογισμούς δεν μηχανοποιήθηκαν αλλά εξ ηλεκτρίστηκαν χάρη στο δυαδικό σύστημα.

Αναλυτικότερα, οι αλγόριθμοι υλοποιήθηκαν σε ηλεκτρικά και ηλεκτρονικά κυκλώματα για ταχύτητα στη λήψη αποτελεσμάτων.

Για την ακρίβεια οι δυαδικοί αριθμοί υιοθετήθηκαν μετά το 1940 με ορόσημο την πτυχιακή εργασία του Claude Shannon με τίτλο 'A Symbolic Analysis of Relay and Switching Circuits',η οποία παρουσιάστηκε στο Τεχνολογικό Ινστιτούτο της Μασαχουσέτης.

Ο Claude Shannon πρότεινε την χρήση των δυαδικών ιδιοτήτων των ηλεκτρικών διακοπτών για αναπαράσταση της προτασιακής λογικής και την εκτέλεση λογικών συναρτήσεων Boole οι οποίες συμβολίζονται ως AND, OR και NOT.

Σε αντίθετη με τους δεκαδικούς αριθμούς, οι δυαδικοί αριθμοί υλοποιούνται ευκολότερα σε ένα ηλεκτρονικό κύκλωμα : οι τιμές μηδέν και ένα μπορούν εύκολα να αντιπροσωπεύονται από τις καταστάσεις "ON" και "OFF" ενός κυκλώματος για παράδειγμα.

Με την εισαγωγή του προγραμματιζόμενου ηλεκτρονικού υπολογιστή δηλαδή ενός υπολογιστή ο οποίος μπορούσε να δεχτεί δεδομένα, να κάνει μαθηματικές πράξεις γρήγορα και να παρουσιάζει τα αποτελέσματα στον χρήστη έλαβε χώρα μία αλλαγή οι οποία είχε πολλές παραμέτρους: οι αριθμοί έγιναν οδηγίες.

Η ονομαζόμενη "κοινωνία της πληροφορίας" δεν έχει μόνο ως αντικείμενο την χρήση των δυαδικών αριθμών για κωδικοποίηση της ανθρώπινης γλώσσας και το αναλογικού περιεχομένου μέσω της ψηφιοποίησης αλλά και την επιτάχυνση των μηχανικών υπολογισμών μέσω της δυαδικής λογικής.

Οι γλώσσες των υπολογιστών και τα προγράμματα σταδιακά κατέστησαν εφικτή την αναπαράσταση περίπλοκων αλγόριθμων των οποίων τα δομικά στοιχεία είναι οι απλές δυαδικές οδηγίες του "κώδικα μηχανής".

Τον εικοστό αιώνα ο δυαδικός κώδικας, η αρχιτεκτονική Von Neumann και η υλοποίηση των "λογικών πυλών" στα έξυπνα microchips έδωσαν ώθηση στη δημιουργία γρήγορων ηλεκτρονικών υπολογιστών και στην διαμόρφωση μεγάλων και περίπλοκων αλγόριθμων.

Για πρώτη φορά στην ιστορία ακολουθίες αριθμών χρησιμοποιήθηκαν όχι μόνο για να συμβολίσουν ποσότητες αλλά και οδηγίες, δηλαδή αλγόριθμους.

Αντίθετα από την πεποίθηση σαφούς διάκρισης μεταξύ υλικού και λογισμικού συμπεραίνουμε ότι κάθε είδος ψηφιακού υπολογισμού είναι στην πραγματικότητα η υλοποίηση στο ίδιο μέσο πληροφορίας και οδηγιών δυαδικών αριθμών και λογικής Boole. Οι δυαδικοί αριθμοί αποτελούν συμπληρωματική μορφή της λογικής Boole.

Με τον ψηφιακό υπολογισμό ο αλγόριθμος αρίθμησης και ο υπολογιστικός αλγόριθμος έγιναν ένα και το αυτό. [6]

1.3 Περιγραφή αλγόριθμου

Ο αλγόριθμος φυσιολογικά θα πρέπει στο τελικό του στάδιο να εκφραστεί σε κάποια γλώσσα προγραμματισμού έτσι ώστε να μπορεί να λυθεί το πρόβλημα με την βοήθεια ενός ηλεκτρονικού υπολογιστή.

Κατά τη διάρκεια του σχεδιασμού ενός αλγορίθμου χρησιμοποιείται για την περιγραφή του είτε η φυσική γλώσσα είτε η ψευδογλώσσα προγραμματισμού(pseudo-programming language) είτε ο ψευδοκώδικας (pseudocode).

Ψευδοκώδικα ονομάζουμε ένα είδος μίξης γλωσσών προγραμματισμού, συγκεκριμένα κάποιων δομών τους και προτάσεων σε φυσική γλώσσα (ελληνικά αγγλικά κλπ.)με την βοήθεια της οποίας περιγράφεται η γενική ιδέα μίας υλοποίησης ενός αλγόριθμου σε πρόγραμμα με την χρήση μίας γλώσσας προγραμματισμού την οποία θα επιλέξει ο προγραμματιστής. Με αυτό τον τρόπο είμαστε σε θέση να εστιάσουμε στις ιδιαιτερότητες του προβλήματος για τη λύση του οποίου αναπτύχθηκε ο αλγόριθμός μας.

Οι δομές γλωσσών προγραμματισμού που χρησιμοποιούνται σε έναν ψευδοκώδικα είναι οι εξής:

- Εκφράσεις :χρησιμοποιούμε μαθηματικά σύμβολα για να εκφράσουμε αριθμητικές και Boole εκφράσεις. Συγκεκριμένα το βέλος με φορά από δεξιά προς τα αριστερά(←) λειτουργεί ως τελεστής ανάθεσης τιμής σε μία μεταβλητή. Εκτός αυτού χρησιμοποιούμε και το σύμβολο ισότητας (=).
- Δήλωση μεθόδων :πρώτα γράφουμε την λέξη Αλγόριθμος, το όνομα που επιθυμούμε να του δώσουμε και μετά ανοίγουμε παρένθεση πληκτρολογούμε τις παραμέτρους μας και φυσικά κλείνουμε παρένθεση.
- Δομές απόφασης : τέτοιες δομές αρχίζουν με την λέξη εάν η οποία ακολουθείται από μία συνθήκη της οποίας έπεται η λέξη τότε ακολουθούμενη από εντολές και κάποιες φορές τη λέξη διαφορετικά.
- Βρόχοι ενώ : πρώτα έχουμε την λέξη ενώ η οποία ακολουθείται από την συνθήκη στην συνέχεια έχουμε την λέξη κάνε και τις εντολές μας
- Βρόχοι επανάλαβε: πρώτα έχουμε την λέξη επανάλαβε μετά την οποία ακολουθούν οι εντολές. Στη συνέχεια έχουμε την λέξη μέχρι και την συνθήκη μας.
- Βρόχοι για: σε αυτή την περίπτωση αρχίζουμε με την λέξη και συνεχίζουμε με την αύξηση του μετρητή μας την λέξη κάνε και φυσικά τις εντολές μας.

- Τοποθέτηση στοιχείων σε θέσεις ενός πίνακα: ας πούμε ότι έχουμε έναν πίνακα έστω Α και έναν δείκτη ο οποίος δηλώνει τη θέση κάθε στοιχείου το στοιχείο οπότε ο όρος Α[i] δηλώνει το κελί που βρίσκεται ο όρος του πίνακα.
- Κλήσεις μεθόδου.
- Η εντολή επίστεφε :σε αυτή την περίπτωση έχουμε την επιστροφή μίας τιμής [1]

1.4 Χαρακτηριστικά αλγορίθμων

Οι αλγόριθμοι έχουν τα ακόλουθα χαρακτηριστικά :

- Καθαρότητα και σαφήνεια: τα βήματα ενός αλγορίθμου πρέπει να είναι ξεκάθαρα διατυπωμένα και να μην επιδέχονται κανενός είδους παρερμηνεία.
- Ορθός καθορισμός δεδομένων εισόδου: αν για τη λειτουργία ενός αλγόριθμου απαιτούνται δεδομένα εισόδου αυτά θα πρέπει να είναι σαφώς καθορισμένα δηλαδή θα πρέπει να δηλώνεται ξεκάθαρα ο τύπος τους.
- Ορθός καθορισμός δεδομένων εξόδου :Θα πρέπει να οριστεί με σαφήνεια ο τύπος των δεδομένων εξόδου του αλγόριθμου.
- Ισχύ ιδιότητας του πεπερασμένου: ο κάθε αλγόριθμος θα πρέπει να τερματίζει μετά από καθορισμένο χρόνο.
- Δυνατότητα υλοποίησης : ο κάθε αλγόριθμος θα πρέπει να είναι απλός και πρακτικός έτσι ώστε να μπορεί να εκτελεστεί με τους διαθέσιμους πόρους.

- Ανεξαρτησία από γλώσσες προγραμματισμού :ένας αλγόριθμος πρέπει να αποτελείται από οδηγίες οι οποίες μπορούν να υλοποιηθούν σε μία οποιαδήποτε γλώσσα προγραμματισμού έτσι ώστε το αποτέλεσμα εξόδου να είναι το ίδιο όπως είναι αναμενόμενο.
- Αριθμός δεδομένων εισόδου: ένας αλγόριθμος μπορεί είτε να μην έχει καθόλου δεδομένα εισόδου είτε να έχει όσα απαιτούνται για την επίλυση του προβλήματος για το οποίο υλοποιήθηκε.
- Αριθμός δεδομένων εξόδου: ένας αλγόριθμος πρέπει να έχει τουλάχιστον ένα αποτέλεσμα στην έξοδό του.
- Σαφήνεια :όλες οι οδηγίες ενός αλγόριθμου πρέπει να έχουν ως χαρακτηριστικά την ευκολία ερμηνείας τους, και την ακρίβειά τους. Κάθε οδηγία θα πρέπει να είναι ξεκάθαρη και έτσι να είναι απόλυτα σαφές αυτό που θα πρέπει να υλοποιηθεί. Κάθε θεμελιώδης τελεστής θα πρέπει επίσης να ορίζεται ξεκάθαρα.
- Πεπερασμένα βήματα: κάθε αλγόριθμος θα πρέπει να τερματίζει αφού συμπληρωθεί καθορισμένος αριθμός βημάτων.

- Αποτελεσματικότητα: κάθε αλγόριθμος πρέπει να υλοποιείται με την χρήση αποτελεσματικών λειτουργιών.[5] [6] [7]

1.5 Εφαρμογές αλγορίθμων

Ο ρόλος που παίζουν οι αλγόριθμοι σε διαφόρους τομείς είναι ιδιαίτερα σημαντικός και έχουν ποικίλες εφαρμογές.

Κάποιοι από τους τομείς στους οποίους εφαρμόζονται είναι οι ακόλουθοι:

- Επιστήμη των ηλεκτρονικών υπολογιστών: στα πλαίσια του προγραμματισμού οι αλγόριθμοι χρησιμοποιούνται για την επίλυση απλών προβλημάτων όπως για παράδειγμα μία απλή ταξινόμηση πινάκων ή αναζήτηση στοιχείων εντός αυτών αλλά και για περίπλοκα προβλήματα που σχετίζονται με την τεχνητή νοημοσύνη και τη μηχανική μάθηση.
- Μαθηματικά: οι αλγόριθμοι χρησιμοποιούνται στην επίλυση μαθηματικών προβλημάτων όπως η εύρεση βέλτιστής λύσης ενός συστήματος γραμμικών εξισώσεων ή η εύρεση συντομότερης διαδρομής σε ένα διάγραμμα.
- Επιχειρησιακή έρευνα: οι αλγόριθμοι χρησιμοποιούνται για την βελτιστοποίηση και την λήψη αποφάσεων σε πεδία όπως οι μεταφορές, η διαχείριση υλικού και η κατανομή πόρων.
- Τεχνητή νοημοσύνη: οι αλγόριθμοι αποτελούν τα θεμέλια της τεχνητής νοημοσύνης και τις μηχανικής μάθησης και χρησιμοποιούνται στην ανάπτυξη ευφυών συστημάτων με την βοήθεια των οποίων μπορεί να γίνει εφικτή η αναγνώριση μέσω εικόνας, η επεξεργασία φυσικής γλώσσας, η λήψη αποφάσεων.
- Επεξεργασία δεδομένων :οι αλγόριθμοι αποτελούν εργαλείο ανάλυσης, επεξεργασίας, και εξαγωγής γνώσεων από μεγάλο όγκο δεδομένων σε τομείς όπως η προώθηση αγαθών (marketing), η οικονομία και η υγεία.
- Οι αλγόριθμοι μπορεί να είναι είτε περίπλοκοι είτε απλοί και αυτό εξαρτάται από τον σκοπό για τον οποίο σχεδιάστηκαν.[1]

1.6 Τύποι αλγορίθμων

Οι σημαντικότεροι τύποι αλγορίθμων είναι οι ακόλουθοι:

- **Αλγόριθμος Brute force(Brute force algorithm):**αποτελεί την απλούστερη προσέγγιση στην λύση ενός προβλήματος.
- **Αναδρομικός αλγόριθμος (Recursive Algorithm)**: ο σχεδιασμός ενός τέτοιου αλγόριθμου είναι βασισμένος, όπως μπορούμε να συμπεράνουμε και από τον επιθετικό προσδιορισμό, σε μία τεχνική που ονομάζεται αναδρομή (recursion).Σε αυτή τη περίπτωση το πρόβλημα διαχωρίζεται σε τμήματα και η συνάρτηση καλείται ξανά και ξανά.

- **Αλγόριθμος οπισθοδρόμησης (Backtracking algorithm):** με την βοήθεια ενός τέτοιου αλγόριθμου δομείται η λύση με αναζήτηση ανάμεσα σε όλες τις πιθανές λύσεις. Συγκεκριμένα, είναι δυνατή η σταδιακή δόμηση της λύσης του προβλήματος ακολουθώντας κάποια κριτήρια. Σε περίπτωση που μία λύση αποτυγχάνει οπισθοδρομούμε στο σημείο της αποτυχίας δομούμε την επόμενη λύση και συνεχίζουμε μέχρι να βρούμε την λύση του προβλήματος.
- **Αλγόριθμος αναζήτησης (Searching algorithm):** αυτός ο αλγόριθμος εφαρμόζεται όταν θέλουμε να αναζητήσουμε ένα στοιχείο ή και μία ομάδα στοιχείων εντός μιας δομής δεδομένων.
- **Αλγόριθμος ταξινόμησης (Sorting algorithm):** με την βοήθεια ενός τέτοιου αλγορίθμου είναι δυνατή η ταξινόμηση κάποιων στοιχείων με ένα συγκεκριμένο τρόπο.
- **Αλγόριθμος κατακερματισμού (Hashing algorithm):**ο συγκεκριμένος αλγόριθμος δουλεύει παρόμοια με τον αλγόριθμο ταξινόμησης αλλά υπάρχει και ένα κλειδί το οποίο ανατίθεται σε ένα συγκεκριμένο δεδομένο
- **Αλγόριθμος διαίρει και βασίλευε (Divide and Conquer algorithm):** με αυτόν τον αλγόριθμο μπορούμε να "σπάσουμε" ένα πρόβλημα σε "ΥΠΟ προβλήματα" να λύσουμε το καθένα από αυτά, να συγχωνεύσουμε τις λύσεις και με αυτό τον τρόπο να φτάσουμε στην τελική λύση. Τα τρία βήματα που πρέπει να ακολουθήσουμε είναι: διαίρεση, λύση, συνδυασμός.
- **Άπληστος αλγόριθμος(Greedy algorithm)**:με αυτό τον αλγόριθμο η λύση δομείται τμηματικά. Η λύση για το επόμενο τμήματα δομείται βασισμένη στο άμεσο όφελος του επόμενου τμήματος. Η λύση που είναι πιο ωφέλιμη διαλέγεται ως λύση για το επόμενο μέρος.
- **Αλγόριθμος δυναμικού προγραμματισμού (Dynamic programming algorithm):**η ιδέα στην οποία βασίζεται ο συγκεκριμένος αλγόριθμος είναι ότι αφού "σπάσει" το πρόβλημα σε μικρότερα
- προβλήματα την χρήση λύση των οποίων "βάζει στην άκρη" για την αποφυγή επαναληπτικών υπολογισμών του ίδιου τμήματος του προβλήματος. Με απλά λόγια ο συγκεκριμένος αλγόριθμος "θυμάται" τη λύση.
- **Τυχαιοποιημένος αλγόριθμος (Randomized algorithm):**σε αυτή την περίπτωση χρησιμοποιείται ένας τυχαίος αριθμός ο οποίος δίνει άμεσο όφελος. [1]

1.6 Επαλήθευση ορθότητας αλγορίθμου

Για να αποδείξουμε ότι ένας αλγόριθμος είναι ορθός δηλαδή ότι παρέχει την σωστή απάντηση στο πρόβλημα για το οποίο αναπτύχθηκε ανεξάρτητα από το μέγεθος των δεδομένων που δέχεται ως είσοδο θα πρέπει να ελεγχθεί.

Οι τρόποι ελέγχου της ορθότητας ενός αλγορίθμου είναι κυρίως δύο, οι ακόλουθοι:

- η απόδειξη με μαθηματικά εργαλεία κυρίως με μαθηματική επαγωγή η οποία χρησιμοποιείται και κατά τη διάρκεια του σχεδιασμού ενός αλγορίθμου

- ο έλεγχος της απάντησης σε ένα στατιστικά σημαντικό πλήθος εναλλακτικών παραμέτρων του προβλήματος

Ας περιγράψουμε την αποδεικτική διαδικασία της επαγωγής λοιπόν: έστω μία πρόταση π η οποία εξαρτάται από ένα φυσικό αριθμό $n \geq n_0$. Εάν αποδειχτεί ότι η πρόταση π :ισχύει για $n=n_0$ όπως επίσης και για n=k+1,εφόσον ισχύει για n=k, τότε η εν λόγω πρόταση ισχύει για κάθε $n \geq n_0$.

Εναλλακτική της παραπάνω διαδικασίας είναι η ισχυρά επαγωγή :έστω μία πρόταση π η οποία εξαρτάται από ένα φυσικό αριθμό $n \geq n_0$. Εάν αποδειχτεί ότι η πρόταση π :ισχύει για $n=n_0$ όπως επίσης και για n=k+1,εφόσον ισχύει για κάθε $n \leq k$, τότε η εν λόγω πρόταση ισχύει για κάθε $n \geq n_0$. [1] [9]

Οι δύο μορφές της επαγωγικής διαδικασίας είναι πλήρως ισοδύναμες και εφαρμόζονται κατά την κρίση του ατόμου που σχεδιάζει έναν οποιονδήποτε αλγόριθμο.

Η ορθότητά τους στηρίζεται στην ιδιότητα η οποία ονομάζεται ιδιότητα της καλής διατάξεως την οποία επιδεικνύει κάθε σύνολο φυσικών αριθμών.

Ας επιδείξουμε λοιπόν επαγωγικά την ορθότητα του αλγόριθμου που ακολουθεί:

ΑΛΓΟΡΙΘΜΟΣ arrayMax(A, n):
Είσοδος : Ένας πίνακας A στον οποίο αποθηκεύσαμε $n \geq 1$ ακέραιους
Έξοδος :Το μέγιστο στοιχείο του πίνακα A
Τρέχον_μέγιστο _στοιχείο ← A[0]
Για i ← 1 μέχρι n – 1 κάνε
Εάν Τρέχον_μέγιστο_στοιχείο < A[i] τότε
Τρέχον_μέγιστο _στοιχείο ← A[i]
Τέλος _Για
Επίστεφε Τρέχον_μέγιστο _στοιχείο

Πίνακας 1.1 Αλγόριθμος arrayMax

Για να δείξουμε λοιπόν ότι ένας βρόχος ικανοποιεί μία πρόταση π η οποία εξαρτάται από το πλήθος των επαναλήψεων n και εκφράζει κάποια σχέση μεταξύ των εμπλεκόμενων μεταβλητών αποδεικνύουμε τα εξής:

- η πρόταση είναι αληθής πριν την εκκίνηση του βρόχου
- εάν είναι αληθής πριν την i-στή επανάληψη του βρόχου παραμένει αληθής και μετά την ολοκλήρωση της i-στής επανάληψης
- αποδεικνύουμε με επιχειρήματα ότι μετά την ολοκλήρωση του βρόχου αληθεύει η πρόταση π

Κάθε πρόταση που αποδεικνύει την ορθότητα υπολογισμού ενός βρόχου καλείται αμετάβλητη του βρόχου (loop invariant).

Έστω λοιπόν ο αλγόριθμος ευρέσεως του μέγιστου στοιχείου ενός πίνακα που αναπτύξαμε λίγο πιο πάνω.

Θέλουμε να αποδείξουμε ότι μετά το πέρας του βρόχου για η μεταβλητή Τρέχον_μέγιστο _στοιχείο έχει αποθηκευμένο το μέγιστο στοιχείο του πίνακα.

Η πρόταση π έχει ως εξής :

Κατά το ξεκίνημα της i-στή επανάληψης η μεταβλητή Τρέχον_μέγιστο _στοιχείο αποθηκεύει το μέγιστο στοιχείο των i πρώτων στοιχείων του πίνακα.

Πριν την εκκίνηση η πρότασή μας είναι προφανώς αληθής. Έστω τώρα ότι είναι αληθής πριν την i-στή επανάληψη.

Θα παραμείνει αληθής και μετά το πέρας αυτής καθώς η τιμή της μεταβλητής Τρέχον_μέγιστο _στοιχείο είναι η μέγιστη μεταξύ του μεγίστου των i πρώτων στοιχείων και του (i +1)-οστού.

Όταν τελειώσει η εκτέλεση του βρόχου το μέγιστο θα έχει βρεθεί αφού θα έχουν εξεταστεί όλα τα στοιχεία του πίνακά μας. [1] [2]

1.7 Ανάλυση αλγορίθμων

Ο εκάστοτε αλγόριθμος αποτελεί, όπως ήδη αναφέραμε εκτός των άλλων, μία σταδιακή διαδικασία η οποία έχει ως στόχο την υλοποίηση συγκεκριμένων εντολών σε πεπερασμένο χρόνο.[4]

1.7.1 Σχέση μεταξύ δομών δεδομένων και αλγόριθμων

Ο πρωτεύων στόχος του προγραμματισμού είναι η ικανοποιητική επεξεργασία των δεδομένων εισόδου για να παραχθεί η επιθυμητή έξοδος, δηλαδή η λύση του προβλήματος. Ο στόχος μπορεί να επιτευχθεί εαν τα δεδομένα εισόδου οργανωθούν σωστά.

Οι δομές δεδομένων δεν είναι τίποτα άλλο παρά τρόποι οργάνωσης δεδομένων οι οποίες καθιστούν δυνατή την εύκολη και αποτελεσματική επεξεργασία τους.

Αυτές υπαγορεύουν τον τρόπο επεξεργασίας των δεδομένων, δηλαδή η επιλογή ενός αλγόριθμου εξαρτάται από τον τρόπο οργάνωσης των δεδομένων προς επεξεργασία.

Ας δούμε ένα απλό παράδειγμα :υποθέτουμε ότι έχουμε μία συλλογή δέκα ακεραίων και επιθυμούμε να ανακαλύψουμε αν ένα συγκεκριμένο κλειδί (key) δηλαδή ένας συγκεκριμένος ακέραιος βρίσκεται στην συλλογή.

Ο αλγόριθμος είναι ο ακόλουθος :

Πίνακας 1.2 Αποθήκευση δεδομένων σε διακριτές μεταβλητές

ΠΡΩΤΟΣ ΤΡΟΠΟΣ ΟΡΓΑΝΩΣΗΣ

Τα δεδομένα είναι αποθηκευμένα σε διακριτές μεταβλητές A0...A9

Αλγόριθμος

```
found = false;
if (key = = A0)                found = true
else if (key = = A1)           found = true
else if (key = = A2)           found = true
else if (key = = A3)           found = true
else if (key = = A4)           found = true
else if (key = = A5)           found = true
else if (key = = A6)           found = true
else if (key = = A7)           found = true
```

```
else if (key = = A8)        found = true
else if (key = = A9)        found = true
```

Μπορούμε να οργανώσουμε τα δεδομένα μας με διαφορετικούς τρόπους. Στο προηγούμενο παράδειγμα αποθηκεύσαμε τα δεδομένα μας σε μεταβλητές ανεξάρτητες μεταξύ τους και χρησιμοποιήσαμε μία αλή δομή if-else για να εξετάσουμε εάν η τιμή που είναι αποθηκευμένη στην μεταβλητή key βρίσκεται ανάμεσα στις τιμές οι οποίες είναι αποθηκευμένες στις μεταβλητές A0 έως A9.

Ένας δεύτερος τρόπος οργάνωσης είναι η δημιουργία ενός πίνακα στον οποίο θα αποθηκευτούν τα δεδομένα κάτι που είδαμε και στον αλγόριθμο που υλοποιήσαμε στην παράγραφο 1.6 [8]

Ας τον δούμε λοιπόν :

Πίνακας 1.3 Αποθήκευση δεδομένων σε πίνακα

ΔΕΥΤΕΡΟΣ ΤΡΟΠΟΣ ΟΡΓΑΝΩΣΗΣ

Τα δεδομένα μας είναι αποθηκευμένα σε έναν πίνακα έστω Α δέκα στοιχείων

ΑΛΓΟΡΙΘΜΟΣ

```
const int N = 10;
found = false;
for (int i = 0; i < N; i++)
if (A[i] == key
{
found = true;
break;
}
```

Στην σταθερά N αποθηκεύεται ο ακέραιος αριθμός που δηλώνει τις θέσεις του πίνακα. Αρχικοποιούμε την μεταβλητή τύπου Boolean αποδίδοντας την τιμή “false”. Στην συνέχεια χρησιμοποιούμε την επαναληπτική δομή for για να ανατρέξουμε τον πίνακά μας. Σε περίπτωση που το στοιχείο βρεθεί ο αλγόριθμός μας τερματίζεται και βγαίνει από το βρόχο αφού υπάρχει η εντολή break.

Ένας άλλος τρόπος οργάνωσης είναι η δυαδική αναζήτηση, δηλαδή η αποθήκευση των τιμών μας σε έναν ταξινομημένο πίνακα είτε σε αύξουσα είτε σε φθίνουσα σειρά :

Πίνακας 1.4 Αποθήκευση δεδομένων σε πίνακα ταξινομημένο σε αύξουσα σειρά

ΤΡΙΤΟΣ ΤΡΟΠΟΣ ΟΡΓΑΝΩΣΗΣ

Τα δεδομένα μας είναι αποθηκευμένα σε έναν πίνακα έστω Α δέκα στοιχείων ο οποίος είναι ταξινομημένος σε αύξουσα σειρά.

ΑΛΓΟΡΙΘΜΟΣ

```
const int N = 10;
found = false;
low = 0;
high = N – 1;
```

```
while (( ! found) && ( low <= high))
{
mid = (low + high)/2;
if (A[mid] = = key)
found =true;
else if (A[mid] > key)
high= mid – 1;
else
low= mid + 1;
}
```

Πάλι έχουμε έναν πίνακα δέκα θέσεων, αυτή τη φορά ταξινομημένο σε αύξουσα σειρά. Στη μεταβλητή low αποθηκεύουμε τη θέση μηδέν, δηλαδή την αρχή του πίνακά μας και στην μεταβλητή high τη θέση N – 1 δηλαδή τη θέση που δηλώνει το τέλος του πίνακά μας. [8]

Στη συνέχεια έχουμε την δομή επανάληψης while με την βοήθεια της οποίας διατρέχουμε τον πίνακά μας.

Ας δούμε λοιπόν αναλυτικά τι συμβαίνει: όσο δεν έχει βρεθεί η τιμή την οποία αναζητούμε και όσο δεν έχουμε διατρέξει τα στοιχεία όλου του πίνακα βρίσκουμε το μεσαίο στοιχείο του (mid = (low + high)/2).

Αν η τιμή του μεσαίου στοιχείου του πίνακα είναι αυτή που αναζητούμε τότε φυσικά λήγει η αναζήτηση και η τιμή της μεταβλητής found τύπου Boolean αλλάζει και γίνεται true.

Σε περίπτωση που η τιμή του μεσαίου στοιχείου είναι μεγαλύτερη από η τιμή του στοιχείου που αναζητούμε, αναθέτουμε στην μεταβλητή high την τιμή mid-1 για να προχωρήσει η αναζήτηση.

Σε διαφορετική περίπτωση, δηλαδή αν η τιμή του μεσαίου στοιχείου είναι μικρότερη από η τιμή του στοιχείου που αναζητούμε αναθέτουμε στην μεταβλητή low την τιμή mid+1 για να προχωρήσει η αναζήτηση.

Είναι ξεκάθαρο ότι ο αλγόριθμος που θα χρησιμοποιήσουμε εξαρτάται άμεσα από τον τρόπο οργάνωσης δεδομένων ή πιο απλά από την δομή δεδομένων που θα χρησιμοποιήσουμε.

Με τον πρώτο τρόπο οργάνωσης δεδομένων είχαμε μόνο μία επιλογή, με τον δεύτερο τρόπο δύο επιλογές ενώ με τον τρίτο τρείς επιλογές.

Το ερώτημα είναι ποιος τρόπος οργάνωσης δεδομένων είναι ο καλύτερος. Μπορούμε εύκολα να διαπιστώσουμε ότι ο δεύτερος τρόπος οργάνωσης είναι καλύτερος από τον πρώτο.

Είναι οφθαλμοφανές ότι ο αριθμός των συγκρίσεων και στις δύο περιπτώσεις παραμένει ο ίδιος αλλά στην δεύτερη περίπτωση ο αλγόριθμος είναι πιο κλιμακούμενος αφού είναι απείρως ευκολότερο να τροποποιηθεί έτσι ώστε να προστεθούν και άλλα δεδομένα. Για παράδειγμα αν θέλουμε να προσθέσουμε ακόμα ενενήντα δεδομένα ή πιο απλά ενενήντα τιμές στον πρώτο αλγόριθμο θα ήταν απαραίτητο να προσθέσουμε ακόμα ενενήντα μεταβλητές και ενενήντα προτάσεις if-else ενώ στην δεύτερη περίπτωση χρειάζεται απλώς να αλλάζουμε την τιμή της σταθεράς N και από 10 να την κάνουμε 100.

Η σύγκριση μεταξύ του δεύτερου και του τρίτου τρόπου παρουσιάζει μεγαλύτερο ενδιαφέρον.

Όπως μπορεί να δει εύκολα κάποιος ο δεύτερος αλγόριθμος (γραμμική αναζήτηση) είναι πιο απλός συγκριτικά με τον τρίτο (δυαδική έρευνα).

Επιπλέον ο αλγόριθμος γραμμικής αναζήτησης θέτει λιγότερους περιορισμούς στα δεδομένα εισόδου από αυτούς που θέτει ο αλγόριθμος δυαδικής έρευνας ο οποίος απαιτεί δεδομένα ταξινομημένα σε αύξουσα ή φθίνουσα σειρά.

Αν δει κάποιος το θέμα από την άποψη της εκτέλεσης του προγράμματος το σώμα του αλγόριθμου στην γραμμική αναζήτηση(δεύτερος τρόπος) δηλαδή από την αρχή της επαναληπτικής δομής for και μετά δείχνει ότι είναι πιο αποδοτικό από το αντίστοιχο σώμα της επαναληπτικής δομής while στην δυαδική αναζήτηση(τρίτος τρόπος).

Γενικά λοιπόν ο γραμμικός αλγόριθμος αναζήτησης φαίνεται καλύτερος από τον δυαδικό τρόπο αναζήτησης.

Και ερχόμαστε στο ακόλουθο ερώτημα: γιατί ασχολούμαστε με τον δυαδικό αλγόριθμο αναζήτησης αφού γραμμική αναζήτηση είναι καλύτερη; Η πραγματικότητα είναι ότι ο δυαδικός αλγόριθμος αναζήτησης είναι κατά πολύ ανώτερος του αλγόριθμου γραμμικής αναζήτησης.

Για να κατανοήσουμε τον λόγο αυτής της παραδοχής θα αναλύσουμε το παρακάτω πρόβλημα.

Η Εθνική Βάση Δεδομένων και Αρχή Καταχώρισης (National Database and Registration Authority, NADRA) της πακιστανικής κυβέρνησης διατηρεί μία βάση δεδομένων με τα στοιχεία των πολιτικών του Πακιστάν οι οποία έχει περίπου 80 εκατομμύρια αρχεία ο αριθμός των οποίων αυξάνεται συνεχώς.

Ας υποθέσουμε ότι θέλουμε να αναζητήσουμε μία εγγραφή σε αυτή την βάση δεδομένων. Επιπλέον ας υποθέσουμε ότι ο ηλεκτρονικός υπολογιστής που έχουμε μπορεί να εκτελέσει 100000 επαναλήψεις στο σώμα ενός βρόχου for στον γραμμικό αλγόριθμο αναζήτησης σε ένα δευτερόλεπτο.

Επίσης υποθέτουμε ότι κατά μέσο όρο κάθε επανάληψη στο σώμα του βρόχου while στο δυαδικό τρόπο αναζήτησης παίρνει τρεις φορές περισσότερο χρόνο από την επανάληψη εντός του σώματος του βρόχου for της γραμμικής αναζήτησης.

Τα αποτελέσματα της ανάλυσης αυτού προβλήματος φαίνονται στον πίνακα που ακολουθεί :

Πίνακας 1.5 Σύγκριση γραμμικής αναζήτησης με δυαδική αναζήτηση

	Γραμμική Αναζήτηση	**Δυαδική Αναζήτηση**
Αριθμοί επαναλήψεων στο σώμα του βρόχου σε ένα δευτερόλεπτο	100,000	~33000
Αναζήτηση εγγραφής μέσα σε 80 εκατομμύρια εγγραφές		
Χείριστη περίπτωση(ήταν αδύνατο να βρεθεί η εγγραφή)	800 seconds	~0.0008 seconds
Περίπτωση μέσου όρου	400 seconds	~0.0004 seconds
Αριθμός αναζητήσεων σε μία ώρα (μέσος όρος)	9	~ 9 million
Αριθμός αναζητήσεων καθημερινά	216	~ 213 Million

Η ανάλυση αποδεικνύει ότι ο αλγόριθμος της δυαδικής αναζήτησης είναι γρηγορότερος από αυτόν της γραμμικής αναζήτησης. [8]

1.7.2 Πειραματική προσέγγιση

Ο χρόνος τρεξίματος ενός αλγορίθμου ή μίας λειτουργίας δομής δεδομένων εξαρτάται από έναν αριθμό παραγόντων.

Αν ένας αλγόριθμος υλοποιηθεί σε μία γλώσσα προγραμματισμού μπορούμε να μελετήσουμε το χρόνο τρεξίματός του εκτελώντας τον με ποικίλα δεδομένα εισόδου και καταγράφοντας τον χρόνο κάθε εκτέλεσης. Τέτοιες μετρήσεις μπορεί να γίνουν με ακρίβεια με χρήση των κλίσεων συστήματος οι οποίες είναι ενσωματωμένες στη γλώσσα προγραμματισμού ή στο λειτουργικό σύστημα για το οποίο αναπτύχθηκε ο αλγόριθμος.

Γενικά όμως καθορίζουμε την εξάρτηση του χρόνου τρεξίματος του αλγόριθμου από το μέγεθος των δεδομένων εισόδου του.

Για να γίνει αυτό μπορούμε να κάνουμε πειράματα με δεδομένα εισόδου διαφορετικού μεγέθους.

Προκειμένου να είναι αυτά τα πειράματα ουσιώδους σημασίας θα πρέπει να διαλέξουμε ενδεικτικά δείγματα δεδομένων εισαγωγής και να δοκιμάσουμε έναν ικανό αριθμό αυτών για να εξαχθούν αξιόπιστα συμπεράσματα.

Η γενική παραδοχή είναι ότι ο χρόνος τρεξίματος ενός αλγόριθμου αυξάνεται αναλογικά με το μέγεθος των δεδομένων εισόδου του.

Επίσης ο χρόνος τρεξίματος εξαρτάται από το υλικό, το λογισμικό στο οποίο υλοποιήθηκε ο αλγόριθμος, μεταγλωττίστηκε και εκτελέστηκε.

Θα ήταν χρήσιμο να τονίσουμε σε αυτό το σημείο ότι ο χρόνος τρεξίματος του ίδιου αλγορίθμου με το ίδιο μέγεθος δεδομένων εισόδου μπορεί να είναι μικρότερος σε περίπτωση που ο ηλεκτρονικός υπολογιστής έχει έναν γρηγορότερο επεξεργαστή. [9]

1.7.3 Προϋποθέσεις για την υλοποίηση της πειραματικής προσέγγισης

Οι πειραματικές μελέτες για τους χρόνους τρεξίματος των εκάστοτε αλγορίθμων είναι χρήσιμες αλλά έχουν κάποιους περιορισμούς, τους εξής :

- Τα πειράματα μπορούν να γίνουν σε περιορισμένο σύνολο δεδομένων εισόδου και θα πρέπει αυτά τα να είναι αντιπροσωπευτικά για να διασφαλιστεί η αξιοπιστία του αποτελέσματος.
- Είναι δύσκολο να συγκριθεί η αποδοτικότητα δύο αλγορίθμων αν τα παραδείγματα για τον χρόνο τρεξίματός τους δεν έχουν γίνει στα ίδια περιβάλλοντα υλικού και λογισμικού.
- Είναι απαραίτητο να υλοποιηθεί και να εκτελεστεί ένας αλγόριθμος προκειμένου να μελετηθεί πειραματικά ο χρόνος τρεξίματός του. Πιο απλά θα πρέπει πρώτα ο αλγόριθμος να αναπτυχθεί σε μία γλώσσα προγραμματισμού και έπειτα να λάβει χώρα η πειραματική διαδικασία.

Συνεπώς, αν και η πειραματική μέθοδος έχει ένα σημαντικό ρόλο στην αλγοριθμική ανάλυση δεν είναι επαρκής. Υπάρχει η ανάγκη ενός επιπρόσθετου πλαισίου ανάλυσης στο οποίο θα πρέπει να ισχύουν τα παρακάτω :

- Θα πρέπει να λαμβάνονται υπόψη όλα τα πιθανά δεδομένα εισόδου.
- Θα είναι επιβεβλημένη η εκτίμηση της σχετικής αποδοτικότητας οποιουδήποτε ζευγαριού αλγορίθμων με τρόπο ανεξάρτητο από το λογισμικό και το υλικό.
- Θα είναι διαθέσιμη μία υψηλού επιπέδου περιγραφή του αλγόριθμου χωρίς την υλοποίησή του ή την διεξαγωγή πειραμάτων.

Η εν λόγω μεθοδολογία έχει ως σκοπό την σύνδεση ενός αλγορίθμου με μία συνάρτηση f(n) η οποία χαρακτηρίζει τον χρόνο τρεξίματος σε σχέση με τα δεδομένα εισόδου τα οποία είναι μεγέθους n.

Γενικά, οι συναρτήσεις συμπεριλαμβάνουν μεγέθη της τάξης n και n^2. Στην αλγοριθμική ανάλυση χρησιμοποιούμε την εξής φράση : "Ο αλγόριθμος Α τρέχει σε χρόνο ανάλογο με τα δεδομένα εισόδου n."

Αυτή η φράση σημαίνει ότι εάν επρόκειτο να πειραματιστούμε θα συμπεραίναμε ότι ο πραγματικός χρόνος τρεξίματος ενός αλγορίθμου Α με οποιαδήποτε δεδομένα εισόδου μεγέθους n δεν ξεπερνά σε καμία περίπτωση το γινόμενο c*n.

Με το γράμμα c συμβολίζουμε μία σταθερά η οποία εξαρτάται από το λογισμικό και το υλικό το οποίο θα μπορούσε να χρησιμοποιηθεί στο υποτιθέμενο πείραμα.

Δεδομένων δύο αλγορίθμων έστω Α και Β, όταν ο Α τρέχει σε χρόνο ανάλογο με μέγεθος δεδομένων εισόδου n, και ο Β "τρέχει" σε χρόνο ανάλογο με μέγεθος δεδομένων εισόδου n^2 εμείς θα προτιμήσουμε τον αλγόριθμο Α έναντι του αλγόριθμου Β αφού η συνάρτηση f(n) αυξάνεται με μικρότερο ρυθμό από την $f(n^2)$.

Για την αλγοριθμική ανάλυση πρέπει να λάβουμε υπόψη τα ακόλουθα στοιχεία:

- Τη γλώσσα περιγραφής των αλγορίθμων
- ένα υπολογιστικό μοντέλο εκτέλεσής τους
- μία μετρική υπολογισμού του χρόνου τρεξίματος των αλγορίθμων
- μια προσέγγιση χαρακτηρισμού χρόνων τρεξίματος και για τους αναδρομικούς αλγόριθμους [9]

1.7.4 Ορισμός ανάλυσης αλγορίθμων :πολυπλοκότητα χώρου και χρόνου

Ανάλυση αλγορίθμου είναι η εύρεση των πόρων οι οποίοι είναι αναγκαίοι για να τρέξει. Αυτό πολύ απλά σημαίνει ότι η ανάλυση σχετίζεται με τον χρόνο περατώσεώς του και τον χώρο ο οποίος είναι αναγκαίος και μετριέται σε αποθηκευτικές θέσεις.

Ανάλυση πολυπλοκότητας ορίζεται ως η τεχνική η οποία χαρακτηρίζει έναν αλγόριθμο αναφορικά με το μέγεθος των δεδομένων εισόδου, ανεξάρτητα από τη μηχανή, τη γλώσσα προγραμματισμού ή τον μεταγλωττιστή.

Χρησιμοποιείται για τις διαφοροποιήσεις του χρόνου εκτέλεσης σε διαφορετικούς αλγόριθμους.

Οι δύο δείκτες μετρήσεως της αποτελεσματικότητας ενός οποιουδήποτε αλγόριθμου συνιστούν την πολυπλοκότητα χρόνου και χώρου του (time and space complexity).

Η ανάλυση πολυπλοκότητας είναι αναγκαία διότι:

- Ορίζει την χρονική διάρκεια και τους πόρους που είναι απαραίτητοι για την εκτέλεση του κάθε αλγόριθμου.
- Χρησιμοποιείται για την σύγκριση διαφορετικών αλγορίθμων με διαφορετικά μεγέθη δεδομένων εισόδου.
- Βοηθάει στον ορισμό της δυσκολίας του προβλήματος
- Συχνά μετράει πόσο χρόνο και χώρο μνήμης χρειάζεται για να λυθεί ένα πρόβλημα.

Η ανάλυση των αλγορίθμων αποτελεί ένα μέσο παροχής μέτρου συγκρίσεως για την επιλογή του καλύτερου αλγόριθμου βάσει των απαιτήσεών μας. Μπορεί κάποιος να ενδιαφέρεται να περατωθεί ο αλγόριθμος γρήγορα και να μην τον νοιάζει το κόστος σε χώρο και σε αυτή την περίπτωση ο χρόνος είναι το μόνο κριτήριο.

Εάν κάποιος όμως επιθυμεί τη δέσμευση όσο το δυνατόν λιγότερου χώρου για την εκτέλεση του αλγορίθμου του, τότε θα πρέπει να ενεργήσει ανάλογα.

Η απλούστερη ανάλυση που μπορεί να γίνει είναι να υλοποιηθεί ο αλγόριθμος και στην συνέχεια να μετρηθεί ο χρόνος τρεξίματός του καθώς και οι απαιτήσεις του σε μνήμη.

Κατά κανόνα πριν την υλοποίηση ενός αλγόριθμου προηγείται μία εκτίμηση κατά τη φάση του σχεδιασμού του. Η εκτίμηση αυτή είναι δυνατό να γίνει υιοθετώντας ένα κατάλληλο μοντέλο υπολογισμού το οποίο αποτελεί την αφαιρετική θεώρηση του υλικού που έχει στη διάθεσή του [2]

1.7.5 Μοντέλα υπολογισμού

Στα πλαίσια της θεωρητικής μελέτης των αλγορίθμων και της κατάταξή τους σε τάξεις πολυπλοκότητας εφαρμόζουμε το μοντέλο της ντετερμινιστικής μηχανής Turing.

Εάν όμως θέλουμε να περιγράψουμε και να αναλύσουμε έναν αλγόριθμο χρησιμοποιούμε το μοντέλο RAM.

1.7.5.1 Ντετερμινιστική Μηχανή Turing (Deterministic Turing Machine)

Το μοντέλο αυτό πήρε το όνομά του από τον ίδιο τον δημιουργό του, τον γνωστό μαθηματικό Alan Turing.Αποτελείται από μία συσκευή πεπερασμένου ελέγχου, μία ταινία με αριστερό όριο που είναι εκτεινόμενη απείρως προς τα δεξιά και μία κεφαλή ανάγνωσης /εγγραφής της ταινίας.

