ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ & ΠΛΗΡΟΦΟΡΙΚΗΣ

ΕΡΓΑΣΙΑ ΕΞΑΜΗΝΟΥ

ΓΙΑ ΤΟ ΜΑΘΗΜΑ

ΔΙΚΤΥΑ ΔΗΜΟΣΙΑΣ ΧΡΗΣΗΣ ΚΑΙ ΔΙΑΣΥΝΔΕΣΗΣ

ΑΣΦΑΛΕΙΑ ΣΤΟ ΔΙΑΔΙΚΤΥΟ

ΚΑΤΣΑΜΠΙΡΗΣ ΣΑΛΓΑΔΟ ΣΠΥΡΙΔΩΝ ΑΝΙΣΕΤΟ

Α.Μ 6078

ΔΙΔΑΣΚΩΝ: ΧΡΗΣΤΟΣ ΜΠΟΥΡΑΣ

ΠΑΤΡΑ 2018

ΠΕΡΙΕΧΟΜΕΝΑ

ΑΚΡΩΝΥΜΙΑ

ARPANET: Advanced Research Projects Agency Network

ISO: International Organization for Standardization

ENISA: European Union Agency for Network and Information Security

HDD:Hard Disc Drive

NIST:National Institute of Standards and Technology

DES: Data Encryption Standard

AES:Advanced Encryption Standard

RSA: Ron Rivest, Adi Shamir, Len Adjeman

ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ

Ένα δίκτυο είναι ένα σύστημα συνδεδεμένων συσκευών (ηλεκτρονικοί υπολογιστές, εκτυπωτές, wearables), εφοδιασμένων με σχετικό λογισμικό για τον έλεγχο των κινήσεων των δεδομένων. Το σημερινό INTERNET αποτελεί μετεξέλιξη του ARPANET, το οποίο είχε αναπτυχθεί στις ΗΠΑ από μια ομάδα πανεπιστημίων και ιδιωτικών ερευνητικών ομάδων, χρηματοδοτούμενο από το υπουργείο άμυνας και χρησιμοποιήθηκε για καθαρά ακαδημαϊκούς και ερευνητικούς σκοπούς. Έκτοτε, η ραγδαία εξέλιξη της επιστήμης και της τεχνολογίας έχει οδηγήσει στην εξάπλωση των υπολογιστών και την εδραίωση του INTERNET στην ζωή μας.

Το διαδίκτυο, όπως προαναφέρθηκε αποτελεί αναπόσπαστο κομμάτι της καθημερινότητας όλου του κόσμου, καθώς το συναντάμε σε όλες τις εκφάνσεις την ανθρώπινης δραστηριότητας. Έχει συντελέσει καταλυτικά στην αναδιαμόρφωση πολλών κατά χρόνια απαράλλακτων συνηθειών όπως για παράδειγμα είναι η επικοινωνία, η ενημέρωση η εκμάθηση-εκπαίδευση κ.ά. άλλοτε έχοντας θετική και άλλοτε αρνητική επίδραση.

Στα θετικά του διαδικτύου μπορούμε να καταλογίσουμε μια πληθώρα στοιχείων. Το ηλεκτρονικό ταχυδρομείο και τα social media έχουν δώσει μία νέα οπτική στην επικοινωνία μέσω της ταχύτατης αποστολής μηνυμάτων, αλλά παρουσιάζοντας επίσης και νέες ελκυστικότερες μορφές που προσομοιάζουν την κατά πρόσωπο συνομιλία όπως για παράδειγμα η βιντεοσυνομιλία. Επιπρόσθετο θετικό στοιχείο που αξίζει να σημειωθεί, είναι η ευκαιρία που έχουν οι χρήστες να έχουν εύκολα διαθέσιμο έναν ατελείωτο όγκο πληροφοριών. Οι χρήστες μπορούν να διευρύνουν τους γνωστικούς τους ορίζοντες εν γένη, αλλά και να εμβαθύνουν σε συγκεκριμένα γνωστικά πεδία που ενδεχομένως να απασχολούνται έχοντας πρόσβαση σε πανεπιστημιακό υλικό από όλο τον κόσμο. Τέλος το διαδίκτυο έχει επαναπροσδιορίσει και διευκολύνει συνήθειες όπως είναι οι αγορές αγαθών από το σπίτι, αλλά και οι οικονομικές συναλλαγές εισάγοντας έννοιες όπως «κρυπτονόμισμα» και «ηλεκτρονικό πορτοφόλι».

Βέβαια, όπως τα νομίσματα έχουν δύο όψεις, έτσι και στην περίπτωση του διαδικτύου, στα προαναφερθέντα θετικά στοιχεία ελλοχεύουν κίνδυνοι σημαντικοί για την φυσική ασφάλεια των χρηστών αλλά και για την ασφάλεια των προσωπικών τους δεδομένων. Δεν είναι λίγα τα περιστατικά κατά τα οποία επιτήδειοι εκμεταλλευόμενοι την έλλειψη φυσικής παρουσίας και χρησιμοποιώντας ψευδή στοιχεία παραπλάνησαν κόσμο είτε μέσω αγορών είτε μέσω των social media. Πολλές φορές, η υπερπληροφόρηση οδηγεί σε παραπληροφόρηση, καθώς στο ίντερνετ η ανάρτηση υλικού στις ιστοσελίδες και η διακίνηση υλικού μεταξύ τους γίνεται πολλές φορές χωρίς έλεγχο για την αξιοπιστία τους, Τέλος, το κακόβουλο λογισμικό και η υποκλοπή των προσωπικών δεδομένων (π.χ. χρήματα, προσωπικές πληροφορίες, κτλ.) απειλούν καθημερινά χιλιάδες χρήστες, ζημιώνοντας τους οικονομικά, αλλά και ψυχικά.

Η ασφάλεια των δικτύων ηλεκτρονικών υπολογιστών έχει αναδειχθεί σε ένα σημαντικό και άκρως ενδιαφέρον θέμα στην τεχνολογία και επιστήμη των ηλεκτρονικών υπολογιστών. Η καθολικότητα του παγκόσμιου ιστού και η καθημερινή χρήση του για οποιαδήποτε δραστηριότητα έχει τονίσει την σημασία της ασφαλούς περιήγησης σε αυτόν. Η ασφάλεια των δικτύων και των διακινούμενων πληροφοριών αποτελεί όμως αρκετά πολύπλοκο θέμα, καθώς αποδίδεται διαφορετικά από διαφορετικές κατηγορίες χρηστών, καθώς μπορεί να αποτελεί απλά τη δυνατότητα ανώνυμης περιήγησης, της ασφαλούς εκτέλεσης χρηματοοικονομικών συναλλαγών ή για τους διαχειριστές των δικτύων της εύρυθμης λειτουργίας του δικτύου τους. Για αυτόν τον λόγο, είναι σημαντική η κατανόηση των απειλών, καθώς επίσης και των επιδημιολογικών μοντέλων, έτσι ώστε να μπορεί να γίνει σωστή πρόβλεψη της διάδοση των απειλών.

ΚΕΦΑΛΑΙΟ 2: ΠΑΡΑΓΟΝΤΕΣ ΚΑΤΑ ΤΗΣ ΑΣΦΑΛΕΙΑΣ

Όπως αναφέρθηκε προηγουμένως, το διαδίκτυο αποτελεί αναπόσπαστο κομμάτι της σημερινής κοινωνίας. Ωστόσο ποιοι είναι οι παράγοντες που το καθιστούν ευάλωτο σε επιθέσεις; Με την πρόσβαση όλων και περισσότερων ανθρώπων στο διαδίκτυο, αναπόφευκτο είναι να έχουν πρόσβαση και άνθρωποι που επιθυμούν να προκαλέσουν κακό σε άλλους ανθρώπους ή οργανισμούς και επιχειρήσεις, μέσα από κάποια «τρωτά σημεία» (vulnerability). *Ορίζουμε ως τρωτό σημείο, ένα αδύναμο σημείο μέσω του οποίου κάποιος χωρίς εξουσιοδότηση και δικαίωμα εισβάλλει σε ένα υπολογιστή/υπολογιστικό σύστημα.* Υπάρχουν διάφοροι ορισμοί από διάφορους οργανισμούς όπως ISO, ENISA, εντούτοις θεωρώ ότι ο παραπάνω ορισμός στα πλαίσια της εργασίας είναι επαρκής. Τα τρωτά σημεία, όπως αντιλαμβανόμαστε από τον ορισμό οφείλεται στην ελλιπή και λανθασμένη υλοποίηση και σχεδιασμό των συστημάτων.

2.1.1 Αδυναμία στο λογισμικό συστήματος

Γενικά στα δίκτυα υπολογιστών, για την εύρυθμη λειτουργία και επικοινωνία μεταξύ των συσκευών υπάρχουν τα λεγόμενα πρωτόκολλα που ορίζουν τους κανόνες με τους οποίους θα γίνει αυτή η επικοινωνία. Πολλές φορές επειδή θέλουμε να φτιάξουμε κάτι για να ικανοποιήσει μια ανάγκη μας, παραβλέπουμε κάποιες περιπτώσεις οι οποίες όμως μπορούν να δώσουν την δυνατότητα σε μη εξουσιοδοτημένο χρήστη να εισβάλλει στο σύστημα. Ακόμη, και στην περίπτωση που ο σχεδιασμός του πρωτοκόλλου είναι σωστός υπάρχουν περιπτώσεις που γίνονται λάθη στην υλοποίηση τους. Οι αναβαθμίσεις του λογισμικού οδηγούν κάποιες φορές σε τρωτά σημεία, διότι καινούρια κομμάτια κώδικα μπορεί να μην αλληλεπιδρούν σωστά με τα παλαιότερα. Ένα διαδεμένο εργαλείο που χρησιμοποιείται για την εύρεση των τρωτών σημείων είναι το Metasploit.

2.1.2 Αφαιρούμενα μέσα

Ένας υπολογιστής ή ένα υπολογιστικό δίκτυο δεν χρειάζεται να είναι συνδεδεμένο στο διαδίκτυο για να προσβληθεί. Ένα χαρακτηριστικό παράδειγμα για αυτής της περίπτωσης αποτελεί το κακόβουλο λογισμικό Stuxnet. Θεωρείται πως αυτό το λογισμικό δημιουργήθηκε από της αμερικάνικες και ισραηλινές μυστικές υπηρεσίες για να πάρουν πληροφορίες και να δημιουργήσουν προβλήματα στο πυρηνικό πρόγραμμα του Ιράν. Σύμφωνα με το ντοκιμαντέρ Zero days, οι υπολογιστές των Ιρανών δεν ήταν συνδεδεμένοι στο διαδίκτυο. Εντούτοις , το κακόβουλο λογισμικό κατάφερε να εισέλθει μέσω των τεχνικών οι οποίοι εν αγνοία τους τοποθέτησαν ένα usb flash σε έναν από τον υπολογιστή τους. Ύστερα αυτός ο ιός διαδόθηκε σε όλο το διαδίκτυο, επηρεάζοντας αρκετές χώρες. Μολύνσεις από αφαιρούμενα μέσα (flash drives, cd, dvd, external hdd), γίνονται σε καθημερινή βάση, συνήθως γιατί κάποιος χρήστης τοποθετεί ένα τέτοιο μέσο σε μία συσκευή η οποία μολύνεται με αποτέλεσμα να μολύνεται και το υπόλοιπο δίκτυο.

