ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

Σχολή Τεχνολογίας

Τμήμα Ψηφιακών Συστημάτων

Τίτλος Πτυχιακής

Κρυπτογραφικά εργαλεία από την θεωρία στην πράξη

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΠΑΝΓΙΩΤΑΚΟΠΟΥΛΟΣ ΓΙΑΝΝΗΣ (ΑΜ: 3119255)

Πατρώνυμο: Παναγιωτακόπουλος Παναγιώτης

Επιβλέπων: ***Απόστολος Ξενάκης, Επίκουρος καθηγητής τμήματος Ψηφιακών Συστημάτων***

ΛΑΡΙΣΑ, Μήνας Έτος

UNIVERSITY OF THESSALY

School of Technology

Digital Systems Department

Title of Thesis
Cryptographic tools from theory to practice

BACHELOR THESIS

Panagiotakopoulos Giannis (RN: 3119255)

Supervisor: Apostolos Xenakis, Assistant Professor, Department of Digital Systems

LARISSA, month year

Υπεύθυνη Δήλωση περί Ακαδημαϊκής Δεοντολογίας και Πνευματικών Δικαιωμάτων

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ρητά ότι η παρούσα πτυχιακή εργασία, καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας, αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον, και πληρούν τους κανόνες της επιστημονικής παράθεσης. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή/και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή.

Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

Ο/Η Δηλών/ούσα

(Υπογραφή)

Παναγιωτακοπουλος Ιωάννης

Ημερομηνία

Εγκρίνεται από την Επιτροπή Εξέτασης:

Επιβλέπων/πουσα	**Απόστολος Ξενάκης** Επίκουρος, Καθηγητής Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Θεσσαλίας
Μέλος	**Ηλίας Σάββας** Καθηγητής, Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Θεσσαλίας
Μέλος	**Κωσταντίνος Χαϊκάλης** Αναπληρωτής Καθηγητής Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Θεσσαλίας

Ημερομηνία έγκρισης: dd-mm-yyyy

Περίληψη

Τα κρυπτογραφικά εργαλεία διαδραματίζουν κρίσιμο ρόλο στην ασφάλεια των πληροφοριών στην ψηφιακή εποχή, γεφυρώνοντας τις θεωρητικές έννοιες με τις πρακτικές εφαρμογές. Η παρούσα μελέτη διερευνά τα βασικά κρυπτογραφικά εργαλεία που έχουν εξελιχθεί για τη διασφάλιση της εμπιστευτικότητας, της ακεραιότητας και της αυθεντικότητας των δεδομένων. Αναλύονται τα κλασικά συστήματα κρυπτογράφησης, συμπεριλαμβανομένων των μονοαλφαβητικών και πολυαλφαβητικών κρυπτογραφήσεων, όπως το one time pad, τα οποία έθεσαν τα θεμέλια για τη σύγχρονη κρυπτογραφία. Καθώς η κρυπτογραφία εξελισσόταν, εμφανίστηκε η κρυπτογραφία μυστικού κλειδιού, γνωστή και ως συμμετρική κρυπτογραφία, η οποία παρέχει έναν ισχυρό μηχανισμό για ασφαλή επικοινωνία. Ωστόσο, η ανάγκη για πιο κλιμακούμενες λύσεις οδήγησε στην ανάπτυξη της κρυπτογραφίας δημόσιου κλειδιού ή ασύμμετρης κρυπτογραφίας, η οποία επιτρέπει την ασφαλή ανταλλαγή δεδομένων χωρίς την ανάγκη για κοινό μυστικό.

Πέρα από την κρυπτογράφηση, η μελέτη εμβαθύνει σε διάφορα κρυπτογραφικά πρωτεύοντα και τους τρόπους λειτουργίας τους, όπως οι κρυπτογραφήσεις μπλοκ και οι κρυπτογραφήσεις ροής, οι οποίες είναι απαραίτητες για την ασφαλή επεξεργασία δεδομένων. Εξετάζονται επίσης οι συναρτήσεις κατακερματισμού, οι κώδικες αυθεντικοποίησης μηνυμάτων (MAC) και οι ψηφιακές υπογραφές, υπογραμμίζοντας τη σημασία τους για τη διασφάλιση της ακεραιότητας και της αυθεντικότητας των δεδομένων. Επιπλέον, συζητούνται οι μηχανισμοί διαχείρισης και ανταλλαγής κλειδιών, οι οποίοι είναι ζωτικής σημασίας για τη δημιουργία ασφαλών διαύλων επικοινωνίας.

Λέξεις κλειδιά

Κρυπτογραφικά εργαλεία, συστήματα κρυπτογράφησης, ψηφιακές υπογραφές, διαχείριση κλειδιών.

Abstract

Cryptographic tools play a critical role in information security in the digital age, bridging theoretical concepts with practical applications. This paper explores the key cryptographic tools that have evolved to ensure confidentiality, integrity and authentic-ty of data. Classical encryption schemes are studied, including single- and multi-alphabetic ciphers, such as the one-time pad, which laid the foundation for modern cryptography. As cryptography evolved, secret key cryptography, also known as symmetric cryptography, emerged, which provides a powerful mechanism for secure communication. However, the need for more scalable solutions led to the development of public key cryptography, or asymmetric cryptography, which allows for secure data exchange without the need for a shared secret.

Beyond encryption, the paper delves into various cryptographic primitives and their modes of operation, such as block ciphers and stream ciphers, which are essential for secure data processing. Hash functions, message authentication codes (MAC) and digital signatures are also examined, highlighting their importance in ensuring data integrity and authenticity. In addition, key management and exchange mechanisms are dis-cussed, which are crucial for the establishment of secure communication channels.

Keywords

Cryptographic tools, Encryption systems, Digital signatures, Key management

Ευχαριστίες

Είναι σημαντική η αναγνώριση της βοήθειας που έλαβε ο σπουδαστής κατά την διάρκεια της προπαρασκευής της εργασίας του. Μπορεί να είναι ακαδημαϊκή, τεχνική, γραμματειακή, διοικητική και προσωπική (π.χ. οικογένεια). Η ενότητα αυτή δεν πρέπει να ξεπερνά τη μισή σελίδα.

ονοματεπώνυμο

ημερομηνία

Περιεχόμενα

1 Εισαγωγή

Ο όρος Η λέξη "κρυπτογραφία", η οποία προέρχεται από τα συνώνυμα "κρυπτός" + "γράφω", αναφέρεται στον κλάδο της επιστήμης που μελετά, αναπτύσσει και εφάρμοζέ-μεθόδους κρυπτογράφησης και αποκρυπτογράφησης για την απόκρυψη του περιεχομένου των επικοινωνιών δύο ομιλητών. Είναι ο τομέας της έρευνας που είναι αφιερωμένος στην ασφαλή επικοινωνία. Ο κύριος στόχος είναι η δημιουργία ενός μέσου επικοινωνίας μεταξύ δύο ή περισσότερων μελών, έτσι ώστε μόνο τα ίδια τα μέλη να μπορούν να διαβάσουν τις πληροφορίες (Rubinstein-Salzedo, 2018).

Στο παρελθόν, τα μηνύματα κρυπτογραφούνταν με τη χρήση κρυπτογραφίας, η οποία μετέτρεπε τα δεδομένα από μια κανονικά κατανοητή μορφή σε έναν γρίφο που θα παρέμενε άλυτος χωρίς τη γνώση της κρυπτογραφικής τροποποίησης. Το κύριο χαρακτηριστικό των προηγούμενων μεθόδων κρυπτογράφησης ήταν ότι η επεξεργασία γινόταν πάνω από το γλωσσικό πλαίσιο. Η προσοχή έχει μεταφερθεί σε διάφορους μαθηματικούς κλάδους, συμπεριλαμβανομένων των διακριτών μαθηματικών, της θεωρίας αριθμών, της θεωρίας πληροφοριών, της υπολογιστικής πολυπλοκότητας, της στατιστικής και της συνδυαστικής ανάλυσης, ως αποτέλεσμα της χρήσης του αριθμητικού ισοδύναμου σε πιο πρόσφατους τύπους κρυπτογραφίας (Babington, 2018).

Τα θεμέλια της ασφαλούς επικοινωνίας στην ψηφιακή εποχή αποτελούνται από κρυπτογραφικές τεχνικές, οι οποίες προσφέρουν βασικές εγγυήσεις για την ακεραιότητα, τη μυστικότητα και την αυθεντικότητα των δεδομένων. Αυτά τα μέσα, τα οποία έχουν τις ρίζες τους σε περίπλοκες μαθηματικές θεωρίες, έχουν αλλάξει δραματικά με την πάροδο του χρόνου για να καλύψουν τις αυξανόμενες ανάγκες για ασφάλεια σε διάφορους κλάδους, συμπεριλαμβανομένων των τραπεζών και της προστασίας της ιδιωτικής ζωής (Šarac et al., 2018).

2 Κρυπτογραφία

2.1 Ιστορική αναδρομή

Κατά την Πρώτη Περίοδο της Κρυπτογραφίας (1900 π.Χ.-1900 μ.Χ.) αναπτύχθηκαν διάφορες κρυπτογραφικές προσεγγίσεις και αλγόριθμοι, οι οποίοι βασίζονταν κυρίως σε βασικές αντικαταστάσεις γραμμάτων. Τα επιτεύγματα αυτά στηρίχθηκαν στην εφευρετικότητα και την επινοητικότητα των δημιουργών τους και όχι σε εξειδικευμένες γνώσεις και περίπλοκη τεχνολογία. Επί του παρόντος, η κρυπτανάλυση έχει διεξαχθεί σε όλα αυτά τα συστήματα, και τα ευρήματα δείχνουν ότι το αρχικό κείμενο μπορεί να ανακτηθεί γρήγορα αν είναι γνωστό ένα σημαντικό τμήμα του κρυπτοκειμένου. Η χρήση της κρυπτογραφίας μπορεί να εντοπιστεί στις κοινωνίες της Μεσοποταμίας ήδη από το 1500 π.Χ., όπως υποδεικνύεται από την ανακάλυψη μιας μικρής σφηνοειδούς επιγραφής στις όχθες του ποταμού Τίγρη (Sahinaslan & Sahinaslan, 2019).

Το παλαιότερο γνωστό βιβλίο κρυπτογράφησης στον κόσμο πιστεύεται ότι είναι μια σφηνοειδής επιγραφή που βρέθηκε στα Σούσα της Περσίας. Οι αριθμοί 1 έως 8 και 32 έως 35 είναι τοποθετημένοι κάθετα, με τα αντίστοιχα σφηνοειδή σύμβολα για κάθε αριθμό τοποθετημένα οριζόντια απέναντί τους. Οι Σπαρτιάτες αναγνωρίζονται ως οι πρωτοπόροι στη χρήση της κρυπτογράφησης στον πόλεμο. Η αρχική κρυπτογραφική συ-σκευή, γνωστή ως "ράβδος", αναπτύχθηκε κατά τον πέμπτο αιώνα π.Χ. και χρησιμοποιούσε τη μέθοδο της κρυπτογραφικής αντικατάστασης. Το σπαρτιατικό Σκυτάλη, όπως περιγράφεται από τον Πλούταρχο, ήταν μια ξύλινη ράβδος με συγκεκριμένη διάμετρο γύρω από την οποία τυλίγονταν σπειροειδώς μια λωρίδα χαρτιού. Σε κάθε στήλη του κειμένου τυπώθηκε ένα γράμμα, και όταν η λωρίδα ξετυλίχθηκε, τα γράμματα αναμείχθηκαν μεταξύ τους και το κείμενο έγινε δυσανάγνωστο (Sahinaslan & Sahinaslan, 2019).

Η διάμετρος της ράβδου χρησίμευε ως "κλειδί". Τα συστήματα που βασίζονται στη στενογραφία κυριαρχούσαν στην αρχαιότητα, ενώ η κρυπτογραφία έπαιζε δευτερεύοντα ρόλο. Οι Έλληνες συγγραφείς δεν ασχολούνται με το αν χρησιμοποιούνταν ή όχι τεχνικές

αντικατάστασης γραπτών γραμμάτων, αλλά οι Ρωμαίοι το έκαναν, ιδίως κατά τη διάρκεια της βασιλείας του Ιουλίου Καίσαρα. Σε γράμματα τρεις θέσεις αργότερα στο λατινικό αλφάβητο, ο Ιούλιος Καίσαρας αντικαθιστούσε τα γράμματα του κειμένου σε επιστολές που έστελνε στον Κικέρωνα και σε άλλους φίλους του. Ως αποτέλεσμα, το κρυπτοσύστημα αντικατάστασης του Καίσαρα είναι η ονομασία που δόθηκε στο σύστημα κρυπτογράφησης που αντικαθιστά αλφαβητικά γράμματα με άλλα που τοποθετούνται έναν προκαθορισμένο αριθμό θέσεων είτε πριν είτε μετά (Sahinaslan & Sahinaslan, 2019).

Στον αραβικό κόσμο, τόσο τα μαθηματικά όσο και η κρυπτολογία είναι ακόμη αναπτυσσόμενοι τομείς. Οι γρίφοι λέξεων, τα αινίγματα, τα λογοπαίγνια και τα αναγράμματα κυριαρχούν σε όλο το γνωστό βιβλίο "Χίλιες και μία νύχτες". Ως αποτέλεσμα, άρχισαν να εμφανίζονται εκδόσεις με κρυπταναλυτικά αλφάβητα -όπως το αλφάβητο "Dawoudi", που πήρε το όνομά του από τον βασιλιά Δαυίδ- (Souza, 2016). Οι πρώτοι που δημιούργησαν και εφάρμοσαν τεχνικές κρυπτανάλυσης ήταν οι Άραβες. Τον δέκατο τέταρτο αιώνα ανέπτυξαν το κύριο εργαλείο για την ανάλυση των κωδίκων, το οποίο περιλαμβάνει τη σύγκριση των συχνοτήτων των γραμμάτων σε ένα δεδομένο κείμενο με τις συχνότητες των γραμμάτων στη γλώσσα στο σύνολό της (Rathidevi et al., 2017). Ως αποτέλεσμα των εξελίξεων στον τομέα του στρατού, η κρυπτογραφία είχε σημαντική ανάπτυξη τις επόμενες δεκαετίες. Ο Giovanni Batista Porta, ένας Ιταλός, δημοσίευσε τη γνωστή πραγματεία κρυπτολογίας "De furtivis literarum notis" το 1563. Περιελάβανε την εισαγωγή πολυαλφαβητικών κρυπτογραφικών συστημάτων και κρυπτογραφήσεων δύο γραμμάτων, οι οποίες αντικαθιστούν ένα γράμμα με δύο. Σημαντική ιστορική σημασία έχει ο Γάλλος Vigenere, ο οποίος ανέπτυξε τον πίνακα πολυαλφαβητικής αντικατάστασης που συνεχίζει να χρησιμοποιείται στη σύγχρονη εποχή (Sahinaslan & Sahinaslan, 2019).

Ο C. Wheatstone, γνωστός για τις ηλεκτρικές του γνώσεις, ανέπτυξε το αρχικό μηχανικό κρυπτοσύστημα. Το σύστημα αυτό αποτέλεσε τη βάση για την εξέλιξη των κρυπτογραφικών μηχανών καθ' όλη τη διάρκεια της δεύτερης εποχής της κρυπτογραφίας.

2.2 Κρυπτογραφία γενικά

Η πληροφορία θα πρέπει να γίνει αντιληπτή ως ένα μετρήσιμο φαινόμενο. Η κατανόηση των θεμάτων που σχετίζονται με την ασφάλεια των πληροφοριών είναι απαραίτητη για

την εισαγωγή μας στην κρυπτογραφία. Αυτή μπορεί να λάβει διάφορες μορφές ανάλογα με τις ανάγκες και τις περιστάσεις μας. Όλοι οι συμμετέχοντες σε μια ανταλλαγή πληροφοριών πρέπει να γνωρίζουν ότι πρέπει να ακολουθούνται ορισμένοι κανόνες προκειμένου να διασφαλίζεται η ασφαλής επικοινωνία. Για να επιτευχθεί αυτό, έχουν αναπτυχθεί στην ιστορία διάφορα πρωτόκολλα και διαδικασίες που αφορούν την ασφάλεια της μεταφοράς πληροφοριών. Δεν αρκεί η χρήση αποκλειστικά μαθηματικών μεθόδων και πρωτοκόλλων όταν συζητάμε για την ασφαλή ανταλλαγή πληροφοριών (Babington, 2018).