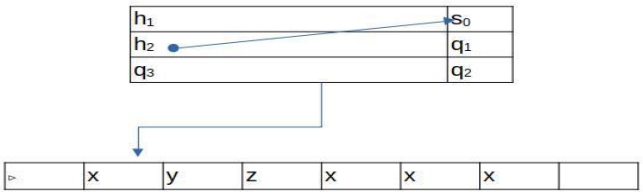
Μία ντετερμινιστική μηχανή Turing M (deterministic Turing Machine) ορίζεται ως εξής:

$$M=(S,\Sigma,\delta,s_0,H)$$

(1.10)

Η εν λόγω συνάρτηση τηρεί τους ακόλουθους περιορισμούς:

- Εάν δ(q,⊳)= (p,b), q ∈ (S - H) τότε b=⇾
- Εάν δ(q,a)= (p,b), q ∈ (S - H) a∈ Σ τότε b≠⊳



Εικόνα 1.1 Στιγμιότυπο ντετερμινιστικής μηχανής Turing

Η είσοδος εγγράφεται στην ταινία η οποία οριοθετείται από τα αριστερά και από δεξιά με άπειρο πλήθος από κενούς χαρακτήρες. Η μηχανή ξεκινάει από την αρχική κατάσταση s_0 με την κεφαλή της τοποθετημένη στο αριστερότερο κελί της ταινίας. Σε κάθε υπολογιστικό βήμα η μηχανή διαβάζει το σύμβολο a του τρέχοντος κελιού της ταινίας το οποίο προσπελαύνει η κεφαλή αναγνώσεως/εγγραφής και, σύμφωνα με την τρέχουσα κατάσταση q, υπολογίζεται η δ δηλαδή έχουμε δ(q,a)=(p,b).

Αυτός ο υπολογισμός υποχρεώνει τη μηχανή να μεταβεί από την κατάσταση q στην κατάσταση p και είτε να εγγράψει το b ∈Σ στο τρέχον κελί διαγράφοντας κατά συνέπεια το προηγούμενο περιεχόμενό του είτε να μετακινήσει την κεφαλή δεξιά στην περίπτωση που ισχύει b=⇾ ή αριστερά εάν b=⇽

Σε περίπτωση που το τρέχον σύμβολο είναι το όριο, δηλαδή το ⊳ η κεφαλή πάει αναγκαστικά προς τα δεξιά. Με αυτό τον τρόπο αποφεύγεται η διαγραφή του ειδικού συμβόλου ή η μετακίνηση της κεφαλής εκτός του αριστερού άκρου της ταινίας.

Όταν έχουμε p∈ H τότε η μηχανή μεταβαίνει σε τερματική κατάσταση.

Σε αυτό το σημείο θα πρέπει να αναφερθεί η θέση Church Turing σύμφωνα με την οποία :Μία μηχανή Turing η οποία τερματίζει σε όλες τις εισόδους αποτελεί την ακριβή, τυπική έννοια που ισοδυναμεί με ό,τι αντιστοιχεί η διαισθητική έννοια του "αλγορίθμου".

1.7.5.2 Το μοντέλο Random Access Machine (RAM)

Αν θέλουμε να αναλύσουμε έναν αλγόριθμο χωρίς να ασχοληθούμε με πειράματα που να σχετίζονται με τον χρόνο τρεξίματός του σε διαφορετικά συστήματα μπορούμε να ασπαστούμε την ακόλουθη προσέγγιση η οποία εφαρμόζεται κατευθείαν στον κώδικα υψηλού επιπέδου ή στον ψευδοκώδικα.

Το συγκεκριμένο μοντέλο που δεν πρέπει να το μπερδεύουμε με το "random access memory," αναφέρεται σε συστήματα του ενός επεξεργαστή γενικού σκοπού στο οποίο δεν υπάρχει δυνατότητα τέλεσης ταυτόχρονων πράξεων. Συγκεκριμένα:

- στο σύστημα υπάρχουν οι αναγκαίοι καταχωρητές, ένας συσσωρευτής καθώς και μία ακολουθία αποθηκευτικών θέσεων με διευθύνσεις 0,1,2.. οι οποίες αποτελούν την κύρια μνήμη του
- στο σύστημα υπάρχει η δυνατότητα εκτέλεσης αριθμητικών πράξεων { +,-,*,/,mod } λήψης αποφάσεων διακλαδώσεως τύπου if έχοντας ως βάση τους τελεστές { =,<,>,≤,≥, ≠ } και ανάγνωσης/γραφής από/προς τις θέσεις μνήμης.

Οι στοιχειώδεις πράξεις (primitive operations) είναι σε μεγάλο βαθμό ανεξάρτητες από γλώσσες προγραμματισμού και μπορούν να υλοποιηθούν και σε ψευδοκώδικα.

Ας τις αναφέρουμε λίγο πιο αναλυτικά :

- ανάθεση τιμής σε μια μεταβλητή
- κλήση μεθόδου
- υλοποίηση μίας αριθμητικής πράξης (το επισημάναμε πιο πάνω)
- σύγκριση δύο αριθμών (το επισημάναμε πιο πάνω)
- εισαγωγή στοιχείων σε ένα πίνακα
- υλοποίηση μιας αναφοράς σε κείμενο
- επιστροφή τιμής μια μεθόδου

Μία οποιαδήποτε στοιχειώδης πράξη ανταποκρίνεται σε οδηγίες χαμηλού επιπέδου.

Επίσης οι στοιχειώδεις πράξεις χρεώνονται κάποιο χρόνο και οι θεωρήσεις επί του θέματος είναι οι εξής :

- Μέτρηση μοναδιαίου κόστους (unit cost measure) :σε αυτή την περίπτωση κάθε πράξη χρεώνεται σταθερό (πεπερασμένο) κόστος κάτι το οποίο δεν εξαρτάται από το μήκος της δυαδικής αναπαράστασης των τελεστών (operands).
- Μέτρηση λογαριθμικού κόστους(logarithmic cost measure):σε αυτή την περίπτωση η πράξη διαρκεί χρονικά όσο το μήκος της δυαδικής αναπαράστασης των τελεστών. Για παράδειγμα η μετακίνηση του αριθμού n από την κύρια μνήμη προς έναν καταχωρητή χρεώνεται $[\log_n]+1$ μονάδες χρόνου.

Συνήθως χρησιμοποιείται η μέτρηση μοναδιαίου κόστους εκτός αν γίνεται εκτεταμένη χρήση πράξεων επί συμβολοσειρών bit.

Αν χρησιμοποιήσουμε ως κριτήριο την υπολογιστική ισχύ η μηχανή RAM είναι ισοδύναμη με τη μηχανή Turing σύμφωνα με το Θεώρημα ισοδυναμίας RAM και μηχανής Turing στο οποίο υποστηρίζεται ότι κάθε υπολογιστική διαδικασία μίας μηχανής Turing με κόστος πολυωνυμικό στο μέγεθος της εισόδου μπορεί να

εξομοιωθεί από μία διαδικασία μίας μηχανής RAM η οποία είναι υπολογιστική, πολυωνυμική στο μέγεθος της εισόδου και αντίστροφα.

Αντί να προσπαθούμε λοιπόν να ορίσουμε τον ακριβή χρόνο εκτέλεσης κάθε στοιχειώδους πράξης απλά μετρούμε πόσες στοιχειώδεις πράξεις εκτελούνται και χρησιμοποιούμε αυτόν τον αριθμό έστω t ως εκτίμηση υψηλού επιπέδου για τον χρόνο τρεξίματος του αλγορίθμου

Αυτή η ιδιότυπη μέτρηση θα συσχετιστεί με έναν πραγματικό χρόνο τρεξίματος σε συγκεκριμένο περιβάλλον λογισμικού και υλικού διότι κάθε στοιχειώδης πράξη ανταποκρίνεται σε μία εντολή συνεχούς χρόνου και υπάρχουν μόνο βασικές λειτουργίες προκαθορισμένου χρόνου.

Η υπόθεση που κατά κάποιον τρόπο υπονοείται σε αυτού του τύπου την προσέγγιση είναι ότι οι χρόνοι τρεξίματος διαφορετικών στοιχειωδών πράξεων θα είναι σχεδόν παρόμοιοι.

Συνεπώς ο αριθμός t στοιχειωδών πράξεων των θα είναι αναλογικός με τον πραγματικό χρόνο τρεξίματος εκείνου του αλγορίθμου.

Κάθε καταχωρητής μνήμης έχει αποθηκευμένη μία λέξη η οποία μπορεί να είναι αριθμός, μία συμβολοσειρά η μία διεύθυνση δηλαδή τη βάση ενός βασικού τύπου. Ο όρος “τυχαία πρόσβαση” (random access) αναφέρεται στην ικανότητα της κεντρικής μονάδας επεξεργασίας να έχει πρόσβαση σε έναν αυθαίρετο καταχωρητή μνήμης με μία στοιχειώδη πράξη.

Υποθέτουμε ότι η κεντρική μονάδα επεξεργασίας στο μοντέλο Random Access Machine μπορεί να κάνει όλες τις στοιχειώδεις πράξεις σε ένα συνεχές αριθμό βημάτων και δεν υπάρχει εξάρτηση από το μέγεθος των δεδομένων εισόδου.

Συνεπώς, ένας ακριβές όριο στον αριθμό των στοιχειωδών πράξεων που μπορεί να κάνει ένας αλγόριθμος ανταποκρίνεται ευθέως στον χρόνο τρεξίματος αυτού του αλγόριθμου στο μοντέλο Random Access Machine[2]

1.7.5.2 Ανάλυση με βάση το μοντέλο RAM

Σε γενικό πλαίσιο, η ανάλυση των αλγορίθμων γίνεται στο μοντέλο RAM βάσει του μοναδιαίου μέτρου πάντα σε συνάρτηση με το μέγεθος των δεδομένων της εισόδου.

Το μέγεθος των δεδομένων εισόδου φυσικά εξαρτάται από το πρόβλημα που καλούμαστε να επιλύσουμε.

Αν θα πρέπει να ασχοληθούμε με έναν αλγόριθμο ταξινόμησης ως μέγεθος δεδομένων εισόδου θεωρούμε το πλήθος n των αντικειμένων προς ταξινόμηση.

Αν τώρα έχουμε ένα αλγόριθμο ο οποίος υπολογίζει ένα ελάχιστο επικαλύπτον δέντρο σε ένα γράφημα G=(V,E) με σύνολο κορυφών V και σύνολο ακμών E είσοδος έχει μέγεθος n=|V|+|E|.

Στην περίπτωση που αλγόριθμός μας έχει ως αντικείμενο την τέλεση πράξεων μεταξύ αριθμών, τότε ως μέγεθος δεδομένων εισόδου πρέπει να θεωρηθεί το συνολικό μήκος σε αριθμό bit της δυαδικής αναπαραστάσεώς τους.

Συνεπώς ο χρόνος και ο χώρος ενός αλγόριθμου για μία συγκεκριμένη είσοδο ονομάζεται αντίστοιχα ο αριθμός των στοιχειωδών πράξεων ή βημάτων που εκτελούνται και ο αριθμός των θέσεων μνήμης που απαιτούνται για την υλοποίησή του στο μοντέλο RAM.

Η απαίτηση που μόλις διατυπώθηκε ικανοποιείται ως εξής:

- σε κάθε εντολή που έχει να κάνει είτε με ανάθεση τιμής σε μία μεταβλητή ή με απλή δήλωση μίας μεταβλητής είτε σε λογική είτε σε αριθμητική πράξη θα χρεώνουμε σταθερό χρόνο
- οι βρόχοι for και while θα χρεώνονται το πλήθος των επαναλήψεων επί το πλήθος των εντολών που εκτελούνται σε κάθε επανάληψη
- Η δέσμευση μεταβλητής απλού τύπου κοστίζει σταθερό O(1) χώρο ενώ η δέσμευση ενός πίνακα έστω k θέσεων κοστίζει χώρο και χρόνο O(k)

Αρκετές φορές θα εντοπίσουμε τις λεγόμενες κυρίαρχες πράξεις (dominant operations) του αλγορίθμου οι οποίες παίζουν εξέχοντα ρόλο στην συμπεριφορά του.

Ας πάρουμε για παράδειγμα τους αλγορίθμους ταξινόμησης στους οποίους οι κυρίαρχες πράξεις είναι οι συγκρίσεις και οι ανταλλαγές στοιχείων μεταξύ δύο θέσεων του πίνακα εισόδου.

Συμπερασματικά, ο χρόνος μετράται σε πλήθος συγκρίσεων και ανταλλαγών που λαμβάνουν χώρα.

Για να κατανοήσουμε αυτά που παραθέσαμε έως τώρα θα αναλύσουμε έναν αλγόριθμο που αναπτύξαμε λίγο πιο πάνω, τον arrayMax(A, n)

Πίνακας 1.6 Μέτρηση στοιχειωδών πράξεων και υπολογισμός κόστους του αλγόριθμου arrayMax

ΜΕΤΡΗΣΗ ΣΤΟΙΧΕΙΩΔΩΝ ΠΡΑΞΕΩΝ /ΚΟΣΤΟΣ	
Τρέχον_μέγιστο _στοιχείο ← A[0]	Η αρχικοποίηση της μεταβλητής Τρέχον_μέγιστο _στοιχείο ← A[0] αντιστοιχεί σε δύο στοιχειώδεις πράξεις :η πρώτη είναι η τοποθέτηση μίας μεταβλητής στη θέση μηδέν και η δεύτερη η ανάθεση αυτής της τιμής σε μία μεταβλητή. Έτσι οι στοιχειώδεις πράξεις μας είναι **2**
Για i ← 1	Είμαστε στην αρχή του βρόχου μας και ο μετρητής αρχικοποιείται στην τιμή 1 και έχουμε μία στοιχειώδη πράξη. Συνεπώς ο αριθμός των στοιχειωδών πράξεων είναι **1**
i < n	Πριν το βρόχο επαληθεύεται η εν λόγω συνθήκη δηλαδή το ότι το πλήθος των στοιχείων του πίνακα είναι μεγαλύτερα από τον μετρητή. Αυτό είναι μία στοιχειώδης πράξη (σύγκριση δύο αριθμών).Αφού

	όμως ο μετρητής αρχίσει από το 1 και αυξάνεται κατά μία μονάδα μέχρι το τέλος του βρόχου η σύγκριση γίνεται n φορές. Συνεπώς ο αριθμός των στοιχειωδών πράξεων είναι **n**
n-1	Ο κώδικας εντός του βρόχου εκτελείται n-1 φορές για τιμές 1,2,3 έως n-1. Σε κάθε επανάληψη το στοιχείο A[i] του πίνακά μας συγκρίνεται με το Τρέχον_μέγιστο _στοιχείο και έχουμε δύο στοιχειώσεις πράξεις, την σύγκριση δύο αριθμών και την τοποθέτηση ενός αριθμού σε μία θέση. Μέχρι εδώ λοιπόν οι στοιχειώδεις πράξεις μας είναι **2.** Μπορεί όμως το στοιχείο A[i] να ανατεθεί στο Τρέχον_μέγιστο _στοιχείο συνεπώς έχουμε άλλες δύο στοιχειώδεις πράξεις (τοποθέτηση σε θέση του πίνακα και ανάθεση τιμής σε μεταβλητή) άρα πιθανώς θα έχουμε άλλες **2** τέτοιες. Τέλος ο μετρητής αυξάνεται κατά μία μονάδα η οποία προστίθεται στο σύνολο των μονάδων του. Άρα έχουμε ακόμα **2** στοιχειώδεις πράξεις. Συνεπώς στο σώμα του βρόχου εκτελούνται 4 ή 6 στοιχειώδεις πράξεις ανάλογα με την περίπτωση. Δηλαδή αν ισχύει A[i] ≤ Τρέχον_μέγιστο _στοιχείο τότε θα έχουμε την σύγκριση δύο αριθμών ήτοι **1** στοιχειώδη πράξη, την ταξινόμηση του πίνακα (indexing) ήτοι άλλη **1,** την αύξηση κατά μία μονάδα του μετρητή δηλαδή ακόμα **1,** και την πρόσθεση αυτής της μονάδας στο σύνολο του μετρητή κάτι το οποίο συνεπάγεται ακόμα **1** και το σύνολο των στοιχειωδών πράξεων είναι **4** Αν ισχύει A[i] > Τρέχον_μέγιστο _στοιχείο τότε έχουμε ακόμα **2 στοιχειώδεις πράξεις** οι οποίες είναι η ανάθεση τιμής σε μεταβλητή και η ταξινόμηση του πίνακα (indexing) και άρα το σύνολό μας είναι **6**
Επέστρεφε Τρέχον_μέγιστο _στοιχείο	Εδώ έχουμε **1** στοιχειώδη πράξη

Το κόστος σε χρονικές μονάδες στην καλύτερη περίπτωση είναι:

$$2+1+n+4*(n-1)+1=3+n+4n-4+1=5n \tag{1.11}$$

Το κόστος σε χρονικές μονάδες στη χειρότερη περίπτωση είναι :

$$2+1+n+6*(n-1)+1=3+n+6n-6+1=3+7n-6+1=7n-2 \tag{1.12}$$

Η καλύτερη περίπτωση είναι φυσικά η T(n)=5n η οποία συμβαίνει όταν πραγματικά το στοιχείο το οποίο βρίσκεται στη θέση A[0] του πίνακά μας είναι το μέγιστο και έτσι δεν γίνεται ποτέ ξανά ανάθεση τιμής την μεταβλητή Τρέχον_μέγιστο _στοιχείο.

Η χειρότερη περίπτωση είναι η T (n)=7n-2 όταν τα στοιχεία του πίνακα ταξινομούνται σε αύξουσα σειρά έτσι ώστε να ανατεθεί άλλη τιμή στη μεταβλητή Τρέχον_μέγιστο _στοιχείο σε κάθε επανάληψη του βρόχου. [10]

Ας δούμε την πολυπλοκότητα χρόνου ενός απλού αλγορίθμου γραμμικής αναζήτησης :

Πίνακας 1.7 Αλγόριθμος γραμμικής αναζήτησης

```
ΑΛΓΟΡΙΘΜΟΣ ΓΡΑΜΜΙΚΗΣ ΑΝΑΖΗΤΗΣΗΣ (A,x)
        Για i ← 1 έως n
        Εάν A[i] =x
                Επέστρεψε “Ναι”
                Τέλος Εάν
                Τέλος Για
        Επέστρεψε “Όχι”
```

Πίνακας 1.8 Μέτρηση στοιχειωδών πράξεων και υπολογισμός κόστους του αλγόριθμου Γραμμικής Αναζήτησης

ΜΕΤΡΗΣΗ ΣΤΟΙΧΕΙΩΔΩΝ ΠΡΑΞΕΩΝ /ΚΟΣΤΟΣ	
Για i ← 1	Είμαστε στην αρχή του βρόχου μας και ο μετρητής αρχικοποιείται στην τιμή 1 και έχουμε μία στοιχειώδη

	πράξη. Συνεπώς ο αριθμός των στοιχειωδών πράξεων είναι **1**
i < n	Αυτό είναι μία στοιχειώδης πράξη (σύγκριση δύο αριθμών).Αφού όμως ο μετρητής αρχίσει από το 1 και αυξάνεται κατά μία μονάδα μέχρι το τέλος του βρόχου η σύγκριση γίνεται n φορές. Συνεπώς ο αριθμός των στοιχειωδών πράξεων είναι **n**
A[i] =x	Σε αυτό το σημείο έχουμε μία ακόμα **(1)**στοιχειώδη πράξη η οποία είναι η σύγκριση δύο αριθμών
Επέστρεψε “Ναι”	Στην περίπτωση που ο αριθμός που αναζητάμε βρεθεί ο αλγόριθμος επιστρέφει θετικό μήνυμα και αυτό αποτελεί μία**(1)** στοιχειώδη πράξη
Επέστρεψε ““Όχι”	Στην περίπτωση που ο αριθμός που αναζητάμε δεν βρεθεί, δηλαδή αφού ελεγχθούν όλα τα στοιχεία του πίνακα του πίνακα τα οποία είναι n τον αριθμό με την βοήθεια της δομής ελέγχου Εάν αποδειχθεί ότι είναι αδύνατη η εύρεση του στοιχείου τότε το μήνυμα που επιστρέφεται από τον αλγόριθμο είναι αρνητικό και αυτό είναι μία (**1)**στοιχειώδη πράξη

Το κόστος σε χρονικές μονάδες στην καλύτερη περίπτωση είναι:

$$T(n)=1+1+1=3 \tag{1.13}$$

Έχουμε την αρχικοποίηση του μετρητή, την σύγκριση του αριθμού τον οποίο επιθυμούμε να βρεθεί και αν αυτός βρεθεί, τότε λαμβάνουμε θετικό μήνυμα.

Το κόστος σε χρονικές μονάδες στην χειρότερη περίπτωση είναι:

$$T(n)=1+n+1=2+n \tag{1.14}$$

Στη χειρότερη περίπτωση διατρέχονται όλα τα στοιχεία του πίνακά μας και εφόσον η εύρεση είναι αδύνατη επιστρέφεται αρνητικό μήνυμα.

Ας δούμε τώρα την χρονική πολυπλοκότητα του αλγόριθμου δυαδικής αναζήτησης, κλασσικό παράδειγμα της αλγοριθμικής οικογένειας διαίρει και βασίλευε. Απαραίτητη προϋπόθεση εφαρμογής αυτού του αλγορίθμου σε έναν πίνακα είναι ο πίνακας αυτός να είναι ταξινομημένος. [12]

Πίνακας1.9 Αλγόριθμος Δυαδικής αναζήτησης

```
ΑΛΓΟΡΙΘΜΟΣ ΔΥΑΔΙΚΗΣ ΑΝΑΖΗΤΗΣΗΣ (Κλειδί)
    Αρχή =1 Τέλος=n
  Ενώ Αρχή ≤Τέλος Κάνε
   Μέση=(Τέλος + Αρχή)/2
 Εάν Α[Μέση]=Κλειδί τότε
   Επέστρεψε Μεσαίο στοιχείο πίνακα
 Διαφορετικά εάν Α[Μέση]>Κλειδί τότε
  Τέλος= Μέση -1
Διαφορετικά Τέλος =Μέση+1
      Τέλος Εαν
         Τέλος Ενω
  Επέστρεψε-1
```

Ας υποθέσουμε ότι αναζητούμε κάποιον ακέραιο αριθμό "Κλειδί" σε έναν πίνακα Α στοιχείων n ο οποίος είναι ταξινομημένος. Συγκρίνουμε το "Κλειδί" με το περιεχόμενο της μεσαίας θέσης του πίνακα.

Και εδώ ο αλγόριθμος παύει να είναι βαρετός:

- εάν τα δύο στοιχεία είναι ίδια τότε ο σκοπό μας επετεύχθη
- Εαν το Κλειδί είναι μικρότερο από το μεσαίο στοιχείο, τότε καταλαβαίνουμε ότι είναι αδύνατο να βρίσκεται στον υπό πίνακα Α[Μέση ..Τέλος] και συνεχίζουμε στον υπό πίνακα Α[Αρχή..Μέση-1] και εξετάζουμε το μεσαίο στοιχείο του
- το Κλειδί είναι μεγαλύτερο του Α[Μέση] συνεπώς δεν βρίσκεται στον υπό πίνακα Α[Αρχή .. Μέση] και έτσι εξετάζουμε το μεσαίο στοιχείο του πίνακα Α[Μέση+1..Τέλος]

Πίνακας 1.10 Αναδρομική εξίσωση πολυπλοκότητας

ΑΝΑΔΡΟΜΙΚΗ ΕΞΙΣΩΣΗ ΠΟΛΥΠΛΟΚΟΤΗΤΑΣ

T(0)=0	Αρχική Συνθήκη
T(n)=1	Αν Α[Μέση]=Κλειδί
=1+T([n+1]/2 -1)	Αν Κλειδί <Α[Μέση]
=1+T(n-[(n+1)/2])	Αν Κλειδί >Α[Μέση]

Απλοποιούμε αυτή τη σχέση θεωρώντας την χειρότερη περίπτωση και πολύ απλά αγνοούμε το δεύτερο σκέλος. Θεωρούμε ότι $n=2^k-1$ για κάποιον ακέραιο k και έτσι αντικαθιστούμε και έχουμε :

$$T(n)=1+(1+T(2^{k-1}-1)) \quad (1.15)$$

Η αρχική μας συνθήκη είναι T(0)=0. Έτσι έχουμε :

$$\begin{gathered} T(n)=1+(1+T(2^{k-2}-1)) \\ 1+(1(1+T(2^{k-3}-1)))..... \\ i+T(2^{k-i}-1) \\ k+T(0)=k=\log(n+1) \end{gathered} \quad (1.16)$$

Αποδεικνύεται ότι για $n=2^{k-1}$ η πολυπλοκότητα της δυαδικής αναζήτησης είναι λογαριθμική.

Αφού λοιπόν υπολογίσουμε την πολυπλοκότητα χρόνου T_1 (n), T_2 (n) ή χώρου S_1(n), S_2(n) δυό αλγορίθμων A_1 και A_2 που επιλύουν το ίδιο πρόβλημα λέμε ότι ο πρώτος είναι γρηγορότερος από τον δεύτερο αν ισχύει T_1 (n)≤T_2 (n) ή οικονομικότερος από τον δεύτερο αν ισχύει S_1(n)≤ S_2(n).

Ας αναλύσουμε λίγο παραπάνω τους δύο τρόπους αναζήτησης των οποίων τις χρονικές πολυπλοκότητες αναλύσαμε παραπάνω.

Θεωρείται ότι η σειριακή αναζήτηση είναι ο ευκολότερος τρόπος αλλά ο λιγότερο αποδοτικός σε σύγκριση με την δυαδική.

Αυτό όμως δεν είναι απόλυτο γιατί θα πρέπει να ελέγξουμε κάποιες παραμέτρους: για παράδειγμα η σειριακή αναζήτηση μπορεί να εφαρμοστεί σε πίνακες οι οποίοι είναι αταξινόμητοι και μικρού μεγέθους δηλαδή τα στοιχεία τους θα πρέπει να είναι κατά κανόνα λιγότερα από είκοσι.

Η δυαδική αναζήτηση είναι αποδοτικότερη μόνο εάν εφαρμόζεται σε ταξινομημένο πίνακα. [13]

1.8 Ασυμπτωτικός ρυθμός Αύξησης

Όταν επιθυμούμε να πούμε κάτι σχετικά με τον χρόνο εκτέλεσης ενός αλγορίθμου για εισόδους μεγέθους n ένα πράγμα στο το οποίο στοχεύουμε, όπως δείξαμε και στην προηγούμενη παράγραφο με τους υπολογισμούς της χρονικής πολυπλοκότητας, είναι μία πρόταση όπως η εξής :για κάθε είσοδο μεγέθους n ο αλγόριθμος θα εκτελεί έστω το πολύ $1,62n^2+3,3n+8$ βήματα.

Αυτού του είδους η πρόταση μπορεί να μην μας ικανοποιεί πλήρως για τους εξής λόγους:

- Ο υπολογισμός ενός τέτοιου ακριβούς ορίου μπορεί να είναι μία κοπιώδης διαδικασία και όχι ακριβώς αυτό που επιθυμούμε.

- Επειδή ο τελικός μας στόχος είναι να αναγνωρίσουμε ευρείες κατηγορίες αλγορίθμων με παρόμοια συμπεριφορά θα θέλαμε να ταξινομήσουμε τους χρόνους εκτέλεσης με μικρότερο πεδίο αναλυτικότητας έτσι ώστε οι ομοιότητες μεταξύ των διαφόρων αλγορίθμων και προβλημάτων να εμφανίζονται με σαφήνεια.
- Οι λεπτομερείς προτάσεις σχετικά με τον αριθμό βημάτων που εκτελεί ένας αλγόριθμος είναι άνευ περιεχομένου.

Έτσι επιχειρούμε να εκφράσουμε το ρυθμό αύξησης του χρόνου εκτέλεσης ενός οποιουδήποτε αλγορίθμου και των άλλων λειτουργιών με ένα τρόπο όσο το δυνατόν ανεξάρτητο από εξωγενείς παράγοντες και όρους χαμηλής τάξης.

Θα θέλαμε να πάρουμε έναν χρόνο εκτέλεσης σαν αυτόν που αναφέρθηκε λίγο πιό πάνω δηλαδή $1,62n^2+3,3n+8$ βήματα και να πούμε ότι αυξάνεται με ρυθμό n^2 αν εξαιρέσουμε τους σταθερούς παράγοντες.

1.8.2 Ασυμπτωτικά άνω όρια

Έστω ότι έχουμε μία συνάρτηση T(n) η οποία εκφράζει την χειρότερη περίπτωση του χρόνου εκτέλεσης ενός αλγορίθμου για μία είσοδο μεγέθους n.

Εάν πάρουμε ως δεδομένη μία άλλη συνάρτηση έστω f(n) λέμε:

Η συνάρτηση T(n) είναι O(f(n)) δηλαδή λέμε ότι η συνάρτηση T(n) είναι τάξης (order of) f(n) αν για μεγάλο n η συνάρτηση φράσσεται εκ των άνω από ένα σταθερό πολλαπλάσιο της f(n).

Κάποιες φορές η παραπάνω πρόταση γράφεται και ως εξής :$T(n)=O(f(n))$

Πιο ξεκάθαρα η συνάρτηση T(n) είναι τάξης O(f(n)) εάν υπάρχουν σταθερές $c>0$ και $n_0>0$ έτσι ώστε για όλα τα $n\geq n_0$ να ισχύει το εξής :

$$T(n)\leq c*f(n) \tag{1.17}$$

Σε αυτή την περίπτωση θα ήταν ορθό να πούμε ότι η T(n) έχει ασυμπτωτικό άνω όριο την f(n). Θα πρέπει να τονίσουμε ότι για να ισχύσει ο παραπάνω ορισμός θα πρέπει να υπάρχει μία σταθερά c που να λειτουργεί για όλα τα n και η c δεν πρέπει να εξαρτάται από την n.

Ας δούμε ένα παράδειγμα του τρόπου με τον οποίο αυτός ο ορισμός μας επιτρέπει να εκφράσουμε άνω όρια σε χρόνους εκτέλεσης.

Θεωρούμε ένα αλγόριθμο του οποίου χρόνος εκτέλεσης έχει την παρακάτω μορφή :

$$T(n)=p*n^2+q*n+r \tag{1.18}$$

Οι σταθερές p,q, και r είναι θετικές. Θα θέλαμε να μπορούμε να προβάλλουμε τον ισχυρισμό ότι κάθε συνάρτηση τέτοιας μορφής είναι $O(n^2)$.

Η παραπάνω πρόταση ισχύει διότι για κάθε n≥1 έχουμε qn≤ qn2 και rn ≤rn^2. Έτσι μπορούμε να γράψουμε :

$$T(n)=p*n^2+q*n+r\leqslant p*n^2+q*n^2+r*n^2=(p+q+r)*n^2 \quad (1.19)$$

Η παραπάνω συνάρτηση ισχύει για όλα τα n≥1 και αυτή η ανισότητα είναι αυτό που απαιτεί ο ορισμός του O(.) ο οποίος είναι:

$$T(n)\leqslant c*n^2 \quad (1.20)$$

Φυσικά ισχύει c=p+q+r. Θα πρέπει να επισημάνουμε ότι ο συμβολισμός O(.) εκφράζει ένα άνω όριο και όχι τον ρυθμό αύξησης της συνάρτησης.

1.8.3 Ασυμπτωτικά κάτω όρια

Υπάρχει μία σημειογραφία για τα κάτω όρια. Αρχικά ας υποθέσουμε ότι έχουμε μόλις αποδείξει ότι ο χρόνος εκτέλεσης της συνάρτησης T(n) στη χειρότερη περίπτωση είναι $O(n^2)$. Επιθυμούμε επίσης να αποδείξουμε ότι αυτό το άνω όριο είναι το καλύτερο δυνατό. Για να αποδείξουμε ότι αυτό ισχύει θα πρέπει να εκφράσουμε την έννοια ότι για αυθαίρετα μεγάλα μεγέθη εισόδου n συνάρτηση T(n) είναι ένα σταθερό πολλαπλάσιο κάποιας συγκεκριμένης συνάρτησης f(n).Στο παράδειγμα που ήδη αναλύσαμε η συνάρτηση f(n) είναι $n^{2.}$

Έτσι λέμε ότι η T(n) είναι Ω(f(n)) κάτι το οποίο επίσης γράφεται T(n)=Ω(f(n)) εάν υπάρχουν σταθερές c>0 και n0 ≥0 έτσι ώστε για όλα τα n≥n0 θα έχουμε T(n)≥c*f(n).

Αναλογικά με την σημειογραφία O(.) μπορούμε να πούμε ότι η συνάρτηση T έχει ασυμπτωτικό κάτω όριο την f.

Και πάλι τονίζουμε ότι η σταθερά c πρέπει να είναι αμετάβλητη και ανεξάρτητη του n.

Αυτός ορισμός λειτουργεί όπως η O(.) στην προηγούμενη παράγραφο με την διαφορά ότι φράσουμε την συνάρτηση T(n) εκ των κάτω και όχι εκ των άνω. Ας επιστρέψουμε λοιπόν στην συνάρτηση που εξετάσαμε στην προηγούμενη παράγραφο:

$$T(n)=p*n^2+q*n+r \quad (1.21)$$

Και πάλι θεωρούμε ότι οι σταθερές p,q,r είναι θετικές και ας υποθέσουμε ότι T(n)=Ω(n2).

Είδαμε στην προηγούμενη παράγραφο ότι η δημιουργία του άνω ορίου περιλάμβανε την ανάπτυξη των όρων της T(n) μέχρι να πάρουν την μορφή μίας σταθεράς c η οποία πολλαπλασιάζεται με n2 τώρα πρέπει να κάνουμε το αντίθετο δηλαδή πρέπει να μειώσουμε το μέγεθος της T(n) μέχρι να μοιάζει με μία σταθερά επί n2. Για όλα τα n≥0 έχουμε:

$$T(n) = p*n^2 + q*n + r \geqslant p*n^2 \quad (1.22)$$

Η παραπάνω συνάρτηση ικανοποιεί τις απαιτήσεις του ορισμού του Ω(.) με c=p>0

1.8.4 Αυστηρά ασυμπτωτικά όρια

Αν υπάρχει η δυνατότητα να δείξουμε, όπως αποδείχτηκε, ότι ένας χρόνος εκτέλεσης T(n) είναι και ο O(f (n)) αλλά και Ω(F(n)) τότε θα έχουμε βρει το "σωστό" όριο :η T(n) αυξάνεται όπως ακριβώς και η f(n) για έναν σταθερό συντελεστή. Αυτό είναι το συμπέρασμα που μπορεί να εξαχθεί από το γεγονός ότι η T(n)=p*n2 +q*n+r είναι και $O(n^2)$ και $\Omega(n^2)$

Υπάρχει μία σημειογραφία με την οποία μπορεί να εκφραστεί η εύρεση των σωστών" ορίων :αν μία συνάρτηση T(n) είναι και O(f(n)) και Ω(f(n)) λέμε ότι είναι και Θ(f(n)).

Σε αυτή την περίπτωση, όταν δηλαδή η συνάρτηση T(n) είναι και Θ(f(n)), λέμε ότι η συνάρτηση f(n) αποτελεί αυστηρό ασυμπτωτικό όριο της συνάρτησης T(n).Έτσι η παραπάνω ανάλυση μας έδειξε ότι η T(n)=p*n2 +q*n+r είναι $\Theta(n^2)$.

Τα αυστηρά ασυμπτωτικά όρια για χρόνους εκτέλεσης χειρότερης περίπτωσης είναι χρήσιμα εργαλεία αφού χαρακτηρίζουν με ακρίβεια την απόδοση της χειρότερης περίπτωσης ενός οπουδήποτε αλγορίθμου μέχρι κάποιους σταθερούς συντελεστές.

Σύμφωνα με τον ορισμό της Θ(.) μπορεί κάποιος να λάβει τέτοια όρια με μία προσπάθεια γεφύρωσης του χάσματος ανάμεσα σε ένα άνω όριο και ένα κάτω όριο.

Ίσως τύχει να διαβάσουμε μία πρόταση της παρακάτω μορφής: "Έχει αποδειχθεί ένα άνω όριο $O(n^3)$ για το χρόνο εκτέλεσης του αλγορίθμου στη χειρότερη περίπτωση αλλά δεν υπάρχει γνωστό παράδειγμα για το οποίο ο αλγόριθμος θα εκτελείται σε περισσότερα από $\Omega(n^2)$".Μία τέτοιου είδους πρόταση αποτελεί παρότρυνση εύρεσης ενός αυστηρότατου ασυμπτωτικού ορίου για το χρόνο εκτέλεσης χειρότερης περίπτωσης ενός αλγόριθμου.

Μπορούμε να πάρουμε άμεσα ένα αυστηρό ασυμπτωτικό όριο υπολογίζοντας το όριο καθώς το n τείνει στο άπειρο. Ουσιαστικά αν ο λόγος των συναρτήσεων f(n) και g(n) συγκλίνει προς μία σταθερά καθώς το n τείνει προς το άπειρο τότε f(n)=Θ(g(n)).

Έστω δύο συναρτήσεις για τις οποίες το όριο

$$\lim n \to \infty \frac{f(n)}{g(n)} \quad (1.23)$$

υπάρχει και είναι ίσο με κάποιον αριθμό c>0.Τότε θα ισχύει f(n)=Θ(g(n)).

Για να δείξουμε ότι f(n)=O(g(n)) και f(n)=Ω(g(n)) ;όπως απαιτείται από τον ορισμό του Θ(.) θα χρησιμοποιήσουμε το γεγονός ότι το όριο υπάρχει και είναι θετικό. Επειδή έχουμε

$$\lim n\to\infty \frac{f(n)}{g(n)}=c>0 \qquad (1.24)$$

από τον ορισμό του ορίου συνεπάγεται ότι υπάρχει κάποιο n_0 πέρα από το οποίο το όριο είναι πάντοτε μεταξύ ½ c και 2c.Έτσι έχουμε

$$f(n)\leqslant 2cg(n) \qquad (1.25)$$

για όλα τα $n>n_0$ κάτι που σημαίνει ότι f(n)=O(g(n)) και

$$f(n)\geq \frac{1}{2}*c*g(n) \qquad (1.26)$$

για όλα τα $n>n_0$ που σημαίνει f(n)=Ω(g(n)).[14]

Επίλογος

Στο πρώτο κεφάλαιο ορίσαμε την έννοια της λέξης "αλγόριθμος" και αμέσως μετά κάναμε μία ιστορική αναδρομή προσπαθώντας να παρουσιάσουμε την εξέλιξή τους μέσα στον χρόνο.

Αναφερθήκαμε στην αυτοματοποίηση των αλγορίθμων, παρουσιάσαμε τους τρόπους περιγραφής τους, τα χαρακτηριστικά τους και τις εφαρμογές τους. Παραθέσαμε τους τύπους των αλγορίθμων και εξηγήσαμε τον

τρόπο απόδειξης της ορθότητάς τους. Εξηγήσαμε την σημασία της πολυπλοκότητας χώρου και χρόνου ενός αλγόριθμου με αναφορά στο μοντέλο RAM.
Τέλος αναλύσαμε τον όρο “ασυμπτωτικός ρυθμός αύξησης” ο οποίος σχετίζεται με το χρόνο εκτέλεσης ενός αλγόριθμου.

Κεφάλαιο 2
Εισαγωγή

Στο δεύτερο κεφάλαιο ξεκαθαρίζουμε τις έννοιες των λέξεων κρυπτογραφία, κρυπτολογία και κρυπτανάλυση. Απαραίτητο είναι να γίνει μία ιστορική αναδρομή η οποία ξεκινά από την Αίγυπτο και συνεχίζει στην Κίνα, στην Ινδία παραθέτοντας ένα μικρό κρυπτογραφικό παράδειγμα αλφαβητικής αντικατάστασης. Επίσης παραθέτουμε ένα παράδειγμα κρυπτογραφίας στα πλαίσια της αναφοράς μας στη Μεσοποταμία. Επιπλέον αναφερόμαστε στην Ελλάδα και υλοποιούμε ένα παράδειγμα με χρήση του Τετραγώνου του Πολύβιου. Στη συνέχεια υλοποιούμε ένα κρυπτογραφικό παράδειγμα, αυτή τη φορά με βάση το Κρυπτογράφημα του Καίσαρα. Αναφερόμαστε στον Αραβικό κόσμο, στα μεσαιωνικά χρόνια και βλέπουμε ένα παράδειγμα με τον πίνακα του Ιωάννη Τριθέμιου. Ένας σημαντικός σταθμός αποτελεί ο δέκατος ένατος αιώνας και η αναφορά μας στο κρυπτογράφημα Playfare. Ιδιαίτερη θέση σε αυτή την αναδρομή έχει η αποκρυπτογράφηση της μηχανής Enigma γεγονός το οποίο επηρέασε σημαντικά την έκβαση του Δευτέρου Παγκοσμίου Πολέμου.
Αναφερόμαστε στις αρχές της μοντέρνας κρυπτογραφίας, στους στόχους της, στις εφαρμογές της και στα είδη κρυπτογραφικών αλγορίθμων. Τέλος αναφερόμαστε περιληπτικά στην κρυπτογραφία μυστικού κλειδιού, δηλαδή τη συμμετρική κρυπτογραφία και στη κρυπτογραφία δημόσιου κλειδού, δηλαδή την ασύμμετρη κρυπτογραφία.

2. Κρυπτογραφία

2.1 Ορισμοί

Πριν αναφερθούμε στην κρυπτογραφία θα ορίσουμε την έννοια της κρυπτολογίας. Η κρυπτολογία είναι ένας κλάδος ο οποίος ασχολείται με ζητήματα ασφάλειας των επικοινωνιών έχει μία ιστορία χιλιάδων ετών κάτι που ισχύει για τους περισσότερους τρόπους επικοινωνίας. Εμφανίστηκε ταυτόχρονα με τους πιο απλούς τρόπους κρυπτογράφησης οι οποίο βασίζονται σε μία απλή αντικατάσταση των συμβόλων του μηνύματος προς αναμετάδοση και συνεχίζεται μέχρι και σήμερα με την ανάπτυξη πληθώρας πολύπλοκων αλγορίθμων κρυπτογράφησης αλλά και σύνθετων πρωτοκόλλων τα οποία στηρίζονται στην απόκρυψη πληροφορίας.

Κατά τις τελευταίες δεκαετίες η ραγδαία ανάπτυξη των τηλεπικοινωνιών έχει αναγάγει την διασφάλιση του απορρήτου των επικοινωνιών σε μέγιστο ζήτημα φέρνοντας στο προσκήνιο την κρυπτολογία και καθιστώντας την πρωταγωνίστρια των τεχνολογικών εξελίξεων.

Η κρυπτολογία πλέον είναι μία επιστήμη με έντονη ερευνητική δραστηριότητα. Τυπικά με τον όρο κρυπτολογία αναφερόμαστε τόσο στην κρυπτανάλυση όσο και στην κρυπτογραφία. Θα μπορούσαμε να πούμε ότι ο όρος κρυπτολογία απορρίφθηκε από τον όρο κρυπτογραφία **[15]**

Σύμφωνα με το Συνοπτικό Λεξικό της Αγγλικής Γλώσσας της Οξφόρδης (Concise Oxford English Dictionary) η κρυπτογραφία ορίζεται ως η τέχνη ανάπτυξης και της ερμηνείας ή πιο απλά ως τέχνη ανάπτυξης και λύσης κωδίκων (the art of writing or solving codes).

Αυτός ο ορισμός είναι ιστορικά ακριβής αλλά δεν συλλαμβάνει το σύγχρονο εύρος του τομέα της κρυπτογραφίας. Είναι φανερό ότι ο ορισμός επικεντρώνεται αποκλειστικά στους κώδικες οι οποίοι χρησιμοποιούνται εδώ και αιώνες για την επίτευξη μυστικής επικοινωνίας.

Αλλά η κρυποτογραφία στις μέρες μας περικλείει πολλά περισσότερα αφού σχετίζεται με μηχανισμούς διασφάλισης ακεραιότητας, τεχνικές ανταλλαγής μυστικών κλειδιών, πρωτόκολλα για αυθεντικοποίηση χρηστών, ηλεκτρονικές δημοπρασίες, εκλογές και πολλά άλλα.

Θα μπορούσαμε λοιπόν να πούμε ότι η μοντέρνα κρυπτογραφία σχετίζεται με τη μελέτη μαθηματικών τεχνικών για την ασφάλεια ψηφιακών πληροφοριών. συστημάτων και κατανεμημένων υπολογιστικών συστημάτων απέναντι σε κακόβουλες επιθέσεις.

Η κρυπτογραφία, όπως είδαμε παραπάνω, ορίζεται ως τέχνη. Πραγματικά έως σχεδόν τα τέλη του εικοστού αιώνα η κρυπτογραφία αποτελούσε ένα είδος τέχνης.

Η ανάπτυξη κωδίκων ή το σπάσιμο των ήδη υπαρχόντων βασιζόταν κατά πολύ στη δημιουργικότητα και σε μία αίσθηση που είχαν κάποια άτομα για το πως λειτουργούσε ο κώδικας που τους ενδιέφερε.

Δεν υπήρχε θεωρητικό υπόβαθρο και ούτε κάποιος ορισμός σχετικά με το ποιος θεωρείται καλός κώδικας.

Στις δεκαετίες 1970 και 1980 αυτή η εικόνα άλλαξε δραστικά.

Ένα πλούσιο θεωρητικό πλαίσιο άρχισε να αναδύεται και έτσι κατέστη δυνατή η εκμάθηση της κρυπτογραφίας με επιστημονικό και μαθηματικό υπόβαθρο.

Αυτή η καινούρια προοπτική, στην συνέχεια επηρέασε τον τρόπο σκέψης των ερευνητών σχετικά με τον τομέα της ασφάλειας των ηλεκτρονικών υπολογιστών.

Μία άλλη διαφορά μεταξύ της κλασσικής και μοντέρνας κρυπτογραφίας είναι οι οργανισμοί οι οποίοι την υιοθέτησαν.