2.1.3 Περιήγησης του χρήστη στο διαδίκτυο

Έκτος από τα αφαιρούμενα μέσα, πολλές μολύνσεις γίνονται μέσω της περίηγησης στο διαδίκτυο. Πολλές φορές οι χρήστες επισκέπτονται διάφορα site από τα οποία να κατεβάζουν μολυσμένα αρχεία, ή ακόμα να κλικάρουν διάφορες διαφημίσεις οι οποίες να κατεβάζουν μολυσμένα αρχεία. Στα πλαίσια της περιήγησης, ένας άλλος τρόπος μόλυνσης αποτελεί και η χρήση του email αλλά και τα social media. Συχνά στο email στέλνονται ηλεκτρονικά μηνύματα από αγνώστους που έχουν συνημμένα μολυσμένα αρχεία. Το ίδιο συμβαίνει και σε social media όπως το Facebook, messenger που επιτρέπεται η αποστολή και λήψη αρχείων.

2.1.4 Μη χρήση μέτρων προστασίας

Η μεγάλη διάδοση του κακόβουλου λογισμικού στο διαδίκτυο, έχει οδηγήσει στην δημιουργία μιας μεγάλης αγοράς που προσφέρουν λύσεις προστασίας όπως αντιικά προγράμματα τείχη προστασίας κλπ. Παρά το γεγονός ότι υπάρχει σήμερα μια κατανόηση των κινδύνων πολλοί καθημερινοί χρήστες δεν λαμβάνουν τα μέτρα τους. Αυτό φαίνεται και στο γεγονός ότι το πιο διαδεδομένο λειτουργικό σύστημα

στον μέσο χρήστη τα Windows της Microsoft αποτελεί το λειτουργικό σύστημα με τις περισσότερους ιούς όπως φαίνεται και από το site της statista.

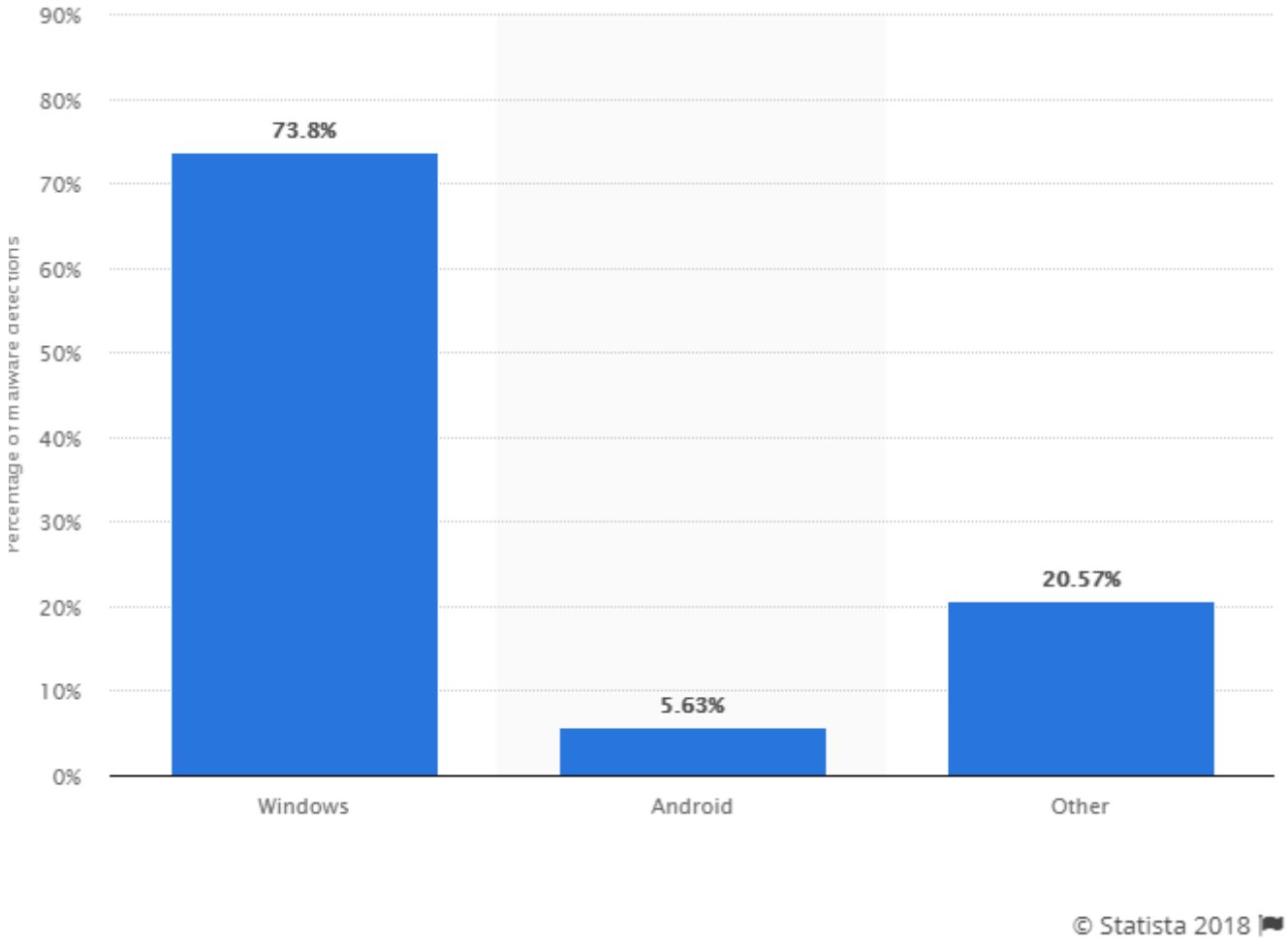


Figure 1 Κατανομή του κακόβουλου λογισμικού ανά λειτουργικό σύστημα src: https://www.statista.com/statistics/680943/malware-os-distribution/

Ένας από τους λόγους που γίνεται αυτό είναι ότι οι μέσοι χρήστες δεν εγκαθιστούν αντιικά προγράμματα και δεν χρησιμοποιούν τείχη προστασίας που βοηθούν στην αποτροπή μολύνσεων και εξάπλωση τους.

ΚΕΦΑΛΑΙΟ 3: ΚΑΚΟΒΟΥΛΟ ΛΟΓΙΣΜΙΚΟ

3.1 Τι είναι το κακόβουλο λογισμικό;

3.1.1 Ορισμός

Κακόβουλο λογισμικό είναι προγράμματα τα οποία έχουν δημιουργηθεί με κακοπροαίρετο σκοπό, είτε για την πρόκληση βλάβης ενός υπολογιστικού συστήματος είτε για την παραβίαση δεδομένων.

Σύμφωνα με αυτό τον ορισμό αντιλαμβανόμαστε ότι υπάρχει ένας διαχωρισμός ανάλογα με την πρόθεση του επιτιθέμενου. Πράγματι, οι προθέσεις των δημιουργών του κακόβουλου λογισμικού οδηγούν στην ύπαρξη διαφορετικών ειδών τέτοιου λογισμικού. Αυτό συμβαίνει διότι άλλες προδιαγραφές και υλοποίηση έχει ο κώδικας που έχει ως στόχο την καταστροφή δεδομένων, διαφορετική αν έχει ως στόχο καταστροφή υλικού και διαφορετική για την απόκτηση απόρρητων πληροφοριών όπως πχ στρατιωτικά μυστικά ή κωδικών πιστωτικών καρτών. Είναι αυτή η ποικιλομορφία στα κίνητρα που έχει οδηγήσει στην ύπαρξη πολλών διαφορετικών ειδών κακόβουλου λογισμικού.

3.2 Είδη κακόβουλου λογισμικού

3.2.1 Ιός (Virus)

Ο Ιός είναι το κακόβουλο λογισμικό, το οποίο όταν εκτελείται, προσπαθεί να αντιγράψει τον εαυτό του σε κάποιο άλλο εκτελέσιμο ή κώδικα, όταν το πετυχαίνει , ο κώδικας λέμε ότι έχει μολυνθεί. Όταν το μολυσμένο πρόγραμμα εκτελείται τότε και ο ιός εκτελείται.[[1]]

Βλέποντας τον ορισμό του ιού στους υπολογιστές, αντιλαμβανόμαστε αμέσως τον παραλληλισμό με τον βιολογικό ιό. Στην βιολογία ο ιός δεν μπορεί να επιτελέσει

τις λειτουργίες της ζωής μόνος του, για αυτό έχει ανάγκη τα κύτταρα ξενιστή. Κατά τον ίδιο τρόπο, ο ιός στους υπολογιστές προσκολλάται σε άλλα προγράμματα τα οποία δεν έχουν κάποιο κακοπροαίρετο σκοπό και τροποποιούν τον κώδικα. Για αυτό τον λόγο όταν εκτελείται το πρόγραμμα ξενιστής, εκτελείται ταυτόχρονα και ο ιός.

Για να μελετηθούν οι ιοί από την σκοπιά της πληροφορικής, πρέπει να εξετάσουμε αρχικά τον μηχανισμό διάδοσης τους. Εδώ θα αναφέρω εν συντομία μερικούς διαδεδομένους τρόπους.

Παρασιτικός: ο ιός προσκολλάται σε άλλα προγράμματα, τα οποία δεν είναι κακοπροαίρετα. Έτσι τους αφήνει να έχουν την αρχική τους λειτουργικότητα, και όταν εκτελούνται τα αρχικά προγράμματα, εκτελούνται και οι ιοί. Υπάρχουν διάφορες τεχνικές για την διάδοση, όπως η companion infection τεχνική, όπου ο ιός βρίσκεται μαζί με το εκτελέσιμο και το μόνο που κάνει είναι να έχει το ίδιο όνομα με το εκτελέσιμο. Στην τεχνική της αντιγραφής, ο ιός αντικαθιστά ένα μέρος του κώδικα του αρχικού προγράμματος. Ανοίγει το πρόγραμμα, όπως θα έκανε με οποιοδήποτε αρχείο και αντιγράφει τον εαυτό του. Στην τεχνική εισαγωγής του κώδικα στην αρχή του προγράμματος, ο ιός εισάγεται στην αρχή για να ενεργοποιηθεί, έτσι ώστε να μπορεί μετά να κάνει άλμα στο πρόγραμμα ξενιστή, για να μην καταλάβει ο χρήστης ότι κάτι πηγαίνει στραβά. Τέλος στην τεχνική εισαγωγής του κώδικα στο τέλος του προγράμματος, ο ιός προσθέτει τον κώδικα του στο τέλος του προγράμματος ξενιστή, τροποποιώντας τον όμως, δημιουργώντας ένα άλμα μέσα στον κώδικα του και να δείχνει στο μέρος του ιού και μετά να επαναφέρει πάλι τον έλεγχο στον ξενιστή.[2]

Ιός μόνιμα εγκατεστημένος στην κύρια μνήμη: Εδώ έχουμε εξειδίκευση όσον αφορά τα αρχεία προσβολής. Εγκαθίσταται στην κύρια μνήμης τμήμα ενός μονίμου προγράμματος συστήματος, με αποτέλεσμα ο ιός από το σημείο αυτό προσβάλει όλα τα άλλα προγράμματα [1].