Η διαδικασία μετατροπής μιας επικοινωνίας με τη χρήση μιας κρυπτογραφικής μεθόδου σε μια ακατανόητη μορφή, έτσι ώστε μόνο ο προοριζόμενος παραλήπτης να μπορεί να την διαβάσει, είναι γνωστή ως κρυπτογράφηση. Η αποκρυπτογράφηση είναι η αντίστροφη διαδικασία που μετατρέπει το κρυπτογραφημένο κείμενο πίσω στο αρχικό μήνυμα. Κρυπτογράφηση είναι η διαδικασία μετατροπής των δεδομένων σε μορφή που περιορίζει την πρόσβαση σε αυτά από μη εξουσιοδοτημένα άτομα. Οι κρυπτογραφικές τεχνικές περιλαμβάνουν μερικές φορές περίπλοκους μαθηματικούς υπολογισμούς (Sharma & Gupta, 2017).

Η διαδικασία κρυπτογράφησης και αποκρυπτογράφησης μιας επικοινωνίας περιλαμβάνει τη χρήση μιας τεχνικής κρυπτογράφησης, γνωστής και ως κρυπτογράφησης, και ενός κλειδιού κρυπτογράφησης. Αυτή η μέθοδος και το κλειδί χρησιμοποιούνται για να διασφαλιστεί η ασφάλεια και η εμπιστευτικότητα της επικοινωνίας (AL-Abiachi et al., 2011). Η ασφάλεια του μεταδιδόμενου κρυπτογραφημένου μηνύματος βασίζεται κυρίως στη μυστικότητα του κλειδιού κρυπτογράφησης, δεδομένου ότι ο μηχανισμός κρυπτογράφησης είναι συχνά γνωστός. Το μέγεθος ενός κλειδιού κρυπτογράφησης καθορίζεται από τον αριθμό των bits που περιέχει. Η επόμενη οδηγία έχει καθολική εφαρμογή: Η δυσκολία των πιθανών χάκερ να αποκρυπτογραφήσουν μια κρυπτογραφημένη επικοινωνία αυξάνεται όσο αυξάνεται το μέγεθος του κλειδιού κρυπτογράφησης. Διαφορετικές τεχνικές κρυπτογράφησης απαιτούν διαφορετικά μήκη κλειδιών προκειμένου να επιτευχθούν ισοδύναμα επίπεδα ανθεκτικότητας της κρυπτογράφησης (Sharma & Gupta, 2017).

2.3 Κρυπτανάλυση γενικά

Η κρυπτανάλυση αποτελεί έναν από τους βασικότερους κλάδους της κρυπτογραφίας, με αντικείμενο τη μελέτη και την ανάλυση κρυπτογραφικών συστημάτων με σκοπό την αποκάλυψη του αρχικού μηνύματος χωρίς γνώση του κλειδιού που χρησιμοποιήθηκε για την κρυπτογράφηση. Με απλά λόγιά, η κρυπτανάλυση είναι τέχνη και η επιστήμη της παράκαμψη της ασφάλειας ενός κρυπτογραφικού συστήματος. Αν και πολλές φορές ταυτίζεται με την έννοια της παραβίασης η του σπασίματος ενός συστήματος, στην πράξη η κρυπτανάλυση είναι ένα αναγκαίο επιστημονικό εργαλείο για την αξιολόγηση της ασφάλειας των κρυπτογραφικών αλγορίθμων και πρωτοκόλλων. Η κρυπτανάλυση είναι τόσο παλιά όσο και η κρυπτογραφία. Ήδη από την αρχαιότητα, πολιτισμοί όπως οι Αιγύπτιοι, οι Έλληνες και οι Άραβες ανέπτυξαν βασικές τεχνικές για την αποκρυπτογράφηση κρυφών μηνυμάτων. Ένα από τα πρώτα ιστορικά παραδείγματα είναι το "Βιβλίο των Κρυφών Γραμμάτων" του Άραβα λόγιου Αλ-Κιντί, ο οποίος περιέγραψε την ανάλυση συχνοτήτων ως μέθοδο επίθεσης σε απλούς υποκαταστατικούς κώδικες.

Κατά τη διάρκεια του Β' Παγκοσμίου Πολέμου, η κρυπτανάλυση έπαιξε καθοριστικό ρόλο, με χαρακτηριστικότερο παράδειγμα τη διάρρηξη του Enigma, της κρυπτομηχανής των Ναζί, από τον Άλαν Τού ρινγκ και την ομάδα του στο Bletchley Park. Σκοπός και σημασία της κρυπτανάλυση είναι η ανακάλυψη του αρχικού μηνύματος χωρίς την κατοχή του κρυπτογραφικού κλειδιού.

Σε ευρύτερο πλαίσιο, η κρυπτανάλυση επιτρέπει:

- Την αξιολόγηση της ασφάλειας ενός αλγορίθμου
- Τον εντοπισμό αδυναμιών και σφαλμάτων στον σχεδιασμό ενός πρωτοκόλλου
- Τη βελτίωση των υπαρχόντων κρυπτογραφικών συστημάτων
- Την ανίχνευση κακόβουλης χρήσης ή πλαστογράφησης δεδομένων

Τεχνικές κρυπτανάλυσης

Ανάλογα με τη φύση του αλγορίθμου και του συστήματος, υπάρχουν διάφορες τεχνικές, όπως:

- **Ανάλυση συχνοτήτων**: Ιδανική για απλούς αλφαβητικούς κώδικες

- **Διαφορική και γραμμική κρυπτανάλυση**: Χρησιμοποιείται σε block ciphers όπως το DES
- **Επιθέσεις χρονισμού (timing attacks)**: Εκμεταλλεύονται τη διαφορά χρόνου εκτέλεσης
- **Αναλυτικά λάθη υλοποίησης**: Εκμετάλλευση ελαττωμάτων στο λογισμικό ή το hardware
- **Brute-force**: Δοκιμή όλων των πιθανών κλειδιών

Σύγχρονες προκλήσεις

Η σύγχρονη κρυπτανάλυση βρίσκεται αντιμέτωπη με νέες τεχνολογίες:

Κβαντικοί υπολογιστές: Η θεωρία προβλέπει ότι οι κβαντικοί υπολογιστές θα μπορούν να "σπάνε" ορισμένους αλγορίθμους, όπως το RSA

Κρυπτογραφία μηδενικής γνώσης (Zero-Knowledge Proofs): Μετατοπίζουν το ενδιαφέρον προς πιο "αποδείξιμες" προσεγγίσεις ασφάλειας

Αλγόριθμοι Post-Quantum: Σχεδιάζονται για να είναι ανθεκτικοί σε επιθέσεις από κβαντικά συστήματα

3 Συμμετρικά συστήματα κρυπτογραφίας

3.1 Εισαγωγή

Τα συμμετρικά συστήματα αναφέρονται σε αλγορίθμους όπου το κλειδί χρησιμοποιείται τόσο για κρυπτογράφηση όσο και για αποκρυπτογράφηση. Αυτό έρχεται σε αντίθεση με την ασύμμετρη κρυπτογραφία όπου χρησιμοποιούνται δυο διαφορετικά κλειδιά το πρώτο είναι ένα δημόσιο κλειδί κρυπτογράφηση και ένα ιδιωτικό κλειδί για αποκρυπτογράφηση. Στη συμμετρική κρυπτογράφηση, τόσο ο αποστολέας όσο και ο παραλήπτης πρέπει να μοιράζονται το ίδιο μυστικό κλειδί πριν από την επικοινωνία. Εάν αυτό το κλειδί εκτεθεί η υποκλαπεί από μη εξουσιοδοτημένο μέρος ,κινδυνεύει η ασφάλεια του συστήματος.

3.2 Μονοαλφβητική αντικατάσταση

Η μονοαλφβητική αντικατάσταση είναι μια μέθοδος κλασικής κρυπτογραφίας στην οποία ο κάθε χαρακτήρας του αρχικού μηνύματος αντικαθίσταται από ένα μόνο χαρακτήρα κάποιου αλλού αλφάβητου σύμφωνα με το σταθερό κλειδί που ορίζεται πριν την έναρξη της επικοινωνίας. Η βασική αρχή αυτής της μεθόδου είναι η εξής: κάθε γράμμα του αρχικού μηνύματος έχει μια μοναδική αντιστοίχιση με ένα γράμμα κρυπτογραφημένου μηνύματος. Αυτή η αντιστοίχιση παραμένει σταθερή σε ολόκληρο το μήνυμα δηλαδή αν το γράμμα Α αντικατασταθεί από το γράμμα Χ μια φορά, τότε κάθε φορά που εμφανίζεται το Α μήνυμα, θα αντικαθίσταται πάντα με το Χ.

3.2.1 Αλγόριθμος Καίσαρα

Ο αλγόριθμος του Καίσαρα είναι μια ειδική περίπτωση κρυπτογραφικού αλγορίθμου αντικατάστασης όπου κάθε σύμβολο του μηνύματος αντικαθίσταται από κάποιο άλλο σύμβολο. Απλά στον αλγόριθμο του Καίσαρα, η αντικατάσταση ακολουθεί έναν πολύ συγκεκριμένο και ειδικό κανόνα, ολίσθηση κατά k θέσεις. Ο κλασικός ιστορικός αλγόριθμος κρυπτογράφησης από τον Ιούλιο καίσαρα είναι κάθε γράμμα του λατινικού αλφάβητου αντικαθίσταται από εκεί που βρίσκεται 3 θέσεις δεξιά του στο αλφάβητο. Στον παρακάτω πίνακα δείχνουμε τα γράμματα του λατινικού αλφαβήτου καθώς και τις θέσεις στις οποίες αντιστοιχούν.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2
										0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5

Ακολουθεί ένα παράδειγμα του αλγόριθμου του Καίσαρα.

Παράδειγμα:

Μήνυμα : this is a car

Κρυπτογραφημένο μήνυμα : wklv lv d fdu

$\boldsymbol{t + 3 = 19 + 3 = 22}$ → **w**

$\boldsymbol{h + 3 = 7 + 3 = 10}$ → **k**

Αν η μετατόπιση μας οδηγήσει σε θέση **> 25** τότε συνεχίζουμε από την αρχή. Κάποιος που δεν γνωρίζει το μηχανισμό (αλγόριθμο) κρυπτογράφησης, δεν μπορεί να διαβάσει το αυθεντικό μήνυμα αν πέσει στα χεριά του αντίστοιχο κρυπτογραφημένο. Θεωρείται χαρακτηριστικό παράδειγμα μιας κατηγορία αλγόριθμοι μετατόπισης η ολίσθησης. Κάθε χαρακτήρας μετατοπίζεται κάποιες θέσεις.

3.2.2 Γραμμικός (Affine) Κρυπταλγόριθμος

Αποτελεί μια επέκταση του αλγορίθμου του Καίσαρα το κλειδί είναι δυο ακέραιοι αριθμοί a, b. Κρυπτογράφηση του χαρακτήρα m: $c = am + b(mod\ N)$ όπου N το πλήθος των συμβόλων του αλφάβητου από το οποίο απαρτίζεται το μήνυμά. Για την αγγλική γλώσσα $N = 26$, για την ελληνική $N = 24$.

Ουσιαστικά, γίνεται ολίσθηση κατά b θέσεις, όχι όμως στον αρχικό χαρακτήρα m αλλά σε έναν άλλον που προκύπτει από τον πολλαπλασιασμό του m με το a. Για $a = 1$,ο γραμμικός αλγόριθμος ταυτίζεται με τον αλγόριθμο του Καίσαρα.

Για την αποκρυπτογράφηση ενός κρυπτογραφημένο χαρακτήρα c έχουμε τον τύπο: $m = a - 1(c - b)\ mod\ N$.

Παράδειγμα:

Μήνυμα M=HELLO Φ Κλειδί: $a = 3$, $b = 1$ Φ Κάθε ένας χαρακτήρας του μηνύματος κρυπτογραφείται ξεχωριστά, με βάση τη μαθηματική σχέση της κρυπτογράφησης $c = am\ +\ b\ (mod\ N)$

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

Το H αντιστοιχεί στον αριθμό 7. Συνεπώς:$3\ \cdot\ 7\ +\ 1\ (mod\ 26)\ =\ 22$. Άρα το H κρυπτογραφείται στο γράμμα W. Φ Αντίστοιχα για το E έχουμε: $3\ \cdot\ 4\ +\ 1(mod\ 26)\ =\ 13$. Άρα το E κρυπτογραφείται στο γράμμα N. Φ Για το L: $3\cdot\ 11\ +\ 1(mod\ 26)\ =\ 34\ mod\ 26\ =8$. Άρατο L κρυπτογραφείται στο γράμμα I. Φ Τέλος, για το O: $3\cdot\ 14\ +\ 1(mod\ 26)\ =\ 43\ mod\ 26\ =\ 17$. Άρα το O κρυπτογραφείται στο γράμμα R. 17 Φ Συνεπώς, το μήνυμα M=HELLO κρυπτογραφείται σε C=WNIIR.

3.3 Πολυαλφαβητική Αντικατάσταση

Σε έναν πολυαλφαβητικό κρυπτογράφηση αντικατάστασης, χρησιμοποιούνται πολλαπλοί κανόνες αντικατάστασης και ο κανόνας που εφαρμόζεται σε κάθε γράμμα του απλού κειμένου αλλάζει σύμφωνα με ένα προκαθορισμένο σχήμα. Ένας από τους πιο γνωστούς πολυαλφαβητικούς κρυπτογράφους είναι ο κρυπτογράφος Vigenère, που πήρε το όνομά του από τον Γάλλο κρυπτογράφο Blaise de Vigenère.

3.3.1 Αλγόριθμος Vigenere

Ο αλγόριθμος δημιουργήθηκε από τον Γάλλο κρυπτογράφο Blaise de Vigenère, Χρησιμοποιεί μια φράση κλειδί για να κρυπτογραφήσει ένα μήνυμα. Κάθε χαρακτήρας του κλειδιού καθορίζει το πόσες θέσεις θα ολισθήσεις ο αντίστοιχος χαρακτήρας του αρχικού μηνύματος. Για παράδειγμα:

-**A**: ολίσθηση 0 θέσεων δεξιά

-**B**: ολίσθηση 1 θέσης δεξιά

-**C** : ολίσθηση 2 θέσεων δεξιά

- ….

Αν η φράση κλειδί έχει μικρότερο μήκος από το αρχικό μήνυμα, η φράση-κλειδί επαναλαμβάνεται.

Παράδειγμα:

Κλειδί: power

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2
										0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5

Πλήθος θέσεων ολίσθησης: Το πλήθος θέσεων ολίσθησης εξαρτάται από τη θέση του αντίστοιχου γράμματος στο αγγλικό αλφάβητο. Για παράδειγμα, το l είναι το 12ο γράμμα, άρα η μετατόπιση του αντίστοιχου γράμματος του αρχικού μηνύματος είναι κατά 11 θέσεις δεξιά κ.ο.κ.

p→15

o→13

w→22

e→4

r→17

Αρά το κλειδί power είναι ισοδύναμο με ολίσθηση θέσεων προς τα δεξιά.

Αρχικό μήνυμα: **simple message**. Επειδή το αρχικό μήνυμα έχει μεγαλύτερο μήκος από το κλειδί, τότε αυτό θα επαναλαμβάνεται όσες φορές χρειαστεί όπως δείχνουμε στον παρακάτω πίνακα

s	I	m	p	l	e	m	e	s	s	a	g	e
p	o	w	e	r	p	o	w	e	r	p	o	w
15	13	22	4	17	15	13	22	4	17	15	13	4
h	w	I	t	c	t	a	a	w	j	p	u	a

Άρα για την κρυπτογράφηση, το “s” θα ολίσθησή κατά 11 θέσεις δεξιά ,το ‘’I’’ κατά 20 θέσεις, Κ.Ο.Κ. Μπορούμε να βρούμε το γράμμα που αντιστοιχεί σε κάθε ολίσθηση, χρησιμοποιώντας τον πίνακα θέσεων των γραμμάτων της αλφάβητου.

ΑΡΧΙΚΟ ΜΗΝΥΜΑ: simple message

ΚΡΥΠΤΟΓΡΑΦΗΜΕΝΟ ΜΥΝΗΜΑ: hwitct aawjpua

Παρατηρώντας το κρυπτοκείμενο, δεν μπορεί να έχει πληροφορία σχετικά με το ποια γράμματα εμφανίζονται πολλές φορές στο αρχικό μήνυμα.