Ιστορικά οι οργανισμοί θεωρούσαν την κρυπτογραφία εργαλείο ζωτικής σημασίας ήταν οι στρατιωτικοί οργανισμοί και κυβερνήσεις κρατών αφού έτσι ήταν δυνατή η κωδικοποίηση ευαίσθητων πληροφοριών οι οποίες μπορούσαν να διαβαστούν μόνο από τον παραλήπτη τους.

Σήμερα η κρυπτογραφία κυριαρχεί παντού. Αν ποτέ χρειάστηκε να πληκτρολογήσετε ένα συνθηματικό (password), να κάνετε μία διαδικτυακή αγορά με τη χρήση της πιστωτικής σας κάρτας ή να κατεβάσετε ένα πιστοποιημένο λογισμικό ενημέρωσης για το λογισμικό σας πέρα από κάθε αμφιβολία χρησιμοποιήσατε κρυπτογραφία.

Όλο και περισσότεροι προγραμματιστές με σχετικά λίγη εμπειρία αναλαμβάνουν το έργο της ασφάλειας των εφαρμογών που αναπτύσσουν ενσωματώνοντας μηχανισμούς κρυπτογραφίας. [16]

Η κρυπτανάλυση αποτελεί κλάδο της κρυπτογραφίας και αντικείμενό της είναι η αποκωδικοποίηση κάποιας κρυπτογραφικής τεχνικής της οποίας το κλειδί δεν είναι γνωστό στο άτομο που προσπαθεί να φέρει αυτό το έργο εις πέρας. Αυτό το άτομο είτε είναι κάποιος ο οποίος επιθυμεί να εντοπίσει κενά ασφαλείας σε κάποιο σύστημα επειδή του ανατέθηκε αυτή εργασία ή κάποιος “εισβολέας”.

Αναλυτικότερα η κρυπτανάλυση προσπαθεί να εντοπίσει προβλήματα στους κρυπτογραφικούς αλγόριθμους που χρησιμοποιούνται.

Θα πρέπει να ξεκαθαρίσουμε ότι η κρυπτανάλυση δεν σχετίζεται μόνο με τους αλγόριθμους κρυπτογράφησης αλλά και με το σύνολο των μηχανισμών της κρυπτογραφίας όπως, για παράδειγμα τις συναρτήσεις κατακερματισμού, τις ψηφιακές υπογραφές κλπ.

Για παράδειγμα η κρυπτανάλυση μίας συνάρτησης κατακερματισμού H έχει ως στόχο την εύρεση δύο τιμών x και y με $x \neq y$ και H (x) = H (y).Στην περίπτωση μίας ψηφιακής υπογραφής η κρυπτανάλυση θα στόχευε στην πλαστογράφησή της, στην έκδοση μίας έγκυρης ψηφιακής υπογραφής άλλου ή ακόμα και να αλλοιώσει ένα ψηφιακά υπογεγραμμένο έγγραφο.

Η κρυπτανάλυση έχει και ως στόχο τον εντοπισμό των αδύναμων σημείων των κρυπταναλυτικών τεχνικών των αλγορίθμων αφού βέβαια αυτοί μελετηθούν προσεχτικά. Αδύναμα σημεία των τεχνικών κρυπνανάλυσης των αλγορίθμων είναι πολύ απλά οι τρόποι “σπασίματος” αυτών. ο εντοπισμός των οποίων μπορεί να οδηγήσει στην ανάπτυξη ασφαλέστερων δομών και τεχνικών έτσι ώστε να αναπτυχθούν ασφαλέστεροι αλγόριθμοι.

Αναφορικά με την ασφάλεια αν κάποιος γνωρίζει ότι ένας αλγόριθμος είναι ευπαθής να μην τον χρησιμοποιεί. [16] [17]

2.2 ιστορική Αναδρομή

Η κρυπτογραφία που ετυμολογικά προέρχεται από τις ελληνικές λέξεις “κρυπτός” και “γράφω” είναι ένα από τα παλαιότερα πεδία τεχνολογικής μελέτης για το οποίο υπάρχουν καταγραφές πριν από 4000 χρόνια και μπορεί εύκολα να χαρακτηριστεί ως η αρχαιότερη μέθοδος επικοινωνίας μεταξύ πολιτισμών. [18] [19]

2.2.1 Αίγυπτος

Η κρυπτογραφία συναντάται στην αρχαία Αίγυπτο περίπου στα 2000 π.Χ. και συγκεκριμένα στα αιγυπτιακά ιερογλυφικά τα οποία διακοσμούσαν τους τάφους των νεκρών βασιλιάδων.

Αυτά τα ιερογλυφικά ήταν ένας τρόπος καταγραφής της ζωής του νεκρού και τόνιζαν τα επιτεύγματά του.

Τα ιερογλυφικά ήταν σκοπίμως κρυπτογραφημένα αλλά όχι τόσο έτσι ώστε να κρυφτεί το κείμενο σε απόλυτο βαθμό και εικάζεται ότι η κρυπογράφηση έγινε για να προσδώσει σπουδαιότητα σε αυτό.

Καθώς περνούσαν τα χρόνια, αυτού του είδους η γραφή γινόταν όλο και περισσότερο περίπλοκη και έτσι το ενδιαφέρον για την αποκρυπτογράφησή της μειώθηκε. [18]

2.2.2 Κίνα

Οι αρχαίοι Κινέζοι χρησιμοποιούσαν την ιδεογραφική φύση της γλώσσας τους ή αλλιώς το ιδεογραφικό σύστημα για κα κρύβουν την σημασία των λέξεων. Συγκεκριμένα, μηνύματα συχνά μετατρέπονταν σε ιδεογραφίες ή αλλιώς ιδεογράμματα (ideographs) για ιδιωτικότητα. Ωστόσο δεν υπάρχει καμία ουσιαστική καταγραφή για την χρήση της κρυπογραφίας στις πρώτες στρατιωτικές επιχειρήσεις.

Ο Τζένγκις Χαν (Genghis Khan) για παράδειγμα δεν αποδεικνύεται ότι χρησιμοποίησε την κρυπτογραφία.[18]

2.2.3 Ινδία

Η κρυπτογραφία στην Ινδία ήταν περισσότερο προηγμένη και η κυβέρνηση χρησιμοποιούσε μυστικούς κώδικες για να επικοινωνεί με ένα δίκτυο κατασκόπων οι οποίοι ήταν διασκορπισμένοι σε όλη την χώρα.

Τα πρώτα ινδικά κρυπτογραφήματα αποτελούνταν κυρίως από απλές αλφαβητικές αντικαταστάσεις οι οποίες ήταν βασισμένες στην φωνητική. Κάποια από αυτά ήταν προφορικά ή χρησιμοποιούνταν ως νοηματική γλώσσα. Ας δούμε ένα μικρό παράδειγμα:

Πίνακας 2.1 Απλό και κρυπτογραφημένο κείμενο βασισμένο στην αλφαβητική αντικατάσταση

Αρχικό κείμενο
pig latin
Κρυπτογραφημένο κείμενο
igpay atinlay

Το πρώτο σύμφωνο της πρώτης λέξης (p) το βάζουμε στο τέλος της και το ίδιο κάνουμε και με το πρώτο σύμφωνο της δεύτερης λέξης (l). Στη συνέχεια προσθέσαμε στο τέλος και των δύο λέξεων τα φωνήεντα a και y.[18]

2.2.4 Μεσοποταμία

Η κρυπτογραφική ιστορία της Μεσοποταμίας ήταν παρόμοια με εκείνη της Αιγύπτου, συγκεκριμένα η σφηνοειδής γραφή χρησιμοποιούταν για την κρυπτογράφηση κειμένων και το ίδιο ίσχυε και για τη Βαβυλώνα και για την Ασσυρία. Στη Βίβλο χρησιμοποιείται μία εβραϊκή κρυπτογραφική μέθοδος η οποία ονομάζεται “atbash”.

Σε αυτή τη μέθοδο το τελευταίο γράμμα της αλφαβήτου αντικαθίσταται από το πρώτο και το αντίστροφο.

Ας δούμε τον κρυπτογραφικό μας πίνακα:

Πίνακας 2.2 Πίνακας κρυπτογράφησης μεθόδου atbash

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

Στην πάνω σειρά έχουμε το αγγλικό αλφάβητο, όπως είναι ευκρινές, και στην κάτω σειρά έχουμε το αλφάβητο προσαρμοσμένο για κρυπτογράφηση. Ας κρυπτογραφήσουμε την λέξη “HELLO”:[18]

Πίνακας 2.3 Αρχική και κρυπτογραφημένη λέξη σύμφωνα με την μέθοδο atbash

ΑΡΧΙΚΗ ΛΕΞΗ				
H	E	L	L	O
ΚΡΥΠΤΟΓΡΑΦΗΜΕΝΗ ΛΕΞΗ				
S	V	O	O	L

Ας δούμε ακόμα ένα παράδειγμα, αυτή τη φορά αποκρυπτογράφησης :

Πίνακας 2.4 Κρυπτογραφημένη και αποκρυπτογραφημένη λέξη σύμφωνα με την μέθοδο atbash

ΚΡΥΠΤΟΓΡΑΦΗΜΕΝΗ ΛΕΞΗ						
W	V	X	I	B	K	G
ΑΠΟΚΡΥΠΤΟΓΡΑΦΗΜΕΝΗ ΛΕΞΗ						
D	E	C	R	Y	P	T

2.2.5 Ελλάδα

Στο ομηρικό έπος “Ιλιάδα” χρησιμοποιήθηκε η κρυπτογραφία όταν, σύμφωνα με την διήγηση του εγγονού του Γλαύκου ο Βελλεροφόντης στάλθηκε στον βασιλιά Ιοβάτη της Λυκίας με μία κρυπογραφημένη επιστολή που στην ουσία ήταν η θανατική του καταδίκη.

Οι Σπαρτιάτες χρησιμοποιούσαν ένα σύστημα το οποίο αποτελούταν από έναν πάπυρο ο οποίος ήταν τυλιγμένος ελικοειδώς, δηλαδή γύρω από έναν κύλινδρο κρυπτογράφησης (staff cipher).

Το μήνυμα ήταν γραμμένο κατά μήκος του κυλίνδρου πράγμα που σημαίνει ότι το μήνυμα ήταν γραμμένο σε στήλες και ο πάπυρος έπρεπε να τυλιχτεί γύρω από αυτόν. Ο αγγελιοφόρος έπαιρνε τον πάπυρο και πήγαινε το μήνυμα στον παραλήπτη.

Για να διαβαστεί το μήνυμα ο πάπυρος θα έπρεπε να τυλιχτεί σε έναν κύλινδρο ίσης διαμέτρου.

Αυτό το κρυπογράφημα ονομαζόταν κρυπογράφημα σκυτάλης (scytale cipher) και χρησιμοποιούταν στον πέμπτο αιώνα π.Χ. για την αποστολή μυστικών μηνυμάτων μεταξύ των Ελλήνων πολεμιστών.

Χωρίς τον σωστό κύλινδρο η αποκωδικοποίηση του μηνύματος θα ήταν πολύ δύσκολη με τις γνωστές τεχνικές εκείνης της εποχής. [18]

Ο δίσκος της Φαιστού είναι ένα από τα πιο φημισμένα αρχαιολογικά ευρήματα της Κρήτης. Πρόκειται για ένα δίσκο από άργιλο διαμέτρου δεκαέξι εκατοστών και πάχους δύο εκατοστών ο οποίος φέρει σημεία και στις δύο όψεις του. Η ιδιαιτερότητα του δίσκου προκύπτει από το γεγονός ότι τα διακόσια σαράντα δύο σημεία έχουν αποτυπωθεί χρησιμοποιώντας σαράντα πέντε διαφορετικές σφραγίδες.

Όλα τα σύμβολα που πρέπει να τονιστεί ότι είναι και στις δύο πλευρές είναι τοποθετημένα σπειροειδώς και χωρίζονται σε ομάδες με μικρές γραμμές οι οποίες κατευθύνονται προς το κέντρο.

Ο δίσκος της Φαιστού χρονολογείται από το 1700 π.Χ. και δεν έχει αποκρυπτογραφηθεί ακόμα [19]

Μία άλλη μέθοδος κρυπτογράφησης αναπτύχτηκε από τον Πολύβιο η οποία ονομάζεται το Τετράγωνο του Πολύβιου (Polybius Square). Τα γράμματα του αλφαβήτου τοποθετούνται σε έναν πίνακα διαστάσεων

πέντε επί πέντε με τους δείκτες i και j να καταλαμβάνουν το ίδιο τετράγωνο. Οι σειρές και οι στήλες αριθμούνται από το ένα έως το πέντε.

Η αποκρυπτογράφηση έχει να κάνει με την χαρτογράφηση των ζευγαριών των ψηφίων στους χαρακτήρες που τους αναλογούν. Ας δούμε ένα παράδειγμα: [18]

Πίνακας 2.5 Πίνακας κρυπτογράφησης σύμφωνα με το Τετράγωνο του Πολύβιου

ΤΕΤΡΑΓΩΝΟ ΠΟΛΥΒΙΟΥ				
1	**2**	**3**	**4**	**5**

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

Πίνακας 2.6 Μήνυμα προς αποκρυπτογράφηση σύμφωνα με το Τετράγωνο του Πολύβιου

ΜΗΝΥΜΑ ΠΡΟΣ ΑΠΟΚΡΥΠΤΟΓΡΑΦΗΣΗ			
54	32	42	44
ΑΠΟΚΡΥΠΤΟΓΡΑΦΗΜΕΝΟ ΜΗΝΥΜΑ			
T	H	I	S

2.2.6 Ρωμαϊκή Αυτοκρατορία

Ο Ιούλιος Καίσαρας χρησιμοποιούσε ένα σύστημα κρυπτογραφίας, το κρυπτογράφημα του Καίσαρα σύμφωνα με το τα γράμματα του αλφαβήτου μετακινούνται δύο θέσεις (για παράδειγμα το Υ μετακινείται στη θέση Α). Αυτό το κρυπτογράφημα εύκολα μπορεί να διδαχτεί και σε παιδιά σχολικής ηλικίας.

Είναι φανερό ότι έχουμε να κάνουμε με ένα κρυπτογράφημα μονοαλφαβητικής αντικατάστασης. [18]

Πίνακας 2.7 Πίνακας κρυπτογράφησης Κρυπτογραφήματος Καίσαρα

ΠΙΝΑΚΑΣ ΚΡΥΠΤΟΓΡΑΦΗΣΗΣ ΚΡΥΠΤΟΓΡΑΦΗΜΑΤΟΣ ΚΑΙΣΑΡΑ

ΑΓΓΛΙΚΟ ΑΛΦΑΒΗΤΟ																									
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

ΠΙΝΑΚΑΣ ΚΡΥΠΤΟΓΡΑΦΗΣΗΣ

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

Πίνακας 2.8 Μήνυμα προς αποκρυπτογράφηση και αποκρυπτογραφημένο μήνυμα σύμφωνα με το Κρυπτογράφημα Καίσαρα

ΜΗΝΥΜΑ ΠΡΟΣ ΑΠΟΚΡΥΠΤΟΓΡΑΦΗΣΗ			
V	J	K	U
ΑΠΟΚΡΥΠΤΟΓΡΑΦΗΜΕΝΟ ΜΗΝΥΜΑ			
T	H	I	S

2.2.7 Αραβικός κόσμος

Κρυπτανάλυση, όπως είδαμε και παραπάνω, ονομάζουμε τη διαδικασία αλλαγής του κρυπτογραφημένου κείμενου σε απλό κείμενο χωρίς την ακριβή γνώση του κρυπογραφήματος και του κλειδιού. Οι ΄Άραβες ήταν οι πρώτοι που έκαναν σημαντική πρόοδο στην κρυπτοανάλυση. [20]

Ένας Άραβας συγγραφέας, ο Qalqashandi έγραψε ένα βιβλίο στο οποίο ανέλυε μία τεχνική λύσης κρυπτογραφημάτων η οποία χρησιμοποιείται και σήμερα και έχει ως εξής: αρχικά σημειώνουμε όλα τα σύμβολα του κρυπτογραφήματος και μετράμε τη συχνότητα του καθενός από αυτά.

Χρησιμοποιούμε το μέσο όρο της συχνότητας εμφάνισης κάθε γράμματος και έτσι μπορούμε να ανακτήσουμε το αρχικό κείμενο.

Αυτή η τεχνική είναι αρκετά ισχυρή για την κρυπνανάλυση οποιουδήποτε κρυπογραφήματος το οποίο βασίστηκε σε μονοαλφαβητική κρυπογράφηση αντικατάστασης εάν έχουμε μεγάλης έκτασης κρυπτογραφημένου κείμενου στη διάθεσή μας. [18]

2.2.8 Μεσαίωνας

Κατά τη διάρκεια του Μεσαίωνα η κρυπτογραφία μπήκε σε μία εξελικτική πορεία. Όλες οι κυβερνήσεις της δυτικής Ευρώπης χρησιμοποιούσαν κάποια μορφή κρυπτογραφίας και οι κώδικες άρχισαν να κερδίζουν σε δημοτικότητα. Κατά κύριο λόγο τα κρυπτογραφήματα αποτελούσαν εργαλείο επικοινωνίας μεταξύ πρεσβευτών.

Το πρώτο μεγάλο βήμα προόδου αναφορικά με την κρυπτογραφία έγινε στην Ιταλία. Στη Βενετία το 1452 δημιουργήθηκε ένας οργανισμός που είχε ως σκοπό την ενασχόληση με την κρυπτογραφία.

Σε αυτό τον οργανισμό υπήρχαν τρεις γραμματείς κρυπτογράφησης οι οποίοι έλυναν και δημιουργούσαν κρυπτογραφήματα τα οποία χρησιμοποιούσαν οι κυβερνήσεις.

Ο Leon Battista Alberti είναι γνωστός ως ο πατέρας της δυτικής κρυπτολογίας εν μέρει λόγω του έργου του αναφορικά με την ανάπτυξη της πολυαλφαβητικής αντικατάστασης.

Πολυαλφαβητική αντικατάσταση ονομάζουμε οποιαδήποτε τεχνική η οποία επιτρέπει σε διαφορετικά σύμβολα ενός κρυπτογραφήματος να αναπαριστούν το ίδιο σύμβολο του απλού κειμένου.

Αυτή η τεχνική δυσκολεύει ακόμα περισσότερο την αποκρυπτογράφηση ενός κρυπτογραφημένου κειμένου με τη χρήση της ανάλυσης συχνότητας των συμβόλων του κρυπτογραφήματος.

Για να βελτιώσει αυτή την τεχνική ο Alberti ανέλυσε μεθόδους σπασίματος κρυπτογραφημάτων και ανέπτυξε ο ίδιος ένα κρυπτογράφημα εξουδετέρωσης αυτών των τεχνικών.

Σχεδίασε δύο χάλκινους δίσκους που εφάρμοζαν μεταξύ τους και ο καθένας από αυτούς είχε χαραγμένο το αλφάβητο επάνω του.

Για να αρχίσει η κρυπτογράφηση, ένα προκαθορισμένο γράμμα του εσωτερικού δίσκου είναι ευθυγραμμισμένο με ένα γράμμα του εξωτερικού δίσκου το οποίο αποτελεί το πρώτο γράμμα του κρυπτογραφημένου κειμένου. Οι δίσκοι παραμένουν ακίνητοι με κάθε γράμμα του απλού κειμένου στον εσωτερικό δίσκο να ευθυγραμμίζεται με ένα γράμμα του κρυπτογραφημένου κειμένου στον εξωτερικό δίσκο.

Μετά από μερικές λέξεις του κρυπτογραφημένου κειμένου οι δίσκοι περιστρέφονται έτσι ώστε το πρώτο γράμμα του εσωτερικού δίσκου να ευθυγραμμιστεί με ένα καινούργιο γράμμα του εξωτερικού δίσκου και με αυτόν τον τρόπο κρυπτογραφείται το μήνυμα. Με την περιστροφή των δίσκων ανά λίγες λέξεις το κρυπτογράφημα άλλαζε σε ικανοποιητικό βαθμό έτσι ώστε να εμποδίσει την αποτελεσματικότητα της ανάλυσης συχνότητας χαρακτήρων.

Αν και αυτή η τεχνική στην αρχική της μορφή ήταν αρκετά αδύναμη η ιδέα της περιστροφής των δίσκων και κατά συνέπεια η αλλαγή του κρυπογραφήματος πολλές φορές εντός του μηνύματος ήταν ένα τεράστιο βήμα στα πλαίσια της κρυπτογραφίας.

Το 1518 ο Ιωάννης Τριθέμιος (Johannes Trithemius)ένας Γερμανός μοναχός ενδιαφερόταν για τον αποκρυφισμό. Έγραψε μία σειρά έξι βιβλίων με τίτλο “Polygraphia” και στο πέμπτο βιβλίο δημιούργησε έναν πίνακα ο οποίος στην πρώτη γραμμή έχει το αλφάβητο και στις υπόλοιπες γραμμές τα γράμματα του αλφάβητου μετατίθενται κατά ένα γράμμα προς τα αριστερά. Παραθέτουμε παρακάτω δύο πίνακες εκ των οποίων ο πρώτος περιέχει με το αγγλικό αλφάβητο και ο δεύτερος το ελληνικό :

Ο Leon Battista Alberti είναι γνωστός ως ο πατέρας της δυτικής κρυπτολογίας εν μέρει λόγω του έργου του αναφορικά με την ανάπτυξη της πολυαλφαβητικής αντικατάστασης.

Πολυαλφαβητική αντικατάσταση ονομάζουμε οποιαδήποτε τεχνική η οποία επιτρέπει σε διαφορετικά σύμβολα ενός κρυπτογραφήματος να αναπαριστούν το ίδιο σύμβολο του απλού κειμένου.

Αυτή η τεχνική δυσκολεύει ακόμα περισσότερο την αποκρυπτογράφηση ενός κρυπτογραφημένου κειμένου με τη χρήση της ανάλυσης συχνότητας των συμβόλων του κρυπτογραφήματος.

Για να βελτιώσει αυτή την τεχνική ο Alberti ανέλυσε μεθόδους σπασίματος κρυπτογραφημάτων και ανέπτυξε ο ίδιος ένα κρυπτογράφημα εξουδετέρωσης αυτών των τεχνικών.

Σχεδίασε δύο χάλκινους δίσκους που εφάρμοζαν μεταξύ τους και ο καθένας από αυτούς είχε χαραγμένο το αλφάβητο επάνω του.

Για να αρχίσει η κρυπτογράφηση, ένα προκαθορισμένο γράμμα του εσωτερικού δίσκου είναι ευθυγραμμισμένο με ένα γράμμα του εξωτερικού δίσκου το οποίο αποτελεί το πρώτο γράμμα του κρυπτογραφημένου κειμένου. Οι δίσκοι παραμένουν ακίνητοι με κάθε γράμμα του απλού κειμένου στον εσωτερικό δίσκο να ευθυγραμμίζεται με ένα γράμμα του κρυπτογραφημένου κειμένου στον εξωτερικό δίσκο.

Μετά από μερικές λέξεις του κρυπτογραφημένου κειμένου οι δίσκοι περιστρέφονται έτσι ώστε το πρώτο γράμμα του εσωτερικού δίσκου να ευθυγραμμιστεί με ένα καινούργιο γράμμα του εξωτερικού δίσκου και με

αυτόν τον τρόπο κρυπτογραφείται το μήνυμα. Με την περιστροφή των δίσκων ανά λίγες λέξεις το κρυπτογράφημα άλλαζε σε ικανοποιητικό βαθμό έτσι ώστε να εμποδίσει την αποτελεσματικότητα της ανάλυσης συχνότητας χαρακτήρων.

Αν και αυτή η τεχνική στην αρχική της μορφή ήταν αρκετά αδύναμη η ιδέα της περιστροφής των δίσκων και κατά συνέπεια η αλλαγή του κρυπογραφήματος πολλές φορές εντός του μηνύματος ήταν ένα τεράστιο βήμα στα πλαίσια της κρυπτογραφίας.

Το 1518 ο Ιωάννης Τριθέμιος (Johannes Trithemius) ένας Γερμανός μοναχός ενδιαφερόταν για τον αποκρυφισμό. Έγραψε μία σειρά έξι βιβλίων με τίτλο “Polygraphia” και στο πέμπτο βιβλίο δημιούργησε έναν πίνακα ο οποίος στην πρώτη γραμμή έχει το αλφάβητο και στις υπόλοιπες γραμμές τα γράμματα του αλφάβητου μετατίθενται κατά ένα γράμμα προς τα αριστερά. Παραθέτουμε παρακάτω δύο πίνακες εκ των οποίων ο πρώτος περιέχει με το αγγλικό αλφάβητο και ο δεύτερος το ελληνικό :

Πίνακας 2.9 Πίνακας κρυπτογράφησης αγγλικού αλφάβητου σύμφωνα με το σύστημα κρυπτογράφησης του Ιωάννη Τριθέμιου

ΑΓΓΛΙΚΟ ΑΛΦΑΒΗΤΟ

	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	26
1	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
2	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
3	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
4	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
5	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
6	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
7	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
8	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
9	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
10	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
11	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
12	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
13	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
14	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
15	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

16	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
17	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
18	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
19	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
20	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
21	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
22	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
23	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
24	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
25	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
26	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	

Πίνακας 2.10 Πίνακας κρυπτογράφησης ελληνικού αλφάβητου σύμφωνα με το σύστημα κρυπτογράφησης του Ιωάννη Τριθέμιου

ΕΛΛΗΝΙΚΟ ΑΛΦΑΒΗΤΟ

Α	**Β**	**Γ**	**Δ**	**Ε**	**Ζ**	**Η**	**Θ**	**Ι**	**Κ**	**Λ**	**Μ**	**Ν**	**Ξ**	**Ο**	**Π**	**Ρ**	**Σ**	**Τ**	**Υ**	**Φ**	**Χ**	**Ψ**	**Ω**
Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Α
Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Α	Β
Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Α	Β	Γ
Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Α	Β	Γ	Δ
Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Α	Β	Γ	Δ	Ε
Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Α	Β	Γ	Δ	Ε	Ζ
Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Α	Β	Γ	Δ	Ε	Ζ	Η
Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ
Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι
Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ
Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ
Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ
Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν
Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ
Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο
Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π
Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ
Τ	Υ	Φ	Χ	Ψ	Ω	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ
Υ	Φ	Χ	Ψ	Ω	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ
Φ	Χ	Ψ	Ω	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ
Χ	Ψ	Ω	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ
Ψ	Ω	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ
Ω	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ

Ας πούμε λοιπόν ότι θέλουμε να κρυπτογραφήσουμε την φράση “Όλα καλά”. Το πρώτο γράμμα παραμένει το ίδιο. Πάμε στη δεύτερη σειρά και το δεύτερο από **λ** γίνεται **μ.**

Πάμε στην τρίτη σειρά και το **α** γίνεται **γ.** Συνεπώς η πρώτη λέξη είναι η **Ομγ**

Τώρα έχουμε τη δεύτερη λέξη και πάμε στην τέταρτη σειρά και το **κ** γίνεται **ν**, και στην συνέχεια πάμε στην πέμπτη σειρά και το **α** γίνεται **ε**. Αμέσως μετά πάμε στην έκτη σειρά και το **λ** γίνεται **π** και τέλος πάμε στην έβδομη σειρά και το α γίνεται η. Άρα η δεύτερη λέξη είναι η Νεπη. [18] [21]

2.2.9 Δέκατος έκτος αιώνας

Ο πιο διάσημος κρυπτογράφος του δέκατου έκτου αιώνα είναι ο Blaise de Vigenere (1523-1596).Το 1585 έγραψε το “Tracte des Chiffres” στο οποίο χρησιμοποίησε τον πίνακα του Ιωάννη Τριθέμιου που είδαμε στην προηγούμενη παράγραφό μας αλλά άλλαξε τον τρόπο που δούλευε το σύστημα κλειδιού. Ο τρόπος χρήσης των κλειδιών ονομάστηκε προγραμματισμός κλειδιών και αποτελεί τμήμα του "Data Encryption Standard" (DES).

Ας δούμε αυτή την τεχνική μέσω παραδείγματος :έστω ότι θέλουμε να κρυπτογραφήσουμε την φράση “simple message” και η λέξη-κλειδί είναι “ lucky”.

Επειδή η λέξη-κλειδί είναι μικρότερη από τη φράση που θέλουμε να κρυπτογραφήσουμε θα πρέπει να την επαναλάβουμε. Έχουμε λοιπόν:

Πίνακας 2.11 Λέξη-κλειδί πριν την επανάληψη

ΛΕΞΗ-ΚΛΕΙΔΙ ΠΡΙΝ ΤΗΝ ΕΠΑΝΑΛΗΨΗ				
L	U	C	K	Y

Πίνακας2 2.12 Κρυπτογραφημένο μήνυμα

ΛΕΞΗ-ΚΛΕΙΔΙ ΜΕΤΑ ΤΗΝ ΕΠΑΝΑΛΗΨΗ												
L	U	C	K	Y	L	U	C	K	Y	L	U	C
ΜΗΝΥΜΑ ΠΡΟΣ ΚΡΥΠΤΟΓΡΑΦΗΣΗ												
S	I	M	P	L	E	M	E	S	S	A	G	E
ΚΡΥΠΤΟΓΡΑΦΗΜΕΝΟ ΜΗΝΥΜΑ												
D	C	O	Z	J	P	G	G	C	Q	L	A	G

Για να αρχίσουμε την κρυπτογράφηση όλου του μηνύματος ενεργούμε ως εξής έχοντας πάντα μπροστά μας τον πίνακα του πίνακα του Ιωάννη Τριθέμιου :στην πρώτη σειρά του πίνακά μας εντοπίζουμε το S, το γράμμα με το οποίο αρχίζει η πρώτη λέξη του μηνύματός μας. Στη συνέχεια πρέπει να εντοπίσουμε που βρίσκεται το αρχικό γράμμα τη λέξης μας που παίζει το ρόλο του κλειδιού. Το εντοπίζουμε και βλέπουμε από τον πίνακά μας ότι βρίσκεται στη σειρά δώδεκα. Βρίσκουμε το σημείο που τέμνεται η στήλη που

βρίσκεται το γράμμα S με την γραμμή που βρίσκεται το γράμμα L και βρίσκουμε το πρώτο γράμμα του κρυπτογραφήματός μας το οποίο είναι το D.

Ακολουθούμε ακριβώς τα ίδια βήματα για κάθε γράμμα και το κρυπογραφημένο μας μήνυμα είναι το :DCOZJPGGCQLAG

Ας δούμε και ένα άλλο παράδειγμα με τη χρήση του ελληνικού πίνακα. Ας θέλουμε να στείλουμε σε κάποιον το μήνυμα αγάπη και ως λέξη -κλειδί έχουμε το “λευκό”. Αμέσως βλέπουμε ότι η λέξη -κλειδί έχει ίδια γράμματα με το μήνυμά μας συνεπώς την αφήνουμε έτσι όπως έχει :

Πίνακας 2.13 Λέξη-κλειδί, λέξη προς κρυπτογράφηση, κρυπτογραφημένη λέξη

		ΛΕΞΗ-ΚΛΕΙΔΙ		
Λ	Ε	Υ	Κ	Ο
		ΛΕΞΗ ΠΡΟΣ ΚΡΥΠΤΟΓΡΑΦΗΣΗ		
Α	Γ	Α	Π	Η
		ΚΡΥΠΟΓΡΑΦΗΜΕΝΗ ΛΕΞΗ		
Μ	Θ	Φ	Β	Χ

Ας εξετάσουμε τώρα το πίνακα με το ελληνικό αλφάβητο του Ιωάννη Τριθέμιου: το πρώτο γράμμα της λέξης μας είναι το Α και φυσικά το εντοπίζουμε στην πρώτη σειρά του πίνακά μας. Το πρώτο γράμμα του κλειδιού μας είναι το Λ και βρίσκεται στην ενδέκατη σειρά.
Στη συνέχεια βρίσκουμε σε ποιο γράμμα τέμνεται κάθετα με την στήλη στην οποία είναι το Α και ανακαλύπτουμε ότι αυτό το γράμμα είναι το Μ, συνεπώς είναι το πρώτο γράμμα της κρυπτογραφημένης λέξης μας. Ακολουθούμε την ίδια διαδικασία και η κρυπτογραφημένη λέξη μας είναι η ΜΘΦΒΧ.

2.2.10 Δέκατος έβδομος αιώνας

Το 1678 ένας Γάλλος ο Antoine Rossignol βοήθησε την χώρα του να νικήσεις τους Ουγενότες αποκρυπτογραφώντας ένα μήνυμα το οποίο είχε στην κατοχή του ένας αιχμάλωτος.
Μετά από αυτή την νίκη κλήθηκε πολλές φορές για να λύσει κρυπτογραφήματα για την γαλλική κυβέρνηση. Χρησιμοποιούσε δύο λίστες για να λύσει τα κρυπτογραφήματά του στη μία εκ των οποίων τα στοιχεία χωρίς κρυπτογράφηση ήταν σε αλφαβητική σειρά και τα κωδικοποιημένα στοιχεία σε τυχαία σειρά και στην άλλη για διευκόλυνση της αποκρυπτογράφησης τα κρυπτογραφημένα στοιχεία ήταν σε αλφαβητική ή αριθμητική σειρά ενώ τα ισοδύναμα μη κρυπτογραφημένα στοιχεία ήταν σε τυχαία σειρά.
Μετά το θάνατό του ο γιός του και ο εγγονός του συνέχισαν την εργασία του. Ως τότε υπήρχαν πολλοί κρυπτογράφοι οι οποίοι πρόσφεραν τις υπηρεσίες τους στη γαλλική κυβέρνηση.
Όλοι μαζί αποτελούσαν το "Cabinet Noir"(Black Chamber) το οποίο ήταν ουσιαστικά ένας κρυπταναλυτικός οργανισμός.[18]

Την δεκαετία του 1700 οι κρυπταναλυτικοί οργανισμοί ήταν κοινό φαινόμενο στην Ευρώπη και ο πιο γνωστός είχε την έδρα του στην Βιέννη. Το όνομα αυτού του οργανισμού ήταν The Geheime Kabinets-Kanzlei και ο διευθυντής του ήταν ο βαρόνος Ignaz de Koch από το 1749 έως το 1763.

2.2.11 Αποικίες

Στις αποικίες δεν υπήρχε κεντρικός κρυπτογραφικός οργανισμός. Η αποκρυπτογράφηση γινόταν κυρίως από άτομα τα οποία είχαν την ικανότητα να το κάνουν και από ιερείς. Το 1775 ένα μήνυμα που υποκλάπηκε από τον Dr. Benjamin Church θεωρήθηκε ότι ήταν κρυπτογραφημένο μήνυμα προς τους Βρετανούς μόνο που οι Αμερικάνοι επαναστάτες δεν μπορούσαν να το αποκρυπτογραφήσουν.

Το πρόβλημά τους λύθηκε από τους Elbridge Gerry ο οποίος έγινε αργότερο ο πέμπτος πρόεδρος των Ηνωμένων Πολιτειών και τον Elisha Porter.

Ο Benedict Arnold χρησιμοποίησε έναν κώδικα και κάθε άτομο που θα ήταν αποδέκτης αλληλογραφίας θα έπρεπε να έχει το ίδιο βιβλίο προκειμένου να είναι σε θέση να αποκρυπτογραφήσει τις επιστολές πράγμα το οποίο δεν ήταν μία επιτυχημένη πρακτική.

Επίσης οι Αμερικάνοι επαναστάτες προσέλαβαν κρυπτογράφους κατά τη διάρκεια του πολέμου δύο από τους οποίους ήταν οι Samuel Woodhull και Robert Townsend οι οποίοι πληροφόρησαν τον στρατηγό General George Washington για τις κινήσεις των βρετανικών στρατευμάτων γύρω και μέσα στην πόλη της Νέας Υόρκης. Ο κώδικας τον οποίο χρησιμοποιούσαν αποτελούταν από αριθμούς οι οποίοι αντικαθιστούσαν τις λέξεις των κειμένων προς κρυπτογράφηση. Ο κώδικας γράφτηκε από τον Benjamin Tallmadge.

Ο πατέρας της αμερικάνικης κρυπτογραφίας είναι ο James Lovell ο οποίος ήταν πιστός στις αποικίες και αποκρυπτογράφησε πολλά κρυπτογραφήματα μερικά από τα οποία βοήθησαν σε νίκες των επαναστατών.

Η κρυπτογράφηση τροχού (wheel cipher) ήταν εφεύρεση του Thomas Jefferson γύρω στα 1795 και παρόλο που ο ίδιος δεν ασχολήθηκε πολύ με την εφεύρεσή του ένα παρόμοιο σύστημα υπήρχε και στο αμερικανικό ναυτικό μόλις λίγα χρόνια πριν. Η κρυπτογράφηση τροχού αποτελούταν από τροχούς καθένας εκ των οποίων είχε χαραγμένο το αλφάβητο σε τυχαία σειρά.

Το κλειδί είναι η σειρά με την οποία τοποθετούνται οι τροχοί στον άξονα. Το μήνυμα κωδικοποιείται με ευθυγράμμιση των γραμμάτων κατά μήκος του άξονα ο οποίος περιστρέφεται μέχρι να σχηματιστεί το μήνυμα που επιθυμούμε. Η αποκρυπτογράφηση γίνεται όταν ο λήπτης ευθυγραμμίσει τα γράμματα του κρυπτογραφήματος.

2.2.12 Δέκατος ένατος αιώνας

Οι Charles Wheatstone and Lyon Playfair εφηύραν το κρυπτογάφημα Playfair το 1854 το οποίο ήταν το πρώτο που χρησιμοποιούσε ζευγάρια συμβόλων για κρυπτογράφηση και πήρε το όνομά του από τον λόρδο Playfair. Αυτό το σύστημα είναι σχετικά γρήγορο και δεν απαιτεί ειδικό εξοπλισμό για την υλοποίησή του.

Χρησιμοποιήθηκε από τους Βρετανούς κατά τη διάρκεια του Πρώτου Παγκόσμιου Πολέμου, του Δεύτερου πολέμου των Μπόερς καθώς και κατά τη διάρκεια του Δευτέρου Παγκόσμιου Πολέμου.

Μετά την εφεύρεση των ηλεκτρονικών υπολογιστών το κρυπτογράφημα Playfair δεν χρησιμοποιούταν πλέον αφού με την χρήση ηλεκτρονικού υπολογιστή και φυσικά του ανάλογο κώδικα ένα τέτοιο κρυπτογράφημα μπορεί εύκολα να σπάσει.

Το κρυπτογράφημα Playfair έχει ένα κλειδί και φυσικά ένα κείμενο προς κρυπτογράφηση. Το κλειδί έχει την μορφή μίας ακολουθίας είκοσι πέντε συμβόλων χωρίς κάποιο από αυτά να επαναλαμβάνεται και μπορούμε να δημιουργήσουμε ένα κλειδί με τη μορφή τετραγωνικού πίνακα, κάτι που θα δούμε παρακάτω.

Η κρυπτογράφηση με χρήση του Playfair έχει τα εξής βήματα:

- Δημιουργία τετραγωνικού πίνακα-κλειδιού στον οποίο υπάρχουν χαρακτήρες οι οποίοι θα μας βοηθήσουν στην κρυπτογράφηση του κειμένου μας.
- Οι χαρακτήρες πρέπει να είναι μοναδικοί δηλαδή να μην επαναλαμβάνονται.
- Στον πίνακα δεν υπάρχει το γράμμα “J” και σε περίπτωση που υπάρχει ένα τέτοιο γράμμα στο κείμενο προς κρυπτογράφηση αντικαθίστατο από το “I”.
- Στον πίνακα δεν πρέπει να υπάρχουν σημεία στίξης και οποιοσδήποτε άλλος μη αλφαβητικός χαρακτήρας.
- Ο πίνακας-κλειδί θα αρχίσει με τη φράση-κλειδί αφού αφαιρεθούν τα επαναλαμβανόμενα γράμματα και στη συνέχεια θα ακολουθήσουν τα υπόλοιπα γράμματα του αλφάβητου χωρίς να διαφοροποιηθεί η σειρά τους

Στη συνέχεια, πριν κρυπτογραφήσουμε το κείμενό μας θα πρέπει να το χωρίσουμε σε διαγράμματα δηλαδή σε συλλαβές οι οποίες αποτελούνται από δύο γράμματα. Σε περίπτωση που ο αριθμός των συμβόλων μας είναι περιττός, τότε μπορούμε να προσθέσουμε το γράμμα “Z” στο τελευταίο γράμμα της φράσης-κλειδιού μας. Σε περίπτωση που υπάρχουν επαναλαμβανόμενα γράμματα στο κείμενο προς κρυπτογράφηση τότε τα αντικαθιστούμε με το γράμμα “Z”.

Οι κανόνες για την κρυπτογράφηση με Playfair είναι τρεις οι εξής :

Εάν και τα δύο γράμματα της συλλαβής μας είναι στην ίδια σειρά, σε περίπτωση που ένα από τα γράμματα είναι δεξιά, τότε αντικαθίσταται από αυτό που είναι αριστερά

Εάν και τα δύο γράμματα της συλλαβής μας είναι στην ίδια στήλη, τότε, αν υποθέσουμε ότι ένα από τα γράμματα είναι στο κάτω μέρος της στήλης αντικαθίσταται με αυτό που βρίσκεται στην κορυφή της ίδιας στήλης.

Εάν τα γράμματα της συλλαβής μας δεν είναι ούτε στην ίδια στήλη ούτε στην ίδια σειρά, τότε τα αντικαθιστούμε με αυτά που βρίσκονται στις αντίθετες θέσεις οριζόντια.

Ας δούμε ένα παράδειγμα για να το καταλάβουμε καλύτερα. Ας υποθέσουμε ότι θέλουμε να κρυπτογραφήσουμε τη φράση “hide the gold” και η φράση-κλειδί μας είναι η “hello world”.

Το πρώτο το μας βήμα είναι να δημιουργήσουμε τον τετραγωνικό πίνακα που θα μας χρησιμέψει ως κλειδί ο οποίος είναι ο εξής :

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

Πίνακας 2.14 Πίνακας -κλειδί για κρυπτογράφηση με την τεχνική Playfair

Η φορά των γραμμάτων είναι από τα αριστερά προς τα δεξιά και αφαιρέσαμε τα γράμματα που εμφανίζονται δύο φορές στη φράση-κλειδί μας και είναι πασιφανές ότι παραλείψαμε το J.

Στη συνέχεια δημιουργούμε το δίγραμμα, δηλαδή αφού θέλουμε να κρυπτογραφήσουμε την φράση " hide the gold" την χωρίζουμε σε συλλαβές των δύο γραμμάτων και αυτές είναι οι εξής :hi-de-th -eg-ol-dz

Τώρα προχωρούμε στη διαδικασία της κρυπτογράφησης.

Τα διαγράμματά μας είναι συνολικά έξι. Ας αρχίσουμε με το πρώτο δηλαδή το "hi" και ας προσπαθήσουμε να το εντοπίσουμε στον παραπάνω πίνακα.

Τα εν λόγω γράμματα δεν είναι ούτε στην ίδια στήλη ούτε στην ίδια σειρά. Συνεπώς θα πρέπει να διαλέξουμε τις αντίθετες γωνίες οριζόντια δηλαδή έχουμε: hi→ lf

Ας συνεχίσουμε με το δίγραμμα "de" τα γράμματα του οποίου βρίσκονται στην ίδια στήλη συνεπώς έχουμε:de→ gd

Πάμε στο δίγραμμα "th" του οποίου τα δύο γράμματα δεν είναι ούτε στην ίδια γραμμή ούτε στην ίδια στήλη συνεπώς και πάλι θα πρέπει να διαλέξουμε τις αντίθετες γωνίες και έτσι :th→ nw

Στη συνέχεια πάμε στο δίγραμμα "eg" και έχουμε :eg→ dp

Ας πάμε στο δίγραμμα "ol" τα γράμματα του οποίου είναι στην ίδια γραμμή και έχουμε:ol→ wo

Τέλος έχουμε το δίγραμμα "dz" και έχουμε :dz→ cv [21]

2.2.13 Εικοστός αιώνας

Στην αυγή του εικοστού αιώνα σύννεφα πολέμου φάνηκαν στην Ευρώπη. Η Αγγλία επικεντρώθηκε στη βελτίωση των κρυπταναλυτικών υποδομών της. Η ομάδα κρυπτανάλυσης ονομαζόταν "Δωμάτιο 40" επειδή εγκαταστάθηκε αρχικά σε ένα κτίριο στο Λονδίνο. Το μεγαλύτερο επίτευγμα αυτής της ομάδας ήταν το σπάσιμο των γερμανικών ναυτικών κρυπτογραφημάτων ένα έργο αρκετά εύκολο διότι οι Γερμανοί χρησιμοποιούσαν πολιτικούς και εθνικιστικούς όρους ως κλειδιά, είχαν την συνήθεια να τα αλλάζουν ανά τακτά χρονικά διαστήματα κλπ.

Το 1895 ο ασύρματος άλλαξε την κρυπτογραφία διότι οι επικοινωνίες ήταν ανοιχτές και μπορούσε να τις υποκλέψει ο οποιοσδήποτε και η ασφάλεια ήταν κάτι σχεδόν αδύνατο.

Το 1917 οι Αμερικανοί σχημάτισαν έναν κρυπτογραφικό οργανισμό ο οποίος ονομάστηκε MI-8 με διευθυντή τον Herbert Osborne Yardley όπου κάθε τύπος κρυπτογραφικού μηνύματος μπορούσε να αναλυθεί. Το 1929 ο Herbert Hoover ο πρόεδρος των Ηνωμένων Πολιτειών αποφάσισε να τερματίσει την

λειτουργία του. Ο Yardley έγραψε ένα βιβλίο στο οποίο περιέγραφε την δουλειά του οργανισμού MI-8 με τίτλο "The American Black Chamber".