Ιός τομέα εκκίνησης: όπως στην περίπτωση του ιού μόνιμα εγκαταστημένου στην κύρια μνήμη, υπάρχει και εδώ εξειδίκευση όσον αφορά τα αρχεία προσβολής. Μολύνεται ο τομέας εκκίνησης και έτσι εκκινείται η διαδικασία διάδοσης με την έναρξη του συστήματος [1].

3.2.2 Σκουλήκι (Worm)

Σκουλήκι είναι ένα πρόγραμμα που μπορεί να δημιουργεί αντίγραφα του εαυτό του και να στέλνει μέσω δικτυακών συνδέσεων σε άλλους υπολογιστές. Όταν φθάσει σε ένα νέο σύστημα, το σκουλήκι μπορεί να ενεργοποιηθεί και να συνεχίσει την αντιγραφή και διάδοση του. Εκτός από την διάδοση, εκτελεί και κάποιές ανεπιθύμητες λειτουργείες [1].

Σύμφωνα με τον παραπάνω ορισμό, παρατηρούμε ότι υπάρχουν μερικές ομοιότητες με τους ιούς, αλλά και κάποιες διαφορές που τα καθιστούν διαφορετικό είδος κακόβουλου λογισμικού. Τα σκουλήκια εισέρχονται βρίσκοντας κάποιο τρωτό σημείο διαδίδονται μέσω πρωτοκόλλων όπως ftp, http και στην συνέχεια ψάχνουν για νέα συστήματα. Η διαφορά τους με τους ιούς είναι ότι οι ιοί μεταδίδονται προσκολλώντας τον εαυτό τους σε άλλα αρχεία ενώ τα σκουλήκια μεταδίδονται μέσω του δικτύου χωρίς την βοήθεια άλλου λογισμικού. Επιπλέον, τα σκουλήκια σε αρκετές περιπτώσεις μπορεί να μεταφέρει πληροφορίες από σύστημα σε σύστημα.

Σε προηγούμενο κεφάλαιο, υπήρξε αναφορά στο κακόβουλο λογισμικό Stuxnet. Το Stuxnet αποτελεί ένα παράδειγμα πολύ πολύπλοκου σκουληκιού. Δεν έκανε καμία ζημιά σε υπολογιστές που δεν πληρούσαν πολύ αυστηρές προϋποθέσεις, και η επίθεση γινόταν σε βήματα. Αρχικά, έβλεπε αν το λειτουργικό σύστημα ήταν windows, μετα έπρεπε να ήταν Plc και να είχε ένα λογισμικό ειδικό για αυτές τις συσκευές[3].

Figure 2 μήνυμα του σκουληκιού Blaster το οποίο αναφέρεται στον Bill Gates src: https://en.wikipedia.org/wiki/Computer_virus

3.2.3 Δούρειοι ίπποι (Trojan horses)

Στην πληροφορική, ο δούρειος ίππος (trojan horse ή απλά trojan) είναι ένα κακόβουλο πρόγραμμα που ξεγελάει τον χρήστη και τον κάνει να πιστεύει ότι εκτελεί κάποια χρήσιμη λειτουργία ενώ στα κρυφά εγκαθιστά στον υπολογιστή του άλλα κακόβουλα προγράμματα [4].

Βλέποντας τον παραπάνω ορισμό, καταλαβαίνουμε πως οι δούρειοι ίπποι χαρακτηρίζονται από το στοιχείο της παραπλάνησης, μιας και βρίσκεται μέσα σε μια νόμιμη και χρήσιμη εφαρμογή. Οι δούρειοι ίπποι μοιάζουν στους ιούς όσον αφορά την χρήση «ξενιστή », όμως δεν αναπαράγεται όπως κάνουν οι ιοί.

Οι δούρειοι ίπποι μπορούν να προκαλέσουν πολλά προβλήματα. Μπορούν να στέλνουν δεδομένα σε κάποιον άλλον υπολογιστή, δεδομένα όπως κωδικοί κλπ. Ακόμη, μπορούν να απενεργοποιούν τα αντιικά προγράμματα, συνήθως για να επιτρέψει την λειτουργία κάποιων άλλων κακόβουλων προγραμμάτων.

Τέλος, μπορούν να λειτουργήσουν σαν proxy servers, με αποτέλεσμα το μολυσμένο μηχάνημα να είναι διαθέσιμο από τον κακόβουλο χρήστη. Συνήθως αυτό επιτυγχάνεται με την δημιουργία μιας «κερκόπορτας» (backdoor).

3.2.4 Λογικές Βόμβες(Logic Bombs)

Η λογική βόμβα αποτελεί ένα από τα πρώτα είδη κακόβουλου λογισμικού και υπάρχει αρκετά πριν από τους ιούς και τα σκουλήκια. Η λειτουργία τους βασίζεται στην εξής λογική: ο κώδικας της λογικής βόμβας ενσωματώνεται σε κάποιο άλλο λογισμικό και εκτελείται όταν πληρούνται κάποια κριτήρια. Αυτά τα κριτήρια μπορεί να είναι κάποια ημερομηνία , ύπαρξη ή απουσία κάποιων αρχείων ή στοιχείων κλπ. Πολλές φορές οι λογικές βόμβες χρησιμοποιούνται σε συνδυασμό με ιούς ή σκουλήκια. Το γεγονός ότι είναι ενσωματωμένα σε κώδικα άλλου λογισμικού που ενίοτε μπορεί να είναι νόμιμο λογισμικό, καθιστούν μια λογική βόμβα πολύ δύσκολο να εντοπιστεί, αφού το μπορεί να είναι κρυμμένη μέσα σε εκατομμύρια γραμμές κώδικα.

3.2.5 Adware

Το adware είναι ένα λογισμικό το οποίο έχει σαν στόχο την προβολή στην οθόνη του υπολογιστή. Διαφέρουν όμως με τις διαφημίσεις που βλέπουμε στο διαδίκτυο. Δεν αποτελούν απλές διαφημίσεις που υπάρχουν στα άκρα μιας ιστοσελίδας. Συνήθως όταν γίνεται εγκατάσταση κάποιου δωρεάν λογισμικού γίνεται εγκατάσταση αυτών των διαφημίσεων οι οποίες εμφανίζονται στην οθόνη ως pop up ανεξάρτητα αν γίνεται περιήγηση στο διαδίκτυο.

3.2.6 Rootkit

Το rootkit αποτελεί έναν συνδυασμό από κακόβουλο λογισμικό, με σκοπό την υπονόμευση ενός υπολογιστή έχοντας δικαιώματα υπερχρήστη εξου και το πρώτο συνθετικό root. Για να καταφέρουν αυτό , τροποποιείται το λειτουργικό σύστημα, γίνεται απεγκατάσταση ή απενεργοποίηση των λογισμικών προστασίας. Τα rootkits χρησιμοποιούνται συνεχώς από τους κακόβουλους εισβολείς με σκοπό την συνεχόμενη πρόσβασή τους στο σύστημα που έχουν εισβάλλει. Τα rootkits από μόνα τους δεν αποτελούν κίνδυνο δηλαδή δεν προκαλούν βλάβη όπως οι ιοί, ωστόσο όταν κάποιος κακόβουλος το χρησιμοποιεί μπορεί να έχει πλήρη έλεγχο του μηχανήματος.

3.2.7 Rabbit ή fork bomb

Σε αυτήν την κατηγορία ανήκει το κακόβουλο λογισμικό το οποίο έχει την ικανότητα να εξαπλώνεται αστραπιαία. Αυτό συμβαίνει, όταν μία διεργασία αναπαράγεται συνεχώς με αποτέλεσμα να καταναλώνονται όλοι οι πόροι του συστήματος με αποτέλεσμα να καταρρέει.

Εδώ δίνονται δύο απλά παραδείγματα fork bomb τα οποία είναι γραμμένα σε JAVA και σε C.

```
public class ForkBomb
{
  public static void main(String[] args)
  {
    while(true)
    {
      Runtime.getRuntime().exec(new String[]{"javaw", "-cp",
System.getProperty("java.class.path"), "ForkBomb"});
    }
  }
}
```

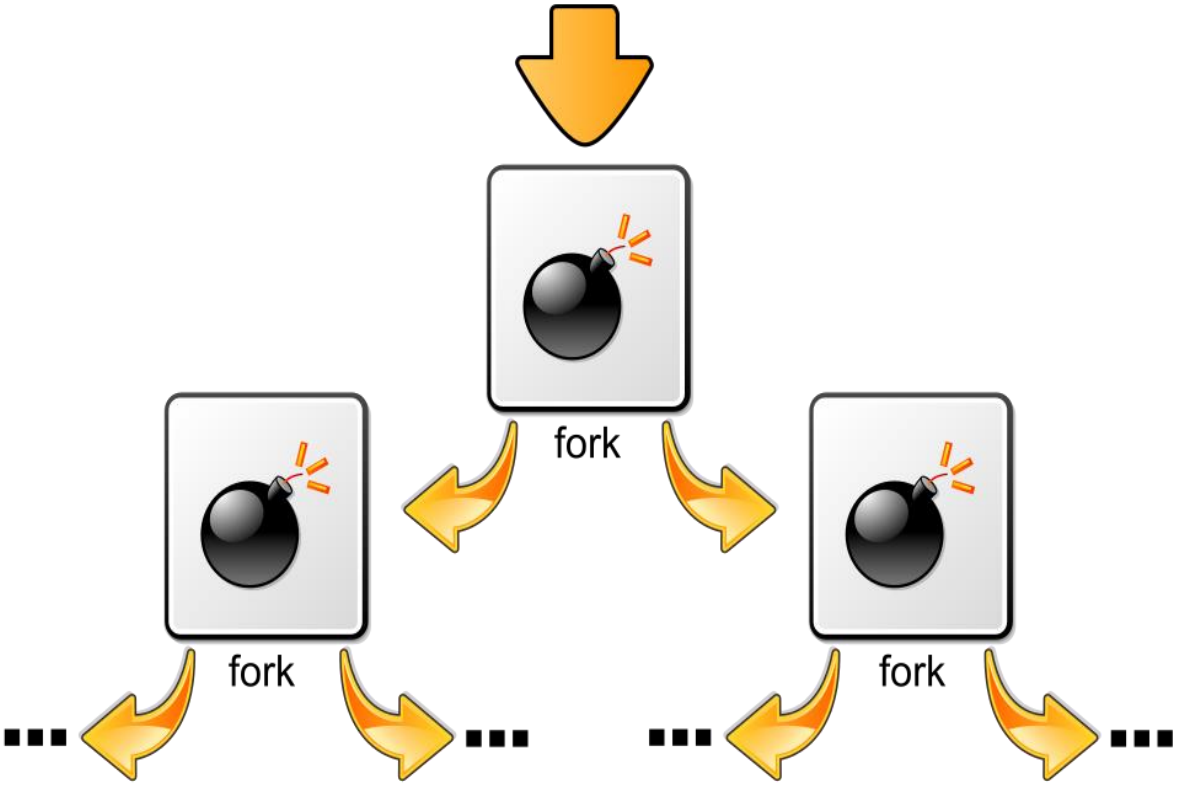
[8]