3.3.2 Αλγόριθμος HILL

Ο αλγόριθμος του HILL αποτελεί επέκταση του γραμμικού αλγορίθμου. Το μήνυμα κωδικοποιείται ανά block L στοιχείων με βάση L γραμμικές εξισώσεις συγκεκριμένα $c = km \ mod \ n$ οπού πίνακας διαστάσεων LxL.

Παράδειγμα εξισώσεων:

Για αρχή χωρίζουμε το μήνυμα που θα είναι σε αγγλικό κείμενο σε μπλοκ μήκους 3. Αν (p1, p2, p3) είναι ένα τέτοιο μπλοκ, τότε αυτό κρυπτογραφείται ως εξής:

$$c1 = 9 * p1 + 18 * p2 + 10 * p3(mod26)$$

$$c2 = 16 * p1 + 21 * p2 + 1 * p3(mod26)$$

$$c3 = 5 * p1 + 12 * p2 + 23 * p3(mod26)$$

$$\begin{pmatrix} c1 \\ c2 \\ c3 \end{pmatrix} = \begin{pmatrix} 9 & 18 & 10 \\ 16 & 21 & 1 \\ 5 & 12 & 23 \end{pmatrix} \begin{pmatrix} p1 \\ p2 \\ p3 \end{pmatrix} (mod\ 26)$$

Παράδειγμα:

Έστω το ακόλουθο μήνυμα :

I can do many things

8 2 0 13 3 14 13 0 14 25 19 8 13 7 19

Η αποκρυπτογράφηση γίνεται πολλαπλασιάζοντας κάθε μπλοκ του κρυπτοκειμενου,όπως αυτά προέκυψαν με την προηγουμένη διαδικασία, με τον αντίστροφο του K. Συνεπώς, ο πίνακας K πρέπει να είναι αντιστρέψιμος $(mod\ n)$. Ο K είναι αντιστρέψιμος αν και μόνο αν ισχύει $gcd(det(K), n) = 1$. Αν ο K δεν είναι αντιστρέψιμος , τότε υπάρχουν ζευγάρια block του μηνύματος, τα οποία κρυπτογραφούνται στο ιδιο block κρυπτοκειμένου – το οποίο βέβαια δεν πρέπει να συμβαίνει.

Παράδειγμα:

bcd → XJR

$$\begin{pmatrix} \mathbf{23} \\ \mathbf{9} \\ \mathbf{17} \end{pmatrix} = \begin{pmatrix} \mathbf{9} & \mathbf{18} & \mathbf{10} \\ \mathbf{16} & \mathbf{21} & \mathbf{1} \\ \mathbf{5} & \mathbf{12} & \mathbf{22} \end{pmatrix} \begin{pmatrix} \mathbf{1} \\ \mathbf{2} \\ \mathbf{3} \end{pmatrix} (\boldsymbol{mod26})$$

hfa → XJR

$$\begin{pmatrix} 23 \\ 9 \\ 17 \end{pmatrix} = \begin{pmatrix} 9 & 18 & 10 \\ 16 & 21 & 1 \\ 5 & 12 & 22 \end{pmatrix} \begin{pmatrix} 7 \\ 5 \\ 0 \end{pmatrix} (mod 26)$$

3.3.3 Αλγόριθμος Play fair

Στον αλγόριθμο play fair χωρίζουμε το μήνυμα σε ζεύγη γραμμάτων. Κάθε ένα ζεύγος το κρυπτογραφούμε με βάση τους κανόνες: Αν υπάρχει ζευγάρι με δυο ιδιά γράμματα τότε αναμεσά τους προστίθεται ένα X. Στην αποκρυπτογράφηση, τα πλεονάζοντα αυτά X ευκολά αναγνωρίζονται και απομακρύνονται. Αν τα δυο γράμματα του ζεύγους εμφανίζονται στην ιδιά γραμμή του πίνακα τότε το καθένα αντικαθίσταται από το πρώτο της γραμμής. Αν τα δυο γράμματα του ζεύγους εμφανίζονται στην ίδια στήλη στον πίνακα τότε το καθένα αντικαθίσταται από αυτό που βρίσκεται αμέσως κάτω του. Αν δεν βρίσκονται ούτε στην ίδια γραμμή ούτε στην ίδια στήλη, τότε φανταζόμαστε το νοητό ορθογώνιο που ορίζουν τα δυο γράμματα και τα αντικαθιστούμε από τα αλλά δυο γράμματα που αντιστοιχούν στις γωνίες του ορθογωνίου θα αντικατασταθεί από εκείνο το γράμμα που βρίσκεται στην ίδια γραμμή.

Παράδειγμα:

q	u	e	r	y
a	b	c	d	f
g	h	i or j	k	l
m	n	o	p	s
t	v	w	x	z

Br → du

Vk → XH

MO → NP

Ασφάλεια του Play fair αλγορίθμου:

Περισσότερη ασφάλεια από τους μονοαλφαβητικους αλγορίθμους αντικατάστασης .Συνεπώς, ένας έλεγχος συχνότητας εμφάνισης ζευγών γίνεται πιο σύνθετος . Χρησιμοποιήθηκε στον πρώτο παγκόσμιο πόλεμο από τον αμερικανικό στρατό και τον αγγλικό

στρατό. Σήμερα, με αρκετό δοθέν κρυπτοκείμενο, μπορεί να σπάσει με ελέγχους συχνότητας εμφάνισης διγραμμάτων.

3.4 Αλγόριθμοι Αντιμετάθεσης

Στους αλγορίθμους αντιμετάθεσης τα γράμματα του αρχικού μηνύματος ανακατεύονται (αναδιατάσσονται). Δηλαδή είναι οι γνωστοί μας αναγραμματισμοί. Σε αυτούς τους αλγορίθμους το μυστικό κλειδί είναι ο τρόπος αναδιάταξης. Παρακάτω δίνουμε 2 παραδείγματα των αλγορίθμων κρυπτογράφησης. Αν και το πλήθος όλων των πιθανών κλειδιών μπορεί να είναι πολύ μεγάλο (θεωρητικά, αν δε υπάρχουν περιορισμοί στις αντιμεταθέσεις, το μέγιστο δυνατό πλήθος είναι N!, οπού N το μέγεθος του μηνύματος), εν τούτοις δεν θεωρούνται ασφαλείς αλγόριθμοι για λόγους που θα αναπτύξουμε μετά.

3.4.1 Αλγόριθμος Rail Fence

Στον αλγόριθμο κρυπτογράφησης Rail Fence το αρχικό κείμενο γράφεται προς τα κάτω και διαγώνια σε k διαδοχικές σειρές (ράγες) ενός φανταστικού φράχτη. Όταν φτάσουμε στην τελευταία σειρά, γυρίζουμε προς τα πάνω κινούμενοι διαγώνια, και αφού φτάσουμε στην πρώτη σειρά η κατεύθυνση αλλάζει ξανά. Στην ουσία τα γράμματα του μνήματος γράφονται σε ζιγκ-ζαγκ. Αφού γραφτεί όλο το μήνυμα όλες οι μεμονωμένες γραμμές του συνδικάζονται ώστε να μπορέσουμε να δημιουργήσουμε το κρυπτογραφημένο μας κείμενο. Το κλειδί του αλγορίθμου είναι το πλήθος των γραμμών k. Για παράδειγμα αν έχουμε k = 3 τότε αυτό σημαίνει ότι θα τοποθετήσουμε όλο το μήνυμα σε τρεις γραμμές. Παρακάτω δείχνουμε ένα παράδειγμα λειτουργίας του αλγορίθμου.

Παράδειγμα κρυπτογράφησης Rail Fence

Έστω ότι θέλουμε κρυπτογραφήσουμε το κείμενο: INFORMATION TECHNIQUES με κλειδί k = 3. Κατασκευάζουμε 3 σειρές αφού το κλειδί είναι 3 και 21 στήλες δηλαδή όσο είναι και το μήκος του αρχικού μηνύματος. Κατόπιν γράφουμε τους χαρακτήρες του αρχικοί κειμένου κινούμενοι ζιγκ-ζαγκ όπως φαίνεται στον παρακάτω πίνακα.

I				R				I				C				Q				
	N		O		M			T		O			E		H		I		U		S
		F				A				N	T				N				E		

Διαβάζουμε οριζόντια για να προκύψει το κρυπτοκείμενο

Κρυπτοκείμενο: **IRIEISNOMTOTCNQEFANHU**

Αποκρυπτογράφηση παράδειγμα

Το κρυπτοκείμενο: **IRIEISNOMTOTCNQEFANHU** έχει μήκος κρυπτοκειμένου: 21 χαρακτήρες και κλειδί να είναι k=3. Κατασκευάζουμε πλέγμα με 3 σειρές (όσο είναι το κλειδί) και 21 στήλες (όσο το μήκος του κρυπτογραφήματος). Χρησιμοποιώντας την ίδια διαδρομή Ζιγκ ζαγκ, μαρκάρουμε στις οποίες θα μπει το κείμενο:

I				R				I				C				Q				
	N		O		M			T		O			E		H		I		U		S
		F				A				N	T				N				E		

Κινούμενοι οριζόντια συμπληρώνουμε τα γράμματα του κρυπτοκειμένου στις επιλεγμένες θέσεις. Διαβάζουμε ζιγκ ζαγκ ώστε να προκύψει το αρχικό κείμενο :

Αρχικό κείμενο: **INFORMATION TECHNIQUES**

3.5 Το Πρόβλημα της Ανταλλαγής Κλειδιών

Οι σύγχρονοι αλγόριθμοι κρυπτογραφίας είναι ασφαλείς αρκεί να μην διαρρεύσει το κλειδί. Συγκεκριμένα όλη η ασφάλεια βρίσκεται στην μυστικότητα του κλειδιού (αρχή του Kerckhoffs, 1883). Με ποιον τρόπο όμως θα μπορούσαμε να μεταδώσουμε με ασφάλεια το κλειδί στον παραλήπτη; Την απάντηση την δίνει ο αλγόριθμος Diffie-Hellman.

3.5.1 Αλγόριθμος Diffie-Hellman

Ένα σενάριο το οποίο ακολουθεί την μέθοδο ανταλλαγής κλειδιών σύμφωνα με τον αλγόριθμο Diffie-Hellman είναι το ακόλουθο:

Έστω ότι ο Bob και η Alice θέλουν να επικοινωνήσουν και ανταλλάξουν μηνύματα μεταξύ τους με ασφάλεια. Έστω ότι χρησιμοποιούν κάποια μέθοδο συμμετρικής κρυπτογράφησης και πρέπει να ανταλλάξουν μεταξύ τους το μυστικό κλειδί. Τα βήματα που ακολουθούνται για την ασφαλή ανταλλαγή του κλειδιού είναι τα παρακάτω:

1. Η Alice και ο Bob συμφωνούν σε δυο αριθμούς, ένα πρώτο αριθμό έστω p, και έναν άλλο όχι αναγκαστικά πρώτο, g έτσι ώστε $2 \leq g \leq p-2$. Έστω ότι p=23 και g=14. Οι αριθμοί p, g δεν χρειάζεται να κρατηθούν μυστικοί.
2. Η Alice επιλεγεί ένα μυστικό αριθμό, έστω a τέτοιο ώστε $1 \leq \alpha \leq p-1$. Έστω ότι έχει επιλέξει a=3. Υπολογίζει τον αριθμό $A = g^a \, mod \, p$ που είναι 14^3 mod 23= 2744 mod 23 = 7.
3. Η Alice στέλνει τον αριθμό A, δηλαδή το 7, μέσω του μη ασφαλούς καναλιού στον Bob.

4. Ο Bob επιλεγεί ένα μυστικό αριθμό, έστω b τέτοιο ώστε $1 \leq b \leq p-1$. Έστω ότι επιλεγεί b=4. Κατόπιν κάνει τους ίδιους υπολογισμούς με την Alice, δηλαδή $B = g^b \, mod \, p$ δηλαδη 14^4mod 23 = 38416 mod 23 = 6.
5. Ο Bob στέλνει τον αριθμό B, δηλαδή το 6 μέσω του μη ασφαλής καναλιού στην Alice.
6. Η Alice υπολογίζει τον αριθμό b^amod $p = 6^3 \; mod \; 23 = 216 \, mod \; 23 = 9$.
7. Ο Bob υπολογίζει τον αριθμό $A^b mod \; p = 7^4 \; mod \; 23 = 2401 \, mod \; 23 = 9$.
8. Το 9 είναι το μυστικό κλειδί της Alice και του Bob.

Σχηματικά τα βήματα περιγράφονται όπως στην παρακάτω εικόνα:

Alice	Bob
Alice and Bob agree on p and g	
Choose a Calculate $A = g^a \bmod p$ Send A to Bob Calculate $s = B^a \bmod p$ $= (g^b)^a \bmod p$ $= g^{ba} \bmod p$	Choose b Calculate $B = g^b \bmod p$ Send B to Alice Calculate $s = A^b \bmod p$ $= (g^a)^b \bmod p$ $= g^{ab} \bmod p$

Εικόνα 1 Τα βήματα του αλγορίθμου Diffie Hellman. Πηγή [1]

Τα προηγούμενα βήματα είναι τα βήματα που προτείνει ο αλγόριθμος Diffie Hellman για την ασφαλή ανταλλαγή του μυστικού κλειδιού. Δεν υπάρχει γνωστός αποτελεσματικός τρόπος να βρεθεί το μυστικό κλειδί μέσω των p, g, A, και B. Αυτό ανάγεται στο πρόβλημα του διακριτού λογάριθμου (discrete logarithm problem), για το οποίο δεν υπάρχει αποτελεσματική λύση. Αν έχουμε έναν πρώτο αριθμό p, έναν αριθμό g, και $y = g^x \bmod p$, το πρόβλημα του διακριτού λογαρίθμου είναι η εύρεση του ακεραίου x, $1 \leq x \leq p-1$. Ο ακέραιος x ονομάζεται διακριτός λογάριθμος του y με βάση g και γράφουμε $x = \log_g y \bmod p$.

Η συνάρτηση $y = g^x mod \; p$ είναι μια μονόδρομη συνάρτηση (one way function). Λέγεται μονόδρομη διότι ενώ είναι πολύ εύκολο να υπολογίσουμε το y αν έχουμε διαθέσιμα τα g, x, p ωστόσο δεν υπάρχει αποτελεσματική και συστηματική μέθοδο

να υπολογίσουμε το x αν γνωρίζουμε τα y, g, p. Ο μόνος τρόπος να το κάνουμε αυτό είναι να δοκιμάσουμε διάφορες τιμές του x μέχρι να βρούμε την σωστή.

Θα πρέπει να εξετάσουμε την συμπεριφορά των αριθμών g^x και $g^x mod\ p$ προκειμένου να βγάλουμε χρήσιμα συμπεράσματα. Για g =2 και p = 13 έχουμε:

x	1	2	3	4	5	6	7	8	9	10	11	12
g^x	2	4	8	16	32	64	128	256	512	1024	2048	4096
$g^x mod\ p$	2	4	8	3	6	12	11	9	5	10	7	1

Από τον παραπάνω πίνακα παρατηρούμε ότι η συμπεριφορά του 2^x είναι προβλέψιμη ενώ η συμπεριφορά του 2^x mod 13 δεν φαίνεται να είναι. Πράγματι το 2^x mod 13 παίρνει όλες τις τιμές από το 1 μέχρι το 13-1=12 πριν αρχίσει να επαναλαμβάνεται. Για την συνάρτηση αυτή το 12 είναι η θεμελιώδης περίοδος δηλαδή δεν υπάρχει περίοδος μικρότερη από αυτή.

Για g = 3 και p = 13 έχουμε:

x	1	2	3	4	5	6	7	8	9	10	11
g^x	3	9	27	81	243	729	2187	6561	19683	59049	177147
$g^x mod\ p$	3	9	1	3	9	1	3	9	1	3	9

Από τον παραπάνω πίνακα παρατηρούμε ότι το $3^x mod\ 13$ δεν παίρνει όλες τις τιμές από το 1 μέχρι το 13-1=12 πριν αρχίσει να επαναλαμβάνεται. Η θεμελιώδης περίοδος του είναι πολύ μικρή, μόνο 3. Συνεπώς αν θα έπρεπε να δοκιμάσουμε να βρούμε το x με g = 3 και p = 13 τότε θα αρκούσε να δοκιμάζαμε μόνο 3 διαφορετικές τιμές ενώ στο προηγούμενο πίνακα θα έπρεπε να δοκιμάσουμε πολύ περισσότερες τιμές μέχρι να βρούμε τον μυστικό αριθμό x.