Μέχρι το 1917 οι επικοινωνίες μέσω τηλέγραφου γινόταν με την χρήση του κώδικα Baudot. Η Αμερικανική Τηλεγραφική και Τηλεφωνική Εταιρία είχε θορυβηθεί αναφορικά με την ευκολία που θα μπορούσε κάποιος να διαβάσει τα μηνύματα των μερών που επικοινωνούσαν και έτσι ο Gilbert S. Vernam ανέπτυξε ένα σύστημα το οποίο πρόσθετε τους ηλεκτρονικούς παλμούς του κειμένου προς μετάδοση με ένα κλειδί το οποίο παρήγαγα παλμούς κρυπτογραφήματος. Ο ίδιος δημιούργησε μία μηχανή για κρυπτογράφηση μηνυμάτων αλλά δεν χρησιμοποιήθηκε ευρέως.

Η χρήση των κρυπτογραφικών μηχανών άλλαξε δραματικά την φύση της κρυπτογραφίας και της κρυπτανάλυσης. Η κρυπτογραφία συνδέθηκε με το σχεδιασμό μηχανής και το προσωπικό ασφάλειας ασχολήθηκε με την προστασία αυτών των μηχανών. Η ουσία δεν άλλαξε αλλά η μέθοδος της κρυπτογράφησης έγινε πιο αξιόπιστη και ηλεκτρομηχανική.[18]

Το 1929 ο Lester S. Hill δημοσίευε ένα άρθρο με τίτλο “Cryptography in an Algebraic Alphabet” (Κρυπτογραφία με αλγεβρικό αλφάβητο) στο περιοδικό "The American Mathematical Monthly”.[22]

Σε κάθε χαρακτήρα ενός κειμένου προς κρυπτογράφηση αποδίδει αριθμητική τιμή. Στη συνέχεια χρησιμοποιεί πολυωνυμικές εξισώσεις για να το κρυπτογραφήσει. Για να απλοποιήσει τις εξισώσεις τις μετέτρεψε σε πίνακες οι οποίοι πολλαπλασιάζονται ευκολότερα. Αυτή η μέθοδος σχεδόν εξαλείφει κάθε επανάληψη του κρυπτογραφήματος και δεν σπάει με επίθεση ανάλυσης συχνότητας χαρακτήρων.

Στη συνέχεια ο Hill κατασκεύασε μία μηχανή κρυπτογράφησης για το σύστημά του χρησιμοποιώντας μία σειρά από τροχούς με γρανάζια συνδεδεμένους μεταξύ τους αλλά η μηχανή μπορούσε να χειριστεί περιορισμένο αριθμό κλειδιών και τελικά δεν χρησιμοποιήθηκε ευρέως.

Κατά τη διάρκεια του Δεύτερου Παγκοσμίου Πολέμου η ουδέτερη Σουηδία ανέπτυξε κατά πολύ τον κρυπτογραφικό της τομέα όπως και οι ΗΠΑ

Κορυφαία στιγμή στην ιστορία της κρυπτογραφίας ήταν η αποκρυπτογράφηση του κώδικα της μηχανής Enigma και αυτό γιατί αυτό το γεγονός έκρινε την έκβαση του Δευτέρου Παγκοσμίου Πολέμου.

Η πρώτη μηχανή Enigma κατασκευάστηκε από τον μηχανικό Άρθρουρ Σέρμπιους λίγο μετά τον Α΄ Παγκόσμιο Πόλεμο το 1918. Ο Σέμπιους έκανε κάποιες τροποποιήσεις προκειμένου να προσελκύει το ενδιαφέρον των στρατιωτικών δυνάμεων με μηδαμινή επιτυχία.

Τα πρώτα χρόνια η μηχανή Enigma χρησιμοποιούταν σε λογιστικά γραφεία, τράπεζες, και μεγάλες εταιρίες για ανταλλαγή εμπιστευτικών μηνυμάτων. Αλλά το 1927 οι γερμανικές στρατιωτικές υπηρεσίες είδαν την εφεύρεση του Σέμπιους με άλλο μάτι λόγω της επιτυχίας των Βρετανών να παραβιάσουν τις γερμανικές επικοινωνίες και έτσι να καθορίσουν την έκβαση του Πρώτου Παγκοσμίου Πολέμου.

Ο Χίτλερ ανέθεσε το 1934 σε έναν αξιωματικό των διαβιβάσεων τον συνταγματάρχη Erich Fellgiebel την παρακολούθηση και την τεχνολογική εξέλιξη της μηχανής.

Η Enigma είχε ένα πληκτρολόγιο όπως ακριβώς μία απλή γραφομηχανή και πάνω από αυτό υπήρχε ένα φωτιζόμενο πάνελ με το αλφάβητο. Η βασική αρχή της ήταν απλή αφού βασιζόταν στην αντικατάσταση γραμμάτων :για κάθε γράμμα που πατούσε ο χρήστης άναβε στο πάνελ ένα άλλο γράμμα το οποίο ήταν και το κρυπτόγραμμά του. Αυτό που δυσκόλευε την κατάσταση ήταν πως αν ο χρήστης πατούσε πολλές φορές το ίδιο γράμμα στο πληκτρολόγιο στο πάνελ θα εμφανιζόταν ένα άλλο κρυπτόγραμμα.

Ας υποθέσουμε δηλαδή ότι ο χρήστης πατούσε το Α την πρώτη φορά στο πάνελ μπορεί να άναβε το Ο και αν ξαναπατούσε το Α στο πάνελ μπορεί να έβλεπε το Χ. Έτσι ήταν εφικτό να έχουν πολλούς συνδυασμούς και να αποφεύγουν το επαναλαμβανόμενο μοτίβο. Αυτό ήταν δυνατό λόγω ενός είδους γραναζιών που ήταν στο

εσωτερικό της συσκευής. Κάθε φορά που κάποιος πατούσε ένα γράμμα τα γρανάζια άλλαζαν θέση και την ίδια στιγμή άλλαζε και η σύνδεση του πλήκτρου με την αντίστοιχη φωτεινή ένδειξη του κρυπτογράμματος.

Υπήρχαν τρία γρανάζια ή αλλιώς ρότορες που το καθένα είχε την δυνατότητα είκοσι έξι περιστροφών πράγμα που σημαίνει ότι πριν ακόμα ξεκινήσει η διαδικασία της κρυπτογράφησης από την αρχή υπήρχαν δεκαεπτά χιλιάδες διαφορετικοί συνδυασμοί.

Η Βέρμαχτ χρησιμοποίησε ένα ακόμα πιο περίπλοκο μοντέλο αφού τοποθετήθηκε στη συσκευή ένας πίνακας βυσμάτων στην πρόσοψη της μηχανής όπου ειδικά ζεύγη συμφωνημένων γραμμάτων εναλλάσσονταν με την εισαγωγή βυσμάτων. Οι συνδυασμοί λοιπόν ουσιαστικά ήταν αμέτρητοι.[23]

Στα τέλη του 1932 ο πολωνός μαθηματικός Μάριαν Ρεζέβσκι (Marian Rejewski, 1905-1980) κατάλαβε την καλωδίωση της συσκευής και έκανε ένα σημαντικό βήμα προς το σπάσιμο του κώδικα.

Ο Ρεζέβσκι παρατήρησε ότι στις δώδεκα τα μεσάνυχτα οι Γερμανοί έστελναν ένα συγκεκριμένο μήνυμα δύο φορές. Αυτό λειτουργούσε ως προειδοποίηση για την αλλαγή των ρυθμίσεων της μηχανής Enigma.

Με υπομονή και επιμονή ανέλυσε χιλιάδες μηνύματα και έτσι έκανε πίνακες με την πιθανή αντικατάσταση των γραμμάτων. Αξίζει όμως να πούμε ότι ο Γερμανός απόστρατος Χανς-Θίλο Σμιντ έδωσε στον αξιωματικό Γκουστάβ Μπερτράν της γαλλικής αντικατασκοπείας αντίγραφα των ρυθμίσεων της μηχανής Enigma.

Στο παιχνίδι της αποκρυπτογράφησης τελικά μπήκαν οι Γάλλοι και οι Βρετανοί οι οποίοι πήραν και μία συσκευή Enigma.

Οι Βρετανοί το 1938 αγόρασαν μία έπαυλη στο Bletchley Park την οποία ονόμασαν Σταθμό Χ.[24]

Εκεί δούλευε και ο Άγγλος μαθηματικός Άλαν Τούριγκ (Alan Turing). Αυτός και η ομάδα του κατάφεραν να σπάσουν τον κώδικα της συσκευής Enigma με την δημιουργία μίας συσκευής πρόδρομο των υπολογιστών την Bombe. [25]

Ο Τούριγκ κατανόησε ότι δεν ήταν απαραίτητη η εξέταση όλων των πιθανών συνδυασμών για να σπάσει κάποιος τον κώδικα της εν λόγω μηχανής. Απέδειξε ότι ήταν δυνατόν να εξεταστούν μόνο οι σωστές τοποθετήσεις των διακοπτών οι οποίες ήταν περίπου ένα εκατομμύριο συνδυασμοί και όχι οι τοποθετήσεις του πίνακα συνδέσεων που ήταν εκατό πενήντα επτά χιλιάδες συνδυασμοί.[25]

Σύντομα οι Άγγλοι κατάλαβαν ότι οι Γερμανοί χρησιμοποιούσαν συγκεκριμένες φράσεις για να ανοίγουν και να κλείνουν τα μηνύματά τους. Οι Γερμανοί προδόθηκαν από τη σιγουριά τους.

Μέχρι το τέλος του πολέμου οι επιστήμονες του Bletchley Park είχαν αποκρυπτογραφήσει περίπου δυόμιση εκατομμύρια μηνύματα συμβάλλοντας αποφασιστικά στην νίκη των συμμαχικών δυνάμεων [25][26].

Το 1948 ο Σάννον (Shannon) εξέδωσε το βιβλίο του "A Communications Theory of Secrecy Systems".

Ο Σάννον ήταν ένας από τους πρώτους μοντέρνους κρυπτογράφους οποίος εφάρμοσε προηγμένες μαθηματικές τεχνικές στην επιστήμη της κρυπτογραφίας.

Η ανάλυση του Σάννον έχει αρκετά σπουδαία χαρακτηριστικά στατιστικής φύσης της γλώσσας τα οποία κάνουν την λύση προηγούμενων κρυπτογραφημάτων αρκετά ξεκάθαρη.

Ίσως το πιο σπουδαίο επίτευγμα του Σάννον είναι η unicity distance η οποία είναι ένας αριθμός ο οποίος δείχνει την ποσότητα του κρυπτογραφημένου κειμένου που απαιτείται για να καθοριστεί μοναδικά το κείμενο που έχει αποσταλεί. [18]

2.3 Αρχές της μοντέρνας κρυπτογραφίας

Ιστορικά η κρυπτογραφία όπως ήδη αναφέραμε ήταν περισσότερο τέχνη παρά επιστήμη αλλά κατά τη διάρκεια των πρόσφατων δεκαετιών υπερίσχυσε η επιστημονική της πλευρά.

Τα διάφορα σχήματα κρυπτογράφησης αναπτύσσονται και αναλύονται με συστηματικότερο τρόπο με στόχο να αποδείξουμε ότι μία οποιαδήποτε κρυπτογραφική δομή είναι ασφαλής. Για να διατυπώσουμε αυτές τις αποδείξεις θα πρέπει να διατυπώσουμε κάποιους ορισμούς για παράδειγμα θα πρέπει να διασαφηνίσουμε έννοιες όπως για παράδειγμα τη σημασία της λέξης “ασφαλής”. Η έμφαση στους ορισμούς, στις υποθέσεις και στις αποδείξεις διακρίνει την μοντέρνα κρυπτογραφία από την κλασσική.

2.3.1 Επίσημοι ορισμοί

Μία από τις σπουδαιότερες συνεισφορές της μοντέρνας κρυπτογραφίας είναι η αναγνώριση ότι οι επίσημοι ορισμοί για την ασφάλειας είναι απαραίτητοι για τον σχεδιασμό, την μελέτη και την αξιολόγηση των κρυπτογραφικών αλγορίθμων. Οι επίσημοι ορισμοί παρέχουν μία ξεκάθαρη κατανόηση αναφορικά με τις επιθέσεις που μπορεί να συμβούν καθώς και με τις επιθυμητές εγγυήσεις ασφάλειας.

Οι ορισμοί επίσης προσφέρουν έναν τρόπο αξιολόγησης του κώδικα που αναπτύσσεται. Με έναν ορισμό κάποιος μπορεί να μελετήσει ένα προτεινόμενο σχέδιο και αξιολογήσει αν αυτό θα έχει το επιθυμητό αποτέλεσμα. Επίσης οι ορισμοί μπορεί να χρησιμοποιηθούν για να αποδείξουν ότι ένα προτεινόμενο σχέδιο δεν είναι ασφαλές σε περίπτωση που το σχέδιο δεν ικανοποιεί τον ορισμό.

Οι ορισμοί προσφέρουν την δυνατότητα σύγκρισης των σχεδίων ανάπτυξης αλγορίθμων. Μπορεί να υπάρχουν πολλοί τρόποι με τους οποίους μπορεί να ικανοποιείται η ασφάλεια. Για παράδειγμα ένα σχέδιο το οποίο μπορεί να ικανοποιεί έναν αδύναμο ορισμό μπορεί να είναι πιο αποτελεσματικό από ένα άλλο το οποίο ικανοποιεί έναν αυστηρότερο ορισμό. Με ακριβείς ορισμούς μπορούμε να αξιολογήσουμε τα πλεονεκτήματα και τα μειονεκτήματα των δύο σχεδίων.

Επίσης οι ορισμοί μπορούν να μας βοηθήσουν να καταλάβουμε πότε μία χρήση ενός σχεδίου είναι ασφαλής. Για παράδειγμα ας θεωρήσουμε ότι πρέπει να αποφασίσουμε πιο σχέδιο ανάπτυξης κρυπτογραφικού αλγορίθμου να επιλέξουμε για μία μεγάλη εφαρμογή. Ένας ασφαλής τρόπος για να
προσεγγίσουμε το πρόβλημα είναι να κατανοήσουμε το είδος της ασφάλειας που απαιτείται γι' αυτή την εφαρμογή και μετά να βρούμε ένα ανάλογο σχέδιο κρυπτογράφησης. Ένα πλεονέκτημα αυτής της
προσέγγισης είναι η ευελιξία δηλαδή ο προγραμματιστής μπορεί να προσθέσει ή να αφαιρέσει κάποια στοιχεία προκειμένου να επιτευχθεί ο στόχος ενώ ένας επίσημος ορισμός τον αναγκάζει να ακολουθήσει συγκεκριμένα βήματα.

2.3.2 Ακριβείς υποθέσεις

Οι περισσότεροι μοντέρνοι κρυπτογραφικοί αλγόριθμοι δεν μπορούν εκ των προτέρων να θεωρηθούν ασφαλείς. Γι' αυτό τον λόγο οι αποδείξεις για την ασφάλεια βασίζονται σε υποθέσεις. Στα πλαίσια της μοντέρνας κρυπτογραφίας απαιτείται τέτοιες υποθέσεις να είναι σαφείς και μαθηματικά ακριβείς.

2.3.3 Αποδείξεις ασφάλειας

Οι δύο αρχές που περιγράψαμε πιο πάνω μας επιτρέπουν να επιτύχουμε τον στόχο μας ο οποίος είναι η παροχή απόδειξης ότι η δομή ενός σχεδίου ικανοποιεί έναν ορισμό υπό κάποιες συγκεκριμένες υποθέσεις. Τέτοιες αποδείξεις είναι σημαντικές σε περίπτωση που πρέπει να αντιμετωπιστεί κάποιος που προσπαθεί να επιτεθεί και να σπάσει κάποιον κρυπτογραφικό κώδικα. Οι αποδείξεις ασφάλειας αποτελούν εγγύηση ότι κανένας δεν θα μπορέσει να σπάσει τον κώδικα. Η εμπειρία έχει αποδείξει ότι η διαίσθηση στην κρυπτογραφία είναι καταστροφική.

2.3.4 Αποδεδειγμένη ασφάλεια και ασφάλεια πραγματικού κόσμου

Το μεγαλύτερο τμήμα της κρυπτογραφίας έχει μαθηματικά θεμέλια αλλά δεν παύει να αποτελεί ένα είδος τέχνης. Η αυστηρή προσέγγιση αφήνει χώρο στη δημιουργικότητα η οποία σχετίζεται με την ανάπτυξη ορισμών που ταιριάζουν στις σύγχρονες εφαρμογές. Πάντα βέβαια θα υπάρχει και η τέχνη της ανάπτυξης σχεδίων επίθεσης στις εφαρμογές ακόμα και αν θεωρούνται ασφαλείς.
Το πεδίο της κρυπτογραφίας έχει εξελιχθεί κατά πολύ λόγω της προσέγγισης που περιγράψαμε και αυτή η εξέλιξη βοήθησε έτσι ώστε να υπάρχει εμπιστοσύνη στα κρυπτογραφικά σχέδια που αναπτύσσονται στον πραγματικό κόσμο. Αλλά είναι σημαντικό να μην βασιζόμαστε απόλυτα στην απόδειξη ασφάλειας ενός κρυπτογραφικού σχεδίου. Μία απόδειξη ασφάλειας έχει πάντα σχέση με τον ορισμό που λάβαμε υπόψη καθώς και με την υπόθεση ή τις υποθέσεις που χρησιμοποιήθηκαν. Αν η εγγύηση ασφάλειας δεν ταιριάζει με αυτά που απαιτούνται ή αν το μοντέλο που αντιμετωπίζει τις επιθέσεις δεν έχει τη δυνατότητα να εντοπίσει τις ικανότητες του αντιπάλου τότε η απόδειξη ασφάλειας δεν εξυπηρετεί τον σκοπό μας. Παρόμοια αν η υπόθεση στην οποία βασιστήκαμε είναι ψευδής, τότε η απόδειξη ασφάλειας είναι χωρίς νόημα. [18]

2.4 Στόχοι της κρυπτογραφίας

Η κρυπτογραφία στοχεύει να κάνει εφικτή την επικοινωνία μεταξύ δύο οντοτήτων μέσω ενός καναλιού το οποίο μπορεί και να μην είναι απόλυτα ασφαλές έτσι ώστε μία τρίτη μη εξουσιοδοτημένη οντότητα να μην μπορεί είτε να κατανοήσει στοιχεία αυτής της επικοινωνίας είτε να έχει δυνατότητα παρεμβολής σε αυτήν. Οι βασικοί στόχοι είναι οι εξής :

- **Ιδιωτηκότητα και εμπιστευτικότητα (privacy and confidentiality)** :τα άτομα και οι οργανισμοί χρησιμοποιούν κρυπτογραφία καθημερινά με σκοπό την προστασία της ιδιωτικότητας και τη διατήρηση της εμπιστευτικότητας των δεδομένων τους. Η κρυπτογραφία αποτελεί μέσο διασφάλισης της εμπιστευτικότητας μέσω της κρυπτογράφησης μηνυμάτων τα οποία αποστέλλονται
- με την εφαρμογή ενός αλγόριθμου με ένα κλειδί το οποίο είναι γνωστό μόνο στον αποστολέα και στον παραλήπτη. Η κρυπτογράφηση επίσης διασφαλίζει την διαδικτυακή περιήγηση (browsing) όπως για παράδειγμα στην
- περίπτωση των εικονικών ιδιωτικών δικτύων(Virtual Private Networks, VPN) στα οποία γίνεται χρήση κρυπτογραφημένων σηράγγων, ασυμμετρικής κρυπτογράφησης και δημοσίων και ιδιωτικών κλειδιών.
- **Ακεραιότητα (integrity):**οι πληροφορίες που περιέχονται στο αποσταλέν μήνυμα μπορούν να αλλάξουν μόνο από τα εξουσιοδοτημένα μέλη αλλά και πάλι κάθε αλλαγή θα είναι ανιχνεύσιμη.
- **Πιστοποίηση (authentication):** πρόκειται για την τεχνική με την οποία πιστοποιείται ότι η οντότητα με την οποία λαμβάνει χώρα η επικοινωνία είναι αυτή που επιθυμούμε και όχι κάποια άλλη. Όταν αρχίσει μία επικοινωνιακή σύνοδος θα πρέπει να γίνει πιστοποίηση των ταυτοτήτων των μελών τα οποία θα επικοινωνήσουν. Πιο απλά το κάθε μέλος θα πρέπει να αποδείξει την ταυτότητά του πριν αρχίσει η ανταλλαγή πληροφοριών.
- **Μη άρνηση (non denial) :**Η κρυπτογραφία απαιτεί ανάληψη ευθύνης από τον αποστολέα ενός μηνύματος και αυτό σημαίνει ότι ο αποστολέας δεν μπορεί να αρνηθεί τις προθέσεις του κατά τη διάρκεια της δημιουργίας ή και της μετάδοσης κάποιου μηνύματος. Οι ψηφιακές υπογραφές μπορούν να διασφαλίσουν το παραπάνω καθώς έτσι ο αποστολέας δεν μπορεί να ισχυριστεί ότι κάποιο μήνυμα, συμβόλαιο ή κάθε φύσης έγγραφο που αυτός δημιούργησε είναι ψευδές.[27]

2.5 Εφαρμογές της κρυπτογραφίας

Οι τηλεπικοινωνίες εξελίσσονται με ραγδαίους ρυθμούς καθώς και η κρυπτογραφίας όπως είναι λογικό. Η ανάγκη για ασφάλεια των μεταδιδόμενων πληροφοριών γίνεται όλο και πιο έντονη και οι εφαρμογές τις κρυπτογραφίας ολοένα και αυξάνονται. Κάποιες από αυτές είναι οι παρακάτω:

- Ασφάλεια σε τραπεζικές συναλλαγές

- Κινητή τηλεφωνία
- Σταθερή τηλεφωνία
- Διασφάλιση εταιρικών πληροφοριών
- Στρατιωτικά δίκτυα
- Διπλωματικά δίκτυα
- Ηλεκτρονικές επιχειρήσεις
- Ηλεκτρονική ψηφοφορία
- Ηλεκτρονική δημοπρασία
- Συστήματα συναγερμών
- Έξυπνες κάρτες
- Ιδιωτικά δίκτυα
- Παγκόσμιος Ιστός (World Wide Web)
- Δορυφορικές εφαρμογές
- Ασύρματα Δίκτυα
- Τηλεδιάσκεψη [27]

2.6 Είδη κρυπτογραφικών αλγορίθμων

Υπάρχουν πολλά είδη κρυπτογραφικών αλγορίθμων τα οποία ποικίλλουν σε περιπλοκότητα, ασφάλεια πάντα σε εξάρτηση με τον τύπο επικοινωνίας και την ευαισθησία των πληροφοριών που μεταδίδονται.
Τα είδη αυτά είναι τα παρακάτω:

- κρυπτογράφηση μυστικού κλειδιού(secret key cryptography) ή συμμετρική κρυπτογραφία(symmetric cryptography) στην οποία χρησιμοποιείται ένα κλειδί για κρυπτογράφηση και αποκρυπτογράφηση.
- κρυπτογράφηση με αλγορίθμους ροής (stream ciphers) οι οποίοι δουλεύουν σε μία ακολουθία από bits ή bytes και συχνά αλλάζουν το κλειδί τους με την χρήση μηχανισμών ανάδρασης. Ένας αυτοσυγχρονιζόμενος αλγόριθμος κρυπτογράφησης ροής διασφαλίζει ότι η διαδικασία της αποκρυπτογράφησης είναι σε συγχρονισμό με την διαδικασία της κρυπτογράφησης.
- κρυπτογράφηση τύπου μπλοκ (block ciphers)στα πλαίσια του οποίου η κρυπτογράφηση εκτελείται σε έναν καθορισμένο αριθμό bits.
- κρυπτογράφηση συνάρτησης κατακερματισμού (hash function) διασφαλίζει την ακεραιότητα των δεδομένων κατά τη διάρκεια της κρυπτογράφησης και της αποκρυπτογράφησης.

- κρυπτογράφηση δημόσιου κλειδιού(public key cryptography) ή ασύμμετρη κρυπτογράφηση (asymmetric cryptography) στα πλαίσια της οποίας χρησιμοποιούνται μαθηματικές μέθοδοι για την δημιουργία κωδίκων οι οποίοι σπάνε εξαιρετικά δύσκολα. Η επικοινωνία των οντοτήτων είναι εφικτή χωρίς μυστικό κλειδί. [28]

2.7 Κρυπτογραφία μυστικού κλειδιού και δημόσιου κλειδιού, μία πρώτη προσέγγιση

Η κλασσική κρυπτογράφηση σχετιζόταν με τον σχεδιασμό και την χρήση κρυπτογραφημάτων τα οποία καθιστούσαν δυνατή την επικοινωνία δύο μελών παρά την παρουσία ωτακουστών που ίσως παρακολουθούν την μεταξύ τους επικοινωνία.

Τα κρυπτογραφήματα στην σύγχρονη ορολογία ονομάζονται σχήματα κρυπτογράφησης (encryption schemes).

Η ασφάλεια στα σχήματα κρυπτογράφησης βασίζεται σε μεγάλο βαθμό σε ένα κλειδί το οποίο δίνεται στα μέρη που επικοινωνούν πριν αρχίσει η επικοινωνία και φυσικά είναι άγνωστο στον ωτακουστή.

Αυτή είναι η ρύθμιση ιδιωτικού κλειδιού (private-key setting) και η κρυπτογράφηση ιδιωτικού κλειδιού αποτελεί μία πλευρά αυτής της ρύθμισης.

Ας δούμε την κρυπτογράφηση ιδιωτικού κλειδιού κάτω από ένα γενικό πρίσμα.

Τα δύο μέρη τα οποία πρόκειται να επικοινωνήσουν μοιράζονται ένα κλειδί και το χρησιμοποιούν όταν θέλουν να επικοινωνήσουν μυστικά.

Το ένα μέρος, ο αποστολέας, μπορεί να στείλει ένα μήνυμα αφού χρησιμοποιήσει για την κρυπτογράφησή του το κλειδί το οποίο πριν έχει κοινοποιήσει στον παραλήπτη.

Στη συνέχεια ο παραλήπτης του μηνύματος χρησιμοποιεί αυτό το κλειδί για αποκρυπτογράφηση του ληφθέντος μηνύματος.

Ας σημειωθεί ότι το ίδιο κλειδί χρησιμοποιείται και για κρυπτογράφηση και για αποκρυπτογράφηση και γι' αυτό η εν λόγω διαδικασία είναι γνωστή ως συμμετρική κρυπτογράφηση όπως ήδη αναφέραμε σε αντίθεση με την κρυπτογράφηση του δημόσιου κλειδιού ή ασύμμετρη κρυπτογράφηση αφού στην κρυπτογράφηση και στην αποκρυπτογράφηση χρησιμοποιούνται διαφορετικά κλειδιά.

Υπάρχουν δύο εφαρμογές στην συμμετρική κρυπτογραφία. Στην πρώτη έχουμε δύο διακριτά μέλη τα οποία χωρίζει κάποια απόσταση, για παράδειγμα το ένα μέλος μπορεί να βρίσκεται στην Αθήνα και το άλλο στη Λάρισα. Υποτίθεται ότι τα δύο αυτά μέλη κατάφεραν, με κάποιον τρόπο να μοιραστούν το ιδιωτικό κλειδί με ασφάλεια πριν την επικοινωνία τους.

Η άλλη εφαρμογή αυτού του είδους της κρυπτογραφίας είναι η επικοινωνία του μέλους με τον εαυτό του μετά από μία χρονική περίοδο.

Ας σκεφτούμε για παράδειγμα την κρυπτογράφηση ενός σκληρού δίσκου :ο χρήστης κρυπτογραφεί ένα κείμενο και αποθηκεύει το κρυπτογράφημα στο σκληρός δίσκο του. Ο ίδιος χρήστης φυσικά μετά από κάποιο χρονικό διάστημα θα θελήσει να ανακτήσει το κείμενο και να το αποκρυπτογραφήσει.[28][29]

Επίλογος

Τα κυριότερα σημεία του δεύτερου κεφαλαίου ήταν η διασαφήνιση των εννοιών των λέξεων κρυπτογραφία και κρυπτανάλυση διότι πολλοί τις μπερδεύουν και είναι σημαντικό για κάποιον που θέλει να έχει μία επαφή με την κρυπτογραφία να γνωρίζει τη σημασία αυτών των όρων. Σημαντική ήταν η αναφορά στην μηχανή Enigma. Αναφερθήκαμε στους στόχους, στις εφαρμογές της μοντέρνας κρυπτογραφίας και στα είδη των κρυπτογραφικών αλγορίθμων αλλά κάναμε και μία μικρή θεωρητική εισαγωγή στους όρους κρυπτογραφία μυστικού κλειδιού και δημόσιου κλειδιού

Κεφάλαιο 3
Εισαγωγή

Σε αυτό το κεφάλαιο εξηγούμε τον όρο συμμετρική κρυπτογραφία ή κρυπτογραφία ιδιωτικού κλειδιού και εξηγούμε την αδυναμία της. Αναφέρουμε τις χρήσεις της συμμετρικής κρυπτογραφίας καθώς και την ταξινόμηση των αλγορίθμων συμμετρικής κρυπτογράφησης σε κατηγορίες οι οποίες είναι οι αλγόριθμοι ροής και οι αλγόριθμοι τμήματος. Εστιάζουμε στους αλγόριθμους τμήματος και τους περιγράφουμε με την

συνδρομή ενός παραδείγματος κρυπτογράφησης και αποκρυπτογράφησης. Αναφέρουμε δύο κατηγορίες κρυπτογραφημάτων τμήματος δηλαδή τα κρυπτογραφήματα Hill και το δίκτυο Feistel.Αναλύουμε τις μεθόδους γεμίσματος των τμημάτων κάτι που είναι ιδιαίτερα σημαντικό για τους κρυπτοαλγόριθμους τμήματος. Δεν παραλείπουμε να εξηγήσουμε τις μεθόδους λειτουργίας των αλγορίθμων κρυπτογράφησης τμήματος. Τέλος περιγράφουμε κάποιους από τους αλγόριθμους συμμετρικής κρυπτογράφησης.

3. Συμμετρική κρυπτογραφία

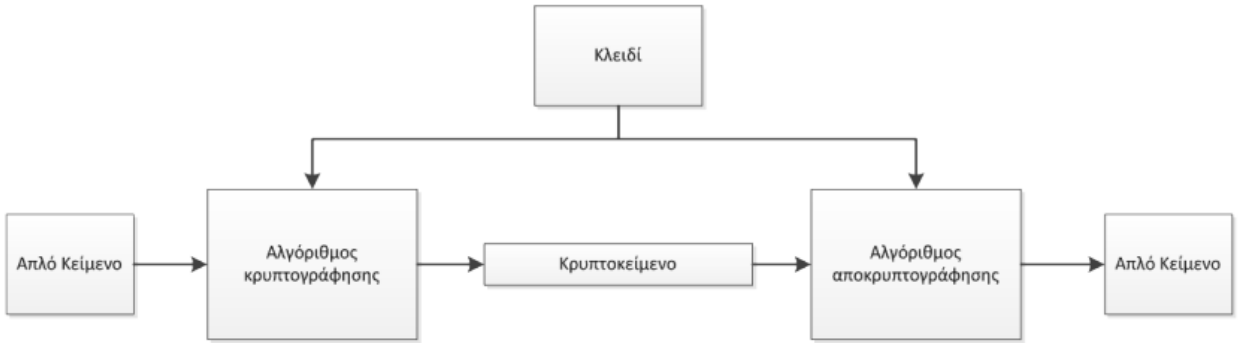
3.1 Ορισμός

Η συμμετρική κρυπτογραφία είναι αρχαιότερη από την ασύμμετρη διότι η πρώτη χρονολογείται από την αρχαία Αίγυπτο ενώ η ασύμμετρη κρυπτογραφία εμφανίστηκε το 1976.[27].Συμμετρική κρυπτογραφία είναι η μετατροπή ενός μηνύματος προς μετάδοση σε κρυπτογράφημα με τη χρήση ενός κλειδιού και από τις δύο πλευρές οι οποίες επιθυμούν να επικοινωνήσουν. Αυτό το κλειδί χρησιμεύει και για την κρυπτογράφηση του μηνύματος αλλά και για την αποκρυπτογράφηση [30]

Ας δούμε τη διαδικασία της συμμετρικής κρυπτογραφία :έχουμε το κείμενο προς κρυπτογράφηση το οποίο εισάγεται μαζί με το κλειδί στον αλγόριθμο κρυπτογράφησης. Το κλειδί δεν έχει καμία απολύτως σχέση με το κείμενο που επιθυμούμε να κρυπτογραφηθεί. Το αποτέλεσμα της εφαρμογής του αλγορίθμου κρυπτογράφησης είναι το κρυπτοκείμενο. Η είσοδος του αλγόριθμου αποκρυπτογράφησης είναι το κρυπτοκείμενο και το κλειδί που είναι το ίδιο με αυτό του αλγορίθμου κρυπτογράφησης.

Στον αλγόριθμο αποκρυπτογράφησης εφαρμόζονται οι αντίστροφοι μηχανισμοί με αυτούς του αλγορίθμου κρυπτογράφησης και το κείμενο επαναφέρεται στην αρχική του μορφή, αυτή του απλού κειμένου.

Τα παραπάνω αποτυπώνονται στο σχήμα που ακολουθεί:



Εικόνα 3.1 Συμμετρικό σύστημα κρυπτογραφίας [35]

3.2 Αδυναμία του συμμετρικού συστήματος κρυπτογράφησης

Η αδυναμία του συμμετρικού συστήματος κρυπτογράφησης έγκειται στο ότι χρησιμοποιείται το ίδιο κλειδί στην κρυπτογράφηση και στην αποκρυπτογράφηση του μηνύματος.

Αυτή η απαίτηση έχει ως αναγκαία συνθήκη ο αποστολέας και ο παραλήπτης να έχουν ένα ασφαλή τρόπο για να μοιραστούν αυτή την πληροφορία. Αρχικά το κλειδί είναι στα χέρια του αποστολέα και αυτός πρέπει να βρει έναν ασφαλή τόπο για να το αποστείλει στον παραλήπτη.

Ο λόγος που το κλειδί, κατά κανόνα, δεν μπορεί να αποσταλεί μέσω ενός καναλιού είναι ότι τέτοιου είδους κανάλια δεν είναι πάντα διαθέσιμα. Ένας τρόπος αποστολής του κλειδιού είναι ο τεμαχισμός του και η αποστολή του μέσω διαφορετικών καναλιών επικοινωνίας. Ένας άλλος είναι να συναντηθεί ο αποστολέας με τον παραλήπτη και ο πρώτος να παραδώσει το κλειδί στον δεύτερο.

3.3 Προϋποθέσεις για την ασφάλεια της επικοινωνίας με τη συμμετρική κρυπτογραφίας

Για μία ασφαλής επικοινωνία με τη χρήση της συμμετρικής κρυπτογραφίας θα πρέπει να ισχύουν τα εξής:

- Θα πρέπει να υπάρχει ένας ισχυρός αλγόριθμος κρυπτογράφησης και έτσι ο επιτιθέμενος δεν θα μπορεί είτε να τον αναλύσει είτε να ανακαλύψει το κλειδί του.
- Ο αποστολέας και ο παραλήπτης πρέπει να έχουν παραλάβει τα κλειδιά με ασφαλή τρόπο.

3.4 Χρήσεις της συμμετρικής κρυπτογραφίας

Μερικά από τα παραδείγματα χρήσης της συμμετρικής κρυπτογραφίας είναι τα ακόλουθα :

- σε εφαρμογές για πληρωμή όπως για παράδειγμα συνδιαλλαγές με κάρτες
- Στην επικύρωση στοιχείων του αποστολέα
- στην παραγωγή τυχαίων αριθμών ή κατακερματισμό (hashing) [30][31]
- πολλά λειτουργικά συστήματα όπως για παράδειγμα τα Microsoft Windows, Apple MacoS χρησιμοποιούν αλγόριθμους συμμετρικής κρυπτογράφησης για τα δεδομένα τα οποία είναι αποθηκευμένα στο σκληρό δίσκο προστατεύοντας τα αποθηκευμένα δεδομένα σε περίπτωση απώλειας ή κλοπής της συσκευής [36]
- στα εικονικά ιδιωτικά δίκτυα(Virtual Private Networks, VPN) χρησιμοποιείται συνεχώς συμμετρική κρυπτογραφία για την κρυπτογράφηση δεδομένων τα οποία μεταδίδονται στο Διαδίκτυο παρέχοντας ασφαλή απομακρυσμένη πρόσβαση στους χρήστες
- σε πρωτόκολλα κινητής τηλεφωνίας όπως για παράδειγμα το iMessage

- σε συμπιέσεις αρχείων αφού μερικές μορφές συμπίεσης αρχείων όπως για παράδειγμα η ZIP χρησιμοποιεί συμμετρική κρυπτογραφία για παροχή επιπλέον ασφάλειας για τα συμπιεσμένα αρχεία

Οι συμμετρικοί κρυπτογραφικοί αλγόριθμοι είναι σημαντικοί για τους εξής λόγους:

- η διαδικασία δημιουργίας ενός ισχυρού κλειδιού για τέτοιου είδους κρυπτογραφήματα (ciphers) είναι σχετικά φθηνή
- τα κλειδιά είναι πολύ μικρότερα συγκριτικά με το επίπεδο της προστασίας που προσφέρουν
- οι αλγόριθμοι έχουν σχετικά μικρό κόστος επεξεργασίας [33]

3.5 Ταξινόμηση συμμετρικών κρυπτογραφικών αλγορίθμων

Οι συμμετρικοί κρυπτογραφικοί αλγόριθμοι ανάλογα με τον τρόπο που επεξεργάζεται ο αλγόριθμος το μήνυμα ταξινομούνται σε δύο κατηγορίες οι οποίες είναι οι εξής :

- Αλγόριθμοι ροής (stream ciphers) χαρακτηριστικό των οποίων είναι η κρυπτογράφηση μίας ροής μηνύματος χωρίς να το διαχωρίζουμε
- Αλγόριθμοι τμήματος (block ciphers) οι οποίοι χωρίζουν το μήνυμα σε κομμάτια καθένα εκ των οποίων κρυπτογραφείται ξεχωριστά

Κάποιοι αλγόριθμοι συμμετρικής κρυπτογράφησης είναι οι εξής :

Πίνακας 3.1 Συμμετρικοί αλγόριθμοι τμήματος και ροής

Συμμετρικοί αλγόριθμοι τμήματος	**Συμμετρικοί αλγόριθμοι ροής**
Data Encryption Standard (DES)	ORYX
Triple DES	RC4
New Des	SEAL
3-Way	EO
Blowfish	A5/X
Twofish	
CAST	
CMEA	
IDEA	
MacGuffin	
Lucifer	
MARS	
RC2	
RC5	
RC6	
Rijndael Advanced Encryption Standard (AES)	
Safer	
Serpent	
Tiny Encryption Algorithm	

3.6 Αλγόριθμοι ροής (Stream Ciphers)

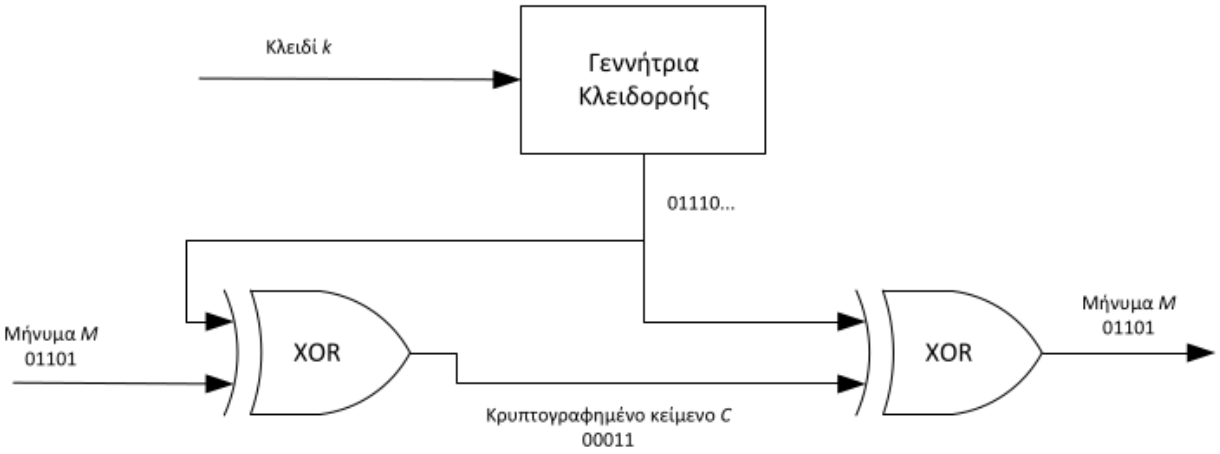
Μία από τις τάξεις τεχνικών συμμετρικής κρυπτογραφίας είναι οι αλγόριθμοι ροής. Λειτουργούν σε μικρές ομάδες οι οποίες μπορεί να είναι bits ή bytes ο μετασχηματισμός των οποίων εξαρτάται από τον χρόνο. [33]

Για να είμαστε πιο ακριβείς σε ένας αλγόριθμος ροής λειτουργεί ως ακολούθως:

Το κείμενο ή γενικότερα τα δεδομένα προς κρυπτογράφηση παριστάνονται ως ακολουθία από δυαδικά ψηφία έστω $m_o, m_1, m_{2...} m_n$

Στη συνέχεια μία γεννήτρια κλειδοροής (key stream generator) δέχεται ως είσοδο ένα κλειδί k και στην έξοδο αυτής παράγεται μία ψευδοτυχαία ακολουθία από bits, η κλειδοροή (key stream).Η γεννήτρια κλειδροής είναι ένας τύπος γεννήτριας ψευδοτυχαίων αριθμών.

Στη συνέχεια το κείμενο κρυπτογραφείται με modulo 2(XOR) πράγμα το οποίο σημαίνει ότι στην κλειδοροή προστίθεται το αρχικό μας κείμενο. Η παραγόμενη ακολουθία bits με αυτόν τον τρόπο αποτελεί το κρυπτογραφημένο κείμενο. [33] [34]



Εικόνα 3.2 Συμμετρικοί αλγόριθμοι ροής

Ο συμβολισμός του κρυπτογραφικού αλγόριθμου ροής θα μπορούσε να είναι (για i≥0):

(3.1)

$$c_i = m_i \otimes k_i$$

- $c_0, c_1 \ldots . c_n$: τα bits του κρυπτογραφημένου μηνύματος
- $m_0, m_1 .. m_n$: τα bits του μηνύματος προς κρυπτογράφηση

- $\oplus$: αυτό το σύμβολο εκφράζεται η πράξη της αποκλειστικής διάζευξης (exclusive or) των bits
- $k_0, k_1 .. k_n$: τα bits της κλειδοροής.

Πάλι για i≥0 στην αποκρυπτογράφηση θα ισχύει :

$$m_i = c_i \otimes k_i \quad (3.2)$$

Πίνακας 3.2 Πίνακας αλήθειας XOR

ΠΙΝΑΚΑΣ ΑΛΗΘΕΙΑΣ XOR

a	**b**	**a⊕ b**
0	0	0
0	1	0
1	0	1
1	1	0

Η γεννήτρια της κλειδοροής πρέπει να έχει τις εξής ιδιότητες :

- η περίοδος κατά την οποία λαμβάνει χώρα η επανάληψη θα πρέπει να είναι μεγάλη αφού η ακολουθία είναι περιοδική και μετά από έναν αριθμό bits θα πρέπει να επαναλαμβάνεται έχοντας ως αφετηρία την αρχή επειδή παράγεται από κάποια συνάρτηση η οποία βασίζεται σε ένα
- κλειδί. Αν υποτεθεί που η περίοδος της επανάληψης είναι μικρή μπορεί κάποιος να υπολογίσει το ακριβές μέγεθος της περιόδου, κάτι που θα κάνει την αποκρυπτογράφηση του κρυπτοκειμένου ενδεχόμενη
- η κλειδοροή πρέπει να μοιάζει με μία πραγματικά τυχαία ακολουθία δηλαδή, όπως αναφέραμε, πρέπει να είναι ψευδοτυχαία. Τρόπος επαλήθευσης αυτού αποτελεί η εφαρμογή ελέγχων τυχαιότητας (random tests) οι οποίοι ελέγχουν κατά πόσο είναι ίδιο το πλήθος των στοιχείων μηδέν και ένα ή ότι κάθε μηδενικό ψηφίο ακολουθείται από το ψηφίο ένα τόσο συχνά όσο και το αντίστροφο
- είναι απαραίτητη η μεγάλη γραμμική ισορροπία (linear equivalence) της κλειδοροής. Η χρήση γραμμικών μεθόδων όπως για παράδειγμα ο υπολογισμός της επόμενης τιμής μίας ακολουθίας βάσει των προηγούμενων τιμών της, είναι ικανή να παράγει οποιαδήποτε ακολουθία δυαδικών ψηφίων. Αν

στον υπολογισμό αυτό χρησιμοποιείται ένας σχετικά μικρός αριθμός προηγουμένων τιμών, τότε λέμε ότι η ακολουθία έχει μικρή γραμμική ισοδυναμία ενώ σε αντίθετη περίπτωση λέμε ότι έχει μεγάλη γραμμική ισοδυναμία. Εάν μία κλειδοροή έχει μεγάλη γραμμική ισοδυναμία εγγυάται μεγαλύτερη ασφάλεια των κρυπτογραφημένων δεδομένων.