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>

int main()
{
    while(1) {
      fork(); /* malloc can be used in order to increase the data
usage */
    }
    return 0;
}
```

[8]

Για την πρόληψη τέτοιων περιστάσεων είναι ο περιορισμός του πλήθους διεργασιών που ένας χρήστης μπορεί να έχει στην κατοχή του.



Γραφική απεικόνιση της λογικής των fork bombs src: https://en.wikipedia.org/wiki/Fork_bomb#/media/File:Fork_bomb.svg

`

3.2.8 Ransomware

Το ransomware αποτελεί ένα σχετικά μοντέρνο είδος κακόβουλου λογισμικού. Η γενική ιδέα πίσω από αυτό το είδος είναι ότι «κλειδώνονται» τα προσωπικά δεδομένα του θύματος και τον απειλεί μέχρι να καταβάλλει λύτρα συνήθως σε κάποιο κρυπτονόμισμα όπως το bitcoin, Ethereum, Litecoin. Το «κλείδωμα» αυτό γίνεται με την χρήση κρυπτογραφικών μεθόδων όπου σημαντική γνώση είναι το κλειδί το οποίο αποτελεί ένα δύσκολο υπολογιστικά πρόβλημα όπως θα δούμε και στο 5 κεφάλαιο. Έτσι , το κλειδί δίνεται αφού πληρωθούν τα λύτρα.

ΚΕΦΑΛΑΙΟ 4: ΤΡΟΠΟΙ ΑΝΤΙΜΕΤΩΠΙΣΗΣ ΚΑΚΟΒΟΥΛΟΥ ΛΟΓΙΣΜΙΚΟΥ

Η αντιμετώπιση του κακόβουλου λογισμικού αποτελεί σημαντική στην σύγχρονη εποχή. Η αντιμετώπιση μπορεί να μελετηθεί από διάφορες σκοπίες, από τον τρόπο που γίνεται η περιήγηση του χρήστη στο διαδίκτυο, σε πακέτα λογισμικού , τείχη προστασίας κλπ.

Η πλοήγηση του χρήστη στο διαδίκτυο αποτελεί έναν λόγο που διακυβεύεται η ασφάλεια. Για αυτό τον λόγο πρέπει να γίνονται ορισμένες ενέργειες για την ασφαλέστερη περιήγηση. Για παράδειγμα, είναι σημαντικό ο χρήστης να γνωρίζει ακριβώς τι ιστοσελίδες ψάχνει και να προσέχει τα url των ιστοσελίδων που τελικά μπαίνει. Αυτό είναι ιδιαίτερα σημαντικό καθώς είναι συνηθισμένο κακόβουλοι να δημιουργούν ιστοσελίδες που μοιάζουν με τις πραγματικές και με αυτόν τον τρόπο να υποκλέβουν ευαίσθητες πληροφορίες όπως email, κωδικούς κλπ.

Επιπλέον, θα πρέπει να αποφεύγεται η απάντηση σε παράθυρα pop up και θα καλό είναι να κλείνονται. Πολλές φορές με την απάντηση ενός τέτοιου παράθυρο μπορεί να αρχίσει η λήψη ενός αρχείου όπου μπορεί να είναι κάποιο κακόβουλο λογισμικό ή να είναι ενσωματωμένο σε ένα κατά τα άλλα σημαντικό λογισμικό. Σε αυτό το σημείο αξίζει να σημειωθεί ότι πολλές φορές ο ίδιος ο χρήστης από αφέλεια και απροσεξία αποτελεί η αιτία να εισέλθει και να εκτελεστεί ένα κακόβουλο λογισμικό. Έτσι, πρέπει να υπάρχει ιδιαίτερη προσοχή όσον αφορά την λήψη αρχείων είτε από διάφορες ιστοσελίδες, είτε από email. Βέβαια πέρα από την προσοχή άλλα μέτρα είναι η εγκατάσταση διάφορων λογισμικών πακέτων όπως antivirus.

4.2 Antivirus προγράμματα

4.2.1 Τρόπος λειτουργίας

Τα αντιικά προγράμματα αποτελούν ένα από τους δημοφιλέστερους τρόπους προστασίας εναντίον του κακόβουλου λογισμικού, καθώς είναι πολύ διαδεδομένα στο λειτουργικό σύστημα Windows, αποτελώντας μία μεγάλη αγορά στον τομέα της πληροφορικής.

Από τους πιο παλιούς τρόπους με τους οποίους τα αντιικά προγράμματα αντιμετωπίζουν τους ιούς είναι βασισμένη στην υπογραφή των ιών. Κάθε φορά που φτάνει ένας ιός στα χέρια κάποιου εργαστηρίου, μελετιέται και στην συνέχεια βγάζουν μια υπογραφή από τον κώδικα που χαρακτηρίζει τον ιό. Έτσι κάθε φορά που έρχεται ένα ύποπτο λογισμικό τότε συγκρίνουν την υπογραφή του με τις υπογραφές που έχουν μέσα στις βάσεις δεδομένων για να βρουν ομοιότητες.

Υπάρχουν ωστόσο, πιο σύγχρονες μέθοδοι για την εύρεση κακόβουλου λογισμικού.

- Heuristics μέθοδος, αποτελεί μια μέθοδος η οποία χρησιμοποιείται για την εύρεση ιών που δεν έχουν γνωστοί μέχρι αυτή την στιγμή, εξομοιώνοντας την αρχή του προγράμματος.
- Sandbox μέθοδος, αποτελεί μια μέθοδος όπου κάθε πρόγραμμα εκτελείται πρώτα σε μια εικονική μηχανή και ανάλογα με την συμπεριφορά του προγράμματος το αφήνει να εκτελεστεί στο κανονικό μηχάνημα ή όχι.

Μερικές εταιρείες antivirus. Src: https://supersonicit.com.au/anti-virus/

4.2.2 Πιθανά προβλήματα

Όπως αναφέρθηκε το 4.1 η προσοχή του χρήστη αποτελεί απαραίτητη προϋπόθεση για την εύρυθμη λειτουργία των διάφορων δραστηριοτήτων του στο διαδίκτυο. Το ίδιο συμβαίνει και στην περίπτωση των αντιικών προγραμμάτων. Ένα από τα κυριότερα προβλήματα που μπορούν να υπάρξουν είναι η λήψη και η εγκατάσταση κακόβουλου λογισμικού που έχει παραπλανητικό όνομα και νομίζει ο χρήστης ότι είναι όντως αντιικό πρόγραμμα. Τέτοια παραδείγματα είναι το MacDefender, WinFixer κλπ.

Όπως συμβαίνει στις περισσότερες περιπτώσεις στην ζωή, δεν υπάρχει πανάκεια και τα σφάλματα αποτελούν μια συνιστώσα που πάντα πρέπει να την λαμβάνουμε υπόψη. Τι γίνεται όταν το αντιικό πρόγραμμα κάνει λάθος;

Ορίζουμε ως false alarm την περίπτωση που ένα αντιικό πρόγραμμα αναγνωρίζει ένα μη κακόβουλο λογισμικό ως κακόβουλο[9].

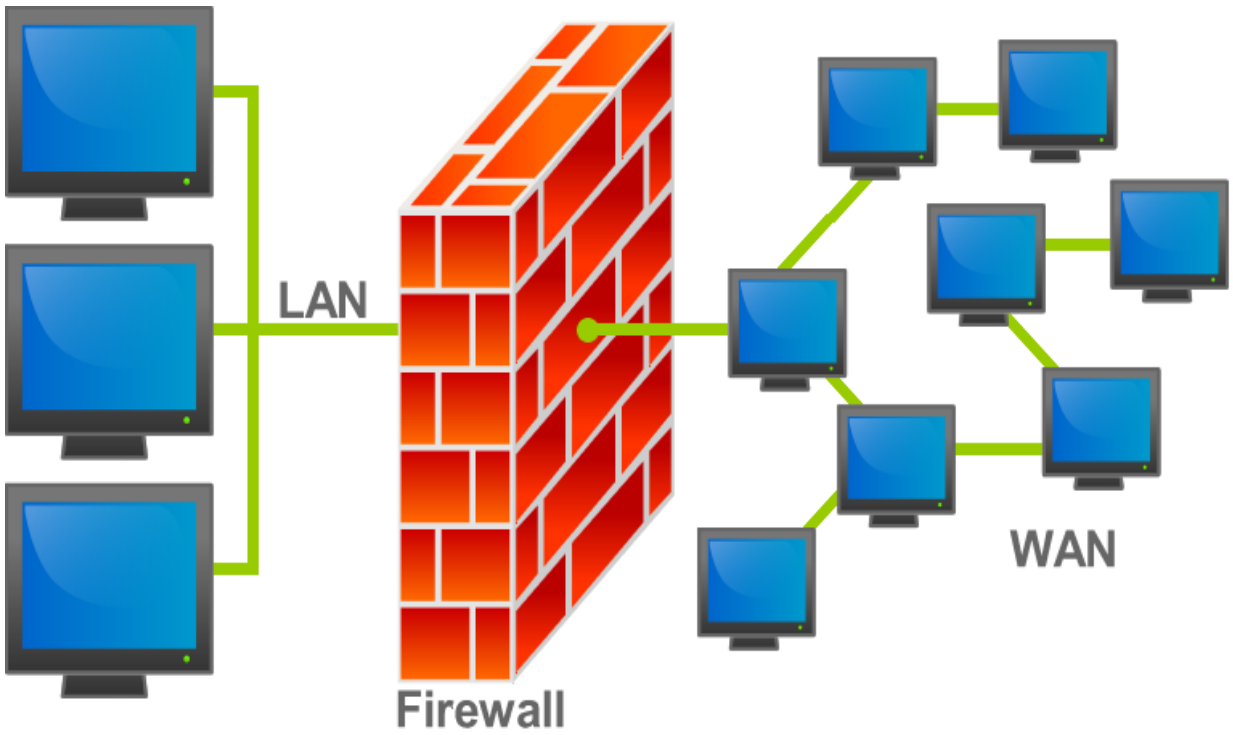
Τα false alarms μπορούν να δημιουργήσουν ποικίλα προβλήματα. Μπορεί απλά να σβήνουν σωστά αρχεία και να προκαλούν εκνευρισμό στον χρήστη, μέχρι να σβηστούν σημαντικά αρχεία για την λειτουργία του λειτουργικού συστήματος. Στην τελευταία περίπτωση η επαναφορά του συστήματος μπορεί να στοιχίζει αρκετά και να χρειάζεται τεχνική υποστήριξη. Ένα τέτοιο παράδειγμα συνέβη το 2007 από την Symantec, όπου λανθασμένα έσβησε σημαντικά αρχεία του λειτουργικού συστήματος