Η επιλογή τον αριθμών g και p θα πρέπει να είναι τέτοια ώστε η θεμελιώδης περίοδος να είναι πάρα πολύ μεγάλη. Γενικά επιλέγουμε τον αριθμό p να είναι ένας πολύ μεγάλος πρώτος αριθμός που να έχει πάνω από 1200 δεκαδικά ψηφιά (ένας τέτοιος αριθμός χρειάζεται 4096 bits για την αναπαράσταση του. Κατόπιν επιλέγουμε ένα αριθμό g σύμφωνα με τις αρχές της θεωρίας αριθμών ώστε η περίοδος του $g^x\ mod\ p$ να είναι πάρα πολύ μεγάλη.

4 Κρυπταλγόριθμοι Τμήματος

Οι Κρυπταλγόριθμοι τμήματος (block ciphers) είναι οι περισσότερο διαδεδομένοι κρυπτογραφικοί αλγόριθμοι. Χαρακτηριστικές τους εφαρμογές:

- Διαδίκτυο (πρωτόκολλο https για κρυπτογραφημένες συνδέσεις)
- Email (πρωτόκολλο S-MIME για κρυπτογράφηση ηλεκτρονικών μηνυμάτων)
- Συναλλαγές στα ATM τραπεζών
- Εικονικά ιδιωτικά δίκτυα (VPNs)
- Λογισμικά κρυπτογραφήσεις σκληρών δίσκων Η/Υ

Οι Κρυπταλγόριθμοι τμήματος είναι συμμετρική αλγόριθμοι κρυπτογραφήσεις όπως και οι Κρυπταλγόριθμοι ροής, αλλά η κρυπτογράφηση γίνεται με διαφορετικό τρόπο.

Σε αυτούς τους αλγορίθμους το αρχικό μήνυμα χωρίζεται σε τμήματα (block) σταθερού μεγέθους και το κάθε τμήμα κρυπτογραφείται ξεχωριστά. Ένα τυπικό μέγεθος τμήματος είναι 128 bits. Η κρυπτογράφηση είναι μια αρκετά συνθέτη διαδικασία (δεν είναι απλή πρόσθεση XOR όπως στους Κρυπταλγόριθμους ροής).

Γνωστοί Κρυπταλγόριθμοι τμήματος είναι οι DES, 3DES, AES. Αυτοί οι αλγόριθμοι έχουν χρησιμοποιηθεί ως διεθνή πρότυπα Κρυπταλγόριθμοι τμήματος.

Ο αλγόριθμος DES (Data Encryption Standard) ορίστηκε σαν πρότυπο κρυπτογράφησης στις 23 Νοέμβριου του 1976. Κάθε 5 χρόνια από τότε γινόταν ανανέωση του DES ως πρότυπο. Το 1987 αν και δεν είχε αποδειχθεί ενδεχομένη μη ασφάλεια του άρχισαν οι πρώτες έντονες αμφισβητήσεις. Πλέον θεωρείται μη ασφαλής αλγόριθμος.

Το μεγάλο μειονέκτημα του DES ως προς την ασφάλεια του ήταν το μέγεθος του κλειδιού που χρησιμοποιούσε το οποίο ήταν 56 bits. Για να αντιμετωπιστεί αυτό το μειονέκτημα του DES καθορίστηκε ένα νέο πρότυπο το 3DES δηλαδή ο τριπλός DES που χρησιμοποιεί τριπλάσιο μέγεθος κλειδιού τον 168 bits. Το 1997 ο NIST προσκάλεσε δημοσιά για ορισμό νέο πρότυπου που θα λάμβανε το όνομα Advanced Encryption Standard (AES) προς αντικατάσταση του DES. Ως ελάχιστο μήκος κλειδιού τέθηκαν τα 128 bits. Παρακάτω περιγράφεται ο αλγόριθμος AES.

4.1 Αλγόριθμος Advanced Encryption Standard (AES)

Ο AES στην σύγχρονη μορφή του υποστηρίζει 3 μήκη κλειδιού: 128, 192, 256 bits.

Το μήκος κάθε μπλοκ είναι τα 128 bits. Ένα μεγάλου του προσόν είναι ότι υλοποιείται εύκολα με hardware. Σύμφωνα με τον αλγόριθμο αυτό απαιτούνται 10 έως 15 γύροι ανάλογα με το μήκος του κλειδιού. Κάθε ένας γύρος αποτελείται από 4 βασικές πράξεις:

- Αντικατάσταση byte (byte substitution) η οποία γίνεται με χρήση s-boxes με καλά χαρακτηριστικά
- Ολίσθηση (shift row)
- Συνδυασμός πολλών bit (mix column)
- Πρόσθεση (XOR) του κλειδιού

Παρακάτω παραθέτουμε τον αλγόριθμο AES σε ψευδοκώδικα

Algorithm: AES cipher algorithm.

```
AESCipher(b, k, n) → s
    Input: b, a block of 16 bytes
           k, the encryption key
           n, the number of rounds
    Data: s, the state
          rk, an array of size n + 1 that will contain the round keys
    Output: s, the ciphertext corresponding to b
1   s ← CreateState(b)
2   rk ← ExpandKey(k)
3   s ← AddRoundKey(s, rk[0])
4   for i ← 1 to n do
5       s ← SubBytes(s)
6       s ← ShiftRows(s)
7       s ← MixColumns(s)
8       s ← AddRoundKey(s, rk[i])
9   s ← SubBytes(s)
10  s ← ShiftRows(s)
11  s ← AddRoundKey(s, rk[n])
12  return s
```

Εικόνα 2 Τα βήματα του αλγορίθμου AES. Πηγή [1]

5 Κρυπταλγόριθμοι Ροής

5.1 One Time Pad

Ο μηχανικός Gilbert Vernam είχε προτείνει από το 1918 ένα σύστημα οπού το κλειδί είχε μέγεθος ίσο με το μήνυμα, και η κρυπτογράφηση ήταν η πρόσθεση xor του κλειδιού με το μήνυμα (αλγόριθμος vernam). Η κρυπτογράφηση περιγράφεται από τη σχέση c= m xor k. Για την αποκρυπτογράφηση, εφαρμόζεται η αντιστροφή σχέση: -m=c xor k.

Παράδειγμα:

Μήνυμα m: 00101101

Κλειδί κ: 11101001

Κρυπτοκείμενο: c = m xor k = 00101101 xor 111011001= 11000100

Ως σημειωματάριο μια χρήσης αποκαλείται το ιδανικό εκείνο κρυπτοσύστημα, γενίκευση του αλγορίθμου του vernam, όπου: το κλειδί είναι μια τυχαία ακολουθία από bits, μεγέθους όσο και το μήνυμα, μη περιοδική. Δεν χρησιμοποιείται ποτέ το ίδιο κλειδί εκ νέου. Το one time pad επιτυγχάνει τέλεια μυστικότητα κατά Shannon. Βασική ιδέα για την απόδειξη : Για κάθε μήνυμα m=m1m2 ... m και κρυπτοκείμενο c = c1c2 ... cn υπάρχει ένα μοναδικό κλειδί k = k1k2 ... kn τέτοιο ώστε Εκ(m)=c.

$$p(m|c) = \frac{p(m)p(c|m)}{p(c)} = \frac{p(m)pk}{\sum_{m,k} p(m)pk} = p(m)$$

Επειδή $p_{k=}1/2^n$ για όλα τα k και $\sum_m p(m) = 1$

Το σημειωματάριο μιας χρήσης δεν μπορεί να υλοποιηθεί πρακτικά για τους δυο εξής λογούς :

1. Δεν μπορούμε να εξασφαλίσουμε ότι θα έχουμε κλειδί ίδιου μεγέθους με το μήνυμα. Πως θα ανταλλάξουμε κλειδί με ασφάλεια; Ας ανακαλέσουμε τον αλγόριθμο Vigenère : Το κλειδί αναγκαστικά θα επαναλαμβάνεται περιοδικά, διότι το μέγεθος του κλειδιού είναι ίδιο με το μήνυμα να γεννάει το πρόβλημα ασφαλούς διανομής του ενώ επίσης το κλειδί << αποφασίζεται >> πριν την έναρξη της επικοινωνίας.

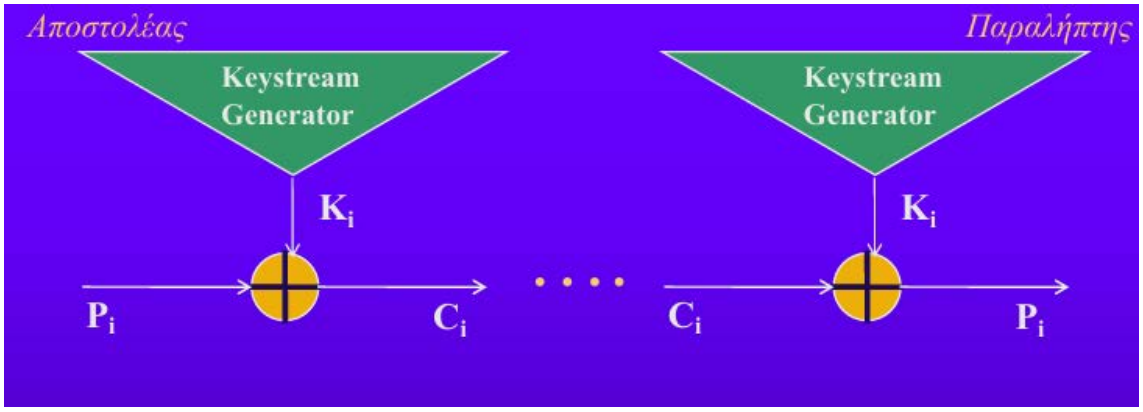
2. Δεν μπορούμε από μια ντετερμινιστική μηχανή να έχουμε παραγωγή απολύτως τυχαίας ακολουθίας. Στόχος των κρυπτογραφικών αλγορίθμων ροής είναι να προσομοιάσουν, κατά το δυνατόν, στο σημειωματάριο μιας χρήσης.

5.2 Κρυπταλγόριθμοι ροής – Πως Λειτουργούν

Οι **Κρυπταλγόριθμοι ροής** (stream ciphers) είναι ένας τύπος συμμετρικού κρυπτογραφικού αλγορίθμου που χρησιμοποιείται για την **κρυπτογράφηση δεδομένων σε ροή**, δηλαδή bit-προς-bit ή byte-προς-byte, σε αντίθεση με τους **Κρυπταλγόριθμοι μπλοκ**, που επεξεργάζονται δεδομένα σε ομάδες (blocks).

Πως λειτουργούν : Ξεκινώντας από ένα μυστικό κλειδί, ο αλγόριθμος παράγει μια ψευδοτυχαια ακολουθία. Η ακολουθία αυτή πρέπει να είναι απρόβλεπτη χωρίς το κλειδί.

Συνδυασμός keystream με το αρχικό κείμενο : κάθε bit του αρχικού μηνύματος συνδυάζεται με το αντίστοιχο από την keystream με απλή λειτουργία, συνήθως XOR. **Αποκρυπτογράφηση :** για να αποκρυπτογραφήσουμε εφαρμόζουμε την ίδια keystream να αναπαραχθεί στο cipher text με ακρίβεια . Δηλαδή: plaintext = ciphertext XOR keystream.



Εικόνα 3 Αποστολή κρυπτογραφημένου μηνύματος στον παραλήπτη. Πηγή [3]

5.3 Τυχαιότητα Κλειδοροής

Η τυχαιότητα αναφέρεται στην ποιότητα και την ασφάλεια της ψευδοτυχαίας ακολουθίας που παράγουν οι Κρυπταλγόριθμοι ροής. Η Κλειδοροή είναι η ροή bit η byte που συνδικάζεται με το αρχικό μήνυμα για να παραχθεί το κρυπτογράφημα.

Για παράδειγμα είναι προφανές ότι η ακολουθία 11111111110 δεν μπορεί να θεωρηθεί τυχαία. Ομοίως, ούτε η ακολουθία 10101010 μπορεί να θεωρηθεί τυχαία. Το πως κατανέμονται τα 0 και 1 πρέπει να έχουν μια τυχαία μορφή, από την άλλη μεριά, μπορούμε να ορίσουμε ποτέ ακριβώς είναι τυχαία ακολουθία;

Κάποια πρώτα κριτήρια τυχαιοτητας προτάθηκαν από τον Golomb

Για μια ακολουθία $a_1,a_2,a_3\ldots$με περίοδο N, πρέπει να ισχύουν τα εξής :

1. Ισοκατανεμημένο πλήθος 0 και 1 (Balance Property)
2. Αν ως διαδρομή (run) ορίζουμε ένα τμήμα της ακολουθίας που αποτελείται μόνο από μηδενικά ή μόνο από άσους (και αμέσως πριν και μετά από αυτά βρίσκονται διαφορετικά bit από αυτά που απαρτίζουν το τμήμα), τότε σε μία περίοδο οι μισές διαδρομές έχουν μήκος 1, το ¼ των διαδρομών έχουν μήκος 2, το 1/8 μήκος 3 Κ.Ο.Κ.
3. Η συνάρτηση αυτοσυσχέτισης

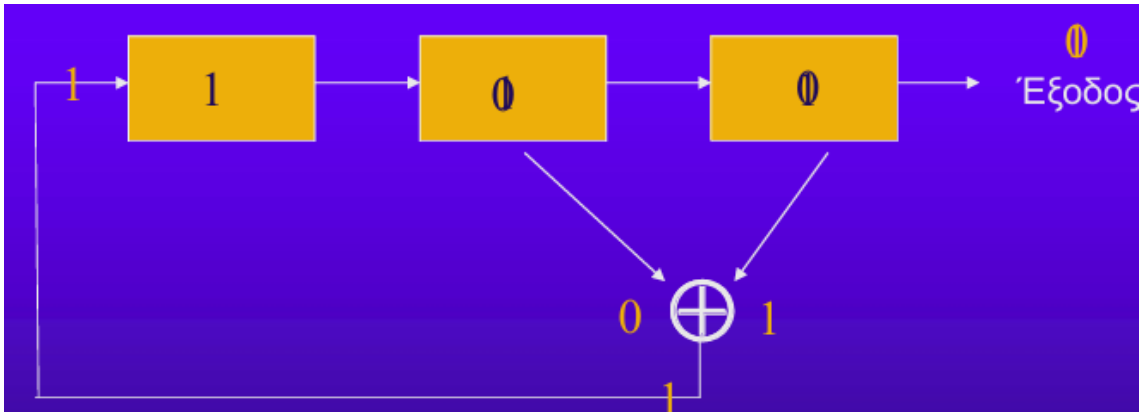
$$C(\tau) = \sum_{i=0}^{N-1} (-1)^{ai\ xor\ ai+r}$$

για την ακολουθία a0a1… περιόδου N μπορεί να πάρει μόνο δύο τιμές: να είναι σταθερή (ίση με K) για $\tau \neq 0$, και τιμή N για $\tau=0$.

5.4 Γραμμικοί Καταχωρητές Ολίσθησης με ανάδραση (Linear Feedback Shift Registers – LFSRs)

Είναι ένα ψηφιακό κύκλωμα, που αποτελείται από n βαθμίδες. Υπάρχει μια πύλη XOR, η οποία έχει ως εισόδους κάποιες από τις βαθμίδες του LSFR. Το σύνολο των τιμών των βαθμίδων του LSFR ονομάζεται κατάσταση του LSFR για την τρέχουσα χρονική στιγμή. Κάθε χρονική στιγμή, η κατάσταση του μεταβάλλεται. Η μεταβολή της κατάστασης ενεργοποιείται με τον παλμό του ρολογιού του συστήματος. Σε κάθε νέα κατάσταση η τιμή της πρώτης βαθμίδας του LFSR προκύπτει από τις τιμές της προηγουμένης κατάστασης, βάσει της πρόσθεσης XOR που υλοποίει ο LFSR. Οι τιμές των άλλων βαθμίδων προκύπτουν από ολίσθηση προς τα δεξιά των τιμών ολκών των βαθμίδων της προηγουμένης κατάστασης.