Οι παραπάνω συνθήκες δεν είναι επαρκείς για να εξασφαλίσουν την αξιοπιστία ενός κρυπτογραφικού αλγορίθμου ροής. Θα πρέπει με κάποιο τρόπο να εξασφαλίζεται ότι ακόμα και αν κάποιος αποκτήσει κάποια πληροφορία για κάποιο κομμάτι της ακολουθίας κλειδοροής να είναι υπολογιστικά αδύνατο να συνάγει άλλα κομμάτια της.

Οι μοντέρνοι κρυπτογραφικοί αλγόριθμοι ροής χρησιμοποιούν γεννήτριες κλειδοροών οι οποίες παράγουν ψευδοτυχαίες ακολουθίες με πολύ μεγάλες περιόδους.

3.7 Αλγόριθμοι δέσμης ή τμήματος (Block Ciphers)

Οι αλγόριθμοι δέσμης συμμετρικού κλειδιού οι οποίοι προσφέρουν εμπιστευτικότητα, υλοποιούνται σε πολλά κρυπτογραφικά συστήματα, χρησιμοποιούνται ως βασικό στοιχείο σε τεχνικές αυθεντικοποίησης μηνύματος, σε μηχανισμούς ακεραιότητας μηνύματος, σε πρωτόκολλα αυθεντικοποίησης οντότητας και σε ψηφιακές υπογραφές[31] [32].

Είναι σπουδαίο να τονίσουμε ότι οι αλγόριθμοι τμήματος είναι απλώς εργαλεία τα οποία συνδυάζονται για να δημιουργήσουν κάτι χρήσιμο. Οι αλγόριθμοι τμήματος μόνοι δεν κάνουν κάτι για το οποίο ενδιαφέρεται ο χρήστης. Είναι πανίσχυρα εργαλεία αλλά για να τα χρησιμοποιήσει κάποιος θα πρέπει να μάθει να τα χειρίζεται.

3.7.1 Έννοιες και ορισμοί

Κάθε αλγόριθμος τμήματος αποτελεί μία συνάρτηση μετάθεσης (permutation function) που αντιστοιχεί τμήματα μήκους n -bits κειμένου προς κρυπτογράφηση σε τμήματα μήκους n-bits κρυπτοκειμένου. Το n καλείται μήκος τμήματος. Η συγκεκριμένη συνάρτηση έχει ως παράμετρο το διάνυσμα- κλειδί (k). [31][32][35]

Ένας αλγόριθμος τμήματος n-bits αποτελεί ένα τμήμα αρχικού κειμένου το οποίο χρησιμοποιείται για να παράγει ένα τμήμα κρυπτογραφήματος ίσιου μήκους.

Η συνάρτηση κρυπτογράφησης η οποία είναι μία αντιστρέψιμη αντιστοιχία έχει την εξής μορφή :

$$C = E_k(P) \tag{3.3}$$

Ας δούμε αναλυτικά τον τύπο μας :

Το C συμβολίζει το κρυπτογραφημένο κείμενό μας.

Το E είναι ο αλγόριθμος.

Το P είναι το κείμενό μας προς κρυπτογράφηση.

Το k είναι το μυστικό κλειδί.

Παρατηρούμε ότι το κρυπτογράφημα, όπως ήταν αναμενόμενο είναι αποτέλεσμα της κρυπτογράφησης του P με τη χρήση του κλειδιού κρυπτογράφησης k.
Η συνάρτηση της αποκρυπτογράφησης, η οποία είναι και η αντίστροφη της παραπάνω συνάρτησης είναι η εξής :

$$P = D_k(E) \tag{3.4}$$

Πίνακας 3.3 Κείμενο προς κρυπτογράφηση

ΚΕΙΜΕΝΟ ΠΡΟΣ ΚΡΥΠΤΟΓΡΑΦΗΣΗ

m o n o a l p h a b e t i c u n i l a t e r a l s u b s t i t u t i o n s y s t e m

Θα κρυπτογραφήσουμε αυτό το κείμενο αφού το χωρίσουμε σε ομάδες πέντε χαρακτήρων ως εξής :

m	o	n	o	a
l	p	h	a	b
e	t	i	c	u
n	i	l	a	t
e	r	a	l	s
u	b	s	t	i

t	u	t	i	o
n	s	y	s	t
e	m	x	x	x

Παρατηρούμε ότι η τελευταία ομάδα έχει μόνο δύο γράμματα, επομένως πρέπει με κάποιον τρόπο να συμπληρώσουμε τα κενά.
Για να το κάνουμε αυτό πρέπει να διαλέξουμε τον σωστό χαρακτήρα έτσι ώστε το άτομο που θα αποκρυπτογραφήσει το μήνυμα να καταλάβει ότι ο συγκεκριμένος χαρακτήρας δεν έχει σχέση με αυτό.
Σε αυτό το παράδειγμα τα τρία κενά συμπληρώνονται με το γράμμα Χ.
Στη συνέχεια επανατοποθετούμε τα γράμματα στα τμήματα σύμφωνα με την ακόλουθη μετάθεση **:**

0	1	2	3	4
3	**4**	**0**	**2**	**1**

Για το πρώτο τμήμα η τοποθέτηση γίνεται ως εξής: το γράμμα m βρίσκεται στη θέση μηδέν (0) μετατίθεται στη θέση τρία (3) και αφού μετράμε αρχίζοντας από το μηδέν ο δείκτης που αντιστοιχεί στην θέση του m είναι ο δύο(2) (0,1,2, τρεις θέσεις). Το γράμμα ο βρίσκεται στη θέση ένα και μετατίθεται στη θέση τέσσερα (4)(1,2,3,4). Με την ίδια λογική γίνονται και οι άλλες μεταθέσεις.

m	o	n	o	a
o	**a**	**m**	**n**	**o**

Για το δεύτερο τμήμα έχουμε:

l	p	h	a	b
a	**b**	**l**	**h**	**p**

Για το τρίτο τμήμα έχουμε:

e	t	i	c	u
c	**u**	**e**	**i**	**t**

Για το τέταρτο τμήμα ισχύει :

n	i	l	a	t
a	**t**	**n**	**l**	**i**

Για το πέμπτο τμήμα έχουμε:

e	r	a	l	s
l	**s**	**e**	**a**	**r**

Για το έκτο τμήμα ισχύει:

u	b	s	t	i

t i u s b

Για το έβδομο τμήμα ισχύει:

t u t i o

o n t t u

Για το όγδοο τμήμα ισχύει:

n s y s t

s t n y s

Για το ένατο τμήμα ισχύει:

e m x x x

x x e x m

Συνολικά το κρυπτογραφημένο κείμενο είναι το εξής :

Πίνακας 3.4 Κρύπτογραφημένο κείμενο

ΚΡΥΠΤΟΓΡΑΦΗΜΕΝΟ ΚΕΙΜΕΝΟ					
1	**o**	**a**	**m**	**n**	**o**
2	**a**	**b**	**l**	**h**	**p**
3	**c**	**u**	**e**	**i**	**t**
4	**a**	**t**	**n**	**l**	**i**
5	**l**	**s**	**e**	**a**	**r**
6	**t**	**i**	**u**	**s**	**b**
7	**o**	**n**	**t**	**t**	**u**
8	**s**	**t**	**n**	**y**	**s**
9	**x**	**x**	**e**	**x**	**m**

Ας δοκιμάσουμε τώρα να αποκρυπτογραφήσουμε το παρακάτω κρυπτογράφημα αντιστρέφοντας φυσικά τον τρόπο κρυπτογράφησης που είδαμε παραπάνω με χρήση πάλι της αντιμετάθεσης πέντε χαρακτήρων :

THNROKEENRCKNUTIVYNUITESRXXYXX

Χωρίζουμε το κρυπτογραφημένο κείμενο σε ομάδες πέντε γραμμάτων και έτσι η πρώτη ομάδα έχει ως εξής :
Χωρίζουμε το κρυπτογραφημένο κείμενο σε ομάδες πέντε γραμμάτων και έτσι η πρώτη ομάδα έχει ως εξής :

T H N R O

Πάμε στη δεύτερη ομάδα:

K E E N R

Η τρίτη ομάδα έχει ως εξής:

C K N U T

Η τέταρτη ομάδα είναι η εξής:

I V Y N U

Η πέμπτη ομάδα έχει ως ακολούθως :

I T E R S

Τέλος, η έκτη ομάδα είναι η ακόλουθη:

X	X	Y	X	X

Στη συνέχεια επανατοποθετούμε τα γράμματα στα τμήματα σύμφωνα με την ακόλουθη μετάθεση **:**

0	1	2	3	4
3	4	0	2	1

Πίνακας 3.5 Αποκρυπτογραφημένο κείμενο

ΑΠΟΚΡΥΠΤΟΓΡΑΦΗΜΕΝΟ ΚΕΙΜΕΝΟ					
1	**N**	**0**	**R**	**T**	**H**
2	**E**	**N**	**R**	**K**	**E**
3	**N**	**T**	**U**	**C**	**K**
4	**Y**	**U**	**N**	**I**	**V**
5	**E**	**S**	**R**	**I**	**T**
6	**Y**	**X**	**X**	**X**	**X**

Ένας αλγόριθμος τμήματος πρέπει να έχει τις εξής ιδιότητες :

➢το κλειδί θα πρέπει να επιλέγει μία τυχαία αντιμετάθεση

➢τα συγγενικά κλειδιά δεν πρέπει να δίνουν συγγενικές μεταθέσεις [32]

Το 1949 ο Shannon διατύπωσε τις έννοιες της διάχυσης (diffusion) και της σύγχυσης (confusion) οι οποίες ουσιαστικά είναι δύο ιδιότητες ενός ασφαλούς κρυπτογραφήματος οι οποίες στοχεύουν στην παρεμπόδιση της στατιστικής ανάλυσης.

Διάχυση σημαίνει ότι εάν αλλάξουμε έναν χαρακτήρα στο κείμενο προς κρυπτογράφηση (plaintext) τότε θα πρέπει να αλλάξουν αρκετοί χαρακτήρες του κρυπτοκειμένου και το ίδιο ισχύει εάν αλλάξουμε έναν χαρακτήρα του κρυπτοκειμένου. [38] [39].Αυτό σημαίνει ότι η στατιστική συχνότητα των γραμμάτων στο κείμενο το οποίο πρόκειται να κρυπτογραφηθεί διαχέεται σε πολλούς χαρακτήρες στο κρυπτοκείμενο και συνεπώς κάποιος πρέπει να έχει στη διάθεσή του μεγάλης έκτασης κρυπτοκειμένου για να κάνει μία στατιστική επίθεση που να έχει πιθανότητες επιτυχίας.

Σύγχυση σημαίνει ότι το κλειδί δεν σχετίζεται με απλό τρόπο με το κρυπτοκείμενο. Συγκεκριμένα κάθε χαρακτήρας του κρυπτοκειμένου θα πρέπει να εξαρτάται από αρκετά τμήματα του κλειδιού και, όπως είναι λογικό, η σχέση μεταξύ κλειδιού και κρυπτοκειμένου γίνεται πολυπλοκότερη. [39]

Η σύγχυση δίνεται με την χρήση μίας αντικατάστασης η οποία ονομάζεται S-Box (Substitution Box) που ουσιαστικά είναι μία μέθοδος και η διάχυση είναι το αποτέλεσμα μίας αντιμετάθεσης.

Θα πρέπει να αναφέρουμε σε αυτό το σημείο ότι υπάρχουν δύο τύποι μεθόδων S-Box οι στατικές και οι δυναμικές [40].Παρακάτω θα αναλύσουμε ένα παράδειγμα κρυπτογραφικού αλγόριθμου τμήματος για να καταλάβουμε τις έννοιες που μόλις περιγράψαμε.

3.7.1.1 Κρυπτογραφήματα Hill

Τα κρυπτογραφήματα Hill τα οποία εφευρέθηκαν το 1929 είναι κρυπτογραφικοί αλγόριθμοι τμήματος και έχουν τις παραπάνω ιδιότητες.

Ο χαρακτήρας του κρυπτογραφήματος ο οποίος αντικαθιστά έναν συγκεκριμένο χαρακτήρα κειμένου προς κρυπτογράφηση κατά τη διαδικασία αυτής θα εξαρτάται από τους γειτονικούς χαρακτήρες του κειμένου που θα κρυπτογραφηθεί και η κρυπτογράφηση γίνεται με την χρήση πινάκων.

Το κλειδί κρυπτογράφησης είναι ένας τετραγωνικός πίνακας ακεραίων. Αυτοί οι ακέραιοι έχουν εύρος τιμής από το μηδέν έως το n-1 {0….n-1}. Το n συμβολίζει το πλήθος των γραμμάτων του αλφάβητου της γλώσσας στην οποία γράφτηκε το κείμενο. Για παράδειγμα για την αγγλική γλώσσα ισχύει n=26.

Ας υποθέσουμε ότι το κλειδί μας είναι το εξής:

Πίνακας 3.6 Κλειδί κρυπτογράφησης

$$k=\begin{pmatrix}1 & 4 & 0\\ 7 & 11 & 2\\ 0 & 5 & 1\end{pmatrix}$$

Επιθυμούμε να κρυπτογραφήσουμε το μήνυμα “time to study”.Ο πίνακας αντιστοίχισης γραμμάτων και αριθμών είναι ο εξής :

Πίνακας 3.7 Αντιστοίχιση γραμμάτων και αριθμών

ΑΝΤΙΣΤΟΙΧΗΣΗ ΓΡΑΜΜΑΤΩΝ ΚΑΙ ΑΡΙΘΜΩΝ

0	1	2	3	4	5	6	7	8	9	10	11	12
A	B	C	D	E	F	G	H	I	J	K	L	M
13	14	15	16	17	18	19	20	21	22	23	24	25
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Με βάση τον παραπάνω πίνακα το κείμενό μας παίρνει τη εξής μορφή:

Πίνακας 3.8 Αντικατάσταση γραμμάτων με αριθμητικά σύμβολα

ΑΝΤΙΚΑΤΑΣΤΑΣΗ ΓΡΑΜΜΑΤΩΝ ΜΕ ΑΡΙΘΜΗΤΙΚΑ ΣΥΜΒΟΛΑ										
t	i	m	e	t	o	s	t	u	d	y
19	**8**	**12**	**4**	**19**	**14**	**18**	**19**	**20**	**3**	**24**

Πίνακας 3.9 Πίνακας αριθμητικών συμβόλων του μηνύματος

$$\begin{pmatrix} 19 & 8 & 12 \\ 4 & 19 & 14 \\ 18 & 19 & 20 \\ 3 & 24 & \dots \end{pmatrix}$$

Είναι ολοφάνερο ότι έχουμε μία κενή θέση στον πίνακά μας η οποία συνηθίζεται να συμπληρώνεται με έναν αριθμό το οποίο εύκολα θα θεωρηθεί ως άσχετος από αυτόν που θα λάβει το κρυπτογραφημένο κείμενο.

Εμείς θα συμπληρώσουμε το κενό του πίνακά μας με τον αριθμό είκοσι τρία που αντιστοιχεί στο γράμμα Χ.

Συνεπώς έχουμε:

Πίνακας 3.10 Ολοκληρωμένος πίνακας αριθμητικών συμβόλων του μηνύματος

$$\mu=\begin{pmatrix}19 & 8 & 12\\ 4 & 19 & 14\\ 18 & 19 & 20\\ 3 & 24 & 23\end{pmatrix}$$

Τώρα θα πρέπει να υπολογίσουμε το γινόμενο των πινάκων k και μ με τον κλασσικό τρόπο πολλαπλασιασμού πινάκων αλλά θα πρέπει να έχουμε στο μυαλό μας ότι κάθε φορά που ένα αποτέλεσμα έστω x είναι πάνω από εικοσιπέντε θα πρέπει να το αντικαταστήσουμε με έναν αριθμό y={0…….25} τέτοιον ώστε y ≡ x (mod 26) Το αποτέλεσμα του πολλαπλασιασμού μας είναι ο εξής πίνακας :

Πίνακας 3.11 Γινόμενο πινάκων k και μ

$$\gamma=\begin{pmatrix}23 & 16 & 2\\ 7 & 9 & 0\\ 21 & 17 & 6\\ 15 & 1 & 19\end{pmatrix}$$

Τώρα γράφουμε τα ψηφία του πίνακά μας ως εξής και ταυτόχρονα τα αντικαθιστούμε με το αντίστοιχο γράμμα του αλφάβητου για να πάρουμε το κρυπτογραφημένο κείμενο και έχουμε:[41]

Πίνακας 3.12 Κρυπτογραφημένο κείμενο

ΚΡΥΠΤΟΓΡΑΦΗΜΕΝΟ ΚΕΙΜΕΝΟ											
23	16	2	7	9	0	21	7	6	15	1	19
X	Q	C	H	J	A	V	R	G	P	B	T

3.7.1.2 Δίκτυο Feistel

Το δίκτυο Feistel αποτελεί μία άλλη μεγάλη κατηγορία αλγόριθμων κρυπτογράφησης τμήματος. Στην εν λόγω περίπτωση έχουμε τον διαχωρισμό του κειμένου προς κρυπτογράφηση σε δύο τμήματα, το κάθε μέρος επιδέχεται διαφορετικές τροποποιήσεις και τελικά το ένα "ενημερώνει' το άλλο.

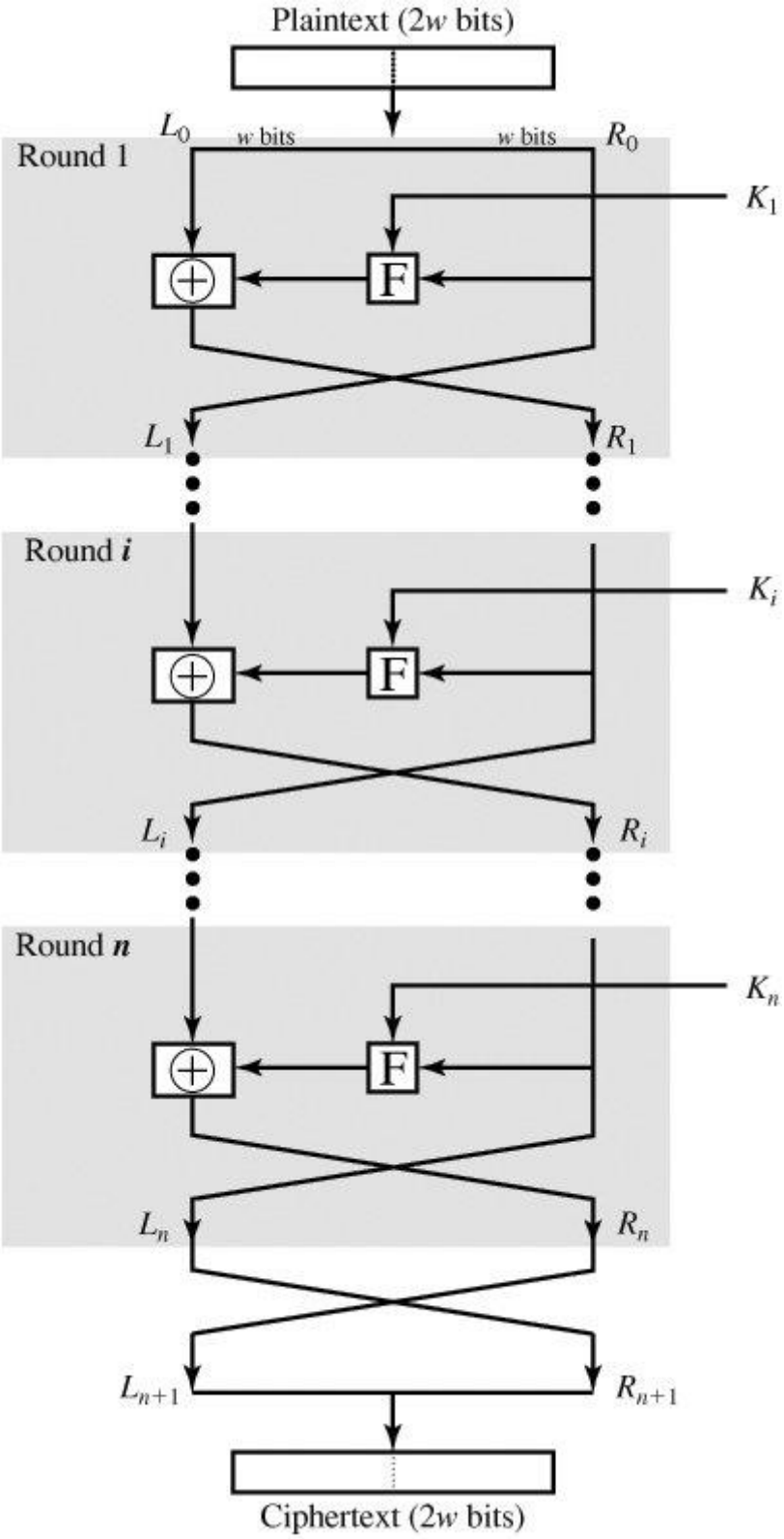
Τα βήματα του αλγόριθμου είναι τα εξής:

- Μετατροπή των χαρακτήρων του κειμένου σε κώδικα ASCII
- Διαίρεση των δεδομένων σε τμήματα τα οποία επεξεργάζονται ένα κάθε φορά.
- Η διαδικασία της κρυπτογράφησης έχει δύο εισόδους η μία είναι τα δεδομένα κρυπτογράφησης και η άλλη το κλειδί
- Όταν το τμήμα δεδομένων είναι έτοιμο για την κρυπτογραφική διαδικασία πρώτα διαιρείται σε δύο τμήματα ίσου μήκους τα οποία εμείς ονομάσαμε L_0, στα αριστερά μας και R_0 στα δεξιά μας (βλέπετε σχήμα που ακολουθεί)
- Εκτός από τα δεδομένα μας η είσοδος δέχεται και το κύριο κλειδί κρυπτογράφησης k.
- Τα δεδομένα περνάνε n γύρους εκτέλεσης και είναι απαραίτητο να επισημάνουμε ότι το n εξαρτάται από τον σχεδιασμό του αλγορίθμου
- Κάθε γύρος κρυπτογράφησης έχει ως εισόδους τα L_{I-1} και R_{i-1} που προέρχονται από τον προηγούμενο γύρο.
- Σε κάθε γύρω χρησιμοποιείται η ίδια μέθοδος κρυπτογράφησης και ένα διαφορετικό υπό κλειδί (sub key) το οποίο παράγεται από το κύριο κλειδί (master key)
- Όλοι οι γύροι επεξεργασίας δομούνται ακριβώς με τον ίδιο τρόπο.
- Στο αριστερό μισό των δεδομένων λαμβάνει χώρα κάποια αντικατάσταση και αυτό γίνεται εφικτό με την εφαρμογή μίας συνάρτησης γύρου (round fuction) F στο δεξί μισό των δεδομένων, με συνδυασμό της εξόδου της συνάρτησης και του αριστερού μισού των δεδομένων με τον τελεστή αποκλειστικής διάζευξης (exclusive OR, XOR).Η συνάρτηση γύρου έχει την ίδια μορφή σε κάθε γύρω αλλά πραγματοποιείται με το υπό κλειδί κάθε γύρου.
- Τέλος, εκτελείται μία μετάθεση η οποία γίνεται με την εναλλαγή των δύο μισών του τμήματος δεδομένων.

  Η υλοποίηση του δικτύου Feistel εξαρτάται από τους παρακάτω παράγοντες :

- Το μέγεθος τμημάτων είναι ζωτικής σημασίας για την συγκεκριμένη υλοποίηση διότι όσο μεγαλύτερο είναι το μέγεθος του τμήματος δεδομένων τόσο πιο αυξημένη είναι η ασφάλεια και η αποφυγή υποκλοπών αλλά μειώνεται η ταχύτητα κωδικοποίησης και αποκωδικοποίησης.
- Ένα κλειδί μεγάλου μήκους σημαίνει αυτόματα μεγαλύτερη ασφάλεια αλλά και πάλι θα έχουμε επιβράδυνση της κωδικοποίησης και της αποκωδικοποίησης.
- Ένας μόνο γύρος δεν προσφέρει επαρκή ασφάλεια. Ο τυπικός αριθμός γύρων είναι δεκαέξι.

- Η μεγάλη πολυπλοκότητα του αλγόριθμου παραγωγής υπό κλειδιών κάνει την κρυπτανάλυση ιδιαίτερα δύσκολη
- Τέλος η μεγάλη πολυπλοκότητα της συνάρτησης γύρου καθιστά δύσκολη την κρυπτανάλυση[41] [42][43][44]



Εικόνα 3.3 Δίκτυο Feistel [44]

3.8 Μέθοδοι Γεμίσματος

Όπως είδαμε στην ανάλυση του αλγόριθμου Hill και όπως θα δούμε παρακάτω για να προχωρήσει η διαδικασία της κρυπτογράφησης θα πρέπει τα δεδομένα που επιθυμούμε να αποστείλουμε να τα χωρίσουμε σε τμήματα καθορισμένου μεγέθους.

3.8.1 Μηδενικό γέμισμα

Ας πούμε ότι τα δεδομένα μας έχουν μήκος σαράντα πέντε bytes.Θα πρέπει να χωρίσουμε τα δεδομένα σε τρία τμήματα των δεκαέξι, δεκαέξι και δεκατριών bytes. Στο μηδενικό γέμισμα το τελευταίο τμήμα των δεδομένων συμπληρώνεται με μηδενικά για να φτάσει τα δεκαέξι bytes διότι πολύ απλά πρέπει να είναι ίσο με τα άλλα δύο. Το συνολικό μήκος του μηνύματός μας θα είναι σαράντα οκτώ bytes.

Εικόνα 3.4 Γέμισμα με μηδενικά bytes

ΑΡΧΙΚΟ ΜΗΝΥΜΑ ΣΑΡΑΝΤΑΠΕΝΤΕ BYTES

ΜΗΝΥΝΑ ΔΕΚΑΕΞΙ BYTES

ΜΗΝΥΝΑ ΔΕΚΑΕΞΙ BYTES

ΜΗΝΥΝΑ ΔΕΚΑΤΡΙΩΝ BYTES

0 0 0

Δεν υπάρχει κάτι στο κρυπτογράφημα που να δηλώνει ότι το μήνυμα είχε αρχικά μήκος σαράντα πέντε bytes.

Το αποκρυπτογραφημένο μήνυμα φυσικά θα είναι μήκους σαράντα οκτώ bytes και θα πρέπει να δούμε πόσα δεδομένα από τα αποσταλέντα αποτελούν το μήνυμά μας με την βοήθεια κάποιας εφαρμογής.

Αυτό δεν αποτελεί πάντα πρόβλημα διότι τα δεδομένα προς αποστολή μπορεί να είναι προκαθορισμένου μήκους για παράδειγμα σαράντα πέντε bytes ή ίσως τα δεδομένα να έχουν ένα πεδίο μήκους.

3.8.2 Τυχαίο γέμισμα

Είναι παρόμοιο με αυτό που περιγράψαμε παραπάνω αλλά οι τιμές γεμίσματος επιλέγονται τυχαία.

Με το μηδενικό γέμισμα ένας οποιοσδήποτε εχθρός γνωρίζει ότι τα τελευταία ψηφία του μηνύματος είναι πιθανόν να είναι μηδέν. Βασικά οποιαδήποτε πληροφορία γνωρίζει κάποιος εν δυνάμει υποκλοπέας σχετικά με ένα μήνυμα προς αποστολή είναι χρήσιμη προκειμένου να την χρησιμοποιήσει για να "σπάσει" την κρυπτογράφησή του.

Θα πρέπει όμως να πούμε ότι αν ο αλγόριθμος κρυπτογράφησης που χρησιμοποιείται είναι ισχυρός, αυτό αποτελεί θέμα ήσσονος σημασίας

3.8.3 Γέμισμα PKCS7

Μία εναλλακτική μορφή γεμίσματος είναι η ονομαζόμενη PKCS7. Αντί για μηδέν ή τυχαία δεδομένα η τιμή του γεμίσματος είναι η τιμή των bytes που πρέπει να γεμίσουν.

Στο παράδειγμα που δείξαμε τα άδεια bytes ήταν τρία συνεπώς η τιμή γεμίσματος κάθε byte θα είναι τρία.

Εικόνα 3.5 Γέμισμα με τρία bytes

3.8.4 Γέμισμα με μονάδες και μηδέν

Μία άλλη μέθοδος είναι το γέμισμα με μονάδες και μηδέν. Σε αυτή τη μέθοδο το πρώτο byte το οποίο γεμίζει έχει την τιμή ένα(1) και όσα bytes απομένουν για γέμισμα έχουν την τιμή μηδέν (0).[45]

Εικόνα 3.6 Γέμισμα με μονάδες και μηδέν

3.9 Μέθοδοι λειτουργίας αλγορίθμων κρυπτογράφησης τμήματος

Μέθοδος λειτουργίας αλγόριθμων κρυπτογράφησης τμήματος ονομάζουμε έναν προτεινόμενο τρόπο χρήσης τους για την κρυπτογράφηση μία σειρά από bits τα οποία αποτελούν το κείμενο προς αποστολή έτσι ώστε να παραχθεί το κρυπτοκείμενο.[46]. Μία τέτοια μέθοδος είναι απαραίτητη αφού οι αλγόριθμοι τμήματος προσφέρουν τρόπους κρυπτογράφησης συμβολοσειρών n-bits καθορισμένου μήκους.

Οι μέθοδοι λειτουργίας είναι οι εξής :

- Electronic Code Book (ECB)

- Cipher Block Chaining (CBC)
- Cipher Feedback (CFB)
- Output Feedback (OFB)
- Counter (CTR)

3.9.1 Electronic Code Book (ECB)

Η μέθοδος Electronic Code Book έχει ως εξής : το κείμενο προς αποστολή χωρίζεται σε τμήματα(blocks) τα οποία κρυπτογραφούνται ατομικά και ανεξάρτητα με τη χρήση του κλειδιού κρυπτογράφησης και έτσι κάθε κρυπτογραφημένο τμήμα μπορεί να αποκρυπτογραφηθεί ατομικά. Επίσης η συγκεκριμένη μέθοδος μπορεί να υποστηρίξει ξεχωριστό κλειδί κρυπτογράφησης για κάθε τμήμα.

Θα πρέπει να επισημάνουμε ότι υπάρχει πιθανότητα σε αυτή την μέθοδο να χρησιμοποιήσουμε μία μέθοδος γεμίσματος από τις προαναφερθείσες.

Ας πούμε λοιπόν ότι το κείμενό μας χωρίζεται στα τμήματα P_1, P_2 …..P_q. Κατά συνέπεια τα κρυπτογραφήματα θα είναι τα C_1,C_2..C_Q [35]

Κάθε τμήμα έχει μία ορισμένη τιμή η οποία συσχετίζεται με ένα κρυπτοκείμενο και το αντίστροφο. Συνεπώς παρόμοια κείμενα με παρόμοια κλειδιά κρυπτογράφησης όταν κρυπτογραφούνται παράγουν παρόμοια κρυπτογραφήματα.[46] [47]

3.9.2 Cipher Block Chaining (CBC)

Στη μέθοδο Αλυσιδωτής Κρυπτογράφησης Τμημάτων πάλι το κείμενο προς αποστολή θα πρέπει να χωριστεί σε ίσα τμήματα.

Σε αυτή τη μέθοδο χρησιμοποιείται το Διάνυσμα Αρχικοποίησης (Initialization Vector, IV) το οποίο μπορεί να είναι ένα τυχαίο τμήμα κειμένου και χρησιμοποιείται για να κάνει το κρυπτογράφημα κάθε τμήματος μοναδικό.

Το πρώτο τμήμα του κειμένου προς κρυπτογράφηση και το Διάνυσμα Αρχικοποίησης συνδυάζονται με χρήση της λειτουργίας XOR. Το αποτέλεσμα αυτού του συνδυασμού κρυπτογραφείται με την χρήση βέβαια του κλειδιού κρυπτογράφησης και αποτελεί το πρώτο κρυπτογράφημα.

Το πρώτο κρυπτογραφημένο κείμενο χρησιμοποιείται ως Διάνυσμα Αρχικοποίησης για το δεύτερο τμήμα του κειμένου προς κρυπτογράφηση και η ίδια διαδικασία θα λάβει χώρα για όλα τα τμήματα του κειμένου προς κρυπτογράφηση.

Στον μηχανισμό λήψης του μηνύματος το κρυπτογράφημα χωρίζεται και πάλι σε τμήματα. Το πρώτο τμήμα του κρυπτοκειμένου αποκρυπτογραφείται με την χρήση φυσικά του ίδιου κλειδιού που χρησιμοποιήθηκε για την κρυπτογράφηση. Το αποτέλεσμα της αποκρυπτογράφησης θα συνδυαστεί με την λειτουργία XOR και έτσι ο λήπτης θα πάρει το αρχικό κείμενο.

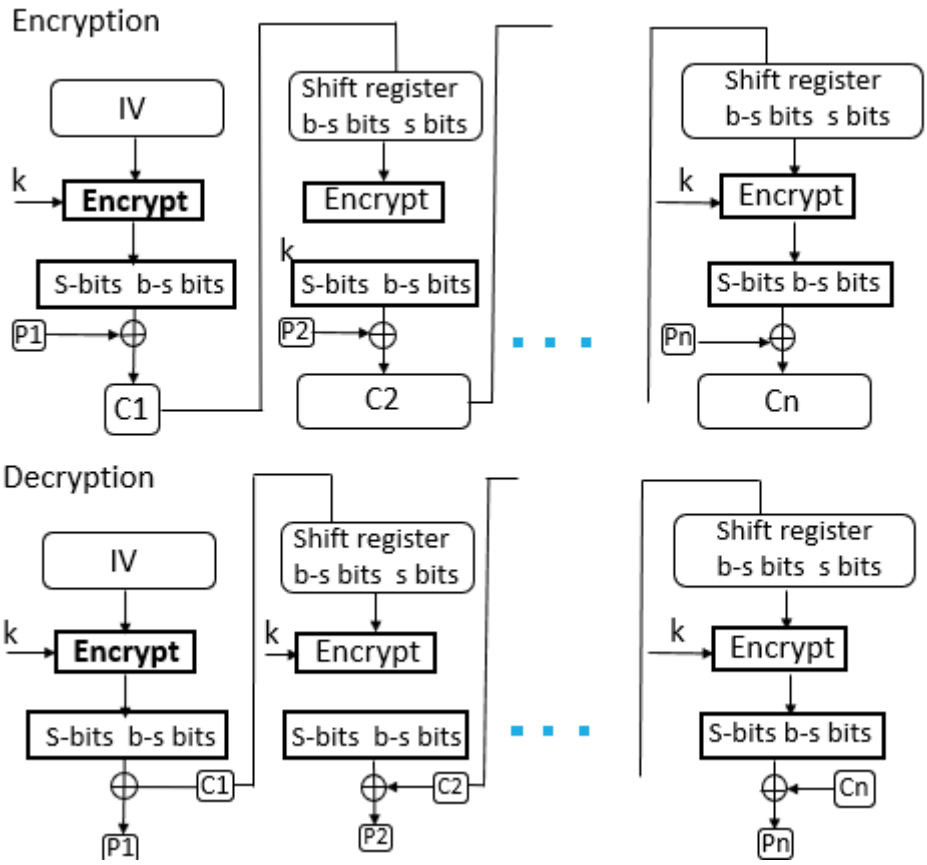
Το δεύτερο τμήμα του κρυπτογραφήματος αποκρυπτογραφείται με την χρήση του ίδιου κλειδιού και το αποτέλεσμα της αποκρυπτογράφησης συνδυάζεται με την λειτουργία XOR και το πρώτο τμήμα του κρυπτογραφήματος και έτσι έχουμε το δεύτερο τμήμα του κειμένου αποκρυπτογραφημένο.
Η ίδια διαδικασία ακολουθείται για όλα τα τμήματα του κρυπτογραφήματος.

3.9.3 Cipher Feedback (CFB)

Στη μέθοδο ανάδρασης κρυπτογραφημάτων τα δεδομένα κρυπτογραφούνται με την μορφή μονάδων των οκτώ bits
Για να αρχίζουμε να υλοποιούμε την συγκεκριμένη μέθοδο θα πρέπει πρώτα να ορίσουνε δύο παραμέτρους, τις εξής:

- την παράμετρο k για την οποία θα ισχύει $1 \leq k \leq n$ και ορίζει το μέγεθος της μεταβλητής ανάδρασης
- την παράμετρο j για την οποία θα ισχύει $1 \leq j \leq k$ η οποία ορίζει τον αριθμό των bits του απλού κειμένου που κρυπτογραφούνται σε κάθε τμήμα. Πρακτικά το πιο διαδεδομένο σχήμα είναι αυτό στο οποίο ισχύει j=k=8 [35]

Και σε αυτή τη μέθοδο χρησιμοποιείται το Διάνυσμα Αρχικοποίησης (Initialization Vector, IV) το οποίο αρχικοποιείται, κρυπτογραφείται με το κλειδί κρυπτογράφησης και έτσι σχηματίζεται το κρυπτογράφημα.
Το εν λόγω κρυπτογράφημα αποθηκεύεται σε έναν καταχωρητή ολίσθησης (shift register).
Στη συνέχεια τα bits που βρίσκονται αριστερά στο Διάνυσμα Αρχικοποίησης (j-bits) συνδυάζονται με χρήση της λειτουργίας XOR με τα πρώτα j- bits του κείμενου προς κρυπτογράφηση
Αυτή η διαδικασία θα σχηματίσει το πρώτο τμήμα του κρυπτογραφήματος το οποίο θα είναι έτοιμο προς αποστολή.
Αμέσως μετά τα περιεχόμενα του καταχωρητή ολίσθησης ολισθαίνουν αριστερά κατά j bits. Συνεπώς τα j bits που βρίσκονται στα δεξιά του καταχωρητή αρχικοποίησης είναι τα λιγότερο σημαντικά δεδομένα. Γι' αυτό ακριβώς τοποθετείται εκεί το κρυπτογράφημα. Αυτή η διαδικασία επαναλαμβάνεται έως το τέλος της κρυπτογράφησης.



Εικόνα 3.7

Μέθοδος ανάδρασης κρυπτογραφημάτων

3.9.4 Output Feedback (OFB) Mode

Η μέθοδος ανάδρασης εξόδου λειτουργεί σε γενικές γραμμές παρόμοια με την μέθοδο ανάδρασης κρυπτογραφημάτων που μόλις περιγράψαμε. Η μόνη διαφορά είναι ότι στη μέθοδο ανάδρασης κρυπτογραφημάτων το κρυπτογράφημα χρησιμοποιείται για το επόμενο στάδιο της κρυπτογραφικής διαδικασίας.

Αντίθετα, στην μέθοδο ανάδρασης εξόδου το κρυπτογράφημα του Διανύσματος Αρχικοποίησης χρησιμοποιείται γι' αυτό το στάδιο.

Αναλυτικότερα, το Διάνυσμα Αρχικοποίησης κρυπτογραφείται, πάντα με την βοήθεια του κλειδιού και έτσι παράγεται το αντίστοιχο κρυπτοκείμενο. Το κείμενο προς κρυπτογράφηση και τα 8bits στα αριστερά του Διανύσματος Αρχικοποίησης συνδυάζονται με τη λειτουργία XOR και παράγουν το κρυπτοκείμενο.

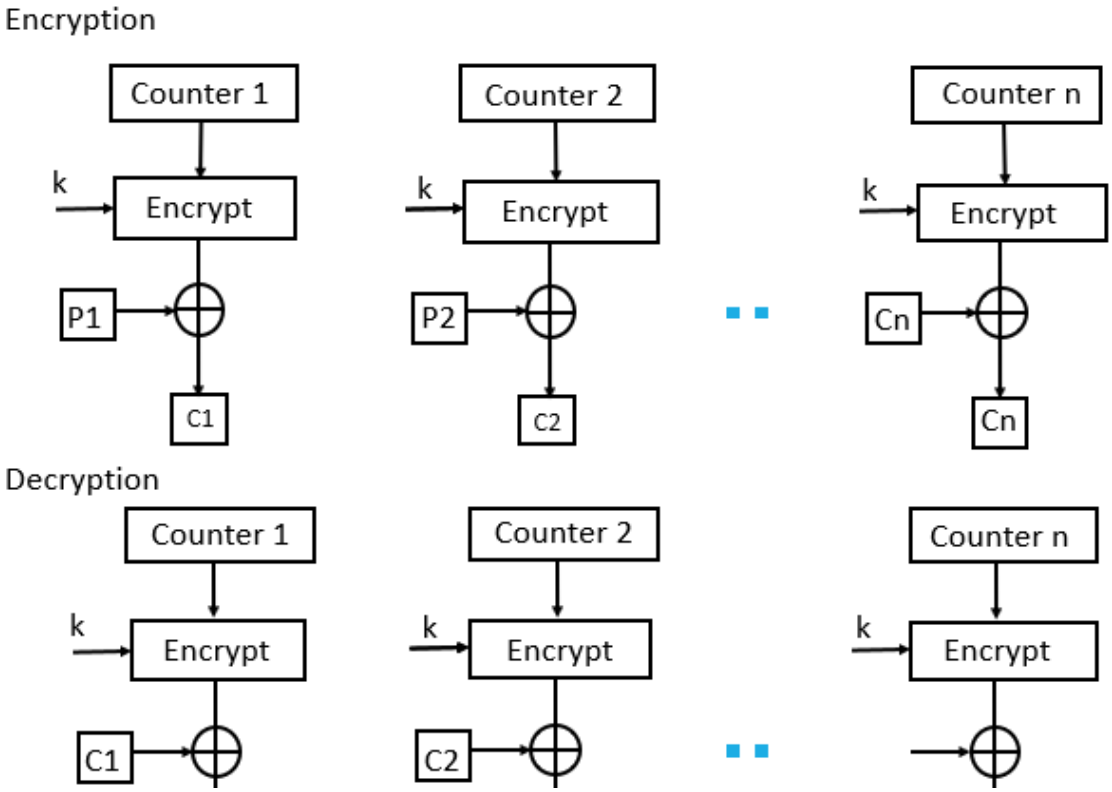
Στο επόμενο στάδιο το κρυπτοκείμενο που παρήχθη στο προηγούμενο στάδιο χρησιμοποιείται ως Διάνυσμα Αρχικοποίησης για την επόμενη επανάληψη και η ίδια διαδικασία ακολουθείται για όλα τα τμήματα.

3.9.5 Counter (CTR) Mode

Σε αυτή την περίπτωση χρησιμοποιείται η ακολουθία των αριθμών σαν είσοδο για τον αλγόριθμο. Όταν ένα τμήμα κρυπτογραφείται για να συμπληρωθεί ο επόμενος καταχωρητής χρησιμοποιείται η επόμενη τιμή του μετρητή. Η τιμή του μετρητή θα αυξάνεται κατά μία μονάδα.

Για την κρυπτογράφηση ο πρώτος μετρητής κρυπτογραφείται με την χρήση κλειδιού και στη συνέχεια το κείμενο συνδυάζεται με τη λειτουργία XOR με το κρυπτογραφημένο αποτέλεσμα για να παραχθεί το κρυπτοκείμενο. Ο μετρητής θα αυξηθεί κατά μία μονάδα για το επόμενο στάδιο και η ίδια διαδικασία θα ακολουθηθεί για όλα τα τμήματα του κειμένου προς αποστολή.

Για την αποκρυπτογράφηση θα χρησιμοποιηθεί η ίδια ακολουθία. Για την μετατροπή ενός κρυπτογραφήματος στο αρχικό κείμενο κάθε τμήμα του κρυπτοκειμένου συνδυάζεται με την λειτουργία XOR με τον κρυπτογραφημένο μετρητή.



Εικόνα 3.8 Μέθοδος μετρητή

3.10 Συμμετρικοί αλγόριθμοι κρυπτογράφησης

Στις παραγράφους που ακολουθούν θα προσπαθήσουμε να περιγράψουμε συνοπτικά κάποιους από τους σημαντικότερους αλγόριθμους κρυπτογράφησης.

3.10.1 Ο αλγόριθμος κρυπτογράφησης DES (Data Encryption Standard)

Ο DES αποτελεί έναν ιστορικό αλγόριθμο και ένα από τα χαρακτηριστικά του οποίου είναι το κλειδί κρυπτογράφησης που έχει μήκος εξήντα τέσσερα bits. Τα πενήντα έξι bits αποτελούν το καθαυτό κλειδί και τα υπόλοιπα οκτώ bits είναι τα bits ισοτιμίας (parity bits).

Ο αλγόριθμος DES έχει παίξει σημαντικό ρόλο στην ασφάλεια δεδομένων. Θα πρέπει να αναφέρουμε ότι ο συγκεκριμένος αλγόριθμος έχει βρεθεί ευάλωτος σε ισχυρές επιθέσεις και γι' αυτό η δημοτικότητα του παράκμασε ελαφρώς.