Windows, καθιστώντας πολλούς υπολογιστές ανίκανους να εκκινήσουν.[9]. Ένα πιο πρόσφατο παράδειγμα συνέβη το 2017 όπου το αντιικό της google για τις android συσκευές (Google Play Protect) αναγνώριζε το λογισμικό Bluetooth του κινητού της Motorola Moto G4, ως κακόβουλο λογισμικό, με αποτέλεσμα να μην λειτουργεί το Bluetooth [9]

4.3 Τείχη προστασίας (Firewalls)

Στην επιστήμη των υπολογιστών ο όρος firewall ή τείχος προστασίας χρησιμοποιείται για να δηλώσει κάποια συσκευή ή πρόγραμμα που είναι έτσι ρυθμισμένο ούτως ώστε να επιτρέπει ή να απορρίπτει πακέτα δεδομένων που περνούν από ένα δίκτυο υπολογιστών σε ένα άλλο[10]

Με άλλα λόγια, το τείχος προστασίας αποτελεί το ενδιάμεσο στοιχείο που ελέγχει την κίνηση από και προς το εξωτερικό δίκτυο, με σκοπό την αποτροπή της εισόδου κακόβουλου λογισμικού. Αυτό γίνεται γιατί τα τείχη προστασίας ελέγχουν της εισερχόμενες συνδέσεις αν είναι αξιόπιστες ή όχι. Όσον αφορά τις εξερχόμενες συνδέσεις, γίνεται έλεγχος για τον εντοπισμό περίεργης σύνδεσης, καθώς όπως αναφέρθηκε και στο κεφάλαιο 3 πολλές φορές σκουλήκια και rootkits προσπαθούν να επικοινωνήσουν με εξωτερικά μηχανήματα για αποστολή προσωπικών δεδομένων στον εισβολέα. Επιπλέον, εκτός από την προσπάθεια αποκλεισμού μεταφοράς των δεδομένων, τα τείχη προστασίας παίζουν καταλυτικό ρόλο στην μείωση του ρυθμού διάδοσης του κακόβουλου λογισμικού όπως για παράδειγμα τους ιούς που χαρακτηρίζονται από μολυσματική συμπεριφορά.



Απεικόνιση λειτουργίας Firewall

src:https://el.wikipedia.org/wiki/Firewall#/media/File:Firewall.png

ΚΕΦΑΛΑΙΟ 5: ΚΡΥΠΤΟΓΡΑΦΙΑ

5.1 Εισαγωγικά

Εκτός από τα πολλά προβλήματα που δημιουργούνται από το κακόβουλο λογισμικό δεν αρκεί να προστατευόμαστε μόνο από αυτά. Πολλές φορές υπάρχουν προβλήματα στο ίδιο το κανάλι μετάδοσης όπου κάποιος μπορεί να επεμβει στην επικοινωνία και να υποκλέψει σημαντικά δεδομένα. Για αυτό τον λόγο υπάρχει η *κρυπτογραφία.*

5.1.1 Ορισμός

Η κρυπτογραφία ασχολείται με την επικοινωνία παρουσία αντιπάλων.[2]

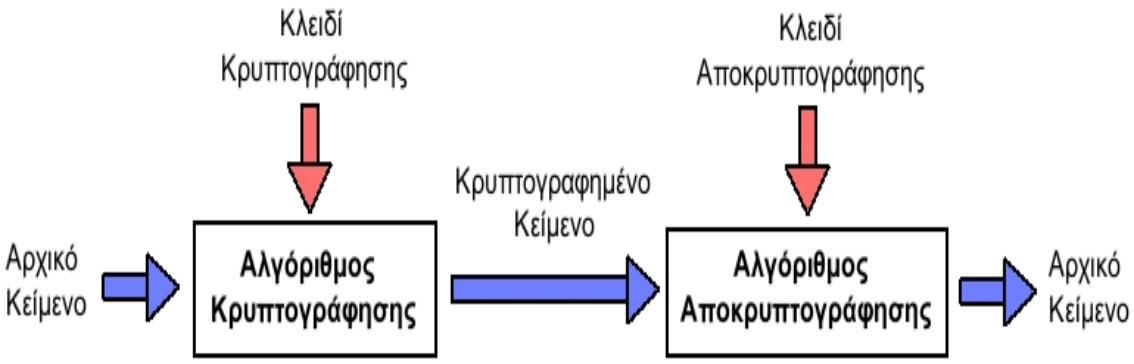
Από τον ορισμό συνειδητοποιούμε πως ο κλάδος της κρυπτογραφίας υπάρχει από την αρχαιότητα, καθώς η δυνατότητα μετάδοσης μηνυμάτων τα οποία θα είναι σε ακατάληπτη μορφή από τους αντιπάλους αποτελεί σημαντικό πλεονέκτημα σε πολεμικές συρράξεις. Μια από τις πιο αρχαίες κρυπτογραφικές μεθόδους θεωρείται πως είναι ο κώδικας του Καίσαρα. Το όνομα το πήρε από τον Ιούλιο Καίσαρα που θεωρείται πως την χρησιμοποιούσε για την αλληλογραφία του. Η λογική του κώδικα αυτού είναι η αντικατάσταση ενός γράμματος με ένα άλλο που βρίσκεται σε σταθερή απόσταση μέσα στο αλφάβητο. Ένας άλλος παρόμοιος κώδικας είναι ο κώδικας Vinegere ο οποίος μοιάζει με τον κώδικα του Καίσαρα. Και οι δύο κώδικες πλέον σπάνε πολύ εύκολα και δεν χρησιμοποιούνται.

Βέβαια, σήμερα η ανάπτυξη της τεχνολογίας, έχει οδηγήσει στην ραγδαία αύξηση της πληροφορίας και των μηνυμάτων που διακινούνται και χρειάζονται να είναι σε ακατάλυπτη μορφή, όπως για παράδειγμα τραπεζικές συναλλαγές.

Γενικά, θέλουμε με ένα κρυπτοσύστημα να πετύχουμε τα εξής:

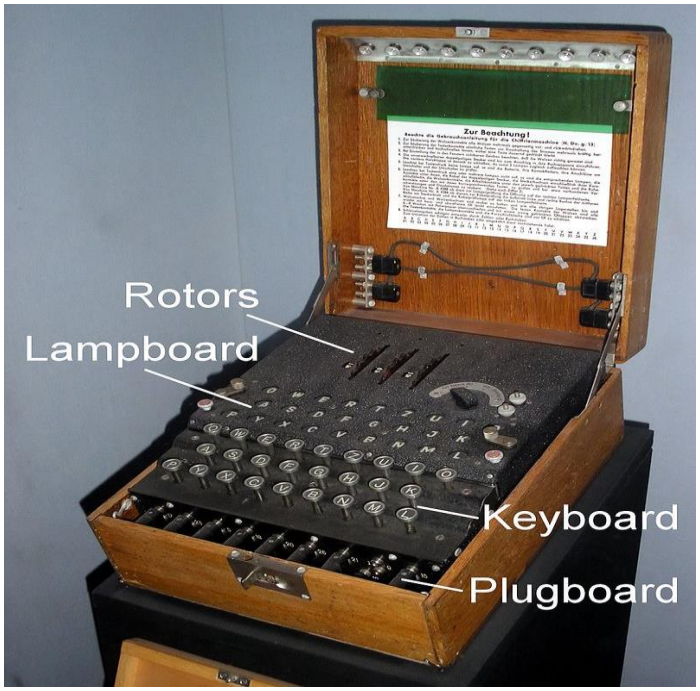
- *Εμπιστευτικότητα*: Η πληροφορία η οποία πρέπει να μεταδοθεί είναι προσβάσιμη μόνο στα εξουσιοδοτημένα μέλη, και με κανέναν τρόπο άλλα άτομα δεν μπορούν να μάθουν την πληροφορία.
- *Ακεραιότητα*: Η πληροφορία μπορεί να αλλοιωθεί μόνο από τα εξουσιοδοτημένα μέλη και δεν μπορεί να αλλοιώνεται χωρίς την ανίχνευση της αλλοίωσης.
- *Μη απάρνηση*: Ο αποστολέας ή ο παραλήπτης της πληροφορίας δεν μπορεί να αρνηθεί την αυθεντικότητα της μετάδοσης ή της δημιουργίας της. Με άλλα λόγια δεν μπορεί να απαρνηθεί ότι έλαβε το μήνυμα ο παραλήπτης και ο αποστολέας ομοίως δεν μπορεί να υποστηρίζει πως δεν το έστειλε ο ίδιος.
- *Πιστοποίηση*: Οι αποστολέας και παραλήπτης μπορούν να εξακριβώνουν τις ταυτότητές τους καθώς και την πηγή και τον προορισμό της πληροφορίας με διαβεβαίωση ότι οι ταυτότητές τους δεν είναι πλαστές. Να γνωρίζει δηλαδή για παράδειγμα ο αποστολέας ότι το μήνυμα που πήρε είναι όντως από τον αποστολέα που πιστεύει ότι το πήρε.[6]

Παρακάτω περιγράφεται γραφικά η λειτουργία ενός κρυπτοσυστήματος



src:https://upload.wikimedia.org/wikipedia/commons/9/98/Encryption_and_decryption_system.png

5.2 Συμμετρικα ασυμμετρα κρυπτοσυστήματα



Η μηχανή Enigma η οποία δημιουργήθηκε από τον γερμανό μηχανικό Arthour Scherbius κατά την διάρκεια του Α παγκοσμίου πολέμου. Χρησιμοποιήθηκε κατά κόρον από την ναζιστική Γερμανία κατά την διάρκειατ ου Β παγκοσμίου πολέμου για την κρυπτογράφηση των στρατιωτικών και διπλωματικών μηνυμάτων. Ο Alan Turing βοήθησε στην αποκρυπτογράφηση αυτής της συσκευής src: https://en.wikipedia.org/wiki/Enigma_machine#/media/File:EnigmaMachineLabeled.jpg

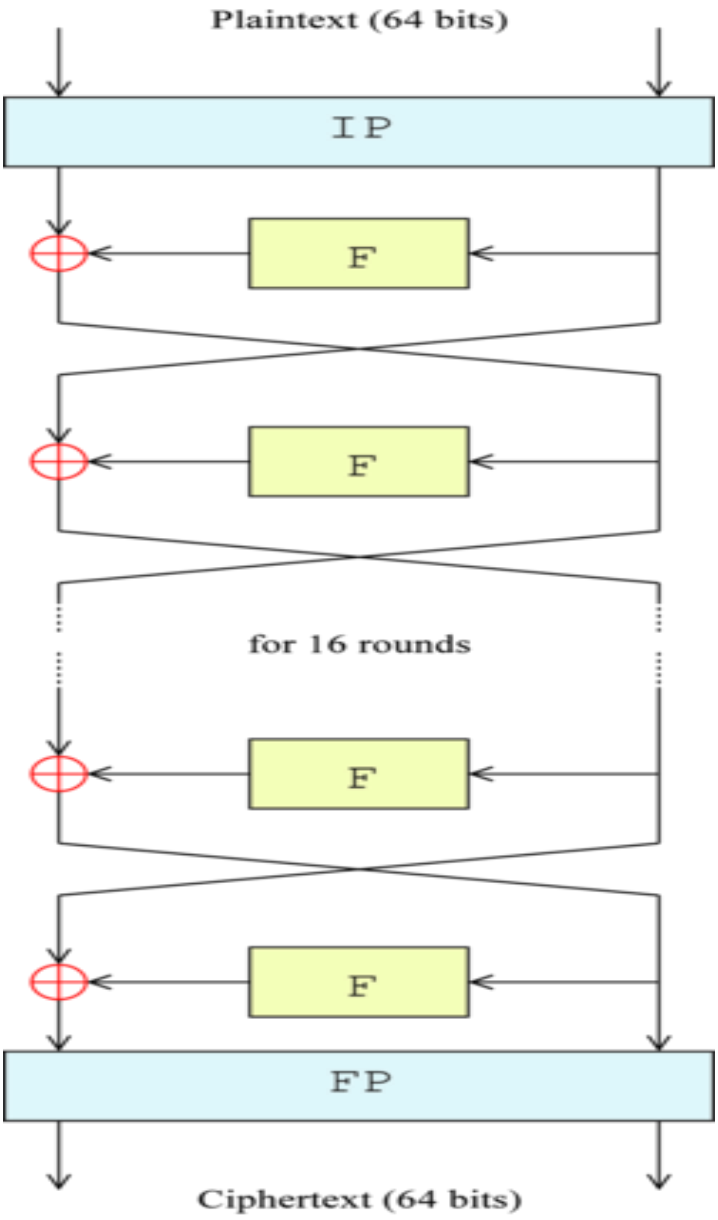
5.2.1 Ορισμός συμμετρικού κρυπτοσυστήματος