Παράδειγμά : LSFR τριών βαθμίδων (n=3)



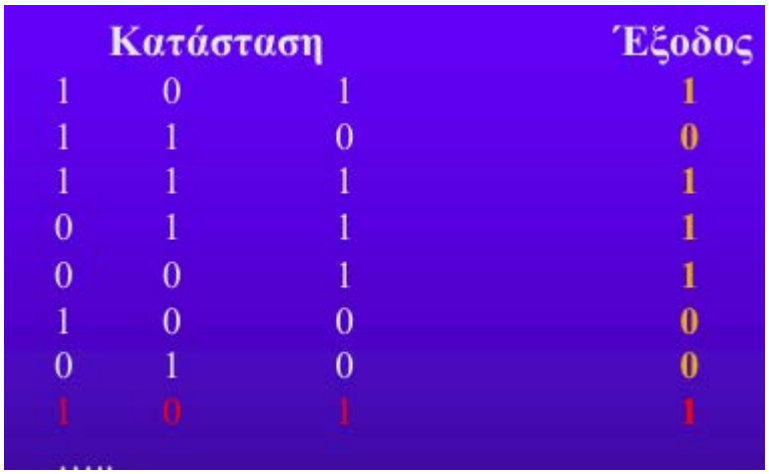
Εικόνα 4 Ένας απλός LSFR με τρεις βαθμίδες. Πηγή [3]

1. Έστω αρχική κατάσταση : 101
2. Η έξοδος του LFSR, για την κατάσταση 101, είναι 1 (το περιεχόμενο της δεξιότερης θέσης μνήμης).

Η επόμενη κατάσταση, μετά τον παλμό του ρολογιού, θα είναι 110

Σχηματική αναπαράσταση της λειτουργίας ενός LFSR

Αν θεωρήσουμε ότι η αρχική του κατάσταση είναι η 101, οι διαδοχικές καταστάσεις από τι οποίες διέρχεται είναι:



Εικόνα 5 Διαδοχικές καταστάσεις ενός LFSR. Πηγή [3]

Αφού συναντήσαμε πάλι την κατάσταση 101, θα έχουμε περιοδική επανάληψη των καταστάσεων. Άρα και περιοδική επανάληψη της ακολουθίας εξόδου Φ Παραγόμενη ακολουθία: k=10111001011100….

Ιδιότητες LFSRs

Ένα LFSR μήκους n(αριθμός βαθμίδων) μπορεί να διέλθει το πολύ από 2n -1 διαφορετικές καταστάσεις – Όλες οι πιθανές n-άδες, πλην της μηδενικής Άρα, μπορεί να παράγει ακολουθίες με μέγιστη δυνατή περίοδο 2n -1. Στη γενική περίπτωση, η ακολουθία εξόδου ενός LFSR εξαρτάται τόσο από την ανάδρασή του (ποιες βαθμίδες προστίθενται με XOR) όσο και από την αρχική του κατάσταση. LFSRs που επιτυγχάνουν τη μέγιστη δυνατή περίοδο ονομάζονται πρωταρχικοί (primitive).Οι ακολουθίες που παράγονται από τέτοιους καταχωρητές ονομάζονται ακολουθίες μέγιστου μήκους (maximal-length sequences ή msequences). Το αν ένας LFSR είναι πρωταρχικός, εξαρτάται αποκλειστικά από την ανάδρασή του και όχι από την αρχική του κατάσταση

6 Κρυπτογραφία Δημοσίου Κλειδιού

6.1 Δημόσιο – Ιδιωτικό κλειδί Κρυπτογράφησης

Φανταστείτε ότι διευθύνετε μια εταιρεία που ονομάζεται Super Trustworthy Boxes (STB). Η εταιρεία σας έχει αναπτύξει ένα πρωτότυπο είδος χρηματοκιβωτίου. Τα κανονικά χρηματοκιβώτια διαθέτουν μια κλειδαριά και ένα κλειδί. Τα STB διαθέτουν μια κλειδαριά και δύο κλειδιά. Τα δύο κλειδιά είναι κατασκευασμένα με τέτοιο τρόπο ώστε αν κλειδώσετε το κουτί με το ένα από τα δύο, μπορείτε να το ξεκλειδώσετε μόνο με το άλλο. Ποιο είναι το πλεονέκτημα του προϊόντος σας σε σύγκριση με τα παραδοσιακά χρηματοκιβώτια ;Αν κάποιος, ας την πούμε Άλις, θέλει να στείλει κάτι σε κάποιον άλλο, ας τον πούμε Μπομπ, μπορεί να το βάλει σε ένα παραδοσιακό χρηματοκιβώτιο και να το στείλει στον Μπομπ. Κανείς δεν μπορεί να σπάσει το κουτί στο δρόμο, και μόνο ο Μπομπ μπορεί να το ανοίξει, αρκεί να έχει ένα αντίγραφο του κλειδιού. Αυτό είναι ένα πρόβλημα επειδή δεν μπορείτε να στείλετε το κλειδί μαζί με το κουτί. Κάποιος στο δρόμο, ας την πούμε Εύα (για ωτακουστή), απλώς θα άνοιγε το κουτί με το κλειδί. Πρέπει να φύγετε για να δώσετε το κλειδί στον Μπομπ, ώστε να μην το πάρει κανείς άλλος. Μπείτε στα χρηματοκιβώτια του STB. Η Αλίκη μπορεί να στείλει το κουτί στον Μπομπ, μαζί με το δεύτερο κλειδί. Κρατάει το πρώτο κλειδί μυστικό. Ο Μπομπ βάζει το μήνυμά του στο κουτί, το κλειδώνει, και το στέλνει πίσω στην Αλίκη. Μόνο η Αλίκη έχει το πρώτο κλειδί, και ούτε το έχει μοιραστεί με κανέναν ούτε το έχει στείλει στον Μπομπ. Έτσι, μόνο η Αλίκη μπορεί να ξεκλειδώσει το κουτί και να ανακτήσει το μήνυμα του Μπομπ. Επιπλέον, τα χρηματοκιβώτια του STB επιτρέπουν στους πελάτες σας κάποια επιπλέον λειτουργικότητα. Η Αλίκη μπορεί να βάλει ένα μήνυμα στο κουτί και να το κλειδώσει χρησιμοποιώντας το πρώτο κλειδί. Στέλνει το κουτί στον Μπομπ μαζί με το δεύτερο κλειδί. Ο Μπομπ μπορεί να το ανοίξει γνωρίζοντας ότι μόνο κάποιος που έχει το πρώτο κλειδί θα μπορούσε να το είχε κλειδώσει. Αν γνωρίζει ότι το πρώτο κλειδί ανήκει στην Αλίκη, μπορεί να είναι σίγουρος ότι η Αλίκη, και όχι οποιοσδήποτε άλλος, ήταν ο αποστολέας του κουτιού.

Από τον αναλογικό στον ψηφιακό κόσμο, και συγκεκριμένα στην κρυπτογραφία. Όταν θέλετε να κρατήσετε ένα μήνυμα μυστικό, κρυπτογραφείτε το μήνυμά σας χρησιμοποιώντας ένα κλειδί, και ο παραλήπτης του μηνύματος χρησιμοποιεί ένα κλειδί για να αποκρυπτογραφήσει το μήνυμα. Σημειώστε ότι χρησιμοποιήσαμε την έκφραση «ένα κλειδί» και όχι «το κλειδί». Εάν τα δύο κλειδιά, για την κρυπτογράφηση και την αποκρυπτογράφηση, είναι πανομοιότυπα, τότε η κρυπτογράφηση είναι συμμετρική. Αλλά αυτό δεν είναι απαραίτητο. Μπορείτε να φανταστείτε ένα σχήμα, όπως αυτό που περιγράψαμε για το STB, όπου χρησιμοποιούνται διαφορετικά κλειδιά. Αυτή είναι η ασύμμετρη κρυπτογραφία. Λειτουργεί έχοντας δύο κλειδιά, ένα που χρησιμοποιείται για την κρυπτογράφηση και ένα που χρησιμοποιείται για την αποκρυπτογράφηση . Για να λειτουργήσει το σχήμα, ένα από τα κλειδιά είναι δημόσιο και το άλλο είναι ιδιωτικό. Η Αλίκη θα κρατάει πάντα μυστικό το ιδιωτικό της κλειδί, αλλά μπορεί να δώσει το δημόσιο κλειδί. Οποιοσδήποτε μπορεί να χρησιμοποιήσει το δημόσιο κλειδί της Αλίκης για να κρυπτογραφήσει ένα μήνυμα. Μόνο η Αλίκη, ωστόσο, μπορεί να αποκρυπτογραφήσει το μήνυμα επειδή μόνο αυτή έχει το αντίστοιχο ιδιωτικό κλειδί. Επειδή το ένα κλειδί είναι δημόσιο, ολόκληρη η προσέγγιση ονομάζεται κρυπτογραφία δημόσιου κλειδιού. Απλά, δεν χρειάζεται να το κάνουν καθόλου αυτό. Ο Μπομπ μπορεί να χρησιμοποιήσει το δημόσιο κλειδί της Αλίκης για να κρυπτογραφήσει μηνύματα προς την Αλίκη, τα οποία μπορεί να αποκρυπτογραφήσει χρησιμοποιώντας το ιδιωτικό της κλειδί. Η Αλίκη μπορεί να χρησιμοποιήσει το δημόσιο κλειδί του Μπομπ για να κρυπτογραφήσει μηνύματα προς τον Μπομπ, τα οποία μπορεί να αποκρυπτογραφήσει χρησιμοποιώντας το ιδιωτικό του κλειδί. Τα κλειδιά διατίθενται σε ζεύγη, επομένως μπορούμε να ονομάσουμε $P(A)$ το δημόσιο κλειδί του ζεύγους A και $S(A)$ το μυστικό κλειδί του ζεύγους A. Μόνο ένα ιδιωτικό κλειδί αντιστοιχεί σε ένα δημόσιο κλειδί, επομένως κάθε ζεύγος κλειδιών είναι μοναδικό. Εάν M είναι το αρχικό μήνυμα απλού κειμένου, τότε η λειτουργία κρυπτογράφησης του μηνύματος χρησιμοποιώντας το δημόσιο κλειδί $S\ (A)$ είναι

$$C\ =\ E_{P(A)}(M)$$

Το κρυπτογραφημένο μήνυμα που προκύπτει μπορεί να αποκρυπτογραφηθεί μόνο με το αντίστοιχο δημόσιο κλειδί:

Η διαδικασία κρυπτογράφησης προκύπτει άμεσα από τα παραπάνω.

1. Ο Μπομπ δημιουργεί ένα ζεύγος κλειδιών $B = (P(B), S(B))$.

2. Η Αλίκη λαμβάνει το δημόσιο κλειδί του Μπομπ $P(B)$. Αυτό μπορεί να συμβεί με διάφορους τρόπους. Για παράδειγμα, ο Μπομπ μπορεί να το δημοσιεύσει σε κάποιον δημόσιο διακομιστή ή να το στείλει στην Αλίκη απευθείας μέσω email.

3. Η Αλίκη κρυπτογραφεί το μήνυμά της χρησιμοποιώντας το $P(B)$:

$$M = D_{S(B)}(C).$$

4. Η Αλίκη στέλνει το C στον Μπομπ.

5. Ο Μπομπ αποκρυπτογραφεί το C χρησιμοποιώντας το μυστικό του κλειδί:

$$M = D_{S(B)}(C)$$

Το γεγονός ότι μόνο ένα μυστικό κλειδί αντιστοιχεί σε ένα δημόσιο κλειδί σημαίνει ότι, όπως η Αλίκη κλειδώνει το χρηματοκιβώτιο, μπορούμε επίσης να χρησιμοποιήσουμε το μυστικό, ιδιωτικό κλειδί για να κρυπτογραφήσουμε ένα μήνυμα:

$$C = E_{S(A)}(M)$$

Το κρυπτογραφημένο μήνυμα που προκύπτει μπορεί να αποκρυπτογραφηθεί μόνο με το αντίστοιχο δημόσιο κλειδί:

$$M = D_{P(A)}(C)$$

Γιατί να θέλει κάποιος να το κάνει αυτό; Επειδή το δημόσιο κλειδί αντιστοιχεί σε ένα μόνο ιδιωτικό κλειδί, ο Μπομπ μπορεί να είναι σίγουρος ότι το μήνυμα που έλαβε έχει κρυπτογραφηθεί από τον κάτοχο του ιδιωτικού κλειδιού. Έτσι, αν γνωρίζει ότι ο κάτοχος του κλειδιού είναι η Αλίκη, τότε γνωρίζει ότι το μήνυμα που έλαβε προέρχεται από την Αλίκη.Έτσι, η κρυπτογράφηση με το ιδιωτικό κλειδί και η αποκρυπτογράφηση με το δημόσιο κλειδί είναι ένας τρόπος για να αποδειχθεί η προέλευση του μηνύματος. Είναι ισοδύναμο με αυτό που κάνουμε όταν υπογράφουμε ένα φύλλο χαρτιού. Επειδή η υπογραφή μας είναι μοναδική, οποιοσδήποτε/οποιαδήποτε γνωρίζει την υπογραφή μας μπορεί να επαληθεύσει ότι έχουμε υπογράψει το φύλλο χαρτιού. Η κρυπτογράφηση με το ιδιωτικό κλειδί ονομάζεται επομένως υπογραφή του μηνύματος και ένα μήνυμα κρυπτογραφημένο με ιδιωτικό κλειδί είναι μια ψηφιακή υπογραφή. Συνολικά, η διαδικασία υπογραφής προχωρά με τρόπο ανάλογο με την κρυπτογράφηση.

1. Η Αλίκη δημιουργεί ένα ζεύγος κλειδιών $A = (P(A), S(A))$. Μοιράζει $P(A)$ στον Μπομπ με κάθε βολικό τρόπο.

2. Η Αλίκη υπογράφει το μήνυμά της M με το μυστικό της κλειδί $S\ (A)$:

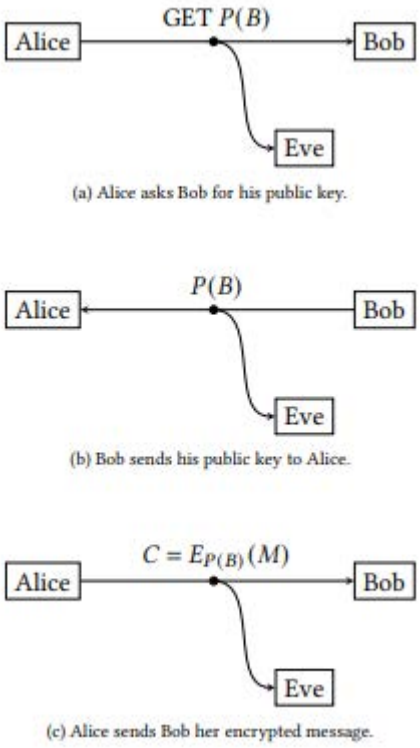
C = $E_{S(A)}(M)$

3. Η Αλίκη στέλνει (M, C) στον Μπομπ.

4. Ο Bob επαληθεύει το C χρησιμοποιώντας το δημόσιο κλειδί της Alice $P(A)$:

$$M = D_{P(A)}\ (C)$$

Συνοψίζοντας, στην κρυπτογραφία δημόσιου κλειδιού, κάθε συμμετέχων έχει ένα σύνολο δύο κλειδιών αντί για ένα μόνο. Το ένα κλειδί μπορεί να κοινοποιηθεί σε οποιονδήποτε ή να τοποθετηθεί σε ένα δημόσιο αποθετήριο. Το άλλο κλειδί πρέπει να κρατηθεί μυστικό. Οποιοσδήποτε μπορεί να χρησιμοποιήσει το δημόσιο κλειδί για να κρυπτογραφήσει ένα μήνυμα, το οποίο μπορεί να αποκρυπτογραφηθεί μόνο χρησιμοποιώντας το ιδιωτικό κλειδί. Εάν Ένας ωτακουστής, ας πούμε η Εύα, ακούει τις επικοινωνίες μεταξύ της Αλίκης και του Μπομπ, μπορεί να λάβει μόνο τα δημόσια κλειδιά και τα κρυπτογραφημένα μηνύματα, επομένως δεν μπορεί να αποκρυπτογραφήσει το υποκείμενο απλό κείμενο, όπως μπορείτε να δείτε στο σχήμα .Επιπλέον, μπορούμε να χρησιμοποιήσουμε το ιδιωτικό κλειδί για να κρυπτογραφήσουμε ένα μήνυμα, δηλαδή να υπογράψουμε ένα μήνυμα. Οποιοσδήποτε μπορεί να λάβει το δημόσιο κλειδί και να επιβεβαιώσει ότι το μήνυμα έχει υπογραφεί χρησιμοποιώντας το ιδιωτικό κλειδί. Εάν γνωρίζει τον κάτοχο του ιδιωτικού κλειδιού, γνωρίζει ότι ο κάτοχος το έχει υπογράψει.