Ο DES είναι ένας αλγόριθμος τμήματος ο οποίος έχει την δυνατότητα κρυπτογράφησης τμημάτων μεγέθους εξήντα τεσσάρων bits. Αυτό σημαίνει πως κείμενο εξήντα τεσσάρων bits αποτελεί την είσοδο του αλγορίθμου και στη συνέχεια παράγεται κρυπτογράφημα ίδιου μεγέθους Αφού ο DES είναι συμμετρικός αλγόριθμος το ίδιο κλειδί χρησιμοποιείται και για την κρυπτογράφηση και για την αποκρυπτογράφηση με μικρές διαφορές. [47]

Πριν ακόμα αρχίσει η διαδικασία κρυπτογράφησης με την βοήθεια του αλγόριθμου DES κάθε όγδοο bit του κλειδιού δηλαδή κάθε bit που βρίσκεται στην όγδοη θέση απορρίπτεται για να παραχθεί το κλειδί της κρυπτογράφησης όπως βλέπουμε παρακάτω(έντονοι αριθμοί):

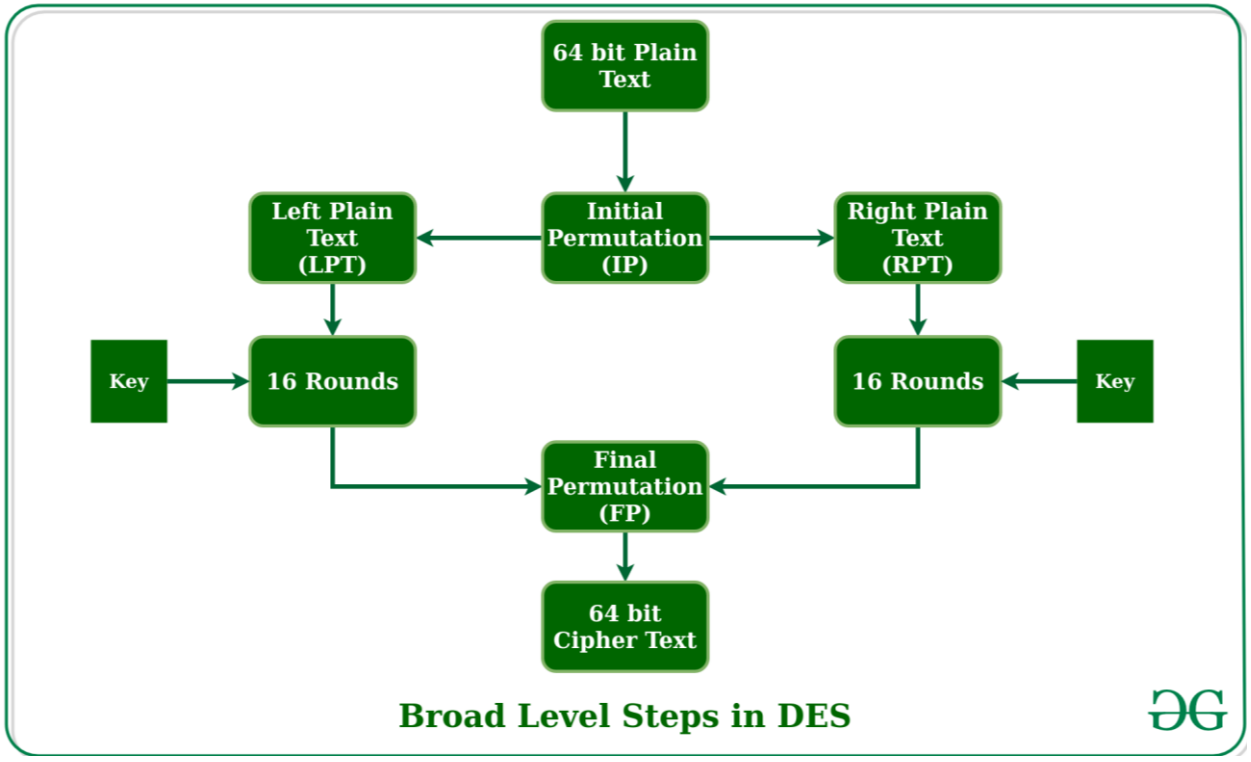
Πίνακας 3.13 Πίνακας bits που απορρίπτονται για την παραγωγή κλειδιού

ΠΙΝΑΚΑΣ ΑΠΟΡΡΙΠΤΕΩΝ BITS															
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	26	27	28	29	30	31	**32**
33	34	35	36	37	38	39	**40**	41	42	43	44	45	46	47	**48**
49	50	51	52	53	54	55	**56**	57	58	59	60	61	62	63	**64**

Με αυτό τον τρόπο παράγεται το κλειδί κρυπτογράφησης των 56 bits. Ο αλγόριθμος DES βασίζεται σε δύο θεμελιώδη χαρακτηριστικά της κρυπτογραφικής επιστήμης τα οποία είναι η αντικατάσταση(substitution) η οποία ονομάζεται και σύγχυση (confusion) και η μετατόπιση (transposition) που ονομάζεται και διάχυση (diffusion).

Ο αλγόριθμος DES αποτελείται από δεκαέξι βήματα ή γύρους σε κάθε ένα εκ των οποίων εφαρμόζονται αντικαταστάσεις και μετασχηματισμοί. Ας δούμε γενικά τη διαδικασία :

- Στο πρώτο βήμα το τμήμα του κειμένου μεγέθους εξήντα τεσσάρων bits περνάει στην αρχική μέθοδο μετασχηματισμού (Initial Premutation Function, IP)
- Ο αρχικός μετασχηματισμός εφαρμόζεται στο κείμενο.
- Στη συνέχεια η μέθοδος μετασχηματισμού παράγει δύο τμήματα των τριάντα δύο bits τα οποία έχουν μετατεθεί και αυτά ονομάζονται Left Plain Text (LPT) και Right Plain Text (RPT) όπως βλέπουνε στο σχήμα που ακολουθεί.
- Τώρα, κάθε Left Plain Text και Right Plain Text περνούν δεκαέξι γύρους κρυπτογραφικής επεξεργασίας
- Τέλος τα Left Plain Text και Right Plain Text ενώνονται και εφαρμόζεται ένας τελικός μετασχηματισμός στο επανενωμένο τμήμα
- Τελικά παράγεται ένα κρυπτοκείμενο εξήντα τεσσάρων bits.



Εικόνα 3.9 Βήματα αλγόριθμου Des

Αρχικός μετασχηματισμός (Initial Premutation)

Όπως έχουμε επισημάνει ο αρχικός μετασχηματισμός (Initial Premutation, IP) εφαρμόζεται μόνο μία φορά και πριν τον πρώτο γύρο και δηλώνει τον τρόπο που πρέπει να προχωρήσει η μετατόπιση (transposition).

Για παράδειγμα, όπως θα δούμε και στον πίνακα που ακολουθεί, στα πλαίσια της μετατόπισης αντικαθίσταται το πρώτο bit του κειμένου προς κρυπτογράφηση με το πεντηκοστό όγδοο του κειμένου προς κρυπτογράφηση, το δεύτερο bit του με το πεντηκοστό και τα λοιπά.

Απ' ότι καταλαβαίνουμε αυτό δεν είναι τίποτα άλλο από μία ανακατάταξη των θέσεων των bits του τμήματος του κειμένου προς κρυπτογράφηση και ο ίδιος κανόνας ισχύει για τις υπόλοιπες θέσεις bits όπως βλέπουμε στον πίνακα που ακολουθεί:

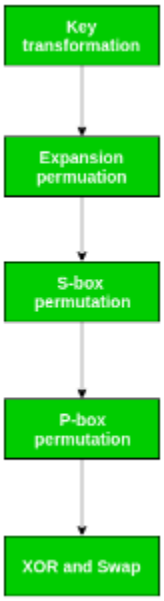
Πίνακας 3.14 Πίνακας ανακατάταξης bits

ΠΙΝΑΚΑΣ ΑΝΑΚΑΤΑΤΑΞΗΣ BITS

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**

Αφού γίνει ο αρχικός μετασχηματισμός το αρχικό, μετασχηματισμένο τώρα, πιά τμήμα κειμένου των εξήντα τεσσάρων bits χωρίζεται σε δύο τμήματα των τριάντα δύο bits.

Σε γενικές γραμμές καθένας από τους δεκαέξι γύρους αποτελείται από τα ακόλουθα βήματα:



Εικόνα 3.10 Βήματα κάθε γύρου του αλγόριθμου Des

Πρώτο βήμα: μετατροπή κλειδιού

Έχουμε επισημάνει ήδη ότι το κλειδί των εξήντα τεσσάρων bits μετατρέπεται σε κλειδί πενήντα έξι bits αφού απορριφθεί κάθε bit το οποίο βρίσκεται στην όγδοη θέση του αρχικού κλειδιού.

Από το κλειδί των πενήντα έξι bits παράγεται ένα άλλο υπό κλειδί των 48 bits κατά τη διάρκεια κάθε γύρου με την χρήση μίας διαδικασίας η οποία ονομάζεται μετατροπή κλειδιού.

Στα πλαίσια αυτής της διαδικασίας το κλειδί των πενήντα έξι bits χωρίζεται σε δύο τμήματα των είκοσι οκτώ bits. Αυτά τα τμήματα μετατοπίζονται κυκλικά κατά μία ή δύο θέσεις ανάλογα με τον γύρο.

Για παράδειγμα εάν βρισκόμαστε στον πρώτο, δεύτερο, ένατο ή δέκατο έκτο γύρω, τότε η μετατόπιση γίνεται μόνο κατά μία θέση ενώ για τους άλλους γύρους η κυκλική μετατόπιση γίνεται κατά δύο θέσεις.

Τα παραπάνω απεικονίζονται στον πίνακα που ακολουθεί :

Πίνακας 3.15 Πίνακας μετατόπισης bits κλειδιού ανά γύρο

ΓΥΡΟΙ															
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ΑΡΙΘΜΟΣ ΤΩΝ BIT ΤΟΥ ΚΛΕΙΔΙΟΥ ΠΟΥ ΜΕΤΑΤΟΠΙΖΟΝΤΑΙ ΑΝΑ ΓΥΡΟ															

1 1 2 2 2 2 2 2 1 2 2 2 2 2 2 1

Μετά από μια κατάλληλη μετατόπιση επιλέγονται σαράντα οκτώ από τα πενήντα έξι bits. Ας δούμε τον πίνακα που ακολουθεί:

Πίνακας 3.16 Πίνακας τελικής επιλογής bits

ΠΙΝΑΚΑΣ ΤΕΛΙΚΗΣ ΕΠΙΛΟΓΗΣ BITS

14	17	11	24	1	5	3	28	15	6	21	10
23	19	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	33	48
44	49	39	56	34	53	46	42	50	36	29	32

Παρατηρούμε ότι μετά την μετατόπιση το δέκατο τέταρτο bit μετακινείται στην πρώτη θέση το δέκατο έβδομο bit μετακινείται στην δεύτερη θέση και τα λοιπά. Παρατηρούμε ότι υπάρχουν μόνο σαράντα οκτώ θέσεις bit.

Εφόσον η διαδικασία μετατροπής του κλειδιού περιλαμβάνει μετασχηματισμό (premutation) όπως επίσης και επιλογή σαράντα οκτώ bits από τα αρχικά πενήντα έξι ονομάζεται Μετασχηματισμός Συμπίεσης (Compression Permutation).

Λόγω αυτής της διαδικασίας σε κάθε γύρο χρησιμοποιείται ένα διαφορετικό υποσύνολο bits για το κλειδί κρυπτογράφησης πράγμα το οποίο δυσκολεύει αρκετά το σπάσιμο του συγκεκριμένου αλγορίθμου.

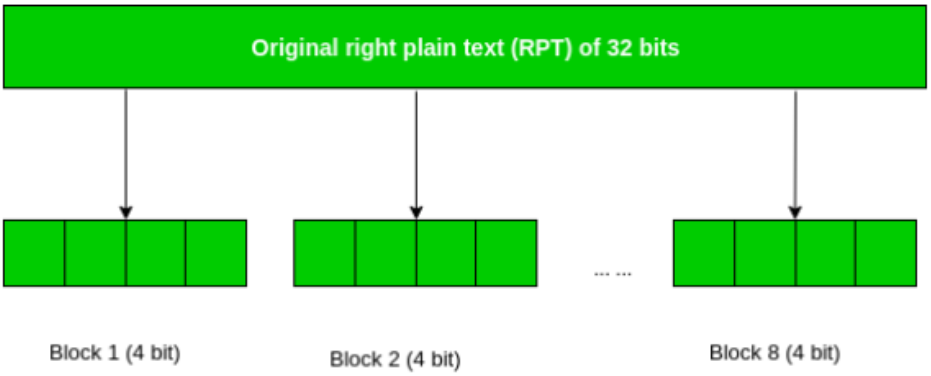
Δεύτερο βήμα: επέκταση μετασχηματισμού

Όπως ήδη αναφέραμε μετά τον πρώτο μετασχηματισμό το κείμενό μας προς κρυπτογράφηση χωρίστηκε σε δύο τμήματα των τριάντα δύο bits έκαστο τα οποία ονομάζονται Left Plain Text(LPT) and Right Plain Text(RPT).Κατά τη διάρκεια της επέκτασης μετασχηματισμού το τμήμα Right Plain Text(RPT) επεκτείνεται από τα τριάντα δύο bits στα σαράντα οκτώ. Αυτό συμβαίνει καθώς το τμήμα Right Plain Text(RPT) χωρίζεται σε οκτώ τμήματα των τεσσάρων bits έκαστο. Στη συνέχεια το καθένα από αυτά τα τμήματα επεκτείνεται σε ένα ανάλογο block των έξι bits.

Αυτή η διαδικασία έχει ως αποτέλεσμα την επέκταση και τον μετασχηματισμό των bits εισόδου ενώ παράλληλα δημιουργείται και έξοδος. Η συνάρτηση μετασχηματισμού του κλειδιού συμπιέζει το κλειδί των πενήντα έξι bits στα σαράντα οκτώ.

Στη συνέχεια η διαδικασία επέκτασης του μετασχηματισμού επεκτείνει το τμήμα Right Plain Text(RPT) από τα τριάντα δύο bits στα σαράντα οκτώ.

Το κλειδί των σαράντα οχτώ bits συνδυάζεται με την λειτουργία XOR με το Right Plain Text(RPT) των σαράντα οκτώ bits και το αποτέλεσμα παραδίδεται στο επόμενο βήμα το οποίο είναι ο μετασχηματισμός S-Box [47]



Εικόνα 3.11Διαίρεση των τριάντα δύο bits σε τμήματα των οκτώ bits

Ο μετασχηματισμός S-Box

Ο μετασχηματισμός S-Box είναι η διαδικασία η οποία δέχεται ως είσοδο τα σαράντα οχτώ bits από την λειτουργία XOR στα οποία συμπεριλαμβάνονται το συμπιεσμένο κλειδί και το τμήμα Right Plain Text(RPT) και δημιουργείται μία έξοδο των τριάντα δύο bits με την χρήση της μεθόδου της αντικατάστασης.

Η αντικατάσταση υλοποιείται από τα οκτώ κουτιά αντικατάστασης (substitution boxes) τα οποία είναι επίσης γνωστά ως S-Boxes. Καθένα από αυτά τα οκτώ κουτιά έχει μία είσοδο των έξι bits και μία έξοδο των τεσσάρων bits. Το τμήμα των σαράντα οκτώ bits διαιρείται σε οκτώ υπό τμήματα καθένα από τα οποία περιλαμβάνει έξι bits και κάθε τμήμα εισέρχεται στο S-Box.

Η αντικατάσταση σε κάθε κουτί τηρεί έναν προκαθορισμένο κανόνα και εξαρτάται από έναν πίνακα τεσσάρων σειρών και δεκαέξι στηλών. Η ακολουθία των bits ένα και έξι της εισόδου αντιπροσωπεύουν τέσσερις σειρές και η ακολουθία των bits δύο έως πέντε αντιπροσωπεύουν δεκαέξι στήλες.

Αφού κάθε S-Box έχει τον δικό του πίνακα απαιτούνται οχτώ πίνακες για να συμπληρωθεί η έξοδός μας

Τα bits εισόδου ορίζουν μία καταχώρηση στον S-Box με έναν ιδιαίτερο τρόπο :θεωρήστε μία είσοδο S-Box των έξι bits με ετικέτες $b_1, b_2, b_3, b_4, b_5, b_6$.

Τα b_1 και b_2 ενώνονται για να σχηματίσουν έναν αριθμό των δύο bits από το μηδέν έως το τρία ο οποίος αντιστοιχεί σε μία σειρά του πίνακα.

Τα μεσαία τέσσερα bits δηλαδή τα b_2, b_3, b_4, b_5 συνδυάζονται για να σχηματίσουν έναν αριθμό τεσσάρων bits από το μηδέν έως το δεκαπέντε ο οποίος αντιστοιχεί σε μία στήλη του πίνακα.

Ας δούμε ένα παράδειγμα :

Πίνακας 3.17 Πρώτο κουτί αντικατάστασης S-Box

S[0]

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	**11**	3	14	10	0	6	13

Ας υποθέσουμε ότι έχουμε έναν αριθμό ο οποίος αποτελείται από έξι bits και είναι ο εξής :1,1,0,0,1,1. Το πρώτο και το τελευταίο bit συνδυάζονται και σχηματίζουν τον αριθμό τρία (11=3) ο οποίος αντιστοιχεί στην τρίτη σειρά και το δεύτερο έως το πέμπτο bit σχηματίζουν τον αριθμό εννέα (1001=9) που αντιστοιχεί στην ένατη στήλη δηλαδή στον αριθμό 11 ο οποίος πρέπει να μετατραπεί στο δυαδικό σύστημα (1011) ο οποίος αποτελεί και την έξοδό μας. Ακολουθούν και οι υπόλοιποι επτά πίνακες.

Πίνακας 3.18 Δεύτερο κουτί αντικατάστασης S-Box

S[1]

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
2	0	14	7	11	11	4	13	1	5	8	12	6	9	3	2	15
3	13	8	10	1	3	15	14	2	11	6	7	12	0	5	14	9

Πίνακας 3.19 Τρίτο κουτί αντικατάστασης S-Box

S[2]

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

Πίνακας 3.20 Τέταρτο κουτί αντικατάστασης S-Box

S[3]

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	5
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	3	15	0	6	10	1	13	18	9	4	5	11	12	7	2	14

Πίνακας 3.21 Πέμπτο κουτί αντικατάστασης S-Box

S[4]

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

Πίνακας 3.22 Έκτο κουτί αντικατάστασης S-Box

S[5]

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

Πίνακας 3.23 Έβδομο κουτί αντικατάστασης S-Box

S[6]

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6

2	1	4	11	13	8	3	7	14	10	15	6	8	0	5	9	2
3	6	11	13	8	1	4	10	7	9	5	0	5	14	2	3	12

Πίνακας 3.24 Όγδοο κουτί αντικατάστασης S-Box

S[7]

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
3	2	1	4	7	4	10	8	13	15	12	9	0	3	5	6	11

Το αποτέλεσμα αυτού του μετασχηματισμού είναι οκτώ τμήματα των τεσσάρων bit τα οποία ενώνονται για να σχηματίσουν ένα block των τριάντα δύο bit και αυτό το τμήμα πάει στο επόμενο βήμα, τον μετασχηματισμό P Box [48][49]

Ο μετασχηματισμός P Box

Η έξοδος των τριάντα δύο bits μετασχηματίζεται σύμφωνα με ένα P Box.Αυτός ο μετασχηματισμός χαρτογραφεί κάθε bit εισόδου σε μία θέση εξόδου. Κανένα bit δεν χρησιμοποιείται δύο φορές και κανένα bit δεν αγνοείται. Αυτή η διαδικασία καλείται ευθύς μετασχηματισμός ή απλά μετασχηματισμός. Ο πίνακας που ακολουθεί δείχνει τη θέση στην οποία μετακινείται κάθε bit (με έντονα γράμματα)

Πίνακας 3.25 Μετασχηματισμός P-Box

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**	**26**	**27**	**28**	**29**	**30**	**31**
15	6	19	20	28	11	27	16	0	14	22	25	4	17	30	9
1	**7**	**23**	**13**	**31**	**26**	**2**	**8**	**18**	**12**	**29**	**5**	**21**	**10**	**3**	**24**

Το αποτέλεσμα του μετασχηματισμού συνδυάζεται με τη λειτουργία XOR με το αριστερό μισό του αρχικού τμήματος των εξήντα τεσσάρων bits. Στη συνέχεια το αριστερό κα το δεξί μισό μετατοπίζονται και αρχίζει ένας καινούργιος γύρος

Ο τελικός μετασχηματισμός

Ο τελικός μετασχηματισμός είναι ο αντίστροφος του αρχικού μετασχηματισμού και περιγράφεται στον ακόλουθο πίνακα :

Πίνακας 3.26 Τελικός μετασχηματισμός

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

Αποκρυπτογράφηση DES

Για την αποκρυπτογράφηση ακολουθείται η ίδια διαδικασία και η μόνη διαφορά είναι ότι τα κλειδιά θα πρέπει να χρησιμοποιηθούν με αντίστροφη σειρά.

3.10.1.1 Ασφάλεια αλγόριθμου DES

Η ασφάλεια του αλγόριθμου DES ήταν πάντα αντικείμενο αμφισβήτησης διότι υπήρχαν ανησυχίες για το μέγεθος του κλειδιού αλλά και για τον ίδιο τον αλγόριθμο και ιδιαίτερα για τις επαναλήψεις και τον σχεδιασμό των S-Boxes. Ο συγκεκριμένος αλγόριθμος αποσύρθηκε το 2005.

3.10.2 Ο αλγόριθμος κρυπτογράφησης Triple DES

Ο αλγόριθμος κρυπτογράφησης Triple DES ή TDEA ή αλλιώς 3DES αποτέλεσε μία πρόταση του W. Tuchman.

Το 1985 προ τυποποιήθηκε στο ANSI X9.17 με σκοπό να χρησιμοποιηθεί στην κρυπτογράφηση οικονομικών εφαρμογών. Το 1999 ενσωματώθηκε ως τμήμα της προτυποποίησης κρυπτογράφησης δεδομένων DES. [31]

Ο συγκεκριμένος αλγόριθμος ωστόσο δεν θα χρησιμοποιείται από τον Δεκέμβριο του 2023 σύμφωνα με το Εθνικό Ινστιτούτο Προτύπων και Τεχνολογίας (National Institute of Standards and Technology, NIST)[58]

3.10.2.1 Ιστορική αναφορά στον αλγόριθμο Triple DES

Όταν το 1994 εκδόθηκε για πρώτη φορά η γραμμική ανάλυση άρχισε η αμφισβήτηση αναφορικά με την ασφάλεια τους αλγόριθμου DES και το 1997 ανακοινώθηκε από το Εθνικό Ινστιτούτο Προτύπων και

Τεχνολογίας ότι προσπάθειες βρισκόταν σε εξέλιξη έτσι ώστε να βρεθεί ένας αλγόριθμος ικανός να τον αντικαταστήσει. Η ανάγκη ενός καινούργιου αλγόριθμου αποτέλεσε επιτακτική ανάγκη αφού η τεχνολογία εξελισσόταν με γοργά βήματα και οι κυβερνό-επιθέσεις ήταν όλο και πιο εντατικές και αποτελεσματικές.
Τελικά το 1998 η distributed.net έσπασε τον αλγόριθμο σε τριάντα εννέα μέρες. [52]
Στην αρχή του 1999 με τη βοήθεια της μηχανής Deep Crack η οποία δημιουργήθηκε από την ομάδα Electronic Frontier Foundation ο αλγόριθμος DES έσπασε σε λίγο περισσότερο από είκοσι δύο ώρες και αυτό το γεγονός αποτέλεσε την ταφόπλακα του συγκεκριμένου αλγόριθμου. [52]
Τελικά ο αλγόριθμος Triple DES προτάθηκε ως μία λύση για έναν πιο ασφαλή αλγόριθμο.

3.10.2.2 Περιγραφή διαδικασίας κρυπτογράφησης του αλγόριθμου Triple DES

Ο αλγόριθμος Triple DES χρησιμοποιεί τρία κλειδιά και τρεις εκτελέσεις του αλγόριθμου DES και τα βήματα είναι τα εξής:

- Κρυπτογράφηση (Encryption)
- Αποκρυπτογράφηση (Decryption)
- Κρυπτογράφηση (Encryption)

Ας δούμε λοιπόν τον τύπο της κρυπτογράφησης :

$$C = E_{K3}[D_{K2}[E_{K1}[P]]] \quad (3.5)$$

Οι όροι του τύπους είναι οι εξής :

- C, είναι το κρυπτογραφημένο κείμενο
- P, είναι το αρχικό κείμενο
- E_K [X], αποτελεί την κρυπτογράφηση του X με το κλειδί K
- D_K [X], είναι η αποκρυπτογράφηση του X με το κλειδί K

Η αποκρυπτογράφηση ακολουθεί ακριβώς την ίδια διαδικασία με αντεστραμμένη χρήση κλειδιών :

$$P = D_{K1}[E_{K2}[D_{K3}[P]]] \quad (3.6)$$

3.10.2.3 Ασφάλεια του κρυπτογραφικού αλγορίθμου Triple DES

Η ασφαλιστική ισχύς ενός κρυπτογραφικού αλγορίθμου ή ενός κρυπτογραφικού συστήματος έχει σχέση με τα bits και το χρόνο που απαιτείται για την κρυπτανάλυση του αλγόριθμου και το σπάσιμό του.

Αν απαιτούνται 2^N εκτελέσεις του αλγόριθμου για να σπάσει και να αποκαλυφθεί το αρχικό κείμενο η ισχύς του αλγόριθμου είναι N bits. Το Εθνικό Ινστιτούτο Προτύπων και Τεχνολογίας έχει ορίσει τα bits ασφάλειας σε 80, 112, 128, 192, 256 αν και τα ογδόντα bits δεν θεωρούνται πλέον ασφαλή. [53]

Η τιμή τυπικά είναι ίση με το μέγεθος του κλειδιού του κρυπτογραφήματος το οποίο είναι ισοδύναμο με την πολυπλοκότητα μίας επίθεσης brute force.

Ένας κρυπτογραφικός αλγόριθμος θεωρείται ότι παραβιάστηκε, ή πιο απλά ότι έσπασε, όταν αποδεικνύεται ότι μία επίθεση έχει λιγότερη τιμή από τη γνωστή του επιπέδου ασφάλειάς του.

Όταν ο Triple DES χρησιμοποιείται με τρία ανεξάρτητα κλειδιά το μήκος του κλειδιού του είναι εκατό εξήντα οκτώ bits αφού κάθε κλειδί έχει πενήντα έξι bits.

Εδώ όμως θα πρέπει να πούμε ότι λόγω των επιθέσεων οι οποίες ονομάζονται επιθέσεις meet-in -the middle (meet-in-the middle attacks) το επίπεδο ασφάλειας του εν λόγω αλγορίθμου είναι εκατό δώδεκα bits.

Επιπλέον το μικρό μέγεθος τμήματος το οποίο είναι εξήντα τέσσερα bits συντελεί ώστε να είναι ευάλωτος σε επιθέσεις block collision όταν χρησιμοποιείται για την κρυπτογράφηση δεδομένων μεγάλου μεγέθους με το ίδιο κλειδί.

Το 2018 το Εθνικό Ινστιτούτο Προτύπων και Τεχνολογίας ανακοίνωσε ότι ο αλγόριθμος δεν μπορούσε να χρησιμοποιηθεί μετά το 2023.

3.10.3 Ο αλγόριθμος κρυπτογράφησης AES (Advanced Encryption Standard)

Το 1997 το Εθνικό Ινστιτούτο Προτύπων και Τεχνολογίας των Η.ΠΑ (National Institute of Standards and Technology, NIST) ξεκίνησε την προσπάθεια ανάπτυξης του κρυπτογραφικού προτύπου AES (Advanced Encryption Standard) με την προκήρυξη ενός διεθνούς διαγωνισμού. [54]

Η βασική μονάδα επεξεργασίας του κρυπτογραφικού αλγορίθμου AES είναι το ένα byte το οποίο, ως γνωστό, αποτελείται από μία ακολουθία οκτώ bits.

Η τιμή ενός byte δηλώνεται από την αλληλουχία bits ανάμεσα σε παρενθέσεις, για παράδειγμα { 10100011 }

Στην περίπτωση που τα bit δηλώνονται από μία μεταβλητή με δείκτη (indexed variable) η σύμβαση είναι να παρουσιάζονται σε φθίνουσα σειρά ως εξής: $\{b_7\, b_6\, b_5\, b_4\, b_3\, b_2\, b_1\, b_0\}$

Ο AES είναι ένας επαναληπτικός αλγόριθμος κρυπτογράφησης τμήματος. Το κείμενο προς κρυπτογράφηση χωρίζεται σε τμήματα των εκατό είκοσι οκτώ bits δηλαδή των δεκαέξι bytes. Το κλειδί, ανάλογα με το επιθυμητό επίπεδο ασφάλειας μπορεί να έχει μήκος εκατό είκοσι οκτώ ή εκατό ενενήντα δύο ή διακόσια πενήντα έξι bits δηλαδή δεκαέξι, είκοσι τέσσερα ή τριάντα δύο bytes.

Κάθε κύκλος του AES αποτελείται από τρεις ομοιόμορφους σχηματισμούς, τα επίπεδα (layers) :

- Το επίπεδο γραμμικής ανάμιξης (linear mixing layer) διασφαλίζει υψηλή διάχυση σε πολλαπλούς κύκλους
- Το μη γραμμικό επίπεδο (non linear layer) σχετίζεται με την παράλληλη εφαρμογή των κουτιών αντικατάστασης (S-Boxes) τα οποία εμφανίζουν άριστες, μη-γραμμικές ιδιότητες.
- Το επίπεδο της πρόσθεσης του κλειδιού (key addition layer) αναφέρεται στη συσχέτιση του ενδιάμεσου αποτελέσματος με το αντίστοιχο υπό κλειδί του κύκλου μέσω της πράξης της αποκλειστικής διάζευξης XOR. [32][54]

3.10.4 Ο αλγόριθμος κρυπτογράφησης IDEA (International Data Encryption Algorithm)

Ο αλγόριθμος International Data Encryption Algorithm (IDEA) είναι ένας συμμετρικός αλγόριθμος κρυπτογράφησης τμήματος ο οποίος δημιουργήθηκε το 1991 από τους X. Lai και J. Massey.

Σχεδιάστηκε με σκοπό την παροχή ασφαλούς κρυπτογράφησης για ψηφιακά δεδομένα και χρησιμοποιείται σε μία ποικιλία εφαρμογών όπως για παράδειγμα στις ασφαλείς επικοινωνίες, στις οικονομικές συναλλαγές και στα ηλεκτρονικά συστήματα ψηφοφορίας. Ο αλγόριθμος IDEA χρησιμοποιεί ένα τμήμα κρυπτογράφησης εξήντα τεσσάρων bit και ένα κλειδί μήκους εκατό είκοσι οκτώ bit.

Στα πλαίσια αυτού του αλγόριθμου χρησιμοποιείται ένας αριθμός μαθηματικών λειτουργιών για να μετασχηματιστεί ένα απλό κείμενο σε κρυπτογράφημα.

Έχει σχεδιαστεί να είναι ασφαλής σε μέγιστο βαθμό και ανθεκτικός σε επιθέσεις όλων των τύπων όπως για παράδειγμα αυτών που σχετίζονται με διαφορική και γραμμική κρυπτανάλυση.

Ένα από τα δυνατά σημεία του IDEA είναι η αποτελεσματική υλοποίησή του σε υλικό και λογισμικό.

Επίσης, είναι ένας αρκετά γρήγορος αλγόριθμος ο οποίος απαιτεί ελάχιστους πόρους μνήμης και μικρή επεξεργαστική ισχύ. Γι’ αυτούς τους λόγους αποτελεί ιδανική επιλογή για τα ενσωματωμένα συστήματα και για άλλες εφαρμογές οι οποίες έχουν περιορισμένους πόρους.

Ο αλγόριθμος IDEA χρησιμοποιείται ευρέως σε εφαρμογές κρυπτογράφησης αν και έχει αντικατασταθεί σε μεγάλο βαθμό από τον κρυπτογραφικό αλγόριθμο AES [55]

3.10.5 Ο αλγόριθμος κρυπτογράφησης RC2

Ο αλγόριθμος κρυπτογράφησης RC2 αναπτύχθηκε από τον Ron Rivest της εταιρίας RSA Security ο οποίος ήταν ο δημιουργός και άλλων αλγορίθμων και κρατήθηκε μυστικός μέχρι και το 1996 όταν στάλθηκε ανώνυμα στην ομάδα sci.crypt. Ο αλγόριθμος RC2 είναι επίσης γνωστός ως ARC2.

Το ακρωνύμιο RC2 ερμηνεύεται ως “Rivest Cipher” ή “Ron’s Code”.

Ο αλγόριθμος κρυπτογράφησης RC2 είναι ένας αλγόριθμος τμήματος του οποίου το τμήμα έχει μέγεθος οκτώ byte δηλαδή εξήντα τέσσερα bit. Αυτό σημαίνει ότι τα δεδομένα εισόδου χωρίζονται σε τμήματα των οκτώ byte και το καθένα από αυτά τυγχάνει ξεχωριστής επεξεργασίας.
Εκτός από τα δεδομένα προς κρυπτογράφηση και αποστολή ο αλγόριθμος RC2 δέχεται ως είσοδο το μυστικό κλειδί χρήστη το οποίο μπορεί να έχει μέγεθος από ένα byte έως και εκατό είκοσι οχτώ byte ή αλλιώς από οκτώ μέχρι 1024 bit. [56] [57]
Ο RC2 είναι επαναλαμβανόμενος αλγόριθμος πράγμα που σημαίνει ότι μετασχηματίζει τμήματα απλού κειμένου προκαθορισμένου μεγέθους σε πανομοιότυπα τμήματα κρυπτοκειμένου με επαναλαμβανόμενες εφαρμογές αναστρέψιμου μηχανισμού που είναι γνωστός ως κυκλική συνάρτηση (round function) αφού κάθε επανάληψη ονομάζεται γύρος.
Υπάρχουν δύο είδη γύρων οι γύροι mixing οι οποίοι είναι συνολικά δεκαέξι και οι γύροι mashing που είναι δύο.

Σε κάθε γύρο η κάθε μία από τις τέσσερις λέξεις που αποτελούνται από δεκαέξι bit οι οποίες βρίσκονται σε μία κατάσταση ενδιάμεσου κρυπτογραφήματος ενημερώνονται ως συνάρτηση των άλλων λέξεων.
Κάθε γύρος mixing δέχεται ένα υπό κλειδί των δεκαέξι bit. Τα εξήντα τέσσερα υποκλείδιά προέρχονται από το κλειδί που επιλέγει ο χρήστης. Τέλος θα πρέπει να σημειώσουμε ότι ο RC2 δεν είναι γρήγορος αλγόριθμος.

3.10.6 Ο αλγόριθμος κρυπτογράφησης RC4

Ο αλγόριθμος κρυπτογράφησης RC4 αναπτύχθηκε το 1987 από τον Ron Rivest για την εταιρία RSA Security και είναι ένας αλγόριθμος ροής ο οποίος χρησιμοποιείται ευρέως κυρίως λόγω της απλότητάς του και της ταχύτητάς του. Είναι ένας αλγόριθμος με μήκος κλειδιού που ποικίλει, συγκεκριμένα χρησιμοποιεί κλειδιά μήκους οκτώ bit ή δύο χιλιάδων σαράντα οχτώ bit. Γενικά χρησιμοποιείται σε εφαρμογές όπως η Secure Socket Layer (SSL) η Transport Layer Security (TLS) κλπ.[58]

3.10.7 Ο αλγόριθμος κρυπτογράφησης RC5

Ο αλγόριθμος κρυπτογράφησης RC5 είναι ένας συμμετρικός αλγόριθμος κρυπτογράφησης τμήματος ο οποίος αναπτύχθηκε το 1994 από τον Ron Rivest. Το κλειδί που χρησιμοποιείται σε αυτόν τον αλγόριθμο είναι ποικίλου μήκους το οποίο μπορεί να είναι από μηδέν έως δύο χιλιάδες σαράντα bit.
Ο αλγόριθμος RC5 λειτουργεί σε γύρους των οποίων ο αριθμός εξαρτάται από το μήκος του κλειδιού.
Τα πλεονεκτήματα του συγκεκριμένου αλγόριθμου είναι τα εξής:

- Είναι ασφαλής και παρέχει προστασία από επιθέσεις

- Είναι γρήγορος και αποτελεσματικός άρα ιδανικός για εφαρμογές μεγάλης κλίμακας
- Υποστηρίζει κλειδιά ποικίλου μήκους
- Είναι κοινόχρηστος αλγόριθμος (Public Domain Algorithm) πράγμα το οποίο σημαίνει ότι μπορεί να χρησιμοποιηθεί από τον οποιονδήποτε
- Η κρυπτογράφηση με την χρήση αυτού του αλγορίθμου είναι εύκολα υλοποιήσιμη πράγμα το οποίο τον κάνει ιδανική επιλογή για οργανισμούς και προγραμματιστές εφαρμογών.[59]

3.10.8 Ο αλγόριθμος κρυπτογράφησης RC6

Ο αλγόριθμος κρυπτογράφησης RC6 είναι ένας συμμετρικός αλγόριθμος κρυπτογράφησης τμήματος ο οποίος αναπτύχθηκε το 1998 από τους Ronald Rivest, Matt Robshaw, Ray Sidney, and Yiqun Lisa Yin.

Ο αλγόριθμος RC6 είναι βασισμένος στον RC5 και αποτελεί μία παραλλαγή του AES.

Ο RC6 χρησιμοποιεί κλειδί και τμήματα δεδομένων ποικίλου μεγέθους καθώς και ποικίλο αριθμό γύρων κάτι που τον καθιστά προσαρμόσιμο.

Θα πρέπει να αναφέρουμε ότι το κλειδί μπορεί να έχει μέγεθος από εκατό είκοσι οκτώ έως διακόσια πενήντα έξι bit.

Τα πλεονεκτήματα του συγκεκριμένου αλγόριθμου είναι τα εξής:

Είναι εξαιρετικά ασφαλής και παρέχει προστασία από επιθέσεις όπως επιθέσεις brute force διαφορικής κρυπτανάλυσης, γραμμικής κρυπτανάλυσης.

Είναι αρκετά γρήγορο και υλοποιείται με ικανοποιητικά αποτελέσματα και σε λογισμικό και σε υλικό [59]

3.10.9 Ο αλγόριθμος κρυπτογράφησης MARS

Ο αλγόριθμος κρυπτογράφησης MARS είναι ένας συμμετρικός αλγόριθμος τμήματος ο οποίος δουλεύει με τμήματα μεγέθους εκατό είκοσι οχτώ bit και κλειδί του οποίου το μέγεθος ποικίλει με εύρος από εκατό είκοσι οχτώ μέχρι και πάνω από τετρακόσια bit. Περιλαμβάνει τριάντα δύο κύκλους μετασχηματισμών από τους οποίους μόνο οι δεκαέξι βασίζονται στο μυστικό κλειδί. Οι πράξεις που πραγματοποιούνται είναι η πρόσθεση με κλειδιά των τριάντα δύο bit, ο πολλαπλασιασμός και η ολίσθηση των δεδομένων. [60][61]

3.10.10 Ο αλγόριθμος κρυπτογράφησης Serpent

Ο αλγόριθμος κρυπτογράφησης Serpent είναι ένας αλγόριθμος τμήματος ο οποίος αναπτύχθηκε στα τέλη της δεκαετίας του 1990 από τους Ross Anderson, Eli Biham, and Lars Knudsen

Έγινε διάσημος πρόσφατα λόγω των δυνατών χαρακτηριστικών του αναφορικά με τη ασφάλεια που προσφέρει και την ανθεκτικότητά του στις επιθέσεις. Τα χαρακτηρίστηκα του είναι τα ακόλουθα:

- Το μήκος του κλειδιού του είναι ποικίλο και μπορεί να είναι εκατό είκοσι οχτώ, εκατό ενενήντα δύο ή διακόσια πενήντα έξι bit και έτσι ο αλγόριθμος είναι ανθεκτικός σε επιθέσεις brute force
- Ο Serpent έχει ένα καθορισμένο μέγεθος τμήματος τα εκατό είκοσι οκτώ bit. Αυτό σημαίνει ότι ένα κείμενο προς κρυπτογράφηση χωρίζεται σε τμήματα τέτοιου μεγέθους πριν την διαδικασία.
- Το επίπεδο ασφάλειας του αλγόριθμου είναι υψηλό.

- Ο Serpent είναι πιο αργός σε σύγκριση με κάποιους άλλους αλγορίθμους όπως ο AES αλλά όχι σε απαγορευτικό βαθμό και μπορεί να υλοποιηθεί σε μία ποικιλία συσκευών. [62]

3.10.11 Ο αλγόριθμος κρυπτογράφησης Twofish

Ο Twofish είναι ένας συμμετρικός αλγόριθμος με ένα κλειδί για κρυπτογράφηση και αποκρυπτογράφηση.
Το μέγεθος τμήματος του αλγορίθμου είναι εκατό είκοσι οκτώ bit και δέχεται ως είσοδο ένα μυστικό κλειδί μήκους διακοσίων πενήντα έξι bit.
Τα δυνατά σημεία του Twofish είναι τα εξής :

- Είναι ασφαλής και ανθεκτικός σε όλες τις επιθέσεις των οποίων ο τύπος είναι γνωστός και έτσι είναι κατάλληλος για χρήση σε εφαρμογές που έχουν ως ζητούμενο την υψηλή ασφάλεια
- Έχει μεγάλο χώρο (bits) για το μυστικό κλειδί κάτι το οποίο σημαίνει ότι ο αριθμός των πιθανών κλειδιών είναι αρκετός έτσι ώστε να αποκρούει αποτελεσματικά τις επιθέσεις brute force.
- Ο Twofish είναι γρήγορος και αποτελεσματικός κάτι που τον καθιστά κατάλληλο για εφαρμογές που απαιτούν υψηλές ταχύτητες για κρυπτογράφηση και αποκρυπτογράφηση δεδομένων.

Τα αδύναμα σημεία του αλγόριθμου είναι τα ακόλουθα:

- Είναι ευάλωτος σε επιθέσεις side-channel όπως για παράδειγμα επιθέσεις συγχρονισμού (timing attacks και επιθέσεις ισχύος (power attacks)
- Η σωστή υλοποίηση του αλγορίθμου Twofish μπορεί να είναι μία πρόκληση αφού τυχόν λάθη στη διαδικασία μπορεί να αποβούν μοιραία διότι κάποιοι μπορεί να τα εκμεταλλευτούν.
- Μπορεί να μην είναι κατάλληλος για συσκευές με χαμηλή ισχύ ή για εφαρμογές με περιορισμένους υπολογιστικούς πόρους λόγω της υπολογιστικής του περιπλοκότητας. [63]

3.10.12 Ο αλγόριθμος κρυπτογράφησης Blowfish

Ο αλγόριθμος Blowfish είναι δημιούργημα του Bruce Schneier ο οποίος τον ανέπτυξε το 1993 ως εναλλακτική πρόταση απέναντι στην τεχνική κρυπτογράφησης DES.
Είναι κατά πολύ γρηγορότερος από τον DES και παρέχει έναν ικανοποιητικό κρυπτογραφικό ρυθμό.
Το μέγεθος τμήματος του αλγόριθμου είναι εξήντα τέσσερα bit, το μέγεθος του μυστικού κλειδιού ποικίλει και μπορεί να είναι από τριάντα δύο bit μέχρι τετρακόσια σαράντα οκτώ bit.
Οι γύροι του αλγορίθμου είναι δεκαέξι και κάθε γύρος επεξεργάζεται ένα τμήμα εξήντα τεσσάρων bit το οποίο χωρίζεται σε δύο κομμάτια των τεσσάρων byte (32-bit words).
Ο αλγόριθμος Blowfish αποτελείται από δύο μεγάλα τμήματα:

- Κρυπτογράφηση δεδομένων η οποία λαμβάνει χώρα μέσω ενός δικτύου Feistel δεκαέξι γύρων και ο κάθε γύρος αποτελείται από μετάθεση εξαρτώμενη από το μυστικό κλειδί και αντικατάσταση εξαρτώμενη και από το κλειδί και από τα δεδομένα. Μεγάλα S-Boxes τα οποία εξαρτώνται από το κλειδί και είναι

δυναμικής φύσεως λειτουργούν με την μέθοδο αντικατάστασης και σχηματίζουν ένα αναπόσπαστο τμήμα του συστήματος κρυπτογράφησης δεδομένων. Όλες οι λειτουργίες κρυπτογράφησης είναι XOR και προσθέσεις σε τέσσερα byte.

- Επέκταση κλειδιού η οποία είναι μία διαδικασία κατά τη διάρκεια της οποίας το κλειδί που έχει μέγιστο μέγεθος τετρακόσια σαράντα οκτώ bit μετατρέπεται σε πίνακες υποκλείδιών το μέγεθος των οποίων είναι τέσσερα χιλιάδες εκατό εξήντα οκτώ byte. Τα υποκλείδιά σχηματίζουν ένα αναπόσπαστο τμήμα του αλγορίθμου Blowfish ο οποίος χρησιμοποιεί έναν μεγάλο αριθμό από αυτά. Αυτά τα κλειδιά υπολογίζονται πριν τις διαδικασίες της κρυπτογράφησης και αποκρυπτογράφησης.

Τα πλεονεκτήματα του Blowfish είναι τα εξής :

- Είναι πιο γρήγορος και πιο αποτελεσματικός από τους DES και IDEA.
- Μπορεί να χρησιμοποιηθεί από όποιον το επιθυμεί χωρίς άδεια.
- Η κρυπτογράφηση είναι ικανοποιητική σε μεγάλους μικροεπεξεργαστές.
- Προσφέρει μεγάλη ασφάλεια σε εφαρμογές οι οποίες αναπτύχθηκαν με γλώσσα προγραμματισμού την Java.
- Υποστηρίζει την ασφαλή αυθεντικοποίηση χρήστη σε περιπτώσεις απομακρυσμένης πρόσβασης.