Ένα κρυπτοσύστημα ονομάζεται συμμετρικό όταν το κλειδί κρυπτογράφησης είναι το ίδιο με το κλειδί αποκρυπτογράφησης

Τα συμμετρικά κρυπτοσυστήματα υπήρξαν πολύ διαδεδομένα, μέχρι την εμφάνιση των ασύμμετρων κρυπτοσυστημάτων. Τα προβλήματα όμως που έχουν είναι τα εξής: η απαίτηση του κρυπτοσυστήματος για ίδιο κλειδί και στην κρυπτογράφηση και στην αποκρυπτογράφηση, έχει το μειονέκτημα της υποκλοπής του κλειδιού, καθώς ο αποστολέας θα πρέπει να στείλει το κλειδί με κάποιο τρόπο και

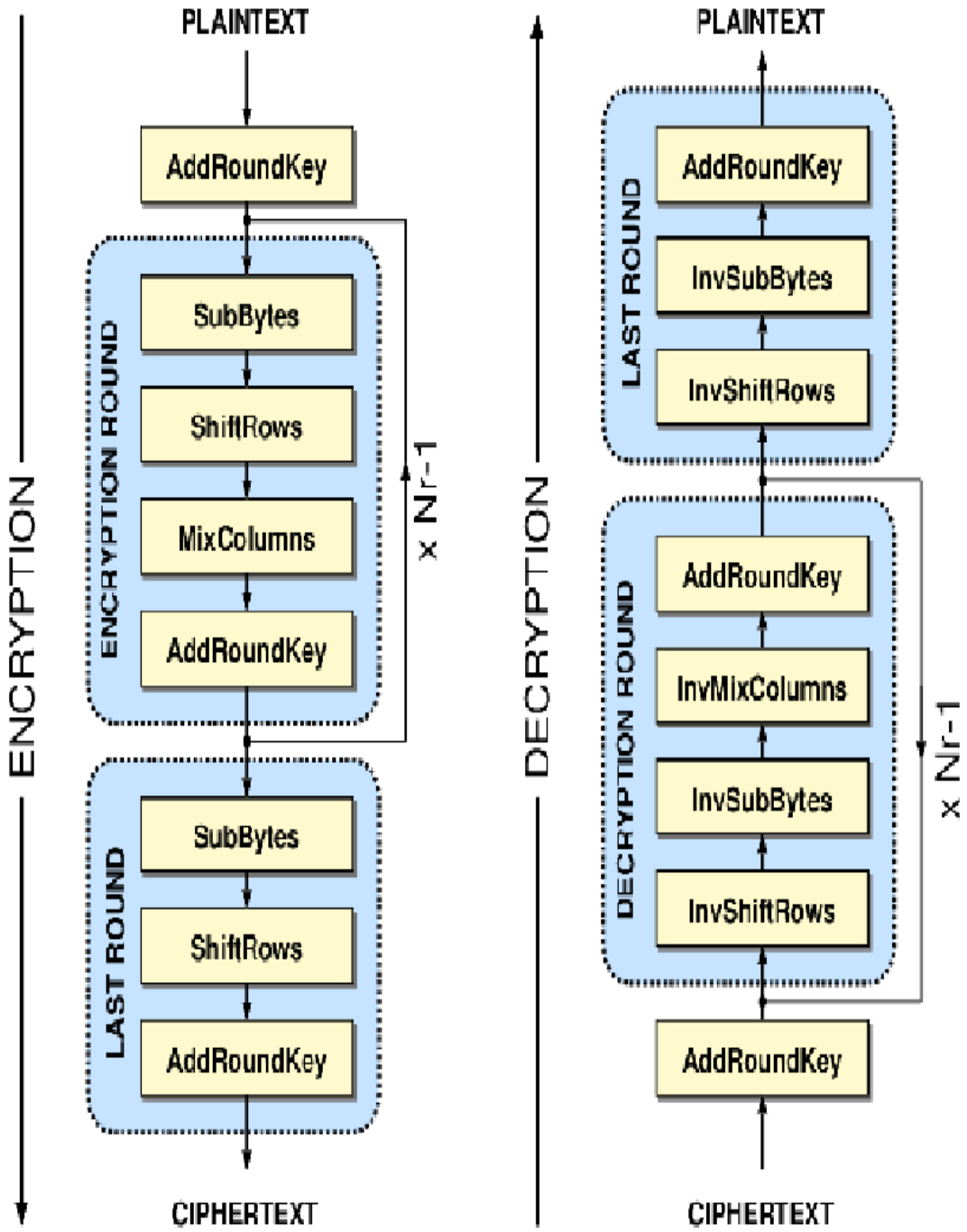
στον παραλήπτη. Επιπλέον, για την επικοινωνία πρέπει να δημιουργούνται πολλά κλειδιά. Για παράδειγμα εφόσον θέλουν να επικοινωνήσουν n άτομα, κάθε άτομο θα πρέπει να αποθηκεύει διαφορετικά κλειδιά για κάθε ένα άτομο που θέλει να επικοινωνήσει. Έτσι συνολικά θα πρέπει να δημιουργηθούν και να αποθηκευτούν n(n-1)/2 κλειδιά. [2]. Όπως βλέπουμε τα κλειδιά αυξάνονται τετραγωνικά συναρτήσει του πλήθους των ατόμων που θέλουν να επικοινωνήσουν. Αυτό στην πράξη κάνει τα συμμετρικά κρυπτοσυστήματα στην σημερινή εποχή απαγορευτικά.

Οι πιο σημαντικοί και γνωστοί συμμετρικοί αλγόριθμοι είναι οι DES, το τριπλό DES και AES. Ο αλγόριθμος DES δημιουργήθηκε το 1977 από την αμερικάνικη κυβέρνηση και επηρεάστηκε στην αρχιτεκτονική του από τα λεγόμενα δίκτυα Feistel. Γενικά σε αυτόν τον αλγόριθμο, κάθε κείμενο ήταν 64bit (αν το κείμενο ήταν μεγάλο έσπαγε σε μικρότερα κομμάτια των 64bit) και το κλειδί ήταν 56 bit (συν 8 parity bits). Το μικρό μήκος του κλειδιού είναι αυτό που έκανε το DES μη ασφαλές, καθώς με τα 56 bit υπάρχουν 2^56 κλειδιά. Υπάρχουν 3 κατηγορίες επιθέσεων στις οποίες είναι ευάλωτο, Α) η διαφορική κρυπτανάλυση, Β) η γραμμική κρυπτανάλυση Γ) η επίθεση Davie. Τελικά, ο DES έσπασε την δεκαετία του 90 με brute force (ωμή βία) από την Electronic Frontier Foundation σε 2 μέρες με ένα μηχάνημα 250.000 δολαρίων [3].



DES src: https://upload.wikimedia.org/wikipedia/commons/6/6a/DES-main-network.png

Ο αλγόριθμος AES αποτελεί εξέλιξη του DES μετά την δημοσιοποιήσει το 1997 από το NIST. Το μήκος κλειδιού στον AES είναι μεταβλητό και μπορεί να είναι ίσο με 128, 192, 256 bits. Ο τεράστιος κλειδοχώρος καθιστά την εξαντλητική αναζήτηση πρακτικώς αδύνατη. Παρακάτω φαίνεται γραφικά εν συντομία η λειτουργία του αλγόριθμου.

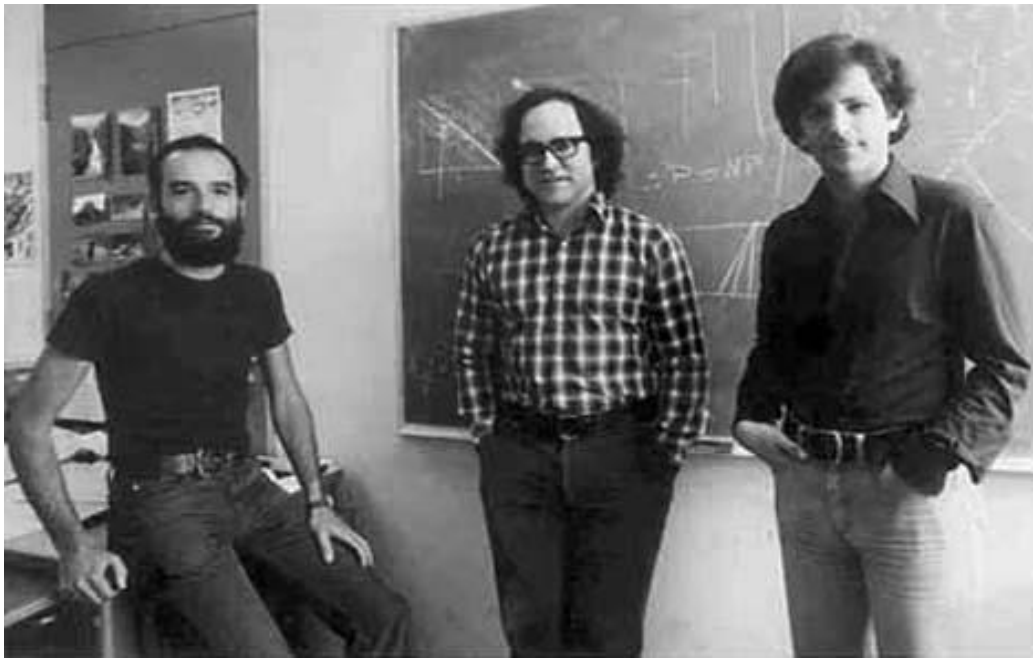


AES src: https://www.researchgate.net/figure/The-basic-AES-128-cryptographic-architecture_fig1_230853805

5.2.1 Ορισμός ασύμμετρου κρυπτοσυστήματος

Ένα κρυπτοσύστημα ονομάζεται ασύμμετρο όταν χρειάζονται δύο κλειδιά ένα για την κρυπτογράφηση και ένα για την αποκρυπτογράφηση.[2]

Η ασύμμετρη κρυπτογραφία δημιουργήθηκε για να καλύψει την αδυναμία μεταφοράς κλειδιών που παρουσίαζαν τα συμμετρικά συστήματα. Χαρακτηριστικό του είναι ότι έχει δυο είδη κλειδιών ένα ιδιωτικό και ένα δημόσιο. Το δημόσιο είναι διαθέσιμο σε όλους ενώ το ιδιωτικό είναι μυστικό. Η βασική σχέση μεταξύ τους είναι : ό,τι κρυπτογραφεί το ένα, μπορεί να το αποκρυπτογραφήσει μόνο το άλλο.[6]. Γενικά, η ασύμμετρη κρυπτογραφία βασίζεται αποκλειστικά σε μαθηματικές υποθέσεις. Δηλαδή , βασίζεται σε υποθέσεις ότι κάποια μαθηματική πράξη είναι δύσκολη, συνεπώς για αυτό τον λόγο η μορφή του κειμένου είναι ακατάληπτη. Μερικά τέτοια μαθηματικά προβλήματα είναι η παραγοντοποίηση ακεραίων αριθμών, το πρόβλημα του διακριτού λογάριθμου κλπ. Στην κατηγορία αυτή γνωστά κρυπτοσυστήματα είναι το κρυπτοσύστημα knapsack, Merkle Hellman, RSA, El Gamal κλπ. Στη συνέχεια παρουσιάζεται ο αλγόριθμος RSA.



Οι δημιουργοί του RSA, Ron Rivest Adi Shamir Leonard Adleman src: http://infosec-regeneration.blogspot.com/2012/12/rsa-encryption-algorithm-or-they-are.html

Το RSA αποτελεί ένα από τα πιο σημαντικά κρυπτογραφικά πρωτόκολλα το οποίο χρησιμοποιείται και σήμερα, κυρίως στο διαδίκτυο. Δημιουργήθηκε από τους Rivest, Shamir, Adleman το 1977. Η ασφάλεια αυτού του συστήματος βασίζεται στο δύσκολο πρόβλημα της παραγοντοποίησης ενός σύνθετου ακεραίου σε γινόμενο πρώτων αριθμών

Για την δημιουργία κλειδιών στο RSA χρησιμοποιείται η εξής διαδικασία.

1. Γίνεται η επιλογή δύο διαφορετικών τυχαίων πρώτων αριθμών p, q
2. Υπολογίζουμε το n=p*q που είναι μια πράξη γρήγορη.
3. Υπολογίζουμε την συνάρτηση Euler φ(n)=(p-1)(q-1)
4. Επιλέγουμε έναν αριθμό e τέτοιο ώστε $e^{\phi(n)} \equiv 1 \pmod{n}$
5. Υπολογίζουμε τον αριθμό d έτσι ώστε $d \equiv e^{-1} \pmod{\varphi(n)}$[7]

Από αυτή την διαδικασία δημιουργούμε το ιδιωτικό κλειδί που είναι η δυάδα n, e και το δημόσιο κλειδί είναι το d και τα p, q.

Αυτό το πρωτόκολλο έχει κάποιος όμως αδυναμίες, οι οποίες το κάνουν επιρρεπές σε επιθέσεις οι οποίες δεν χρειάζονται γνώση για την παραγοντοποίηση των ακεραίων. Αυτές είναι οι

- Επίθεση με κοινό modulus
- Επίθεση επαναληπτικής κρυπτογράφησης

Παρακάτω υπάρχει ένα παράδειγμα κώδικα σε Java που δείχνει πως υλοποιείται το πρωτόκολλο RSA. Σε καμία περίπτωση δεν προτείνειται να χρησιμοποιηθεί σε κάποιο πραγματικό σύστημα