Εικόνα 6 Απόκτηση δημοσίου κλειδιού του Bob και αποστολή κρυπτογραφημένου μηνύματος. Πηγή [1]

Έχοντας τις δύο λειτουργίες, κρυπτογράφηση και υπογραφή, μπορείτε να τις συνδυάσετε, ώστε να μπορείτε να κρυπτογραφήσετε ένα μήνυμα και να το υπογράψετε. Αρχικά, υπογράφετε το μήνυμα, χρησιμοποιώντας το ιδιωτικό σας κλειδί. Στη συνέχεια, κρυπτογραφείτε το μήνυμα και την υπογραφή με το δημόσιο κλειδί του παραλήπτη σας. Ο παραλήπτης πρώτα αποκρυπτογραφεί το κρυπτογραφημένο μήνυμα, χρησιμοποιώντας το αντίστοιχο ιδιωτικό κλειδί, λαμβάνοντας το μήνυμα απλού κειμένου και την υπογραφή. Στη συνέχεια, ο παραλήπτης αποκρυπτογραφεί το μήνυμα υπογραφής με το δημόσιο κλειδί σας. κλειδί και επαληθεύστε ότι ταιριάζει με το αποκρυπτογραφημένο μήνυμα απλού κειμένου. Η Αλίκη και ο Μπομπ θα εκτελέσουν τα ακόλουθα βήματα:

1. Η Αλίκη δημιουργεί ένα ζεύγος κλειδιών $A = (P(A), S(A))$. Διανέμει το $P(A)$ στον Μπομπ με οποιονδήποτε βολικό τρόπο.

2. Ο Μπομπ δημιουργεί ένα ζεύγος κλειδιών $B = (P(B), S(B))$. Διανέμει το $P(B)$ στην Αλίκη με οποιονδήποτε βολικό τρόπο.

3. Η Αλίκη υπογράφει το μήνυμά της M με το μυστικό της κλειδί P (B):

$$C1 \ = \ E_{S(A)} \ \ (M)$$

4. Η Αλίκη κρυπτογραφεί το μήνυμά της και την υπογραφή που υπολόγισε, δηλαδή, (M,C1), χρησιμοποιώντας το δημόσιο κλειδί του Μπομπ P (B):

$$C2 \ = \ \ (E_{P(B)} \ (M, C1).$$

5. Η Αλίκη στέλνει το C2 στον Μπομπ.

6. Ο Μπομπ αποκρυπτογραφεί το C2 χρησιμοποιώντας το μυστικό του κλειδί:

$$(M, C1) \ = \ \ D_{S(B)}(C2).$$

7. Ο Μπομπ επαληθεύει το C1 χρησιμοποιώντας το δημόσιο κλειδί της Αλίκης P(A):

$$M \ = \ D_{P(A)} \ (C1).$$

Για να ολοκληρώσουμε την εικόνα, πρέπει τώρα να δούμε πώς πραγματικά λαμβάνουν χώρα αυτές οι κρυπτογραφήσεις, οι αποκρυπτογραφήσεις και οι υπογραφές. Ακολουθούν απλές διαδικασίες που βασίζονται σε αποτελέσματα που λαμβάνονται από τη θεωρία αριθμών.

6.2 Μέθοδος Κρυπτογράφησης RSA

Ένα από τα πρώτα κρυπτοσύστημα δημόσιου κλειδιού που χρησιμοποιήθηκαν στην πράξη και σε ευρεία χρήση σήμερα είναι το κρυπτοσύστημα RSA. Το όνομα προέρχεται από τους εφευρέτες του, Ron Rivest, Adi Shamir και Leonard Adleman, οι οποίοι δημοσίευσαν για πρώτη φορά το σύστημα το 1977, αν και είχε ανακαλυφθεί ανεξάρτητα νωρίτερα, το 1973 από τον Cliord Cocks που εργαζόταν για την Κεντρική Υπηρεσία Γενικών Επικοινωνιών της Κυβέρνησης του Ηνωμένου Βασιλείου (GCHQ). Το έργο του Cock ήταν, ωστόσο, απόρρητο μέχρι το 1997. Είδαμε στην ενότητα 4.4 ότι το ίδιο συνέβη και με τον μηχανισμό ανταλλαγής κλειδιών Die-Hellman, ο οποίος είχε ανακαλυφθεί από τον Malcolm Williamson στο GCHQ πριν από τον Die και τον Hellman. Στην πραγματικότητα, η ανακάλυψη του Cocks ήταν πρώτη. Ο Williamson ήταν φίλος του και έμαθε την ανακάλυψή του, κάτι που τον ώθησε να προχωρήσει και να αναπτύξει την ανταλλαγή κλειδιών Die-Hellman. Το RSA παρέχει μια μέθοδο για τη δημιουργία ενός ζεύγους δημόσιου-ιδιωτικού κλειδιού. Πριν προχωρήσουμε στα βήματα της μεθόδου, απαιτείται μια αριθμητική αναφορά. Εάν δύο αριθμοί είναι πρώτοι μεταξύ τους, δηλαδή, αν δεν έχουν κοινούς διαιρέτες εκτός από το 1, τότε ονομάζονται σχετικά πρώτοι ή πρώτοι. Με αυτόν τον ορισμό, ο RSA περιλαμβάνει την ακόλουθη σειρά βημάτων

1. Choose two large prime numbers, say p and q for which $p \neq q$.
2. Calculate their product, $n = pq$.
3. Choose an integer e that is relatively prime to $(p-1)(q-1)$.
4. Find a number d, $1 \leq d < (p-1)(q-1)$ such that:

$$(e \cdot d) \bmod [(p-1)(q-1)] = 1$$

5. The public-private key pair is $A = (P(A), S(A)) = ((e,n),(d,n))$.
6. The tuple $P(A) = (e,n)$ is your RSA public key.
7. The tuple $S(A) = (d,n)$ is your RSA private key.

Εικόνα 7 Τα βήματα του RSA. Πηγή [1]

Πρέπει να εξηγήσουμε πώς επιλέγουμε στο βήμα 4. Θα δούμε πώς, αλλά πρώτα ας δούμε πώς λειτουργεί το RSA στην πράξη. Είπαμε ότι οι δύο αριθμοί και πρέπει να είναι μεγάλοι. Όσο μεγαλύτεροι, τόσο το καλύτερο, αλλά όσο μεγαλύτεροι, τόσο περισσότερη υπολογιστική προσπάθεια απαιτείται για την κρυπτογράφηση και την αποκρυπτογράφηση. Δεν υπάρχει λόγος, ωστόσο, για τον οποίο δεν πρέπει να είναι 2048 bit ο καθένας,

και το 4096 είναι καλύτερο .Δεν υπάρχει τέτοια απαίτηση μεγέθους για το . μπορεί ακόμη και να είναι ίσο με 3. Μια δημοφιλής επιλογή είναι $e= 2^{16} + 1= 65537$. αυτός ο αριθμός έχει κάποιες καλές ιδιότητες που καθιστούν πιο απίθανο να σπάσουν μηνύματα κρυπτογραφημένα με RSA χρησιμοποιώντας αυτό. Τόσο η κρυπτογράφηση όσο και η αποκρυπτογράφηση χρησιμοποιούν την ίδια συνάρτηση: $f(m,k,n) = m^k \bmod n$

Τα ορίσματα που διαβιβάζονται στη συνάρτηση διαφέρουν ως προς την κρυπτογράφηση και την αποκρυπτογράφηση. Έτσι, κατά την κρυπτογράφηση, το μήνυμα απλού κειμένου είναι ίσο με το από το δημόσιο κλειδί του συμμετέχοντα. Στην αποκρυπτογράφηση, το κρυπτογραφημένο κείμενο είναι ίσο με το από το ιδιωτικό κλειδί του συμμετέχοντα. Με άλλα λόγια, με αυτό το σχήμα, η κρυπτογράφηση ενός μηνύματος γίνεται με τον υπολογισμό του κρυπτογραφημένου κειμένου χρησιμοποιώντας: $C\ E_{p(A)}(M) = M^E \bmod n$

Η αποκρυπτογράφηση γίνεται με αυτό τον τρόπο : $M = D_{S(A)}(C) = C^d\ mod\ n$

Signing a message is

$$C = E_{S(A)}(M) = M^d \bmod n$$

and the verification of the signature is

$$D_{P(A)}(C) = C^e \bmod n$$

Because for any integers u, v, it holds:

$$(u \bmod n)(v \bmod n) \bmod n = uv \bmod n$$

decryption is:

$$M = D_{S(A)}(C) = C^d \bmod n = (M^e \bmod n)^d \bmod n = M^{ed} \bmod n$$

and similarly signature verification is:

$$D_{P(A)}(C) = C^e \bmod n = (M^d \bmod n)^e \bmod n = M^{de} \bmod n$$

Εικόνα 8 Υπογραφή μηνύματος και αυθεντικοποίηση

Μπορείτε να δείτε ότι στην πραγματικότητα υπολογίζουν το ίδιο πράγμα, όπως θα έπρεπε. Όταν αποκρυπτογραφείτε ένα κρυπτογραφημένο μήνυμα, λαμβάνετε πίσω το απλό κείμενο. Όταν επαληθεύετε μια υπογραφή, λαμβάνετε ξανά πίσω το αρχικό μήνυμα.

Το μόνο πράγμα που αλλάζει είναι η ακολουθία εφαρμογής των κλειδιών. Στην κρυπτογράφηση και την αποκρυπτογράφηση, πρώτα εφαρμόζεται το δημόσιο κλειδί και στη συνέχεια το ιδιωτικό κλειδί. Στην υπογραφή και την επαλήθευση, πρώτα εφαρμόζεται το ιδιωτικό κλειδί και στη συνέχεια το δημόσιο κλειδί.1 και Ας υποθέσουμε τώρα ότι η Αλίκη θέλει να κρυπτογραφήσει ένα μήνυμα. Αν τα και είναι 2048 bit το καθένα, τότε

το 1 πρέπει να είναι 2047 bit το καθένα, και πρέπει να είναι 2047 = 4094. Επομένως, το

πρέπει να έχει μήκος μικρότερο από 4094 bit. Αν δεν είναι, τότε η Αλίκη πρέπει να χωριστεί σε κομμάτια μικρότερα από 4094 bit το καθένα. Η Αλίκη γνωρίζει το δημόσιο κλειδί του Μπομπ =. Για να στείλει στον Μπομπ, η Αλίκη θα υπολογίσει όπως παραπάνω και ο Μπομπ θα το λάβει και θα το αποκρυπτογραφήσει.

Για να λειτουργήσουν οι υπολογισμοί, το μήνυμα απλού κειμένου πρέπει να είναι ένας ακέραιος αριθμός. Αυτό δεν περιορίζει τη χρησιμότητα του RSA. Ένα μήνυμα κειμένου σε έναν υπολογιστή είναι στην πραγματικότητα μια σειρά από bits, που κωδικοποιούν κείμενο μέσω μιας κατάλληλης κωδικοποίησης, όπως το ASCII. Στη συνέχεια, χωρίζουμε το κείμενο σε κομμάτια bits του απαιτούμενου μήκους. Η δεκαδική τιμή ενός κομματιού είναι ο αριθμός .Μπορείτε να επαληθεύσετε ότι το RSA λειτουργεί πραγματικά με ένα παράδειγμα. Ο Μπομπ θέλει να κρυπτογραφήσει το μήνυμα = 314 και να το στείλει στην Αλίκη.

1. Alice takes $p = 17$ and $q = 23$.
2. Alice calculates $n = pq = 17 \times 23 = 391$.

3. She chooses $e = 3$. You can check that e is relatively prime to
$$(p-1)(q-1) = (17-1)(23-1) = 16 \times 22 = 352$$
4. Alice finds $d = 235$, which satisfies $1 \leq d < 352$ and
$(e \cdot d) \bmod [(p-1)(q-1)] = 1$,
or equivalently $(3 \times 235) \bmod 352 = 1$.
5. The public-private key pair is $A = ((3, 391), (235, 391))$.
The tuple $P(A) = (3, 391)$ is the RSA public key.
The tuple $S(A) = (235, 391)$ is the RSA private key.
6. Bob gets $P(A)$ from Alice and encrypts M with the operation:
$$C = M^e \bmod n = 314^3 \bmod 391 = 155$$
7. Bob sends C to Alice.
8. Alice decrypts C with the operation:
$$C^d \bmod n = 155^{235} \bmod 391 = 314$$

Εικόνα 9 Παράδειγμα λειτουργίας RSA

που είναι το αρχικό μήνυμα απλού κειμένου.

Αν μια έκφραση όπως 155^{235} mod 391 = 314 φαίνεται απαγορευτική, μπορείτε να δείτε πώς ακριβώς μπορεί να υπολογιστεί αποτελεσματικά στον πίνακα 4.10 της ενότητας 4.5. Αυτό είναι νέο, αλλά το βήμα 4 παραμένει μυστήριο. Για να καταλήξουμε στο πώς βρίσκεται, χρειαζόμαστε κάποιο υπόβαθρο από τη θεωρία αριθμών. Γνωρίζουμε ότι το πολλαπλασιαστικό αντίστροφο, ή αντίστροφο, ενός αριθμού , είναι ο αριθμός 1/x ή x^{-1} τέτοιος ώστε xx^{-1} = 1. Στην αρθρομετρία, ο μηχανιστικός πολλαπλασιαστικός αντίστροφος ενός ακεραίου modulo n (n>0) είναι ο ακέραιος x^{-1}, 1< x^{-1} < n-1 έτσι ώστε xx^{-1} mod n = 1. Αυτό είναι ισοδύναμο με xx^{-1}= kn + 1, or 1 = xx^{-1} – kn για κάποιο ακέραιο k. Οπότε όταν στο βήμα 4 βρίσκουμε τον αριθμό d, 1< d < (p-1)(q-1) έτσι ώστε (e*d) mod [(p-1)(q-1)] = 1 , βρίσκουμε πραγματικά το modulo πολλαπλασιαστικός αντίστροφος του modulo (p -1) (q-1) για το οποίο κρατά: 1 = ed + k(p-1) (q-1)