Τα μειονεκτήματα του Blowfish είναι τα ακόλουθα :

- Επηρεάζεται η ταχύτητα με την αλλαγή των κλειδιών
- Ο προγραμματισμός των κλειδιών διαρκεί πολύ
- Το μικρό μέγεθος των τμημάτων τον κάνει ευάλωτο σε επιθέσεις birthday οι οποίες ανήκουν στην ομάδα των επιθέσεων brute force.
- Κάθε καινούργιο κλειδί απαιτεί προ εξεργασία η οποία ισοδυναμεί με τέσσερα KB κειμένου γεγονός που επηρεάζει την ταχύτητά του. [63][64][65][66]

3.10.13 Ο αλγόριθμος κρυπτογράφησης CAST-128

Ο αλγόριθμος CAST-128 είναι ένας συμμετρικός αλγόριθμος ο οποίος χρησιμοποιείται σε ποικίλες εφαρμογές όπως για παράδειγμα στα εικονικά ιδιωτικά δίκτυα ( ) και την ασφαλή ηλεκτρονική αλληλογραφία.

Τα χαρακτηριστικά του είναι τα εξής :

- Λειτουργεί με τμήματα των εξήντα τεσσάρων bit απλού κειμένου και παράγει τμήματα κρυποκειμένου των εξήντα τεσσάρων bit
- Το κλειδί που χρησιμοποιείται έχει μήκος μέχρι και εκατό είκοσι οχτώ bit πράγμα το οποίο παρέχει ασφάλεια υψηλού επιπέδου

- Χρησιμοποιεί μία δομή δικτύου Feistel επομένως λαμβάνουν χώρα πολλοί γύροι κρυπτογράφησης και αποκρυπτογράφησης

Τα πλεονεκτήματα του CAST-128 είναι τα εξής :

- Παρέχει ικανοποιητικό βαθμό ασφάλειας.
- Είναι αποδοτικός και υλοποιείται και στο υλικό και στο λογισμικό

Τα μειονεκτήματα του CAST-128 είναι τα ακόλουθα :

- Είναι ευάλωτος σε επιθέσεις συγκεκριμένου είδους όπως για παράδειγμα σε επιθέσεις διαφορικής κρυπτανάλυσης
- Δεν είναι κατάλληλος για χρήση σε εφαρμογές που σχετίζονται με την Άμυνα ή με κυβερνητικές υπηρεσίες
- Δεν παρέχει αυθεντικοποιημένη κρυπτογράφηση πράγμα που σημαίνει ότι το άτομο που επιτίθεται μπορεί να τροποποιήσει το κρυπτοκείμενο χωρίς να ανιχνευτεί [67]

Επίλογος

Τα κυριότερα σημεία του κεφαλαίου στα οποία πρέπει να κάνουμε αναφορά είναι η ανάλυση του όρου συμμετρική κρυπτογραφία την παράθεση των χρήσεων της και την περιγραφή των κάποιων από τους αλγόριθμους συμμετρικής κρυπτογράφησης.

Κεφάλαιο 4
4. Ασύμμετρη κρυπτογραφία
Εισαγωγή

Σε αυτό το κεφάλαιο εξηγούμε αρχικά τι σημαίνει ο όρος ασύμμετρη κρυπτογραφία ή κρυπτογραφία δημόσιου κλειδιού. Αναφέρουμε τα χαρακτηριστικά της στα οποία διασαφηνίζεται ότι σχεδιάστηκε για ανταλλαγή κλειδιών σε δημόσια κανάλια. Στην συνέχεια αναφερόμαστε στις χρήσεις των μεθόδων της ασύμμετρης κρυπτογραφίας και τονίζουμε τα πλεονεκτήματά και τα μειονεκτήματά της. Τέλος αναλύουμε τους κρυπτογραφικούς αλγόριθμους RSA, DSA και ECDSA.

4.1 Ορισμός

Ασύμμετρη κρυπτογραφία ή αλλιώς κρυπτογραφία δημόσιου κλειδιού ονομάζουμε την κρυπτογραφία για την οποία χρησιμοποιούμε ένα ζευγάρι κλειδιών για κρυπτογράφηση και αποκρυπτογράφηση δεδομένων. Αναλυτικότερα, το ζευγάρι των κλειδιών αποτελείται από ένα δημόσιο κλειδί το οποίο διανέμεται ελεύθερα σε όλους και ένα ιδιωτικό. Ο ιδιοκτήτης του ιδιωτικού κλειδιού δεν το αποκαλύπτει σε κανέναν.

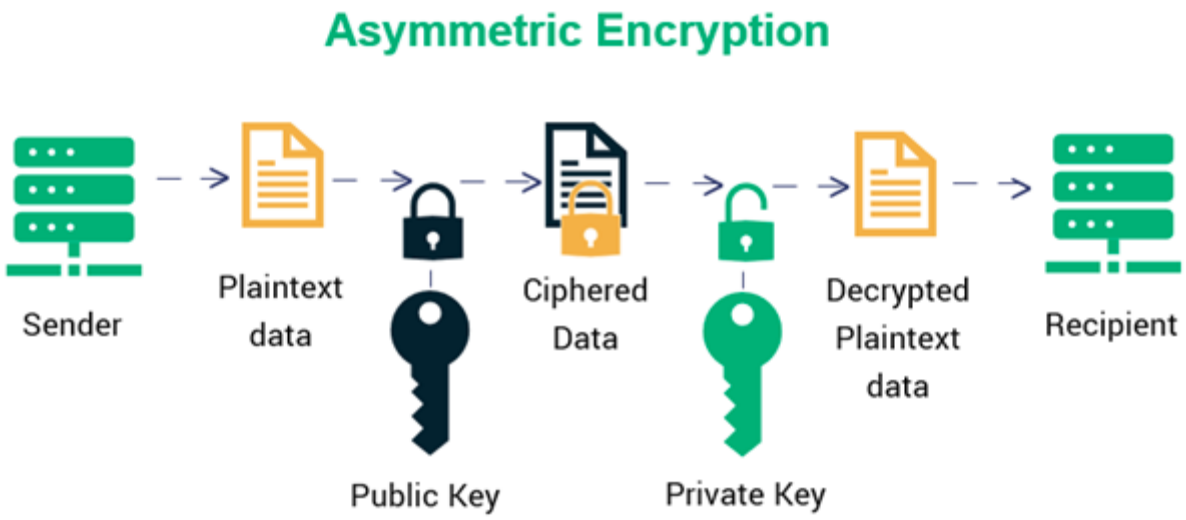
Σε περίπτωση που κάποιος αποφασίσει να στείλει ένα κρυπτογραφημένο μήνυμα μπορεί να βρει το δημόσιο κλειδί του παραλήπτη σε ένα δημόσιο κατάλογο.

Το κάθε μέρος που συμμετέχει σε μία επικοινωνία έχει ένα δημόσιο και ένα ιδιωτικό κλειδί που συνδέονται με κάποια μαθηματική σχέση. Ας δούμε αυτή την διαδικασία αναλυτικότερα:

Αρχικά ο αποστολέας λαμβάνει το δημόσιο κλειδί του παραλήπτη

Ο αποστολέας χρησιμοποιεί το δημόσιο κλειδί του παραλήπτη για να κρυπτογραφήσει τα δεδομένα που πρόκειται να στείλει και έτσι δημιουργείται το κρυπτογραφημένο κείμενο

Ο παραλήπτης χρησιμοποιεί το ιδιωτικό του κλειδί για να τα αποκρυπτογραφήσει τα ληφθέντα δεδομένα και έτσι τα ανακτά.[68]



Εικόνα 4.1 Απεικόνιση της ασύμμετρης κρυπτογραφικής διαδικασίας [69]

Λόγω της μονόδρομης φύσης της κρυπτογραφικής συνάρτησης ένας αποστολέας δεν μπορεί να διαβάσει τα μηνύματα ενός άλλου αποστολέα ακόμα και αν και οι δύο έχουν το δημόσιο κλειδί του παραλήπτη.

Αυτή η επικοινωνιακή προσέγγιση καθιστά δυνατή την ασφαλή επικοινωνία δύο οποιωνδήποτε μερών χωρίς να υπάρχει η ανάγκη να έχουν και τα δύο το ίδιο μυστικό κλειδί.

Ένα από τα πλεονεκτήματα της ασύμμετρης κρυπτογραφίας είναι ότι πρακτικά εκμηδενίζει, όπως είναι ολοφάνερο της ανάγκη ανταλλαγής μυστικών κλειδιών μία διαδικασία αρκετά δύσκολη ιδιαίτερα όταν πρέπει να επικοινωνήσουν πολλά μέρη. Επιπροσθέτως η ασύμμετρη κρυπτογραφία επιτρέπει την δημιουργία ψηφιακών υπογραφών οι οποίες μπορούν να χρησιμοποιηθούν για να πιστοποιήσουν την αυθεντικότητα των δεδομένων.

4.2 Χαρακτηριστικά της ασύμμετρης κρυπτογραφίας

Τα χαρακτηριστικά της ασύμμετρης κρυπτογραφίας είναι τα ακόλουθα:

- Σχεδιάστηκε για ασφάλεια δεδομένων και ανταλλαγή κλειδιών σε δημόσια κανάλια
- Τα κλειδιά της ασύμμετρης κρυπτογραφίας είναι μεγάλα σε μέγεθος

- Οι ασύμμετροι αλγόριθμοι κρυπτογραφίας είναι ισχυροί.

4.3 Ασύμμετρη κρυπτογραφία και Διαδίκτυο

Ουσιαστικά η ασύμμετρη κρυπτογραφία είναι ένας τρόπος πιστοποίησης της ταυτότητάς μας σε μέλη με τα οποία επιθυμούμε να επικοινωνήσουμε μέσω καναλιών τα οποία είναι μη ασφαλή. Αυτός είναι ο λόγος που η κρυπτογράφηση δημόσιου κλειδιού θεωρείται θεμέλιος λίθος στην ασφάλεια του Διαδικτύου.

Η δομή του δημοσίου κλειδιού (Public key infrastructure, PKI) η οποία είναι ένα πλαίσιο κανόνων, λειτουργιών και τεχνολογιών κάνει την ασφαλή διαδικτυακή επικοινωνία εφικτή.

Η ασφάλεια της διαδικτυακής επικοινωνίας βασίζεται και στην συμμετρική αλλά και στην ασύμμετρη κρυπτογραφία.

Οι μέθοδοι της ασύμμετρης κρυπτογραφίας χρησιμοποιούνται για τα εξής:

- Αυθεντικοποίηση μερών που πρόκειται να επικοινωνήσουν.
- Αυθεντικοποίηση ακεραιότητας δεδομένων
- Ανταλλαγή κλειδιών συμμετρικής κρυπτογράφησης

4.4 Χρήσεις της ασύμμετρης κρυπτογραφίας

Η ασύμμετρη κρυπτογραφία χρησιμοποιείται σε πολλές εφαρμογές όπως για παράδειγμα στην ασφαλή επικοινωνία ενώ είμαστε συνδεδεμένοι στο Διαδίκτυο, στις ασφαλείς μεταδόσεις δεδομένων και στις ψηφιακές υπογραφές.

Πολλά πρωτόκολλα είναι βασισμένα στην ασύμμετρη κρυπτογραφία όπως το TLS (Transport Layer Security) και το SSL (Secure Sockets Layer) τα οποία αποτελούν απαραίτητα συστατικά στοιχεία για τη λειτουργία του πρωτοκόλλου https (Hypertext Transfer Protocol Secure)

Μία ψηφιακή υπογραφή είναι μία μαθηματική τεχνική η οποία χρησιμοποιείται για επικύρωση της αυθεντικότητας και της ακεραιότητας ενός μηνύματος, λογισμικού ή ψηφιακού εγγράφου. Μπορούμε να πούμε ότι είναι το ψηφιακό ισοδύναμο μίας χειρόγραφης υπογραφής ή μίας σφραγίδας.

4.5 Πλεονεκτήματα της ασύμμετρης κρυπτογραφίας

Τα πλεονεκτήματα της ασύμμετρης κρυπτογραφίας είναι τα εξής :

- Είναι ασφαλέστερη από την συμμετρική κρυπτογραφία.
- Είναι χρήσιμη όταν υπάρχουν πολλοί παραλήπτες.
- Διευκολύνει των διαμοιρασμό κλειδιών
- Είναι ιδανική για ψηφιακές υπογραφές.

4.6 Μειονεκτήματα της ασύμμετρης κρυπτογραφίας

Τα μειονεκτήματα της ασύμμετρης κρυπτογραφίας είναι τα εξής :

- Είναι μία αργή διαδικασία συγκριτικά με την συμμετρική κρυπτογραφία
- Είναι δύσκολο να υλοποιηθεί σε μεγάλες εφαρμογές λόγω όγκου

Στη συνέχεια αυτού του κεφαλαίου θα προσπαθήσουμε να αναλύσουμε τους εξής αλγόριθμους :

- Αλγόριθμος Rivest–Shamir–Adleman, (Rivest–Shamir–Adleman cryptosystem, RSA)
- Αλγόριθμος Ψηφιακής Υπογραφής, (Digital Signature Algorithm, DSA)
- Αλγόριθμος Ψηφιακής Υπογραφής Ελλειπτικής Καμπύλης, (Elliptic Curve Digital Signature Algorithm, ECDSA) [69]

4.7 Ο αλγόριθμος δημόσιου κλειδιού RSA

Το 1978 οι Ron Rivest, Adi Shamir και Leonard Adleman εισήγαγαν έναν κρυπτογραφικό αλγόριθμο, τον RSA (Rivest-Shamir-Adleman) ο οποίος έπρεπε να αντικαταστήσει τον αλγόριθμο National Bureau of Standards (NBS) αφού ήταν λιγότερο ασφαλής. Ο αλγόριθμος RSA εφαρμόζει ένα σύστημα κρυπτογράφησης (κρυπτοσύστημα) δημόσιου κλειδιού (public key cryptosystem) όπως επίσης και ψηφιακές υπογραφές.

Αναλυτικότερα, ο αλγόριθμος RSA εφαρμόζει δύο σημαντικές ιδέες :

- Κρυπτογράφηση δημόσιου κλειδιού (public key encryption) κάτι που σημαίνει ότι δεν υπάρχει πλέον η ιδέα ενός "ταχυδρόμου" ο οποίος θα πρέπει να παραδώσει κλειδιά σε κάποιον παραλήπτη μέσω ενός ασφαλούς καναλιού πριν τη μετάδοση του μηνύματος προς σ' αυτόν. Στον αλγόριθμο RSA τα κλειδιά κρυπτογράφησης είναι δημόσια ενώ τα κλειδιά αποκρυπτογράφησης είναι ιδιωτικά όπως φυσικά ισχύει για κάθε αλγόριθμο ασύμμετρης κρυπτογραφίας συνεπώς μόνο το πρόσωπο που έχει το σωστό κλειδί αποκρυπτογράφησης μπορεί να αποκρυπτογραφήσει το αποσταλέν μήνυμα. Η κάθε οντότητα η οποία είναι πιθανόν να επικοινωνήσει έχει τα δικά της κλειδιά κρυπτογράφησης και αποκρυπτογράφησης. Θα πρέπει το δημόσιο κλειδί κρυπτογράφησης να έχει δημιουργηθεί με τέτοιον τρόπο ώστε να είναι αδύνατο να βγει μέσω αυτού κάποιο συμπέρασμα για την μορφή του κλειδιού αποκρυπτογράφησης.
- Οι ψηφιακές υπογραφές είναι πολύτιμα εργαλεία όταν αυτός που λαμβάνει το αποσταλέν μήνυμα επιθυμήσει να πιστοποιήσει την ταυτότητα του αποστολέα του. Αυτό γίνεται με την χρήση του κλειδιού της αποκρυπτογράφησης του αποστολέα και η υπογραφή μπορεί αργότερα να επαληθευτεί

από οποιονδήποτε με την χρήση του δημόσιου κλειδιού κρυπτογράφησης και, όπως καταλαβαίνουμε, οι υπογραφές δεν μπορούν να πλαστογραφηθούν κατά κανόνα. Επίσης καμία οντότητα δεν μπορεί να ισχυριστεί ότι δεν υπέγραψε το μήνυμα.

Τα παραπάνω χαρακτηριστικά δεν είναι μόνο χρήσιμα στα πλαίσια του ηλεκτρονικού ταχυδρομείου αλλά και σε άλλες ηλεκτρονικές συνδιαλλαγές όπως για παράδειγμα σε μεταφορές χρημάτων. Η ασφάλεια του αλγορίθμου RSA έχει αξιολογηθεί πολλές φορές.

Οι παραπάνω διαδικασίες σχετίζονται με τα κλειδιά τα οποία στον αλγόριθμο RSA είναι ένα ζευγάρι δύο ειδικών αριθμών. Το ένα κλειδί είναι δημόσιο και υπάρχει σε δημόσιο φάκελο ενώ το ιδιωτικό κλειδί κρατιέται μυστικό.

Στα πλαίσια της κρυπτογραφίας έχει επικρατήσει τα δύο μέρη που επικοινωνούν να αποκαλούνται με τα ονόματα Αλίκη και Μπομπ ενώ ο υποκλοπέας ονομάζεται Εύα [2] [70]

4.7.1 Κρυπτογράφηση/αποκρυπτογράφηση

Η Αλίκη θέλει να στείλει ένα απλό κείμενο M στον Μπομπ δίχως να το καταλάβει η Εύα ακόμα και αν καταφέρει να το υποκλέψει. Για να γίνει κάτι τέτοιο το μήνυμα κρυπτογραφείται και μετατρέπεται σε κρυπτογραφημένο κείμενο (ciphertext) έστω C το οποίο δεν μπορεί να κατανοηθεί από την Εύα. Όταν ο Μπομπ λάβει το C θα πρέπει να το αποκρυπτογραφήσει δηλαδή να το μετατρέψει στην αρχική του μορφή.

4.7.2 Ψηφιακές υπογραφές

Η Αλίκη επιθυμεί να υπογράφει τα κείμενά της. Όταν δηλαδή στέλνει κάτι κρυπτογραφημένο ο Μπομπ πρέπει να είναι σε θέση και να το αποκρυπτογραφήσει αλλά και να είναι βέβαιος ότι το μήνυμα προέρχεται από την Αλίκη και όχι από την Εύα.

Ας πούμε λοιπόν ότι η Αλίκη επιθυμεί να στείλει ένα μήνυμα στον Μπομπ. Αυτό έχει ως προϋπόθεση να διαλέξουν και οι δύο ένα ζευγάρι κλειδιών δηλαδή ένα δημόσιο έστω P (public) και ένα ιδιωτικό έστω S(secret). Έχουμε λοιπόν :

- S_B είναι το μυστικό κλειδί του Μπομπ
- P_B είναι το δημόσιο κλειδί του Μπομπ
- S_A είναι το μυστικό κλειδί της Αλίκης
- P_A είναι το δημόσιο κλειδί της Αλίκης

Η Αλίκη χρησιμοποιεί το δημόσιο κλειδί του Μπομπ για να μετατρέψει το κείμενο Μ σε κρυπτογράφημα C. Δηλαδή έχουμε :

$$C = P_B(M) \quad (4.1)$$

Όταν ο Μπομπ λάβει το κρυπτογραφημένο μήνυμα C θα το αποκρυπτογραφήσει με το μυστικό του κλειδί και θα ανακτήσει το μήνυμα δηλαδή θα έχουμε :

$$M = S_B(C) \quad (4.2)$$

Τα κλειδιά πρέπει να είναι τέτοια ώστε :

- S_B (P_B (M))= P_B (S_B (M))
- Ο υπολογισμός τους να είναι εύκολος
- Η εξαγωγή του S_B από το P_B να είναι πρακτικά αδύνατη.

Η πρώτη ιδιότητα κάνει άμεσο τον σχηματισμό ψηφιακής υπογραφής : αν η Αλίκη υπογράψει ψηφιακά το μήνυμα Μ με το μυστικό της κλειδί S_A λογικά θα στείλει μαζί με το μήνυμα και το μυστικό της κλειδί δηλαδή θα ισχύει S_A (M).

Στη συνέχεια ο Μπομπ χρησιμοποιώντας το δημόσιο κλειδί της Αλίκης, δηλαδή το P_A θα καταλάβαινε ότι το έστειλε η Αλίκη αφού P_A (S_A (M)) =M.

Συμπερασματικά μία πλήρης επικοινωνία μεταξύ της Αλίκης και του Μπομπ προκύπτει αν η Αλίκη στείλει στον Μπομπ το μήνυμα P_B(M,S_A (M)) το οποίο θα είναι κρυπτογραφημένο με το δημόσιο κλειδί του Μπομπ και με ψηφιακή υπογραφή με την βοήθεια του μυστικού κλειδιού της Αλίκης.

Ο Μπομπ θα λάβει το αποσταλέν μήνυμα και θα το αποκρυπτογραφήσει χρησιμοποιώντας το μυστικό του κλειδί δηλαδή S_B (P_B(M,S_A (M))) και μετά θα επιβεβαιώσει ότι μόνο αυτός που έχει το συγκεκριμένο ιδιωτικό κλειδί θα μπορούσε να δημιουργήσει την συγκεκριμένη ψηφιακή υπογραφή αφού P_A(S_A (M))=M

4.7.3 Περιγραφή του αλγόριθμου RSA

4.7.3.1 Επιλογή δημόσιου και ιδιωτικού κλειδιού

Το δημόσιο και το ιδιωτικό κλειδί επιλέγεται ως εξής :

- Επιλέγουμε δύο μεγάλους πρώτους αριθμούς p και q τουλάχιστον εκατό δεκαδικών ψηφίων.
- Υπολογίζουμε το γινόμενο n=p*q
- Επιλέγουμε ένα μικρό περιττό ακέραιο e ο οποίος είναι σχετικά πρώτος με το φ(n)=(p-1)(q-1)

Για να ισχύει αυτό για τον αριθμό e θα πρέπει να έχουμε:

$$gcd(e(p-1)(q-1)) \tag{4.3}$$

Το ακρωνύμιο gcd σημαίνει μέγιστος κοινός διαιρέτης (gcd, Greatest Common Divisor). Θέλουμε να υπολογίσουμε τον αριθμό e σε συνάρτηση με τους αριθμούς d, p και q. Ο αριθμός e είναι πολλαπλασιαστικός αντίστροφος του d. Αυτό σημαίνει ότι πρέπει να ικανοποιηθεί η παρακάτω εξίσωση:

$$e*d=1*(mod\,\varphi(n)) \tag{4.4}$$

Η συνάρτηση φ(n) είναι η συνάρτηση Όιλερ(Euler totient function φ(n)) της οποίας η έξοδος είναι το πλήθος των φυσικών αριθμών οι οποίοι είναι σχετικά πρώτοι με τον n δηλαδή πολύ απλά έχουν με τον n μέγιστο κοινό διαιρέτη τη μονάδα. Ας δούμε ένα πρακτικό παράδειγμα πριν αναλύσουμε διεξοδικά αυτό που μόλις αναφέραμε :για n=9 θα ισχύει φ(9)=6 αφού από το ένα έως το εννέα έξι
αριθμοί οι 1, 2, 4, 5, 7 και 8 είναι πρώτοι προς το εννέα. Τώρα, αναφορικά για τους πρώτους αριθμούς p το πλήθος του είναι το εξής:

$$\varphi(p)=p-1 \tag{4.5}$$

Για n, από τις στοιχειώδεις ιδιότητες της συνάρτησης φ του Όιλερ (Euler totient function φ(n)) έχουμε :

$$\varphi(n)=\varphi(p)*\varphi(q) \tag{4.6}$$

Προχωράμε σε αντικατάσταση και έχουμε :

$$\varphi(n)=(p-1)*(q-1) \tag{4.7}$$

Κάνουμε τις πράξεις και έχουμε :

$$\varphi(n)=p*q-p-q+1 \tag{4.8 }$$

Ξέρουμε ότι n=p*q και έτσι έχουνε

$$\varphi(n)=(p+q)+1 \tag{4.9}$$

Ας σταθούμε σε αυτό το σημείο για να εξηγήσουμε κάποια πράγματα: η σημειολογία a≡b (mod n) έχει την έννοια ότι ο αριθμός n είναι διαιρέτης της διαφοράς των a και. Δηλαδή για ένας αριθμό n για τον οποίο ισχύει $n > 1$ λέμε ότι δύο ακέραιοι αριθμοί, έστω a και b χαρακτηρίζονται **congruent** modulo n αν για κάποιον ακέραιο αριθμό k ισχύει :

$$a-b=k*n \tag{4.10}$$

Συνεπώς η εξίσωση e·d≡1(mod φ(n)) δηλώνει ότι η συνάρτηση φ(n) διαιρεί τη διαφορά e·d-1.Αυτό σημαίνει ότι η διαφορά e·d-1 η οποία είναι πολλαπλάσια του φ(n) έχει την μορφή k · φ(n) για κάποιον ακέραιο k. Σύμφωνα λοιπόν με αυτά που είπαμε έχουμε:

$$e*d\equiv 1*(mod\,\varphi(n)) \tag{4.11}$$

Μεταφέρουμε την μονάδα στο πρώτο μέλος της εξίσωσης και εισάγουμε στο δεύτερο μέλος της τον ακέραιο k και έχουμε :

$$e*d-1\equiv k*\varphi(n) \tag{4.12}$$

Μεταφέρουμε την μονάδα στο δεύτερο μέλος της εξίσωσης και τελικά έχουμε :

$$e*d\equiv k*\varphi(n)+1 \tag{4.13}$$

Γενικά λοιπόν, σύμφωνα με τους νόμους της αρθρωτής αριθμητικής το πολλαπλασιαστικό αντίστροφο (multiplicative inverse) a modulo m υπάρχει μόνο εάν οι a και m είναι σχετικά πρώτοι.
Για παράδειγμα για m=10 και a=3, ο a έχει αντίστροφο $x^{-1}=7$ αφού 3*7=21 ≡1 (mod 10) δηλαδή αν διαιρέσουμε το 21 με το δέκα το υπόλοιπό μας είναι 1.

- Υπολογίζουμε το $d=(e^{-1} \bmod \varphi(n))$
- Ανακηρύσσουμε το ζεύγος P=(e,n) ως δημόσιο κλειδί

➢Ανακηρύσσουμε το ζεύγος S=(d,n) ως μυστικό κλειδί

Συνεπώς, το κρυπτογράφημα C=P(M) το οποίο αντιστοιχεί σε ένα μήνυμα M<n με δημόσιο κλειδί P=(e,n) ορίζεται ως εξής :

$$C=P(M)=M^{e}(mod\,n) \tag{4.14}$$

Το αποκρυπτογράφημα M=S (C) το οποίο αντιστοιχεί σε ένα κρυπτογράφημα C όταν το μυστικό κλειδί είναι S=(d,n) ορίζεται ως εξής :

$$M=S(C)=C^{d}(mod\,n) \tag{4.15}$$

4.7.3.2 Απόδειξη της ορθότητας της διαδικασίας επιλογής δημόσιου και ιδιωτικού κλειδιού

Η ορθότητα της παραπάνω διαδικασίας αποδεικνύεται με το εξής θεώρημα :
Οι συναρτήσεις C και S του RSA είναι αντίστροφες η μία της άλλης, δηλαδή θα πρέπει να ισχύει :

$$P(S(M))=S(P(M))=M \tag{4.16}$$

Για κάθε αριθμό M ∈ $Z_{n,}$ όπου M το μήνυμά μας το n το πλήθος των bit που το απαρτίζουν, το Z το σύνολο των ακεραίων αριθμών ισχύει :

$$P(S(M))=S(P(M))=M^{ed}(mod\,n) \tag{4.17}$$

Μας δόθηκε ότι ο αριθμός d υπολογίζεται από τον εξής τύπο :

$$d=(e^{-1}\,mod\,\varphi(n)) \tag{4.18}$$

Με αυτό ως δεδομένο και τον τύπο 4.13 (e*d=k*φ(n) +1) ο εκθέτης του M στην εξίσωση 4.17 θα είναι :

$$e*d=1+k*\varphi(n)=1+k(p-1)*(q-1) \quad (4.19)$$

Σε αυτό το σημείο θα αναφέρουμε το Μικρό θεώρημα του Fermat σύμφωνα με το οποίο αν p πρώτος και a φυσικός αριθμός ισχύει :

- $a^p \equiv a \pmod p$
- $a^{p-1} \equiv 1 \pmod p$

Βάσει αυτού και της 4.19 έχουμε την εξής μεταβολή στον εκθέτη του M της 4.17 αφού αντικαταστήσουμε το e*d :

$$M^{ed}=M(M^{p-1})^{k(q-1)} mod\ p \quad (4.20)$$

$$M^{ed}=M*1^{k(q-1)} mod\ p \quad (4.21)$$

$$M^{ed}=M(mod\ p) \quad (4.22)$$

Έχοντας πάντα ως βάση το Μικρό Θεώρημα του Fermat έχουμε :

$$M^{ed}=M*(M^{q-1})^{k(p-1)} mod\ q \quad (4.23)$$

$$M^{ed}=M*1^{k(p-1)} mod\ q \quad (4.24)$$

$$M^{ed}=M(mod\ q) \quad (4.25)$$

Προκύπτει $M^{ed} = M \; \forall$ (για κάθε) M. [2]

4.7.4 Παράδειγμα κρυπτογράφησης με τον αλγόριθμο RSA

4.7.4.1 Παραγωγή κλειδιών

Για την παραγωγή κλειδιών ακολουθούμε τα παρακάτω βήματα:

- Διαλέγουμε δύο πρώτους αριθμούς p και q. Αν και καθένας από αυτούς τους αριθμούς θα πρέπει να είναι αρκετά μεγάλος όπως ήδη αναφέραμε για χάρη του παραδείγματος θα διαλέξουμε μικρές τιμές δηλαδή p=7 και q=19.
- Υπολογίζουμε το γινόμενο n ως εξής :

$$n = p * q = 7 * 19 = 133 \qquad (4.26)$$

- Υπολογίζουμε την συνάρτηση του Όιλερ ως εξής :

$$\varphi(n) = (p-1) * (q-1) = (7-1) * (19-1) = 108 \qquad (4.27)$$

- Τώρα θα επιλέξουμε ένα δημόσιο κλειδί το οποίο πρέπει να είναι πρώτος αριθμός, πρέπει να είναι μικρότερος από το αποτέλεσμα της συνάρτησης φ(n) και δεν θα είναι παράγοντες του αποτελέσματός της. Για παράδειγμα δεν μπορούμε να επιλέξουμε το τρία διότι αν και είναι πρώτος αριθμός και μικρότερος από εκατό οκτώ είναι παράγοντάς του αφού αν πολλαπλασιάζουμε το τρία με το τριάντα έξι το γινόμενο είναι ακριβώς αυτός ο αριθμός. Διαλέγουμε λοιπόν το 29 το οποίο θα το συμβολίζουμε από εδώ και πέρα με το γράμμα e.

Είναι λοιπόν η στιγμή να υπολογίσουμε το ιδιωτικό μας κλειδί ως εξής :

$$e \cdot d \equiv 1 (mod\, \varphi(n)) = (29 * d) mod\, 108 = 1 = 41 \quad (4.28)$$

Άρα το ιδιωτικό μας κλειδί είναι το 41 και θα το συμβολίζουμε με το γράμμα d

4.7.4.2 Κρυπτογράφηση κειμένου

Επιλέγουμε να κρυπτογραφήσουμε τον αριθμό 99 και θυμίζουμε ότι ο τύπος κρυπτογράφησης είναι ο $C= M^e$ mod n (τύπος 4.14). Ας δούμε ποιο θα είναι το κρυπτοκείμενό μας :

$$C = M^e (mod\, n) = 99^{29} (mod\, 133) = 92 \quad (4.29)$$

Το κρυπτοκείμενό μας είναι αυτό που θα σταλεί το οποίο μόνο ο κάτοχος του συ σχετιζόμενου ιδιωτικού κλειδιού θα μπορέσει να αποκρυπτογραφήσει. Θυμίζουμε σε αυτό το σημείο ότι το δημόσιο κλειδί είναι το e=29 και το ιδιωτικό κλειδί είναι το d=41 και ο τύπος αποκρυπτογράφησης είναι ο M=Cd (mod n) (4.15) συνεπώς θα έχουμε:

$$M = C^d (mod\, n) = 92^{41} (mod\, 133) = 99 \quad (4.30)$$

4.7.4.3 Ψηφιακή υπογραφή μηνύματος

Μπορούμε φυσικά να χρησιμοποιήσουμε το ίδιο ζευγάρι κλειδιών για να δημιουργήσουμε την ψηφιακή υπογραφή του μηνύματός μας. Χρησιμοποιούμε τον τύπο της κρυπτογράφησης του μηνύματός μας μόνο που αυτή τη φορά αντί για το δημόσιο κλειδί θα χρησιμοποιήσουμε το ιδιωτικό, δηλαδή θα έχουμε:

$$S = M^d (mod\, n) = 99^{41} (mod\, 133) = 36 \quad (4.31)$$

Αν εμείς ως παραλήπτες μπορούμε να χρησιμοποιήσουμε το συ σχετιζόμενο δημόσιο κλειδί και την ψηφιακή υπογραφή για να ανακτήσουμε το μήνυμα, τότε θα ξέρουμε ότι μόνο αυτός που έχει το αυθεντικό ιδιωτικό κλειδί θα μπορούσε να την δημιουργήσει. Χρησιμοποιούμε τον τύπο της αποκρυπτογράφησης (4.15),στη θέση του μηνύματος βάζουμε την ψηφιακή υπογραφή μας και χρησιμοποιούμε το δημόσιο κλειδί. Άρα έχουμε:[71]

$$V = M^e (mod\, n) = 36^{29} (mod\, 133) = 99 \quad (4.32)$$

4.7.4 Ασφάλεια αλγορίθμου RSA

Η ασφάλεια του RSA σχετίζεται με την δυσκολία της παραγοντοποίησης μεγάλων αριθμών ενώ η δημοτικότητα του είναι άρρηκτα συνδεδεμένη με την ευκολία της πράξεως υψώσεως σε δύναμη mod n και της ευρέσεως δύο μεγάλων πρώτων αριθμών.

Η κριτική που δέχεται αφορά τους μεγάλους χρόνους κρυπτογράφησης ακόμα και μέτριων σε μέγεθος μηνυμάτων και την έλλειψη μαθηματικής απόδειξης ότι μόνο αν καταστεί εφικτή η παραγοντοποίηση μεγάλων αριθμών είναι δυνατόν να σπάει αλγόριθμος. [2] [72]

4.8 Ο αλγόριθμος ψηφιακής υπογραφής DSA (Digital Signature Algorithm, DSA)

4.8.1 Ορισμοί

4.8.1.1 Κρυπτοσυστήματα ελλειπτικής καμπύλης

Τα κρυπτοσυστήματα ελλειπτικής καμπύλης (Elliptic curve cryptosystems, ECC) εφευρέθηκαν από τους Neal Koblitz και Victor Miller το 1985. Αυτά τα κρυπτοσυστήματα μπορεί να θεωρήσουμε ότι είναι ανάλογα του παλαιότερου διακριτού αλγόριθμου (discreet logarithm,DL) στον οποίο οι υποομάδες του συνόλου Z^*_P αντικαθιστούνται από την ομάδα των σημείων μίας ελλειπτικής καμπύλης σε ένα πεπερασμένο πεδίο.

Η μαθηματική βάση για την ασφάλεια των κρυπτοσυστημάτων ελλειπτικής καμπύλης (Elliptic curve cryptosystems, ECC)είναι η υπολογιστική δυσκολία επίλυσης του προβλήματος του διακριτού λογαρίθμου ελλειπτικής καμπύλης (elliptic curve discrete logarithm problem,ECDLP).

Εφόσον το πρόβλημα του διακριτού λογαρίθμου ελλειπτικής καμπύλης (ECDLP) φαίνεται δυσκολότερο από το πρόβλημα του διακριτού λογαρίθμου (DLP) **η ισχύς ανά bit κλειδιού (strength-per-key-bit)** δηλαδή ο αριθμός τον λειτουργιών που θα πρέπει να υλοποιήσει κάποιος προκειμένου να σπάσει τον αλγόριθμο είναι μεγαλύτερος στα συστήματα ελλειπτικής καμπύλης από ότι στα συμβατικά διακριτά λογαριθμικά συστήματα.

Συνεπώς, στα συστήματα ελλειπτικής καμπύλης μπορούν να χρησιμοποιηθούν μικρότερες παράμετροι από ότι στα διακριτά λογαριθμικά συστήματα και να έχουμε το ίδιο επίπεδο ασφάλειας.

Τα πλεονεκτήματα των μικρότερων παραμέτρων είναι η μεγαλύτερη ταχύτητα της εφαρμογής μας και μικρότερου μεγέθους κλειδιά και πιστοποιητικά.

Αυτά τα πλεονεκτήματα είναι σημαντικά σε περιβάλλοντα όπου η επεξεργαστική ισχύ,ο χώρος αποθήκευσης, το εύρος ζώνης ή η κατανάλωση ισχύος υπάγονται σε περιορισμούς. [73]

4.8.1.2 Ορισμός του αλγόριθμου DSA

Ο αλγόριθμος DSA αναπτύχθηκε από την αμερικανική Υπηρεσία ασφάλειας (National Security Agency, NSA). Προτάθηκε από τον David W. Kravitz,το 1991 από το Εθνικό Ινστιτούτο Προτύπων και Τεχνολογίας των ΗΠΑ (National Institute of Standards and Technology, NIST) και πιστοποιήθηκε το 1994.

Ο αλγόριθμος Ψηφιακής Υπογραφής (Digital Signature Algorithm, DSA) ορίστηκε σε ένα Κυβερνητικό Ομοσπονδιακό Πρότυπο Επεξεργασίας Πληροφοριών των Η.ΠΑ (Government Federal Information Processing Standard, FIPS) το οποίο ονομάζεται Πρότυπο Ψηφιακών Υπογραφών (Digital Signature Standard,DST). Σε αυτό το πρότυπο ορίζονται οι αλγόριθμοι οι οποίοι χρησιμοποιούνται για να παράγουν ψηφιακές υπογραφές με την βοήθεια των αλγορίθμων κατακερματισμού για αυθεντικοποίηση των ηλεκτρονικών εγγράφων.[72][73]

Πρόκειται για έναν αλγόριθμο ψηφιακής υπογραφής και όχι κρυπτογράφησης και χρησιμοποιεί κρυπτογραφία δημόσιου κλειδιού για την παραγωγή ψηφιακών υπογραφών [74]

Η ασφάλεια του DSA είναι βασισμένη στην υπολογιστική δυσκολία υπολογισμού διακριτών λογαρίθμων μέσα σε ένα πεπερασμένο σώμα [75]

4.8.2 Παράμετροι του αλγόριθμου DSA

Για την παραγωγή ψηφιακών υπογραφών με την βοήθεια του αλγορίθμου DSA χρειαζόμαστε τη βοήθεια μίας ομάδας παραμέτρων τομέα (domain parameters), ένα ιδιωτικό κλειδί έστω x, έναν μυστικό αριθμό ανά μήνυμα (per -message) έστω k, δεδομένα προς υπογραφή και μία συνάρτηση κατακερματισμού (hash function).

Η διαδικασία της αυθεντικοποίησης της ψηφιακής υπογραφής γίνεται με χρήση των ίδιων παραμέτρων τομέα, ένα δημόσιο κλειδί έστω y το οποίο σχετίζεται μαθηματικά με το ιδιωτικό κλειδί x το οποίο χρησιμοποιήθηκε για να παράξει την ψηφιακή υπογραφή, τα δεδομένα που πρέπει να επαληθευτούν και την ίδια συνάρτηση κατακερματισμού που χρησιμοποιήθηκε για την παραγωγή της ψηφιακής υπογραφής.

Οι παράμετροι είναι ως εξής :

- Ένας ακέραιος αριθμός p για τον οποίο ισχύει $2^{L-1} < p < 2^{L}$. Το L είναι το μήκος σε bit που θα πρέπει να έχει ο αριθμός p. Η τιμή του L θα αναφερθεί στην επόμενη παράγραφο.
- Ένας ακέραιος αριθμός q ο οποίος θα είναι ένας πρώτος διαιρέτης του (p-1) για τον οποίο θα ισχύει $2^{N-1} < q < 2^{N}$. Το N είναι το μήκος σε bit που θα πρέπει να έχει ο αριθμός q και η τιμή του θα αναφερθεί στην επόμενη παράγραφο.
- Μία γεννήτρια (g) υποομάδας τάξης q στην πολλαπλασιαστική ομάδα GF(p) για την οποία θα πρέπει να ισχύει 1 < g< p.
- Το ιδιωτικό κλειδί έστω x το οποίο πρέπει να παραμείνει μυστικό και είναι ένας τυχαίος ή ψευδοτυχαίος ακέραιος για τον οποίο θα πρέπει να ισχύει 0 < x < q πράγμα που σημαίνει ότι το ιδιωτικό κλειδί θα ανήκει στο εύρος των αριθμών [1, q–1].
- Το δημόσιο κλειδί έστω y το οποίο θα πρέπει να υπολογιστεί ως συνάρτηση του ιδιωτικού.
- Ένας μυστικός ακέραιος αριθμός έστω k ο οποίος είναι μοναδικός για το κάθε μήνυμα ο οποίος παράγεται τυχαία ή ψευδοτυχαία και για τον οποίο θα πρέπει να ισχύει 0 < k < q δηλαδή θα πρέπει να ανήκει στο εύρος των αριθμών [1, q–1] [76]

4.8.2.1 Παραγωγή Κλειδιών

Το πρότυπο Digital Signature Standard (DSS) ορίζει τις ακόλουθες επιλογές για τιμές του L και του N

- L,N = (1024, 160)
- L,N = (2048, 224)
- L,N = (248, 256)
- L, N = (3072, 256) [77]

Το επόμενο βήμα της διαδικασίας είναι να ορίσουμε τους αριθμούς p και q σύμφωνα με τις προϋποθέσεις που αναπτύξαμε στην προηγούμενη παράγραφο.

Στη συνέχεια θα πρέπει να υπολογιστεί η τιμή της γεννήτριας δηλαδή ο αριθμός g και θα χρησιμοποιήσουμε τον ακόλουθο τύπο :

$$g = h^{e} \bmod p = h^{\frac{(p-1)}{q}} \bmod p \qquad (4.33)$$

Ο αριθμός h μπορεί να είναι ένας οποιοδήποτε ακέραιος αριθμός για τον οποίο πρέπει να ισχύει 1 < h < (p – 1). Θα πρέπει να πούμε ότι ο αριθμός h μπορεί να παραχθεί από μία γεννήτρια τυχαίων αριθμών.

Στην συνέχεια θα πρέπει να επιλεγεί ένας αριθμός x ως ιδιωτικό κλειδί ο οποίος θα πρέπει να τηρεί τις προϋποθέσεις που ορίστηκαν στην προηγούμενη παράγραφο.