```
public static void main(String[] args) {

    BigInteger n = new
BigInteger("95163118457906561534997167608470014334413357");
    BigInteger e = new BigInteger("65537");
    BigInteger d = new
BigInteger("561784318784495317030846362230283376298685");
    Charset c = Charsets.UTF_8;
    String plainText = "Rosetta Code";
    System.out.println("PlainText : " + plainText);
    byte[] bytes = plainText.getBytes();
    BigInteger plainNum = new BigInteger(bytes);
    System.out.println("As number : " + plainNum);
    BigInteger Bytes = new BigInteger(bytes);
    if (Bytes.compareTo(n) == 1) {
        System.out.println("Plaintext is too long");
        return;
    }
    BigInteger enc = plainNum.modPow(e, n);
    System.out.println("Encoded: " + enc);
    BigInteger dec = enc.modPow(d, n);
    System.out.println("Decoded: " + dec);
    String decText = new String(dec.toByteArray(), c);
    System.out.println("As text: " + decText);
}
```

[11]

Εκτός από την παραγοντοποίηση ακεραίων που χρησιμοποιείται από τον RSA, υπάρχουν και άλλες μαθηματικές θεωρήσεις στις οποίες βασίζονται διάφορα κρυπτογραφικά πρωτόκολλα. Ένα τέτοιο παράδειγμα είναι το κρυπτοσύστημα Merkle-Hellman. Το όνομα αποδίδεται στους δημιουργούς τους που είναι οι Ralph Merkle και Martin Hellman το 1978. Αυτό το κρυπτοσύστημα βασίζεται στο μετασχηματισμό υπεραύξουσας ακολουθίας σε αύξουσα ακολουθία.

- Έστω η υπεραύξουσα ακολουθία $b = (b_1, b_2, ..., b_n)$.

  Επιλέγουμε modulus m, τέτοιο ώστε m>$\sum_1^n wi$

- Στην συνέχεια επιλέγουμε ακέραιο w, τέτοιον ώστε 1<w<mκαι gcd(w,m)=1, και υπολογίζουμε την ακολουθία (a'1, a'2,….a'n) έτσι ώστε:

  $\alpha_i^{'}$ = wb_i mod m

- τέλος αναδιατάσουμε τα στοιχεία της α σε αύξουσα διάταξη. Η διάταξη που προκύπτει είναι γνησίως αύξουσα ακολουθία (όχι όμως υπεραύξουσα) και αντιστοιχέι σε δύκολο υπολογιστικά πρόβλημα.

- Από τα παραπάνω έχουμε ότι η (a_1 a_2 ….a_n) αποτελεί το δημόσιο κλειδί. Το σύνολο {(b_1 b_2 ….b_n), w, m} είναι το ιδιωτικό κλειδί. [2]

Η κρυπτογράφηση πραγματοποιείται σε τμήματα απλού κειμένου μεγέθους n bits, όπου n το πλήθος των στοιχείων της ακολουθίας. Έστω, (p_1 p_2 …p_n) το απλό κείμενο εκφρασμένο σε δυαδικά ψηφία. Η κρυπτογράφηση ορίζεται από την εξής πράξη

C=$\sum_{i=1}^{n} pi * ai$ [2]

5.3 Διαχείριση κλειδιών

Όπως έχει ήδη φανεί, κομβικό σημείο για την ασφάλεια ενός κρυπτοσυστήματος είναι τα κλειδιά. Συνεπώς, η σωστή δημιουργία, διαχείριση και διανομή των κλειδιών αποτελεί ζωτικής σημασίας για την ασφάλεια.

Η κρυπτοπερίοδος ενός κλειδιού είναι ο χρόνος ο οποίος περιλαμβάνει τη δημιουργία, διανομή και χρήση ενός κλειδιού. [2]

Η κρυπτοπερίοδος ενός κλειδιού εξαρτάται από τις παρακάτω παραμέτρους

1. Μήκος κλειδιού
2. Ευαισθησία του απλού κειμένου ως προς την εμπιστευτικότητα
3. Τύπος του κλειδιού
4. Από το ίδιο το κρυπτοσύστημα

Ανάλογα από το είδος του κρυπτοσυστήματος τα κλειδια χωρίζονται στις εξής κατηγορίες:

1) Μυστικό κλειδί, το οποίο υπάρχει στο συμμετρικό κρυπτοσύστημα
2) Δημόσιο κλειδί, το οποίο ορίζεται στο ασύμμετρο κρυπτοσύστημα
3) Ιδιωτικό κλειδί το οποίο ορίζεται και αυτό στο ασύμμετρο κρυπτοσύστημα.[2]

Ακόμη μια κατηγοριοποίηση των κλειδιών έχει να κάνει και με την χρήση για την οποία προορίζονται τα παραπάνω κλειδιά:

1) Κλειδί συνόδου, το οποίο χρησιμοποιείται μόνο για μια περίοδο επικοινωνίας.
2) Κλειδί τερματικού, το οποίο είναι όταν ένα κλειδί συνόδου χρησιμοποιείται περισσότερες από μια επικοινωνίες κάποιου μέλους.
3) Κύριο κλειδί. Όταν υπάρχει περίπτωση κάποιο μέλος να έχει πολλά κλειδιά συνόδου και τερματικού. Έχοντας ως σκοπό την απλούστευση της διαχείρισης όλων αυτών των κλειδιών χρησιμοποιείται το κύριο κλειδί.

Ο λόγος για τον οποίο υπάρχουν τόσα είδη κλειδιών είναι για λόγους πρακτικότητας. Ουσιαστικά με αυτούς τους τρόπους καταφέρνουμε με την διαχείριση κλειδιών, χρησιμοποιώντας μια ποσότητα πληροφορίας που το λέμε κλειδί να προστατέψουμε άλλα κλειδιά.

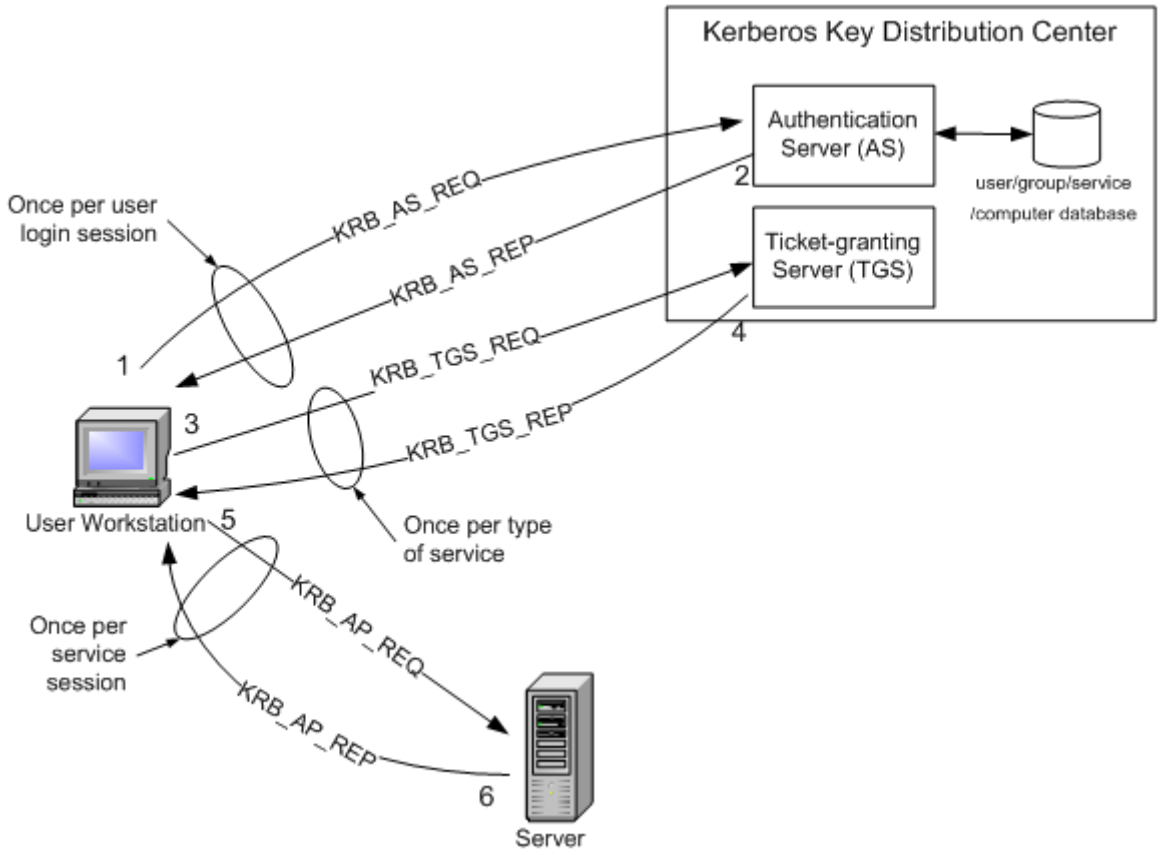
5.4 Εδραίωση κλειδιών

Εκτός από την διαχείριση των κλειδιών ένας άλλος σημαντικός παράγοντας για την κρυπτογραφημένη επικοινωνία είναι η εδραίωση των κλειδιών (key establishment).