για κάποια ακέραια k . Αυτό προκύπτει άμεσα από 1 = xx^{-1} -kn , με διαφορετικό πρόσημο πριν , το οποίο δεν έχει σημασία γιατί είναι απλώς ένα ακέραιο, οπότε μπορούμε να πάρουμε το αρνητικό του. Στον τομέα των πραγματικών αριθμών, μπορούμε πάντα να βρούμε τον αντίστροφο οποιουδήποτε αριθμού x =! 0. Αλλά πώς βρίσκουμε τον αρνητικό με Modulo; Επιπλέον, υπάρχει πάντα ένας αρνητικός με Modulo; Ο αρμονικός πολλαπλασιαστικός αντίστροφος modulo υπάρχει αν και μόνο αν είναι αμοιβαίως πρώτοι. Γι' αυτόν τον λόγο όταν επιλέξαμε, επιμείναμε ότι θα πρέπει να είναι πρώτοι με 1. Με κάτι τέτοιο, γνωρίζουμε ότι υπάρχει μια με τις ιδιότητες που θέλουμε. Πρέπει να δούμε πώς μπορούμε να την βρούμε. Ο μεγαλύτερος κοινός διαιρέτης (gcd) δύο ακέραιων αριθμών είναι ο μεγαλύτερος ακέραιος που τους διαιρεί και τους δύο. Για να βρούμε τον gcd δύο θετικών ακέραιων αριθμών, χρησιμοποιούμε έναν αρχαίο αλγόριθμο, που βρίσκεται στα Στοιχεία του Ευκλείδη, που υποδεικνύεται στον αλγόριθμο 5.1. Ο αλγόριθμος του Ευκλείδη προκύπτει από το γεγονός ότι ο gcd δύο ακεραίων a>0 και b> 0 είναι ο gcd του b και το υπόλοιπο της διαίρεσης του δια του b , εκτός αν διαιρεί το a , οπότε κατά τον ορισμό, ο gcd είναι ο ίδιος. Ο αλγόριθμος 5.1 λειτουργεί κάνοντας αναδρομικές κλήσεις, αντικαθιστώντας και το mod με και σε κάθε κλήση (γραμμή 4). Η αναδρομή σταματά όταν b= 0 (γραμμές 1–2), όπου είναι το αποτέλεσμα του mod της προηγούμενης αναδρομικής κλήσης. Η απόδειξη ότι και το υπόλοιπο έχουν τον ίδιο μεγαλύτερο κοινό διαιρέτη βασίζεται στις βασικές ιδιότητες της διαίρεσης. Αν είναι κοινός διαιρέτης του a και του b, τότε $a = k1d$ και το $k2d$ για μερικούς θετικούς ακέραιους k1,k2. Επίσης αν $r = a \bmod b$, έχουμε $r = a - k3b$ για κάποιους θετικούς ακέραιους k3. Αντικαθιστώντας τις τιμές για a και b , παίρνουμε $r = k1d - k3k2d = (k1 - k3k2)d$, οπότε το d διαιρεί το r που σημαίνει ότι όλοι οι διαιρέτες του και είναι επίσης διαιρέτες του b και $a \bmod b$. Αντίθετα, αν είναι κοινός διαιρέτης και των δύο b και $r = a \bmod b$, τότε $b = z1d$ και $r = z2d$ αρκετοί θετικοί ακέραιοι z1,z2. Την ίδια ώρα έχουμε το a = z3b, για τους για κάποια θετική ακέραιους z3. Αντικαθιστώντας τις τιμές για το r και το b εμείς παίρνουμε a = z3z1d + z2d, οπότε το d διαιρεί το a, που σημαίνει ότι όλοι οι διαιρέτες του b και a mod b είναι επίσης διαιρέτες του a και του b. Άρα a και b έχουν ακριβώς τους ιδίους διαιρέτες μαζί με το b και a mod b, και τότε πρέπει να έχουν τον ίδιο μεγαλύτερο κοινό διαιρέτη. Ο πίνακας 5.1 δείχνει ένα παράδειγμα της λειτουργίας του αλγορίθμου του Ευκλείδη για να προσδιορίσει το μέγιστο κοινό διαιρέτη (gcd) των 160 και 144, ο οποίος είναι 16. Κάθε σειρά του πίνακα δείχνει τι συμβαίνει σε κάθε αναδρομική κλήση. Μπορεί να αποδειχθεί ότι, αν a > b

Table 5.1

Example run of Euclid's algorithm.

a	b	$a \bmod b$
160	144	16
144	16	0
16	0	

Εικόνα 10 Παράδειγμα αλγορίθμου του Ευκλείδη

ο αριθμός των αναδρομικών κλήσεων που απαιτεί ο αλγόριθμος είναιο O(lgb). Αν τότε b>a είναι O(lga). Αυτό συμβαίνει επειδή η πρώτη κλήση απλά αναστρέφει και: παρακολουθεί την εκτέλεση του αλγορίθμου για να προσδιορίσει το gdc των 144 και 160 για να το δει. Μια γενίκευση του αλγορίθμου του Ευκλείδη, ο επεκτεινόμενος αλγόριθμος του Ευκλείδη, μπορεί να χρησιμοποιηθεί για να βρει, εκτός από το gcd δύο θετικών ακεραίων, τον τρόπο με τον οποίο μπορούμε να βρούμε το gcd αυτών των αριθμών συνδυάζοντάς τους με πολλαπλασιασμό και πρόσθεση. Συγκεκριμένα, αν r = gcd(a,b) τότε μπορούμε να βρούμε ακέραιους και τέτοιους ώστε:

$$r = gcd(a, b) = xa + yb$$

Ο επεκταμένος αλγόριθμος του Ευκλείδη είναι ο αλγόριθμος 5.2 και είναι το κλειδί για την εύρεση των αρνητικών πολλαπλασιαστικών αντιστρόφων. Ο αριθμός των αναδρομικών κλήσεων είναι πάλι O(lgb), Αν a > b, η O(lga) , εάν a< b . Μπορεί να αποδειχθεί ότι:

$$|x| <= b/gcd(a, b) \text{ και } |y| <= a/gcd(a, b)$$

όπου η ισότητα ισχύει όταν είναι πολλαπλάσιο του b , ή b είναι πολλαπλάσιο του a. Αν και είναι σχετικά πρώτοι, τότε έχουμε gcd(a,b) = 1 Αυτό σημαίνει ότι μπορούμε να χρησιμοποιήσουμε τον επεκταμένο αλγόριθμο του Ευκλείδη για να βρούμε δύο ακεραίους και έτσι ώστε:

$$1 = xa + yb \ με \ |x| < b \ και \ |y| < a$$

Αυτό με τη σειρά του σημαίνει ότι αν 0 < x < b , όταν x είναι ο πολλαπλασιαστικός αντίστροφος του a modulo b. Εάν x < 0 τότε προσθέτοντας και διαγράφοντας ab η παραπάνω φόρμουλα εμείς $get\ 1 = xa + ab + yb - ab = (x + b)a + (y - a)\,b$, με 0 < x+b < b. Εάν x < 0 τότε ο πολλαπλασιαστικός αντίστροφος του υπολοίπου b is x + b.

Ο αλγόριθμος 5.3 δείχνει τη διαδικασία εύρεσης του αρνητικού πολλαπλασιαστικού αντίστροφου σε αλγοριθμική μορφή. Μπορείτε να παρατηρήσετε ότι αν ο εκτεταμένος αλγόριθμος του Ευκλείδη επιστρέφει ένα gcd διαφορετικό από το 1, τότε ο αρνητικός πολλαπλασιαστικός αντίστροφος δεν υπάρχει, οπότε επιστρέφουμε το μηδέν, το οποίο είναι μια άκυρη τιμή. Ένα παράδειγμα εκτέλεσης του εκτεταμένου αλγόριθμου του Ευκλείδη για τους αριθμούς 3 και 352, οι οποίοι είναι σχετικά πρώτοι, βρίσκεται στο σχήμα 5.2. Για να διαβάσετε αυτόν τον πίνακα, θα πρέπει να...

Algorithm 5.2: Extended Euclid's algorithm.

```
ExtendedEuclid(a, b) → (r, x, y)
    Input: a, b, positive integers
    Output: r, x, y, such that r = gcd(a, b) = xa + yb
1   if b = 0 then
2       return (a, 1, 0)
3   else
4       (r, x, y) = ExtendedEuclid(b, a mod b)
5       return (r, y, x − ⌊a/b⌋ · y)
```

a	b	$a \bmod b$	$(r, y, x - \lfloor a/b \rfloor \cdot y)$
3	352	3	$(1, -117, 1 - \lfloor 3/352 \rfloor \times (-117)) = (1, -117, 1)$
352	3	1	$(1, 1, 0 - \lfloor 352/3 \rfloor \times 1) = (1, 1, -117)$
3	1	0	$(1, 0, 1 - \lfloor 3/1 \rfloor \times 0) = (1, 0, 1)$
1	0		$(1, 1, 0)$

Εικόνα 11 Ο εκτεταμένος αλγόριθμος του Ευκλείδη. Πηγή [1]

πρέπει να διαβάσετε τις στήλες, και να τροποποιήσετε από την κορυφή προς τα κάτω, και έπειτα την τελευταία στήλη από το κάτω προς την κορυφή, καθώς έτσι κατασκευάζεται το τριάδα (r,x,y)κατά την επιστροφή κάθε αναδρομικής κλήσης. Στο τέλος του αλγορίθμου έχουμε: $1 = 3 * (-117) + 352 * 1$

Συνοπτικά, για να βρούμε τον αριθμό d από το μυστικό κλειδί RSA πρέπει να βρούμε τον πολλαπλασιαστικό αντίστροφο του modulo (p-1) (q-1) του e. Για να το κάνουμε αυτό, χρησιμοποιούμε το εκτενής αλγόριθμος του Ευκλείδη με είσοδο και (p-1)(q-1). Η

έξοδος είναι μια τριπλετα (1,x,y). An x >0, x είναι ο αριθμός που ψάχνουμε. Εάν x<0 , x + (p – 1)(q -1) είναι ο αριθμός που ψάχνουμε. Έχουμε τώρα εξηγήσει πώς υλοποιούνται τα διαφορετικά βήματα του RSA; Η ερώτηση που παραμένει είναι γιατί το RSA λειτουργεί καταρχάς. Χρειαζόμαστε κάποια περισσότερη αριθμητική θεωρία για αυτό. . Για κάθε θετικό ακέραιο, ο αριθμός των ακεραίων με 1< k < = n έτσι ώστε k και n είναι ο αριθμός που είναι πρώτος προς το 1, είναι η αξία μιας συνάρτησης, που ονομάζεται συνάρτηση του Euler ή λειτουργία φ του Euler και συμβολίζεται ως φ(n). Αν είναι πρώτος αριθμός, τότε είναι σχετικά πρώτος με όλους τους μικρότερους θετικούς φυσικούς αριθμούς, οπότε για τον πρώτο έχουμε (φn) = n-1. Στον RSA χρησιμοποιούμε p-1= φ(ρ) και q-1 =φ(q). Όταν δυο αριθμοί p και q είναι σχετικά πρώτοι , φ(pq) = φ(ρ)φ(q) = (p-1)(q-1).

Και οι δύο είναι πρώτοι αριθμοί και επομένως επίσης συμπληρωματικοί, πράγμα που σημαίνει ότι στο βήμα 3 επιλέγουμε έναν ακέραιο αριθμό που είναι σχετικά πρώτος προς φ(pq) = φ(n). Στην πραγματικότητα, θα μπορούσαμε να είχαμε υποκαταστήσει φ(n) για το πιο ογκώδες (p-1)(q-1) στην παρουσίαση μας . Η συνάρτηση φ του Euler εμφανίζεται στο θεώρημα του Euler, το οποίο δηλώνει ότι για κάθε θετικό ακέραιο, και για οποιοδήποτε 0 < α < n, $a^{\varphi(n)}$ mod n = 1. Ένα ειδικό περιστατικό του θερίσματος του Euler προκύπτει όταν είναι πρώτος. Αν χρησιμοποιούμε το p να αντιπροσωπεύει τέτοια n, έχουμε a^{p-1}mod p = 1. Αυτό ονομάζεται το Μικρό Θεώρημα του Φερμά, για να διακριθεί από το άλλο Θεώρημα του Φερμά, γνωστό ως το Τελευταίο Θεώρημα του Φερμά, το οποίο βασάνισε τους μαθηματικούς για αιώνες. Επιστρέφοντας στο RSA, για να δείξουμε ότι λειτουργεί, πρέπει να επαληθεύσουμε ότι πράγματι το αποτέλεσμα της αποκρυπτογράφησης είναι το αρχικό μήνυμα. Δεδομένου ότι η υπογραφή και η επαλήθευση είναι το ίδιο με την κρυπτογράφηση και την αποκρυπτογράφηση, χρειάζεται μόνο να αποδείξουμε την κρυπτογράφηση και την αποκρυπτογράφηση. Επειδή έχουμε

$D_{S(A)}$ (C) = C^d mod n = ($M^e mod\ n^d$) = M^{ed} mod n

πρέπει να δείξουμε ότι:

M^{ed} mod n = M mod n = M

Κατ' αρχάς, εξέτασε την περίπτωση που M = 0 .Τότε έχουμε

M^{ed} mod n = 0 mod n = 0 mod p = 0 = M

είμαστε έτοιμοι; έχουμε ακόμα να δείξουμε ότι κρατάει εάν M !=0. Εάν M != 0 και το p είναι πρώτος, τότε από το Θεώρημα του Φερμά MP^{-1} mod p = 1. Να θυμάστε ότι:

$$1 = ed + k\,(p-1)\,(q-1)$$

για κάποιο ακέραιο , ή ισοδύναμα:

$$ed = 1+k'\,(p-1)\,(q-1)$$

με την χρήση k′= -k . Οπότε παίρνουμε

M^{ed} mod p = $M^{1+k'(p-1)(q-1)}$mod p

= $MM^{1+k'(n-1)(q-1)}$mod p

= [(M mod p) ((M^{P-1})$^{k'(q-1)}$ mod p)] mod p

= [(M mod p) ((M^{P-1} mod p) $^{k'(q-1)}$ mod p)] mod p

= [(M mod p) ((1 mod p) $^{k'(q-1)}$ mod p)] mod p

= [(M mod p) (1 $^{k'(q-1)}$ mod p)] mod p

= M mod p

Με τον ακριβώς ίδιο τρόπο, χρησιμοποιώντας το γεγονός ότι προκύπτει από το Θεώρημα του Φερμά. M^{q-1}mod q = 1 έχουμε:

M^{ed} mod q = M mod q

Γι' αυτό δουλέψαμε με το προϊόν (p-1)(q-1): χρησιμοποιούμε το Θεώρημα του Φερμά με p-1 και q-1. Τώρα από ένα άλλο θεώρημα της αριθμητικής θεωρίας, το Θεώρημα των Κινεζικών Υπολοίπων, που δημοσιεύθηκε από τον Κινέζο μαθηματικό Σουν Τζου κάπου μεταξύ του 3ου και 5ου αιώνα μ.Χ., δεδομένου ότι έχουμε βρει:

M^{ed} mod p = M mod p

Και : M^{ed} mod p = M mod q

Παίρνουμε : M^{ed} mod p = M mod pq

Αλλά pq = n, έτσι :

M^{ed} mod n = M mod n = M

αυτό είναι που θέλαμε να δείξουμε.

Η RSA λύνει το πρόβλημα της κατανομής κλειδιών και έχει αναλυθεί εδώ και χρόνια. Η ασφάλειά της στηρίζεται στη δυσκολία της παραγοντοποίησης με πρώτους αριθμούς. Ένας παράγοντας ενός αριθμού είναι ένας ακέραιος αριθμός που διαιρεί τον αριθμό χωρίς να αφήνει υπόλοιπο. Ένας πρώτος παράγοντας είναι ένας παράγοντας που είναι επίσης πρώτος. Η παραγοντοποίηση με πρώτους αριθμούς είναι η εύρεση των πρώτων παραγόντων ενός αριθμού και η έκφρασή τους ως γινόμενο αυτών. Οι τρέχοντες γνωστοί αλγόριθμοι για την επίλυση του προβλήματος είναι υπολογιστικά πολύ απαιτητικοί, επομένως χρειάζεται πολύς χρόνος για να παραγοντοποιηθεί ένας μεγάλος ακέραιος αριθμός. Στην περίπτωσή μας, δεδομένου του , είναι πολύ δύσκολο να βρεθούν οι αριθμοί και . Εάν είχατε μια αποτελεσματική μέθοδο παραγοντοποίησης με πρώτους αριθμούς στη διάθεσή σας, τότε θα τη χρησιμοποιούσατε απλώς για να βρείτε τους αριθμούς και . δεδομένου ότι το είναι δημόσιο, θα υπολογίζατε μόνοι σας όπως στο βήμα 4 του RSA, και θα είχατε το ιδιωτικό κλειδί. Φυσικά, κάποιος θα μπορούσε απλώς να αρχίσει να ελέγχει όλους τους θετικούς ακέραιους αριθμούς μικρότερους από ελπίζοντας να βρει τους παράγοντες του. Γι' αυτόν τον λόγο είναι σημαντικό τα και να είναι μεγάλα. Όπως και στο Die-Hellman, δεν ψάχνουμε τυφλά για πρώτους αριθμούς. Μπορούμε να χρησιμοποιήσουμε αλγόριθμους για να βρούμε τους πρώτους αριθμούς που χρειαζόμαστε. Η ενότητα 16.4 δείχνει μια αποτελεσματική μέθοδο εύρεσης πρώτων αριθμών.

Το RSA είναι δυνητικά ευάλωτο στις εξελίξεις στον κβαντικό υπολογισμό, καθώς είναι δυνατό να πραγματοποιηθεί παραγοντοποίηση σε πολυώνυμο χρόνο σε έναν κβαντικό υπολογιστή. Προς το παρόν, όμως, το RSA είναι ασφαλές.

Εφόσον το RSA παραμένει ασφαλές και λύνει το πρόβλημα κατανομής κλειδιών, μπορεί να αναρωτηθείτε γιατί εξακολουθούμε να χρησιμοποιούμε συμμετρική κρυπτογραφία. Θα μπορούσατε να κρυπτογραφήσετε οποιοδήποτε μήνυμα χρησιμοποιώντας RSA ούτως ή άλλως. Η απάντηση είναι ότι το RSA είναι πολύ πιο αργό από τη συμμετρική κρυπτογραφία όπως το AES. Έτσι, συνήθως το RSA χρησιμοποιείται με υβριδικό τρόπο. Σε ένα τέτοιο σύστημα, οι συμμετέχοντες χρησιμοποιούν το RSA για να διαπραγματευτούν ένα κλειδί, το οποίο στη συνέχεια χρησιμοποιούν με το AES, αντί να το κάνουν με την ανταλλαγή κλειδιών Die-Hellman.