Έπειτα θα πρέπει να υπολογίσουμε το δημόσιο κλειδί με τον ακόλουθο τύπο:

$$y = g^{x} \bmod p$$

(4.34)

Το αποτέλεσμα αυτής της διαδικασίας είναι το ακόλουθο:

- Δημόσιο Κλειδί: (p, q, g, y)
- Ιδιωτικό κλειδί (x)

Ας δούμε ένα αριθμητικό παράδειγμα: για να απλοποιήσουμε τους υπολογισμούς μας ας θεωρήσουμε ότι : (L, N) = (12, 8)

Επιλέγουμε p=2467 και ας δούμε αν η επιλογή μας είναι ορθή:

$$2^{L-1}<p<2^{L}=2^{12-1}<2467<2^{12}=2^{11}<2467<2^{12}=2048<2467<4096 \quad (4.35)$$

Αφού δεν χρειάζεται να αλλάξουμε την τιμή του p επιλέγουμε q=137 και θα πρέπει και εδώ να δούμε εάν επιλέξαμε σωστά:

$$2^{N-1}<q<2^{N}=2^{8-1}<137<2^{8}=2^{7}<137<2^{8}=128<137<256 \quad (4.36)$$

Θα πρέπει να αποδείξουμε ότι ο αριθμός q=137 αποτελεί πρώτο διαιρέτη του p-1. Έχουμε λοιπόν:

$$p-1=2467-1=2466 \quad (4.37)$$

Και πραγματικά αυτό ισχύει αφού :

$$2466=137*18 \quad (4.38)$$

Για να υπολογίσουμε την τιμή του g θέτουμε h=3 και έχουμε :

$$g=h^{\frac{p-1}{q}} \bmod p=3^{18} \bmod 2467=342 \quad (4.39)$$

Επιλέγουμε x=8 και υπολογίζουμε το δημόσιο κλειδί ως εξής :

$$y = g^x \bmod p = 342^7 \bmod 2467 = 282 \quad (4.40)$$

Τα κλειδιά είναι τα εξής :

- Δημόσιο κλειδί: (2467, 137, 342, 282)
- Ιδιωτικό κλειδί: 7 [78]

4.8.2.2 Υπογραφή μηνύματος

Για την παραγωγή της ψηφιακής υπογραφής ενός μηνύματος M, M = $(m_1 \ldots m_n)_2$η οποία είναι το ζεύγος τιμών r, s, ακολουθούμε τα εξής βήματα :

- Επιλέγουμε μία συνάρτηση SHA −i για την οποία ισχύει $\{0, 1\}^l \rightarrow \{0, 1\}^e$ i ∈ {0, 1, 2, 3} l ≥ n και e ≥ N.
- Επιλέγουμε τυχαία έναν ακέραιο k για τον οποίο ισχύει k ∈ {1, ... q − 1} που αποτελεί εφήμερο κλειδί και κρατείται μυστικός.
- Υπολογίζουμε το r με την χρήση του τύπου r = $(g^k \bmod p) \bmod q$.
- Υπολογίζουμε το k^{-1} και το SHA − i(M)
- Υπολογίζουμε τον ακέραιο H(M) ο οποίος είναι τα πρώτα N bits αριστερά του SHA − i(M) (συμπύκνωση)
- Υπολογίζουμε το s με τη χρήση του τύπου s = k^{-1} (H(M) + xr) mod q.
- Σε περίπτωση που το αποτέλεσμά μας είναι r = 0 ή s = 0 θα πρέπει να επαναλάβουμε τη διαδικασία

Ας δούμε ένα παράδειγμα το οποίο αποτελεί συνέχεια του προηγούμενου παραδείγματός μας. Ας θεωρήσουμε ένα μήνυμα M =$(m_1 \ldots m_{58})_2$

Επιλέγουμε την SHA − 2 με l = 2^{64} και e = 224. Η συνάρτηση αυτή είναι η SHA − 224[11].Στη συνέχεια θέτουμε k=6

Τώρα θα υπολογίσουμε το r με τον τύπο που προαναφέραμε ως εξής:

$$r=(g^{k} mod\, p) mod\, q=(342^{6} mod\, 2467) mod\, 137=953 mod\, 137=1 \quad (4.41)$$

Στη συνέχεια υπολογίζουμε την τιμή του όρου k^{-1} που είναι το πολλαπλασιαστικό αντίστροφο του k mod q (k^{-1}*k=1 mod q) ο ο οποίος είναι ο αριθμός είκοσι τρία και αυτό διότι αν το πολλαπλασιάσουμε με το k=6 το αποτέλεσμα είναι 137 και το υπόλοιπο είναι μονάδα και συνεπώς ικανοποιείται η παραπάνω εξίσωση

Ας υποθέσουμε τώρα ότι ισχύει ο ακόλουθος τύπος για τη συνάρτηση κατακερματισμού :

$$SHA_{224}(M)=\underbrace{00011100\,Q}_{224\,bits} \quad (4.42)$$

Υπολογίζουμε τον ακέραιο H(M) ως εξής :

$$H(M)=00011100=2^{4}*1+2^{3}*1+2^{2}*1=2 \quad (4.43)$$

Τέλος, θα υπολογίσουμε τον όρο s με την εφαρμογή του τύπου που ήδη αναφέραμε ως εξής :

$$s=k^{-1}(H(M)+xr) mod\, q=23(28+7*131) mod\, 137=23*945 mod\, 137=89 \quad (4.44)$$

Το αποτέλεσμα της υπογραφής του μηνύματος M είναι το ζευγάρι r=131, s=89 [78]

4.8.2.3 Γνησιότητα υπογεγραμμένου μηνύματος

Η διαδικασία της αποδοχής ή της απόρριψης ενός μηνύματος $M=(m_1 \ldots m_n)_2$ έχει ως εξής :

- Αρχικά θα πρέπει να ελέγξουμε εάν 0 < r < q και 0 < s < q διότι αν ένα από τα δύο δεν ισχύει θα πρέπει να απορριφθεί η υπογραφή.
- Υπολογίζουμε $w = s^{-1} \bmod q$.
- Υπολογίζουμε SHA − i(M).

- Υπολογίζουμε H(M).
- Υπολογίζουμε u_1 = H(M)w mod q.
- Υπολογίζουμε u_2 = rw mod q.
- Υπολογίζουμε v = ($g^{u1} y^{u2}$ mod p) mod q

Η υπογραφή θεωρείται γνήσια αν ισχύει v = r. Ας συνεχίσουμε το παράδειγμα των δύο προηγουμένων παραγράφων.

Διαπιστώνουμε ότι και 0 < r < q και 0 < s < q ισχύουν, αφού 0 < 131 < 137 και 0 < 89 < 137 οπότε προχωράμε στα επόμενα βήματα της διαδικασίας

Υπολογίζουμε το w = s^{-1} mod q ως εξής :

$$w = s^{-1} mod\, q = 89^{-1} mod\, 137 = 117 \quad (4.45)$$

Ας υποθέσουμε τώρα ότι ισχύει ο ακόλουθος τύπος για τη συνάρτηση κατακερματισμού όπως και στην προηγούμενη παράγραφό μας :

$$SHA_{224}(M) = \underbrace{00011100\,Q}_{224\,bits} \quad (4.46)$$

Ο όρος H(M) υπολογίστηκε στην προηγούμενη παράγραφο και ισχύει H(M)=28

Υπολογίζουμε u_1 = H(M)w mod q ως εξής :

$$u_1 = H(M)\, w\, mod\, q = 28 * 117\, mod\, 137 = 3276\, mod\, 137 = 125 \quad (4.47)$$

Υπολογίζουμε u_2 = rw mod q ως ακολούθως:

$$u_2 = rw\, mod\, q = 131 * 117\, mod\, 137 = 15327\, mod\, 137 = 120 \quad (4.48)$$

Τέλος, υπολογίζουμε v = ($g^{u1} y^{u2}$ mod p) mod q όπως βλέπουμε παρακάτω:

$$v = (g^{u_1} y^{u_2} mod\, p)\, mod\, q = (342^{125} * 282^{120} mod\, 2467)\, mod\, 137 = 953\, mod\, 137 = 131 \quad (4.49)$$

Παρατηρούμε ότι v=r=131. Συνεπώς η ψηφιακή υπογραφή είναι γνήσια και γίνεται αποδεκτή. [78]

4.8.2.4 Πρόταση προς απόδειξη

Αν M είναι ένα μήνυμα, τότε η υπογραφή του είναι το ζεύγος (r, s), αν και μόνο αν, v = r. Έχουμε λοιπόν τον τύπο v = ($g^{u1} y^{u2}$ mod p) mod q και υποθέτουμε ότι (r, s) είναι η υπογραφή του M, άρα:

$$v=(g^{u_1} y^{u_2} \bmod p) \bmod q \qquad (4.50)$$

Γνωρίζουμε ότι u_1 = H(M)w mod q και ότι u_2 = rw mod q συνεπώς αντικαθιστούμε αυτούς τους όρους και έχουμε :

$$v=(g^{H(M)w} y^{rw} \bmod p) \bmod q \qquad (4.51)$$

Μας είναι ήδη γνωστό ότι $y=g^x$ και έτσι, για να έχουμε ίδια βάση αντικαθιστούμε αυτό τον όρο ως εξής :

$$v=((g^{H(M)w} g^{xrw}) \bmod p) \bmod q \qquad (4.52)$$

Τώρα μπορούμε να προσθέσουμε τους εκθέτες και έχουμε:

$$v=(g^{H(M)w+xrw} \bmod p) \bmod q \qquad (4.53)$$

Βγάζουμε στον εκθέτη μας κοινό παράγοντα το w και το αποτέλεσμα είναι το εξής :

$$v=(g^{w(H(M)+xr)} \bmod p) \bmod q \qquad (4.54)$$

Αφού w=s-1 mod q αντικαθιστούμε το w και έχουμε:

$$v=(g^{s^{-1}(H(M)+xr)} \bmod p) \bmod q \qquad (4.55)$$

4.9 Ο Αλγόριθμος Ψηφιακής Υπογραφής Ελλειπτικής Καμπύλης (Elliptic Curve Digital Signature Algorithm ECDSA)

4.9.1 Ορισμός

Ο Αλγόριθμος Ψηφιακής Υπογραφής Ελλειπτικής Καμπύλης (Elliptic Curve Digital Signature Algorithm ECDSA) είναι ο ανάλογος ελλειπτικός αλγόριθμος καμπύλης του Αλγόριθμου Ψηφιακής Υπογραφής (Signature Algorithm, DSA) όπως εξηγήσαμε στην προηγούμενη παράγραφο.

Αρχικά προτάθηκε το 1992 από τον Scott Vanstone σε απάντηση της παράκλησης του Εθνικού Ινστιτούτου Προτύπων και Τεχνολογίας (National Institute of Standards and Technology,NIST) για δημόσια σχόλια για την πρώτη τους πρόταση για Πρότυπα Ψηφιακής Υπογραφής (Digital Signature Standard, DSS)

Έγινε αποδεκτός το 1998 ως πρότυπο ISO (International Standards Organization), έγινε αποδεκτός το 1999 ως πρότυπο ANSI (American National Standards Institute, ANSI X9.62) και τέλος έτυχε αποδοχής 2000 ως πρότυπο IEEE (Institute of Electrical and Electronics Engineers, IEEE 1363-2000) και ως πρότυπο FIPS (Federal Information Processing Standards IPS 186-2).

Τα κλειδιά παράγονται από κρυπτογραφία ελλειπτικής καμπύλης και είναι μικρότερα σε μέγεθος από τον μέσο όρο των κλειδιών που παράγονται από τους αλγόριθμους ψηφιακής υπογραφής.

Η κρυπτογραφία ελλειπτικής καμπύλης είναι μία μορφή κρυπτογραφίας δημόσιου κλειδιού η οποία είναι βασισμένη στην αλγεβρική δομή των ελλειπτικών καμπύλων σε πεπερασμένα πεδία.

Η κρυπτογραφία ελλειπτικής καμπύλης χρησιμοποιείται κυρίως για την δημιουργία ψευδοτυχαίων αριθμών ψηφιακών υπογραφών και άλλων στοιχείων.

Όπως αναφέραμε και πιο πάνω μία ψηφιακή υπογραφή αποτελεί μία μέθοδος αυθεντικοποίησης η οποία χρησιμοποιείται στην ασύμμετρη κρυπτογραφία. [78]

4.9.2 Χρήσεις αλγορίθμου ECDSA

Ο αλγόριθμος ECDSA κάνει ό,τι ακριβώς κα οι άλλοι ασύμμετροι αλγόριθμοι αλλά είναι περισσότερο αποτελεσματικός. Αυτό οφείλεται στο ότι ο ECDSA χρησιμοποιεί κλειδιά μικρότερου μεγέθους για να δημιουργήσει το ίδιο επίπεδο ασφάλειας με τους άλλους αλγόριθμους ασύμμετρης κρυπτογραφίας.

Ο ECDSA χρησιμοποιείται για την δημιουργία πιστοποιητικών ECDSA τα οποία είναι ηλεκτρονικά έγγραφα με τα οποία πιστοποιείται η ταυτότητα των ιδιοκτητών. Ένα τέτοιο πιστοποιητικό μπορεί να περιέχει πληροφορίες για το κλειδί που χρησιμοποιήθηκε για την δημιουργία του πιστοποιητικού, για τον ιδιοκτήτη του πιστοποιητικού και την υπογραφή του εκδότη του πιστοποιητικού ο οποίος αποτελεί μία πιστοποιημένη οντότητα.

Γενικά, ο τρόπος που δουλεύει ο αλγόριθμος ECDSA βασίζεται στην ανάλυση μίας ελλειπτικής καμπύλης και στην επιλογή ενός σημείου της. Το σημείο που επιλέγεται πολλαπλασιάζεται με έναν άλλο αριθμό και έτσι δημιουργείται ένα καινούργιο σημείο στην καμπύλη. Το καινούργιο σημείο είναι δύσκολο να βρεθεί ακόμα και αν έχουμε στη διάθεσή μας το αρχικό σημείο. Η πολυπλοκότητα του ECDSA φανερώνει ότι ο συγκεκριμένος αλγόριθμος είναι πιο ασφαλής απέναντι στις σύγχρονες μεθόδους σπασίματος.

4.9.3 Πλεονεκτήματα του αλγορίθμου ECDSA

Τα πλεονεκτήματα του αλγορίθμου ECDSA είναι :

- Συγκριτικά με τους αλγόριθμους κρυπτογράφησης RSA, DSA ο αλγόριθμος ECDSA που είναι ένας αλγόριθμος ελλειπτικής καμπύλης θεωρείται ασφαλέστερος για συγκεκριμένο μέγεθος κλειδιών. Το ίδιο ισχύει και για κλειδιά μικρότερου μεγέθους τα οποία είναι πιο ευάλωτα συγκριτικά με αυτά των οποίων το μέγεθος είναι μεγαλύτερο.
- Ο χρόνος και ο χώρος στη μνήμη που απαιτείται για την παραγωγή και διακίνηση μηνυμάτων είναι λιγότερος συγκριτικά με παλαιότερα εργαλεία
- Απαιτείται λιγότερη υπολογιστική ισχύ για την υλοποίηση του ECDSA κάτι το οποίο καθιστά την υλοποίησή του ελκυστική στους ερευνητές.
- Γενικά, τα συστήματα που βασίζονται σε αλγορίθμους ελλειπτικών καμπυλών είναι οικονομικότερα.

4.9.4 Μειονεκτήματα του αλγορίθμου ECDSA

Τα μειονεκτήματα του αλγορίθμου ECDSA είναι :

- Ο αλγόριθμος ECDSA είναι ανοιχτού λογισμικού και κατά συνέπεια πολλοί χρήστες μπορεί να έχουν πρόσβαση κάποιοι εκ των οποίων μπορεί να δημιουργήσουν εργαλεία παραβίασής του.
- Μία κρυπτογραφημένη πληροφορία μπορεί να είναι δύσκολο να ανακτηθεί ιδιαίτερα σε περίπτωση παραβίασης της πλατφόρμας.

4.8.5 Η Ψηφιακή Υπογραφή ECDSA

4.8.5.1 Παραγωγή κλειδιών

Αρχικά επιλέγουμε μία ελλειπτική καμπύλη E στο σύνολο Z_P τέτοια ώστε:

$p \in P$

$(2^{80} + 1)^2 < p$

Επιλέγουμε ένας ν ακέραιο αριθμό q με τα εξής κριτήρια :

- $q \in P$
- $q \mid |E(Zp)|$
- $2^{N-1} < q < {}^{2N}$ έτσι ώστε:

  $N \in [160, 223]$ όταν $|E(Zp)| \leq 2^{10} q$

  $N \in [224, 255]$ όταν $|E(Zp)| \leq 2^{14} q$.

  $N \in [256, 383]$ όταν $|E(Zp)| \leq 2^{16} q$.

  $N \geq 512$ όταν $|E(Zp)| \leq 2^{32} q$
- Επιλέγουμε ένα σημείο $P = (X_P, Y_P) \in E(Zp)$ τάξης q.
- Επιλέγουμε έναν ακέραιο $A \in \{0, ..., q-1\}$.
- Υπολογίζουμε $Q = (X_Q, Y_Q) = AP$.

Το αποτέλεσμα είναι :

➢Δημόσιο κλειδί: (P, Q)

➢Ιδιωτικό κλειδί: A

Ας δούμε ένα παράδειγμα επιλέγοντας πρώτα την ελλειπτική καμπύλη E :

$$E: y^2 \equiv x^3 + 2 \bmod 7 \qquad (4.57)$$

Τα σημεία της ελλειπτικής καμπύλης είναι τα διατεταγμένα ζεύγη (x, y) mod 7 τα οποία ικανοποιούν
την εξίσωση, και το σημείο στο άπειρο τα οποία μπορούν να υπολογιστούν ως εξής: οι δυνατές περιπτώσεις για το x mod 7 είναι 0, 1, 2, 3, 4,5,6. και αντικαθιστώντας
κάθε μια από τις τιμές αυτές στην εξίσωση, βρίσκουμε τις αντίστοιχες τιμές του y που την επαληθεύουν
Τα σημεία της ελλειπτικής καμπύλης είναι :

$$E(Z_7)=(0,3),(0,4),(3,1),(3,6),(5,1),(5,6),(6,1),(6,6) \cup O\infty$$

Επιλέγουμε A = 2,και υπολογίζουμε Q = 2 (5, 1) = (5, 6).

Τα ζητούμενα κλειδιά είναι:

- Δημόσιο κλειδί: ((5, 1), (5, 6))
- Ιδιωτικό κλειδί: 2 [78]

4.9.5.2 Υπογραφή μηνύματος

Για την υπογραφή ενός μηνύματος M = $(m_1 \ldots m_n)_2$ η διαδικασία είναι η ακόλουθη :

- Επιλέγουμε μια συνάρτηση SHA − i : {0, 1}l −→ {0, 1}e για την οποία i ∈ {0, 1, 2, 3}, l ≥ n και e ≥ N.
- Επιλέγουμε τυχαία ένας ακέραιος k ∈ {1,... q − 1} (εφήμερο κλειδί) ο οποίος κρατείται μυστικός.
- Υπολογίζουμε kP = (X_{kP}, Y_{kP}).
- Υπολογίζουμε r = X_{kP} mod q.
- Υπολογίζουμε k^{-1} και SHA − i(M).
- Υπολογίζουμε τον ακέραιος H(M), που είναι τα πρώτα N bits από αριστερά του SHA − i(M).
- Υπολογίζουμε s = k^{-1} (H(M) + Ar) mod q.
- Το αποτέλεσμα της διαδικασίας είναι η υπογραφή του μηνύματος M η οποία αποτελείται από το ζεύγος (r,s)

Ας συνεχίσουμε το παράδειγμα της προηγούμενης παραγράφου θεωρώντας το μήνυμα M = $(m_1 \ldots m_{47})_2$
Επιλέγουμε την SHA − 2 με l = 2^{64} και e = 2^{24} δηλαδή τη SHA − 224.Στη συνέχεια επιλέγουμε k=2.
Το 2P υπολογίστηκε στην προηγούμενη παράγραφο και ισχύει 2P = (5, 6).
Στη συνέχεια, αφού υπενθυμίσουμε ότι q=3, υπολογίζουμε το r ως εξής :

$$r = X_{kP} \bmod q = 5 \bmod 3 = 2 \qquad (4.58)$$

Στη συνέχεια υπολογίζουμε την τιμή του όρου k^{-1} που είναι το πολλαπλασιαστικό αντίστροφο του k mod q (k^{-1}*k=1 mod q) Θυμίζουμε ότι k=2 q=3.Το πολλαπλασιαστικό αντίστροφο είναι ο αριθμός δύο (k^{-1}=2)

Υποθέτουμε ότι έχουμε:

$$SHA_{224}(M) = \underbrace{01}_{224-bits}Q \qquad (4.59)$$

Το επόμενο βήμα είναι ο υπολογισμός του H(M) ο οποίος έχει ως εξής :

$$H(M)=01=0*2^{1}+1*2^{0}=1 \tag{4.60}$$

Αμέσως μετά αφού θυμηθούμε ότι ισχύει A=2, r=2 θα πρέπει να υπολογίσουμε το s οπότε έχουμε :

$$s=k^{-1}*(H(M)+A*r)\,mod\,q=2^{-1}*(1+2\cdot 2)\,mod\,3=2 \tag{4.61}$$

Το αποτέλεσμα της υπογραφής του M, είναι το ζεύγος: (2, 2).

4.9.5.3 Γνησιότητα υπογραφής μηνύματος

Η διαδικασία αποδοχής ή απόρριψης της υπογραφής (r, s) ενός μηνύματος $M = (m_{1.} \dots m_{n})_{2,}$ αποτελείται από τα εξής βήματα:

- Ελέγχουμε αν ισχύει $0 < r < q$ και $0 < s < q$. αφού αν δεν ισχύει ένα από αυτά η υπογραφή απορρίπτεται
- Υπολογίζουμε $w = s^{-1} \bmod q$.
- Υπολογίζουμε $SHA - i(M)$.
- Υπολογίζουμε H(M).
- Υπολογίζουμε $u_1 = H(M)w \bmod q$.
- Υπολογίζουμε $u_2 = rw \bmod q$.
- Υπολογίζουμε $v = X_R \bmod q$ όπου $(X_{R,} Y_R) = u_1 P + u_2 Q$.

Η υπογραφή (r, s) είναι γνήσια, αν v = r.

Ας συνεχίσουμε το παράδειγμα βλέποντας αρχικά ότι ισχύει $0 < r < q$ και $0 < s < q$ αφού $0 < 2 < 3$ και $0 < 2 < 3$ συνεπώς συνεχίζουμε τη διαδικασία.

Υπολογίζουμε τον όρο w ως εξής :

$$w=s^{-1}mod\,q=2^{-1}mod\,3=2\,mod\,3=2 \tag{4.61}$$

Αναφορικά με τ η συνάρτηση $SHA - i(M)$ ισχύει ό,τι και προηγουμένως δηλαδή :

$$SHA_{224}(M)=\underbrace{01\,Q}_{224-bits} \tag{4.62}$$

Ο όρος H(M) υπολογίστηκε στην προηγούμενη παράγραφο και ισχύει H(M)=1. Το επόμενο βήμα είναι ο υπολογισμός του u_1 ως εξής :

$$u_1 = H(M) w \bmod q = 1 * 2 \bmod 3 = 2 \tag{4.63}$$

Υπολογίζουμε το u2 με τον ακόλουθο τρόπο :

$$u_2 = rw \bmod q = 2 * 2 \bmod 3 = 1 \tag{4.64}$$

Υπολογίζουμε (XR, YR) = u_1 P + u_2 Q ως εξής :

$$(X_R, Y_R) = u_1 P + u_2 Q = 2(5,1) + (5,6) = (5,6) \tag{4.65}$$

Ο όρος v υπολογίζεται ως εξής :

$$v = X_R \bmod q = 5 \bmod 3 = 2 \tag{4.66}$$

4.9.6 Πρόταση προς απόδειξη

Αν M είναι ένα μήνυμα, τότε η υπογραφή του είναι το ζεύγος (r, s), αν και μόνο αν, v = r.

Υποθέτουμε ότι (r, s) είναι η υπογραφή του M. Έχουμε :

$$v = X_R \bmod q \tag{4.67}$$

Αφού ξέρουμε ότι (XR, YR) = u1 P + u2 Q μπορούμε να αντικαταστήσουμε τον όρο και να έχουμε :

$$v=(u_1 P+u_2 Q)_{x,x} \bmod q \tag{4.68}$$

Είναι γνωστό σε εμάς ότι u1 = H(M)w mod q και u2 = rw mod q οπότε αν αντικαταστήσουμε θα έχουμε:

$$v=(H(M)w\,P+rwQ)_{x,x} \bmod q \tag{4.69}$$

Βγάζουμε κοινό παράγοντα το w και έχουμε:

$$v=(w(H(M)P+rQ))_{x,x} \bmod q \tag{4.70}$$

Γνωρίζουμε ότι w = s−1 mod q και ότι Q=AP, συνεπώς αντικαθιστούμε και έχουμε:

$$v=(s^{-1}(H(M)P+A\,rP))_{x,x} \bmod q \tag{4.71}$$

Έχουμε δει ότι s = k^{-1} (H(M) + Ar) mod q, οπότε αντικαθιστούμε και το αποτέλεσμα είναι :

$$v=(kP)_{x,x} \bmod q \tag{4.72}$$

Ισχύει kP = (X_{kP}, Y_{kP}) και έχουμε μετά από αντικατάσταση :

$$v=X_{kP} \bmod q \tag{4.73}$$

Άρα συμπεραίνουμε ότι r=v. [78]

Επίλογος

Ξεκαθαρίσαμε στα πλαίσια αυτού του κεφαλαίου έννοιες όπως ασύμμετρη κρυπτογραφία και δημόσιο κλειδί. Διευκρινίσαμε πως οι αλγόριθμοι δημόσιου κλειδιού χρησιμεύουν στην παραγωγή κλειδιών και στην δημιουργία ψηφιακών υπογραφών.

Πέρα από τα θεωρητικά σημεία που αναλύθηκαν στο τέταρτο κεφάλαιο ξεχωριστό ενδιαφέρον παρουσιάζουν οι μαθηματικές αναλύσεις των κρυπτογραφικών αλγορίθμων Rsa, Dsa και Ecdsa.
Αυτές οι μαθηματικές προσεγγίσεις μας έδειξαν μία μαθηματική οπτική των κρυπτογραφικών αλγορίθμων.

Κεφάλαιο 5
Συμπεράσματα

5.1 Αλγόριθμοι

Οι αλγόριθμοι ουσιαστικά ήταν σύντροφοι του ανθρώπου προτού καν ο ίδιος ανακαλύψει ότι πρέπει να τους ονομάσει έτσι. Και τους χαρακτηρίζουμε έτσι διότι αλγόριθμος είναι ένα σύνολο οδηγιών οι οποίες μπορεί

να χαρακτηρίζουν ρουτίνες όπως για παράδειγμα μαγείρεμα πλύσιμο ρούχων. Τώρα μαθηματικά μιλώντας ένας αλγόριθμος περιγράφει τα βήματα που χρειάζονται για να λυθεί ένα πρόβλημα. Τα βήματα αυτά πρέπει να είναι πεπερασμένα δηλαδή ο αριθμός τους πρέπει να είναι ορισμένος. Στον αλγόριθμος υπάρχει η είσοδος, η υποδοχή δηλαδή των δεδομένων του προβλήματος και η έξοδος δηλαδή η λύση τους προβλήματος.

Καθ' όλη τη διάρκεια της ιστορίας πολλοί επιστήμονες ασχολήθηκαν με τα μαθηματικά τα οποία ουσιαστικά αποτελούν την μήτρα των αλγορίθμων.

Στα μεσαιωνικά λατινικά ο όρος algorismus αναφερόταν στους υπολογισμούς με το χέρι με αραβικούς αριθμούς θέσεως δηλαδή ένας υπολογισμός ο οποίος έχει μία διαδικασία για να υλοποιηθεί. Τελικά ο όρος αλγόριθμος υιοθετήθηκε από ευρωπαίους λόγιους.

Το επόμενο βήμα ήταν οι αλγόριθμοι να αυτοματοποιηθούν να εκτελούνται δηλαδή με την βοήθεια υπολογιστικών μηχανών μία προσπάθεια που διήρκεσε χρόνια και πολλοί επιστήμονες αγωνίστηκαν για κάποια πράγματα που θεωρούμε δεδομένα όπως για παράδειγμα οι ηλεκτρονικοί υπολογιστές και οι εφαρμογές που είναι εγκατεστημένες σε αυτούς.

Οι αλγόριθμοι ουσιαστικά περιγράφουν την διαδικασία επίλυσης ενός προβλήματος με μαθηματική λογική σε αφαιρετικό επίπεδο. Υπάρχουν κανόνες στην περιγραφή τους φυσικά οι οποίοι δεν μπορούν να παραβιαστούν διότι έτσι δεν θα επιτελέσουν τον σκοπό τους που είναι η λύση ενός προβλήματος αλλά έχουμε μία μαθηματική ελευθερία έκφρασης όσο αντικρουόμενο και να ακούγεται αφού η ψευδογλώσσα προγραμματισμού ή η φυσική γλώσσα διέπονται από κανόνες αλλά όχι τόσο αυστηρούς όσο μία γλώσσα προγραμματισμού. Οι κανόνες ωστόσο πρέπει να τηρούνται διότι λειτουργούν ως μέσο διαφύλαξης των χαρακτηριστικών ενός αλγόριθμου. Οι αλγόριθμοι αποτελούν εργαλείο για τον προγραμματισμό για τα μαθηματικά αλλά και για την Τεχνική Νοημοσύνη.

Υπάρχουν βέβαια τρόποι επαλήθευσης της ορθότητας ενός αλγόριθμου οι οποίοι λειτουργούν ως αποδεικτικό στοιχείο της αποτελεσματικότητάς του.

Ένας αλγόριθμος πρέπει να διακρίνεται από σαφήνεια και οργάνωση δηλαδή ξεκάθαρες εντολές και χρήση οργανωμένων δομών δεδομένων όπως για παράδειγμα ταξινομημένων πινάκων.

Βασικό ρόλο παίζει στον τομέα του προγραμματισμού ο υπολογισμός του χρόνου τρεξίματος ενός προγράμματος το οποίο επιθυμούμε να υλοποιήσουμε με την βοήθεια ενός αλγόριθμου διότι οι πόροι ενός συστήματος δεν είναι άπειροι και θα πρέπει να αξιοποιούνται με σύνεση και προσοχή.

Οι αλγόριθμοι αποτελούν εργαλείο δημιουργικότητας το οποίο οφείλει ο γνώστης να το αξιοποιεί με σύνεση και τήρηση κανόνων.

5.2 Κρυπτογραφία

Κρυπτολογία κρυπτογραφία κρυπτανάλυση :τρεις λέξεις το νόημα των οποίων έχει ως κοινή συνισταμένη την ασφάλεια των επικοινωνιών κάτι το οποίο ήταν ζητούμενο ανέκαθεν. Στην σύγχρονη εποχή η κρυπτογραφία δηλαδή η ανάπτυξη κωδίκων για την ασφαλή διαδικτυακή επικοινωνία των χρηστών είναι ζητούμενο ζωτικής σημασίας. Η αναζήτηση τρόπων ασφαλούς επικοινωνίας όπως και η υποκλοπή μηνυμάτων από "εχθρούς" αποτελεί ένα διαχρονικό φαινόμενο αλλά εμείς θα σταθούμε στο σπάσιμο του κώδικα της μηχανής Enigma από μία χούφτα επιστημόνων στα χέρια των οποίων ήρθε η μηχανή και πολύτιμες πληροφορίες που τους βοήθησαν να σπάσουν των κώδικά της.

Το αν εκτιμήθηκε η προσπάθειά τους από τους ηγέτες όπως θα έπρεπε είναι ένα άλλο θέμα. Εμείς θέλουμε να εστιάσουμε στο ρόλο που μπορεί να διαδραματίσει η επιστήμη γενικά και ειδικά η κρυπτογραφία σε συνθήκες πολέμου, έναν ρόλο διαμετρικά αντίθετο με αυτόν του Οπενχάιμερ.

Η συμμετρική κρυπτογραφία και η ασύμμετρη είναι αποτελούν εργαλεία διαφύλαξης του απόρρητου των επικοινωνιών μέσω διαδικτυακών καναλιών. Τα κλειδιά και οι ψηφιακές υπογραφές είναι τρόποι πιστοποίησης της ταυτότητας των μερών.

Τα μαθηματικά και εδώ, όπως είναι φυσικό, παίζουν πρωτεύοντα ρόλο αφού αποτελούν το θεμέλιο λίθο της δημιουργίας τους. Η ασφάλεια πάλι παίζει πρωτεύοντα ρόλο.

Συμπερασματικά σε έναν ψηφιακό κόσμο η κρυπτογραφία είναι απαραίτητο να εξελίσσεται για την διασφάλιση της ακεραιότητας των επικοινωνιών. Το μέλλον μας επιφυλάσσει προκλήσεις με τις οποίες η επιστημονική κοινότητα πρέπει να αναμετρηθεί. Ας ελπίσουμε ότι αυτό θα γίνει με επιτυχία.

Βιβλιογραφία

[1] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Cliford Stein, Introduction to Alorithms Fourth Edition, The MIT Press Cambridge, Massachusetts London, England 2022

[2] Παναγιώτης Δ. Μποτζάνης, Αλγόριθμοι, Εκδόσεις Τζιόλα, Θεσσαλονίκη 2005

[3]" What is Algorithm | Introduction to Algorithms",https://www.geeksforgeeks.org/introduction-to-algorithms/

[4]Jeff Erickson, Algorithms, Copyright 2019 Jeff Erickson

[5] Matteo Pasquinelli, From algorism to algorithm: A brief history of calculation from the Middle Ages to the present day, https://www.academia.edu/71766198

[6] History of Algorithms and Algorithmic Thinking, https://www.cs.ubbcluj.ro/~forest/hcs/cursuri/course-notes.pdf

[7] "Characteristics of an Algorithm",https://medium.com/@bhattshlok12/characteristics-of-an-algorithm-49cf4d7bcd9

[8] Chapter 2 2.1 "Algorithms and complexity analysis",https://fall14cs.files.wordpress.com/2016/04/chapter2-algorithms-and-complexity-analysis.pdf

[9] Α. Ντελόπουλος, Υπολογιστικά Συστήματα : Αλγόριθμοι, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης Νοέμβριος 2003 https://www.eng.auth.gr/ad/17/notes/17-algorithms-notes-v1.0.pdf

[10]Michael T. Goodrich, Roberto Tamassia, "Algorithm Design and Applications", Wiley, https://canvas.projekti.info/ebooks/Algorithm%20Design%20and%20Applications%5BA4%5D.pdf

[11]Δημήτρης Ψούνης, "Ενότητα 1 Ανάλυση αλγορίθμων", https://www.slideshare.net/DimitrisPsounis/30-11-50926747

[12] "Αλγόριθμοι αναζήτητσης", http://repfiles.kallipos.gr/html_books/4410/Ch7.html

[13] "Είναι η Δυαδική Αναζήτηση πάντα ποιο αποδοτική απο τη Σειριακή;" http://www.algorithmos.gr/seiriaki-vs-diadiki-anazitisi.html

[14] Jon Kleinberg, Eva Tardos, "Σχεδιασμός αλγορίθμων, Εκδόσεις Κλεδάριθμος, 2008

[15] Pagourtzis, Aristeidis, Zachos, Efstathios,Grontas, Panagiotis " Εισαγωγή στην Κρυπτολογία" https://repository.kallipos.gr/bitstream/11419/5440/2/ch1.pdf

[16] Jonathan Kartz, Yehuda Lindell, Introduction to Modern Cryptography CRC Press, https://eclass.uniwa.gr/modules/document/file.php/CSCYB105/Reading%20Material/%5BJonathan_Katz%2C_Yehuda_Lindell%5D_Introduction_to_Mo%282nd%29.pdf

[17] "Βασικές αρχές κρυπτανάλυσης ", https://static.eudoxus.gr/books/52/chapter-2252.pdf

[18] Fred Cohen &Associates Specializing in Information Protection Since 1977, "A Short History of Cryptography", https://web.itu.edu.tr/~orssi/dersler/cryptography/Chap2-1.pdf

[19] " Έργα και ιστορίες από τις ανασκαφές της Ιταλικής Αρχαιολογικής Σχολής Αθηνών " https://www.scuoladiatene.it/images/documents/Il_disco_di_Festo%CC%80s_1908.pdf

[20] "History of cryptography", https://uwillnvrknow.github.io/deCryptMe/pages/history.htm

[21] "Playfair Cipher: Beginner's Guide | Unext", https://u-next.com/blogs/cyber-security/playfair-cipher/

[22] Lester Hill, "Cryptography in Algebraic Alphabet", "The American Mathematical Monthly, Vol. 36 (Jun.-Jul,1929) p.p 306-312, April 2009, https://tleise.people.amherst.edu/Math272Spring2014/HillCipherMonthlyArticle.pdf

[23] " Enigma: Η μηχανή που νικήθηκε από τους μαθηματικούς, https://www.ma8imatikos.gr/enigma-η-μηχανή-που-νικήθηκε-από-τους-μαθημ/

[24] “Αυτό ήταν το μυστικό “όπλο” των Γερμανών που κρυπτογραφούσε τα μηνύματα στον Β Παγκόσμιο Πόλεμο. Έμοιαζε με γραφομηχανή. Πώς “έσπασε” το Enigma”,https://www.mixanitouxronou.gr/ayto-itan-to-mystiko-quot-oplo-quot-ton-germanon-poy-kryptografoyse-ta-minymata-ston-v-pagkosmio-polemo-emoiaze-me-grafomichani-pos-quot-espase-quot-to-enigma/
[25] Britanica, Sanat Pai Raikar, “Bombe, code-breaking machine”, https://www.britannica.com/topic/Bombe
[26] “Alan Turing. Ο μαθηματικός που “έληξε” τον Β’ Παγκόσμιο Πόλεμο”, https://e-noesis.gr/alan-turing-o-mathimatikos-poy-quot-elixe-quot-ton-v-pagkosmio-polemo/
[27] “What Is Cryptography?”, https://www.fortinet.com/resources/cyberglossary/what-is-cryptography
[28] “ Συμμετρική Κρυπτογραφία”, http://utopia.duth.gr/~vkatos/documents/the_book/ch5.pdf
[29]Mohammad Ubaidullah Bokhari Dept. of Computer Science,Aligarh Muslim University Aligarh, India, A Review on Symmetric Key Encryption Techniques in Cryptographynternational Journal of Computer Applications (0975 – 8887) Volume 147 – No.10, August 2016,https://www.researchgate.net/publication/333118027_A_Review_on_Symmetric_Key_Encryption_Techniques_in_Cryptography
[30] Peter Smirnoff & Dawn M. Turner, “Symmetric Key Encryption - why, where and how it’s used in banking”, https://www.cryptomathic.com/news-events/blog/symmetric-key-encryption-why-where-and-how-its-used-in-banking
[31] Λυκούδης Κων/νος Συμμετρικοί αλγόριθμοι Κρυπτογράφησης δεδομένων -Η περίπτωση του αλγόριθμου Aes
[32] Blockchain Technology, “An Exploration of Symmetric Key Cryptography: History, Working, and Applications”,https://www.zeeve.io/blog/an-exploration-of-symmetric-key-cryptography-history-working-and-applications/,Published on: March 20, 2023
[33] IBM Domumentation “Symmetric Cryptography”, https://www.ibm.com/docs/en/ztpf/2023?topic=concepts-symmetric-cryptography
[34] Anne Canteaut,“Stream cipher”, https://www.rocq.inria.fr/secret/Anne.Canteaut/encyclopedia.pdf
[35] “Block Ciphers”, https://www.cs.ucdavis.edu/~rogaway/classes/227/fall03/book/bc.pdf
[36] Fauzan Mirza, “ Block Ciphers and Cryptanalysis”, National University of Sciences and Technology,Islamabad, Pakistan,July 1999, https://www.zeeve.io/blog/an-exploration-of-symmetric-key-cryptography-history-working-and-applications/
[37]Chris Christensen, “ Permutation Ciphers”, Spring 2015, https://www.nku.edu/~christensen/1402%20permutation%20ciphers.pdf
[38] “Difference between Confusion and Diffusion”, https://www.geeksforgeeks.org/difference-between-confusion-and-diffusion/

[39] C. E. S HANNON, “ Communication Theory of Secrecy Systems”, Bel System Technical Journal vol 28-4, page 656-715,Oct. 1949, https://pages.cs.wisc.edu/~rist/642-spring-2014/shannon-secrecy.pdf
[40]Kamsiah Mohamed, Fakariah Hani Mohd Ali,Suriyani Ariffin, M. N. M Pauzi Pauzi, “Study of S-box Properties in Block Cipher”, INTERNATIONAL CONFERENCE ON COMPUTER, COMMUNICATION AND CONTROL TECHNOLOGY, IEEE, 2014
[41] “ Hill Ciphers”, https://math.asu.edu/sites/default/files/hill.pdf
[42] Raj Jain “Block Ciphers and DES”, Washington University in Saint Louis, Saint Louis, MO 63130 https://www.cse.wustl.edu/~jain/cse571-14/ftp/l_03bc.pdf
[43] “What is the Feistel cipher structure?,https://www.educative.io/answers/what-is-the-feistel-cipher-structure
[44] ΚΑΝΤΑΣ ΠΑΝΑΓΙΩΤΗΣ, Ε Ρ ΓΑ Σ Ι Α Ε Ξ Α Μ Η Ν Ο Υ ΓΙΑ ΤΟ ΜΑΘΗΜΑ ΔΙΚΤΥΑ ΔΗΜΟΣΙΑΣ ΧΡΗΣΗΣ ΚΑΙ ΔΙΑΣΥΝΔΕΣΗ ΔΙΚΤΥΩΝ, “ΚΡΥΠΤΑΛΓΟΡΙΘΜΟΣ DES”,Πάτρα 2018, https://telematics.upatras.gr/telematics/system/files/bouras_site/ergasies_foithtwn/Κρυπταλγόριθμος DES - Κάντας Παναγιώτης.pdf
[45] Martin McBride, “Block padding methods”, **https://www.tenminutetutor.com/data-formats/cryptography/block-padding-methods/**
[46] Swati Tawde,“Block Cipher modes of Operation”, https://www.educba.com/block-cipher-modes-of-operation/
[47] “ Data encryption standard (DES) | Set 1” https://www.geeksforgeeks.org/data-encryption-standard-des-set-1/
[48] “What is S-Box Substitution?”, https://www.tutorialspoint.com/what-is-s-box-substitution
[49] “THE DES S-BOXES, P-BOX, AND INITIAL PERMUTATION (IP)” https://www.oreilly.com/library/view/computer-security-and/9780471947837/sec9.3.html
[50] “DATA ENCRYPTION ALGORITHM”, https://www.umsl.edu/~siegelj/information_theory/projects/des.netau.net/Dataencryptionalgorithm.html
[51] “Descripttion of Des”, https://www.nku.edu/~christensen/DESschneier.pdf
[52] “ What is 3DES encryption and how does DES work?”, https://www.comparitech.com/blog/information-security/3des-encryption/
[53] Michael Cobb, “What is Triple DES and why is it being disallowed?”, https://www.techtarget.com/searchsecurity/tip/Expert-advice-Encryption-101-Triple-DES-explained
[54] Federal Information Processing Standards Publication, “ Advanced Encryption Standard (AES)”,May 2003 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197-upd1.pdf
[55] “Simplified International Data Encryption Algorithm (IDEA)”, https://www.geeksforgeeks.org/simplified-international-data-encryption-algorithm-idea/

[56] "RC2",http://www.crypto-it.net/eng/symmetric/rc2.html

[57] "RC2",https://www.thalesdocs.com/gphsm/ptk/5.7/docs/Content/PTK-J/Ciphers/RC2.htm

[58] " What is RC4 Encryption?", https://www.geeksforgeeks.org/what-is-rc4-encryption/

[59] Karthikeyan Nagaraj, "An Introduction to RC6 Encryption: How It Works and Why It Matters | 2023" https://cyberw1ng.medium.com/an-introduction-to-rc6-encryption-how-it-works-and-why-it-matters-2023-c70bbef868bc, March 2023

[60] **Reto Galli, "MARS encryption algorithm", ECE 575 Project - Winter 2000**
http://reto.orgfree.com/us/projectlinks/MARSReport.html

[61] Carolynn Burwick, Don Coppersmith, Edward D'Avignon, Rosario Gennaro, Shai Halevi, Charanjit Jutla, Stephen M. Matyas, Jr. Luke O'Connor, Mohammad PeyravianDavid Safford, Nevenko Zunic "MARS - a candidate cipher for AES", Revised, September, 22 1999, https://shaih.github.io/pubs/mars/mars.pdf

[62] Karthikeyan Nagaraj, "A Comprehensive Guide to Serpent Encryption: A Powerful Cipher for Securing Your Data | 2023",March 2023, https://cyberw1ng.medium.com/a-comprehensive-guide-to-serpent-encryption-a-powerful-cipher-for-securing-your-data-2023-61e8f5957880

[63] Karthikeyan Nagaraj, "TwoFish Encryption: A Comprehensive Guide | 2023", March 2023 https://cyberw1ng.medium.com/twofish-encryption-a-comprehensive-guide-2023-b3ad0f844870

[64] " Blowfish Algorithm with Examples", https://www.geeksforgeeks.org/blowfish-algorithm-with-examples/

[65] "Rahul Awati, "Definition Blowfish", https://www.techtarget.com/searchsecurity/definition/Blowfish

[66] " Blowfish Algorithm in Cryptography",https://www.javatpoint.com/blowfish-algorithm-in-cryptography

[67] " CAST Algorithm in Cryptography",https://www.geeksforgeeks.org/cast-algorithm-in-cryptography/

[68] " What is Asymmetric Encryption?",https://www.geeksforgeeks.org/what-is-asymmetric-encryption/

[69] "What Is Asymmetric Encryption & How Does It Work?", https://sectigostore.com/blog/what-is-asymmetric-encryption-how-does-it-work/

[70] " Κρυπτογραφία,Εργαστηριακό μάθημα 6, Αλγόριθμοι δημόσιου κλειδιού RSA", https://cgi.di.uoa.gr/~klimn/cryptography/Lab/Lab-6.pdf

[71] Ed Harmoush, "RSA Example",December 2021,

https://www.practicalnetworking.net/series/cryptography/rsa-example/

[72] "Ασύμμετρη κρυπτογραφία", http://utopia.duth.gr/~vkatos/documents/the_book/ch6.pdf

[73] Don Johnson and Alfred Menezes, and Scott Vanstone, Certicom Research, Canada,Dept. of Combinatorics & Optimization, University of Waterloo,Canada “The Elliptic Curve Digital

Signature Algorithm (ECDSA),”https://www.yumpu.com/en/document/read/17549274/the-elliptic-curve-digital-signature-algorithm-ecdsa

[74] “Digital Signature Standard (DSS)”,https://www.geeksforgeeks.org/digital-signature-standard-dss/

[75] https://www.simplilearn.com/tutorials/cryptography-tutorial/digital-signature-algorithm

[76] Παναγιώτης Γροντάς - Άρης Παγουρτζής, “Κρυπτοσυστήματα Διακριτού Λογαρίθμου”,

https://courses.corelab.ntua.gr/pluginfile.php/2948/course/section/453/crypto2018_19_EG.pdf

[76] FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION, “Digital Signature Standard (DSS)”, https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.186-4.pdf

[77] “DSA”, https://android.googlesource.com/platform/external/wycheproof/+/HEAD/doc/dsa.md

[78] Αναστάσιος Σερετίδης, “ ECDSA,Η Ψηφιακή Υπογραφή της NSA”, Μεταπτυχιακή Διατριβή, https://dias.library.tuc.gr/view/68235