Με τον όρο εδραίωση κλειδιών εννοούμε το σύνολο των μηχανισμών που εκτελούνται προκειμένου να αποκτήσουν τα επικοινωνούντα μέλη τα απαιτούμενα κλειδιά για κρυπτογραφική επικοινωνία. Οι μηχανισμοί εδραίωσης κλειδιών περιλαμβάνουν τη δημιουργία, μεταφορά, και εγκατάσταση κλειδιών, και αναφέρονται περισσότερο στα κλειδιά συνόδου. Οι μηχανισμοί εδραίωσης κλειδιών οι οποίοι είναι κρυπτογραφικής φύσης, αποτελούν τα πρωτόκολλα εδραίωσης κλειδιών. [2]

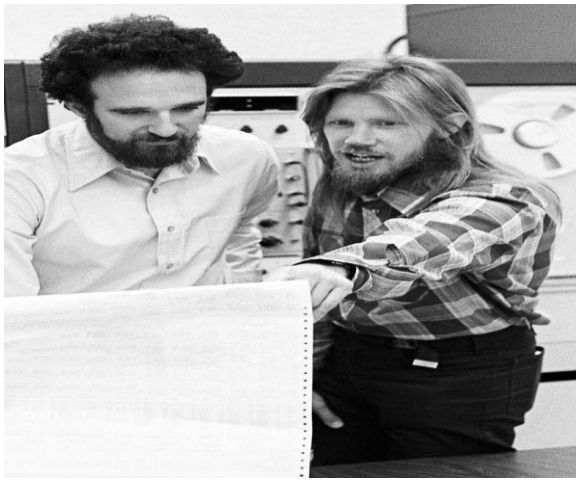
Όπως έχουμε περιγράψει η συμμετρική κρυπτογραφία έχει την αδυναμία της ανάγκης για ένα ασφαλές κανάλι, ώστε να μάθουν όλα τα εμπλεκόμενα μέλη τα κλειδιά. Για να μπορέσουμε να ξεπεράσουμε αυτό το τροχοπέδη, χρησιμοποιούμε τα εξής: 1) το κέντρο διανομής κλειδιών 2) το κέντρο μετάφρασης κλειδιών. Αυτές οι δύο λύσεις αποτελούν λύσεις συγκεντρωτικής διαχείρισης κλειδιών, καθώς εμπεριέχεται η έννοια του έμπιστου μέλους, μέσω του οποίου γίνεται η διανομή των

κλειδιών. Το βασικό πλεονέκτημα με την χρήση αυτών είναι η μείωση του πλήθους των κλειδιών που πρέπει να αποθηκευτούν. Το μειονέκτημα, εντούτοις είναι η προυπόθεση της εμπιστοσύνης που πρέπει να διακατέχει το ενδιάμεσο μέλος, το οποίο είναι δύσκολο να υπάρχει πάντα.[2]



Το σύστημα αυθεντικοποίησης Kerberos το οποίο δημιουργήθηκε από το MIT το οποίο περιέχει KDC. Src:
https://software.intel.com/sites/manageability/AMT_Implementation_and_Reference_Guide/default.htm?turl=WordDocuments%2Fintroductiontokerberosauthentication.htm

5.5 Ανταλλαγή κλειδιών Diffie-Hellman

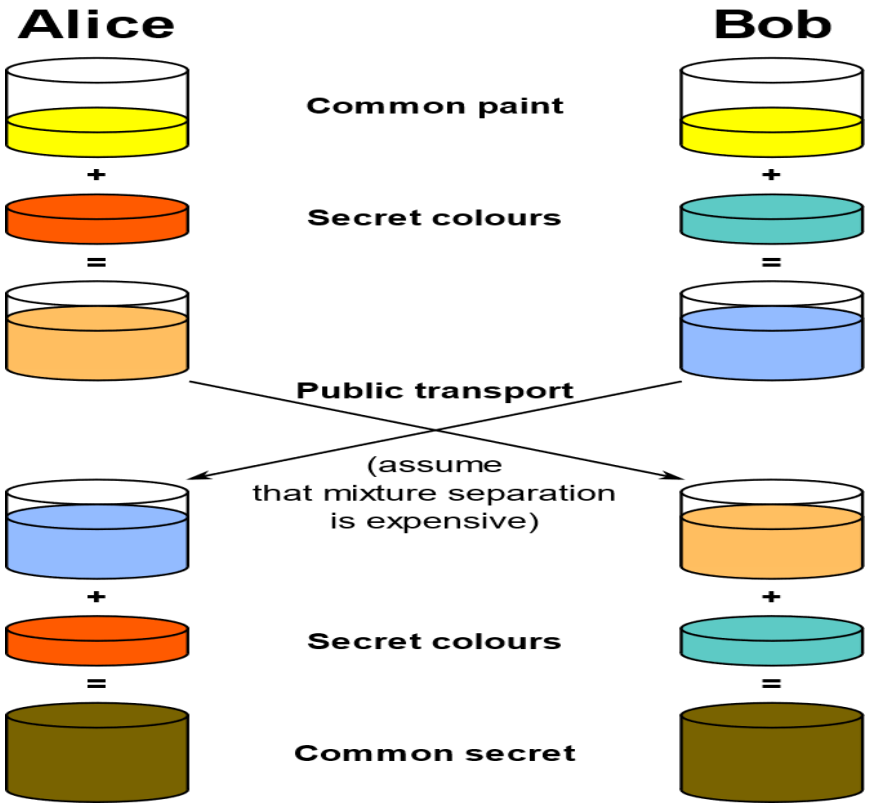


Οι Diffie και Hellman src: https://medium.com/cermati-tech/a-quick-look-into-diffie-hellman-key-exchange-24f32391b41e

Μέχρι τώρα η εδραίωση των κλειδιών γινόταν με συμμετρικό τρόπο, για αυτό υπήρχε η ανάγκη κάποιου «κέντρου». Έτσι το 1976 οι Whitefield Diffie και Martin Hellman δημιούργησαν το πρώτο ασύμμετρο πρωτόκολλο ανταλλαγής κλειδών που λέγεται Diffie-Hellman. Αυτό το πρωτόκολλο βασίζεται στο δύσκολο πρόβλημα του διακριτού λογαρίθμου. Τα βήματα που ακολουθούνται είναι τα εξής.

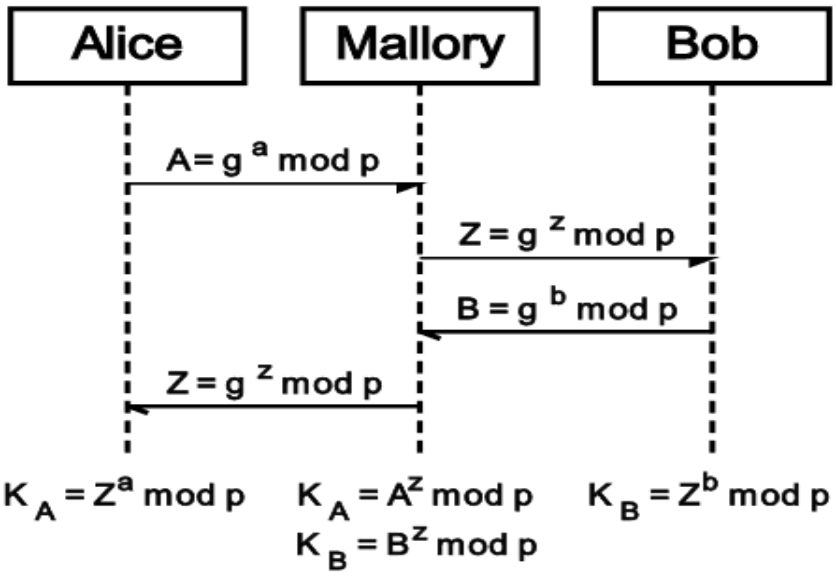
1) Σε πρώτο βήμα τα μέλη συμφωνούν σε ένα τεράστιο πρώτο αριθμό p, και έναν γεννήτορα g του Z_p^*
2) Κάθε μέλος διαλέγει αυθαίρετα έναν αριθμό από το Z_p^*. Έστω ο χρήστης A διαλέγει τον α και ο B το β.
3) Υπολογίζει ο καθένας τα g^amodp και g^bmodp αντίστοιχα.
4) Ανταλλάσσουν αυτά που υπολόγισαν.
5) Ο A υπολογίζει μετά $(g^b\text{mod}p)^a$ και ο B υπολογίζει $(g^\alpha\text{mod}p)^b$.
6) Το τελικό κλειδί είναι το g^{ab}modp.

Για να μπορέσει κάποιος κακόβουλος να αποκρυπτογραφήσει την επικοινωνία πρέπει να γνωρίζει τα a, b. Όμως για να το κάνει αυτό θα πρέπει να μπορεί να λύσει το πρόβλημα του διακριτού λογαρίθμου σε πολυωνυμικό χρόνο, το οποίο όμως δεν ισχύει.



Γραφικό παράδειγμα υλοποίησης του Diffie-Hellman. Src: https://upload.wikimedia.org/wikipedia/commons/4/46/Diffie-Hellman_Key_Exchange.svg

Το βασικό μειονέκτημα αυτού του πρωτοκόλλου είναι ότι είναι επιρρεπής στην επίθεση του «ενδιάμεσου μέλους». Δεν υπάρχει καθόλου μέριμνα για την περίπτωση που κάποιος ενδιάμεσος υποδύεται πως είναι ο παραλήπτης και ο αποστολέας δεν το καταλαβαίνει. Σε αυτή την περίπτωση μπορεί να αλλάξει τα μηνύματα και να ξανά κάνει την ίδια διαδικασία που περιγράφηκε στο πρωτόκολλο Diffie-Hellman (ο ενδιάμεσος χρήστης) με τον παραλήπτη. Παρακάτω υπάρχει ένα γραφικό παράδειγμα της επίθεσης του ενδιάμεσου χρήστη.



Παράδειγμα επίθεσης src:

https://upload.wikimedia.org/wikipedia/commons/5/50/Man-in-the-middle_attack_of_Diffie-Hellman_key_agreement.svg

ΒΙΒΛΙΟΓΡΑΦΙΑ

Βιβλία:

[1] Computer Security: Principles and Practices by William Stallings, Lawrie Brown

[2] Τεχνικές κρυπτογραφίας & κρυπτανάλυσης, Β.Α Κάτος Γ.Χ Στεφανίδης

URLs:

[3] https://en.wikipedia.org/wiki/Stuxnet

[4] https://en.wikipedia.org/wiki/Trojan_horse_(computing)

[5] https://www.kaspersky.com/resource-center/threats/trojans

[6] https://en.wikipedia.org/wiki/Cryptography

[7] https://el.wikipedia.org/wiki/RSA

[8] https://en.wikipedia.org/wiki/Fork_bomb

[9] https://en.wikipedia.org/wiki/Antivirus_software

[10] https://el.wikipedia.org/wiki/Firewall

[11] https://rosettacode.org/wiki/RSA_code#Java

[12] https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange

[13] https://el.wikipedia.org/wiki/Data_Encryption_Standard