1. Η Αλίκη, η οποία θέλει να στείλει ένα μεγάλο μήνυμα στον Μπομπ, οπότε το RSA θα ήταν αργό, επιλέγει ένα τυχαίο κλειδί που θα χρησιμοποιήσει με ένα συμμετρικό κρυπτογράφημα όπως το AES.

2. Η Αλίκη κρυπτογραφεί το μήνυμα με το δημόσιο κλειδί του Μπομπ:

$$CK = E_{P(B)}(K)$$

3. Η Αλίκη κρυπτογραφεί το μήνυμα χρησιμοποιώντας M και το K με AES:

$C_M = E_{AES(K)}(\mathrm{M})$

4. Η Αλίκη στέλνει (CK, CM) στον Μπομπ.

5. Ο Μπομπ αποκρυπτογραφεί το μήνυμα CK, το οποίο περιέχει το κλειδί AES, χρησιμοποιώντας το ιδιωτικό του κλειδί:

$$K = D_{SB}$$

6. Ο Μπομπ χρησιμοποιεί το κλειδί K για να αποκρυπτογραφήσει το μήνυμα M:

$$M = D_{AES(K)}(CM)$$

Με αυτόν τον τρόπο συνδυάζουμε τα καλύτερα και των δύο κόσμων: αξιοποιούμε το RSA για να αποκτήσουμε ένα ασφαλές μυστικό κλειδί για την επικοινωνία μας και στη συνέχεια το AES για να κρυπτογραφήσουμε το μεγαλύτερο μέρος των δεδομένων μας. Αντί για το RSA, θα μπορούσαμε να χρησιμοποιήσουμε την ανταλλαγή κλειδιών Diffie-Hellman για τον ίδιο σκοπό. Και οι δύο μέθοδοι είναι δημοφιλείς. Ωστόσο, αν θέλουμε να υπογράψουμε και να κρυπτογραφήσουμε με την ίδια μέθοδο, τότε επιλέγουμε το RSA.

6.3 Message Hashing

Η ίδια ανησυχία σχετικά με την ταχύτητα ισχύει και για την ψηφιακή υπογραφή. Μπορεί να είναι αργή η υπογραφή ολόκληρου του μηνύματος εάν το μήνυμα είναι μεγάλο. Η Alice μπορεί στη συνέχεια να υπογράψει ένα ψηφιακό αποτύπωμα, ή απλώς ένα αποτύπωμα, του μηνύματος. Ένα αποτύπωμα ενός μηνύματος είναι ένα σύντομο κομμάτι δεδομένων αναγνώρισης που λαμβάνουμε εφαρμόζοντας μια ειδική γρήγορη συνάρτηση $h(\alpha)$ στο μήνυμα. Ονομάζεται επίσης σύνοψη μηνύματος. Η συνάρτηση είναι τέτοια ώστε από οποιοδήποτε μήνυμα να παράγει μια ακολουθία bit, μικρού σταθερού μεγέθους, ας πούμε 256 bit. Η συνάρτηση ονομάζεται συνάρτηση κατακερματισμού. για αυτόν τον λόγο το αποτύπωμα μπορεί επίσης να ονομαστεί κατακερματισμός μηνύματος. Απαιτούμε επίσης να είναι πολύ δύσκολο να βρεθούν δύο μηνύματα με το ίδιο αποτύπωμα, δηλαδή, h και $h0$ έτσι ώστε $h(\alpha) = h(\alpha)0$). Ονομάζουμε συναρτήσεις κατακερματισμού με αυτήν την ιδιότητα «συναρτήσεις κατακερματισμού ανθεκτικές σε συγκρούσεις». Με αυτόν τον τρόπο, το δακτυλικό αποτύπωμα αναγνωρίζει το μήνυμα, επειδή είναι δύσκολο να πλαστογραφήσει ένα άλλο μήνυμα με το ίδιο δακτυλικό αποτύπωμα.

Η Αλίκη θα υπογράψει ακολουθώντας την παρακάτω ακολουθία βημάτων.

1. Η Αλίκη υπολογίζει το αποτύπωμα του Μ :

$$H = h(M).$$

2. Η Αλίκη υπογράφει το αποτύπωμα του Μ:

$$C = E_{S(A)}(H)$$

3. Η Αλίκη στέλνει $M, C = E_{S(A)}$(Η) στον Μπομπ.

4. Ο Μπομπ υπολογίζει $H = h(M)$ ο ίδιος.

5. Ο Μπομπ επαληθεύει ότι η υπογραφή που έλαβε προέρχεται από την Αλίκη:

Η = $DD_{P(A)}$(C)

6. Ο Μπομπ επαληθεύει ότι το αποτύπωμα που υπολόγισε στο βήμα 4 είναι το ίδιο με την υπογραφή που επαλήθευσε στο βήμα 5.

Φυσικά, είναι αδύνατο να μην υπάρχουν συγκρούσεις, δηλαδή διαφορετικά μηνύματα που έχουν το ίδιο αποτύπωμα. Για παράδειγμα, αν τα μηνύματα έχουν μήκος 10 Kbytes,

τότε υπάρχουν 280.000 πιθανά μηνύματα επειδή 10 Kbytes έχουν 10.000 bytes ή 80.000 bits. Η συνάρτηση κατακερματισμού αντιστοιχίζει αυτά τα $2^{80.000}$ μηνύματα σε έναν πολύ μικρότερο αριθμό πιθανών αποτυπωμάτων. Αν παράγει συνοπτικά στοιχεία μήκους 256 bits, τότε ο αριθμός των πιθανών αποτυπωμάτων είναι 2256. Έτσι, θεωρητικά υπάρχει ένας εξαιρετικά υψηλός αριθμός συγκρούσεων: $2^{80.000}/2^{256} = 2^{79.744}$. Ωστόσο, στην πράξη θα συναντήσουμε μόνο μια μικρή μειοψηφία από τα θεωρητικά πιθανά 280.000 μηνύματα. Θα θέλουμε να υπογράψουμε τα μηνύματα μόνο σε αυτήν τη μειοψηφία. Θέλουμε να είναι εξαιρετικά απίθανο να αντιστοιχίσουμε δύο από αυτές στο ίδιο δακτυλικό αποτύπωμα, και αποδεικνύεται ότι υπάρχουν τέτοιες συναρτήσεις, αλλά πρέπει να είμαστε προσεκτικοί ώστε να επιλέξουμε μία που η ερευνητική κοινότητα έχει κρίνει αρκετά ανθεκτική στις συγκρούσεις.

Θα πρέπει να σημειώσουμε ότι ο όρος hash είναι υπερφορτωμένος και σημαίνει πολλά σχετικά, διακριτά, πράγματα. Ο κατακερματισμός είναι μια σημαντική τεχνική για την αποθήκευση και ανάκτηση δεδομένων. Θα εξερευνήσουμε αυτόν τον κατακερματισμό στο κεφάλαιο 13. Γενικά, η συνάρτηση κατακερματισμού είναι μια συνάρτηση που αντιστοιχίζει δεδομένα αυθαίρετου μεγέθους σε δεδομένα σταθερού, πολύ μικρότερου μεγέθους. Οι καλές συναρτήσεις κατακερματισμού είναι ανθεκτικές στις συγκρούσεις. Οι συναρτήσεις κατακερματισμού που χρησιμοποιούμε για ψηφιακές υπογραφές και εκτύπωση έχουν την πρόσθετη ιδιότητα ότι είναι μονόδρομες συναρτήσεις: δεδομένης της εξόδου της συνάρτησης κατακερματισμού, δεν είναι δυνατό να εξαχθούν τα δεδομένα εισόδου που δόθηκαν στη συνάρτηση κατακερματισμού. Μια τέτοια συνάρτηση κατακερματισμού ονομάζεται κρυπτογραφική συνάρτηση κατακερματισμού. Μια οικογένεια κρυπτογραφικών συναρτήσεων κατακερματισμού με καλό ιστορικό ασφαλείας σε ευρεία χρήση είναι η SHA-2 (Secure Hash Algorithm 2)

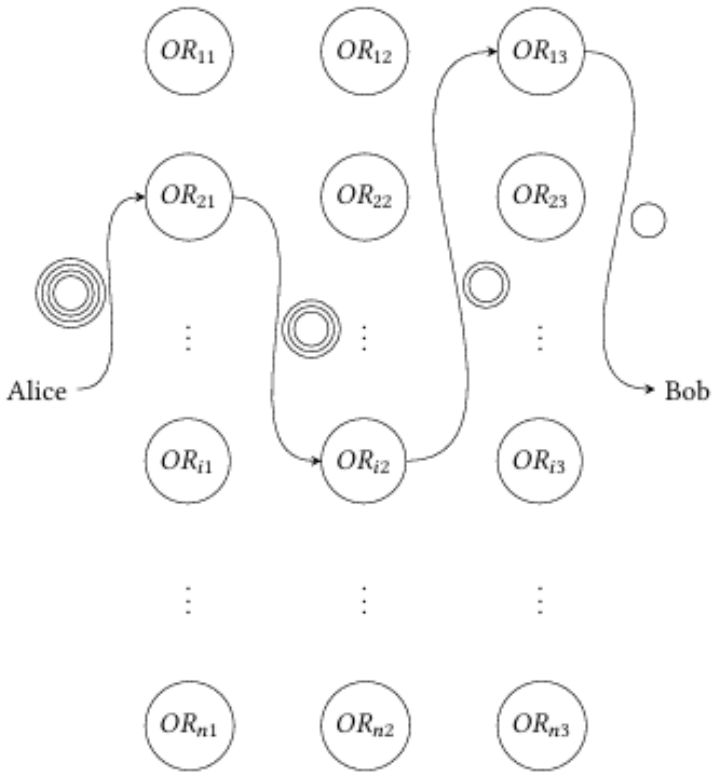
6.4 Ανωνυμία στο Internet

Ο συνδυασμός συμμετρικής και δημοσίας κρυπτογραφίας μπορεί να λύσει και άλλα προβλήματα. Ένα παράδειγμα, το οποίο συνδυάζει RSA, AES και την ανταλλαγή κλειδιών Diffie Hellman, περιλαμβάνει την ανωνυμοποίηση της διαδικτυακής κίνησης. Αυτό είναι ένα παράδειγμα ενός πιο γενικού προβλήματος που σχετίζεται με την κρυπτογράφηση όχι μόνο των δεδομένων αλλά και των μεταδεδομένων. Τα μεταδεδομένα αναφέρονται όχι στα ίδια τα δεδομένα αλλά σε δεδομένα που σχετίζονται με τα δεδομένα. Σε μια τηλεφωνική επικοινωνία, τα μεταδεδομένα περιλαμβάνουν τις ταυτότητες των ατόμων που πραγματοποιούν την κλήση και την ημερομηνία και ώρα της κλήσης. Στο διαδίκτυο, τα μεταδεδομένα σχετικά με την κίνηση μπορεί να αναφέρονται όχι στις πραγματικές πληροφορίες. Αυτό που μεταφέρεται είναι οι λεπτομέρειες των εμπλεκόμενων μερών στην επικοινωνία. Αν η Άλις στείλει ένα ηλεκτρονικό ταχυδρομείο στον Μπομπ, το περιεχόμενο του μηνύματος ηλεκτρονικού ταχυδρομείου είναι τα δεδομένα; η ημερομηνία και η ώρα του ηλεκτρονικού ταχυδρομείου καθώς και το γεγονός ότι η Άλις στείλει ένα ηλεκτρονικό ταχυδρομείο στον Μπομπ είναι μεταδεδομένα. Αν η Άλις επισκεφτεί μια ιστοσελίδα, το περιεχόμενο της ιστοσελίδας είναι τα δεδομένα. Τα γεγονότα ότι η Άλις επισκέφτηκε την συγκεκριμένη ιστοσελίδα σε συγκεκριμένη ημερομηνία είναι μεταδεδομένα. Χρησιμοποιώντας το Diffie - Hellman ή το RSA, μπορούμε να αποκτήσουμε ασφαλή ιδιωτικά κλειδιά με τα οποία να κρυπτογραφήσουμε το περιεχόμενο των επικοινωνιών μας. Κανένας αποσπάτης δεν θα είναι σε θέση να ξέρει τι λέει η Άλις στον Μπομπ. Αλλά ένας αποσπάτης θα ξέρει ότι η Άλις μιλάει με τον Μπομπ, και αυτό καθαυτό μπορεί να είναι σημαντικό. Ακόμα και αν δεν είναι, μπορεί να μην είναι δουλειά κανενός παρά της Άλις και του Μπομπ. Όταν η Άλις επισκέπτεται τον ιστότοπο του Μπομπ, οι πληροφορίες μεταξύ Άλις και Μπομπ ταξιδεύουν σε πακέτα, πηδώντας από υπολογιστή σε υπολογιστή, από τον υπολογιστή της Άλις στον υπολογιστή του Μπομπ και πίσω. Κάθε υπολογιστής στην διαδρομή που κατευθύνει την κίνηση του διαδικτύου, που ονομάζεται δρομολογητής, γνωρίζει ότι μεταφέρει δεδομένα από την Άλις στον Μπομπ. Ακόμα κι αν η επικοινωνία μεταξύ τους είναι κρυπτογραφημένη, η κρυπτογράφηση ισχύει για το περιεχόμενο της επικοινωνίας, δηλαδή, τα δεδομένα. Οι διευθύνσεις της Άλις και του Μπομπ στα πακέτα που μεταδίδουν τα κρυπτογραφημένα δεδομένα είναι ακόμα σε καθαρό κείμενο και μπορούν να καταγραφούν από οποιονδήποτε παρακολουθεί τους υπολογιστές στο ενδιάμεσο; βλ. παρακάτω σχήμα για ένα παράδειγμα μηνύ-

ματος που ταξιδεύει από την Άλις στον Μπομπ μέσω τριών δρομολογητών. Είναι δυνατόν να ανωνυμοποιηθεί η επικοινωνία στο Διαδίκτυο χρησιμοποιώντας την κρυπτογραφία με μια ιδέα που ονομάζεται δρομολόγηση κρεμμυδιού. Λειτουργεί περιλαμβάνοντας το πακέτο σε μια σειρά από συρματοποιημένα πακέτα, όπως ένα κρεμμύδι, όπως φαίνεται στο σχήμα. Ο πρώτος δρομολογητής που λαμβάνει το πακέτο από την Άλις μπορεί να διαβάσει μόνο το εξωτερικό στρώμα, το οποίο δίνει τη διεύθυνση του επόμενου δρομολογητή στον οποίο θα προωθήσει το πακέτο. Ο δεύτερος δρομολογητής θα λάβει το πακέτο χωρίς το εξωτερικό στρώμα και θα μπορεί να διαβάσει μόνο το νέο εξωτερικό στρώμα (δηλαδή, το δεύτερο στρώμα του αρχικού στρωματοποιημένου πακέτου). Αυτό θα δώσει στον δεύτερο δρομολογητή τη διεύθυνση του τρίτου δρομολογητή στον οποίο θα προωθήσει το πακέτο. Η ίδια διαδικασία, απογυμνώνοντας ένα ακόμη στρώμα, θα συμβεί στον τρίτο δρομολογητή, και ούτω καθεξής μέχρι το πακέτο να φτάσει στον Μπομπ. Παρατηρήστε τώρα ότι ο πρώτος δρομολογητής γνωρίζει μόνο ότι έλαβε ένα πακέτο από την Άλις προς τον δεύτερο δρομολογητή. Ο δεύτερος δρομολογητής γνωρίζει μόνο ότι έλαβε ένα πακέτο από τον πρώτο δρομολογητή προς τον τρίτο δρομολογητή. Ο Μπομπ θα γνωρίζει μόνο ότι έλαβε ένα πακέτο από κάποιον δρομολογητή. Κανείς δεν ξέρει, εκτός από την Άλις, ότι το πακέτο αποστάλθηκε αρχικά από την Άλις προς τον Μπομπ.

Πώς δημιουργεί η Άλις το πακέτο με τις στρώσεις κρεμμυδιού και πώς δημιουργεί τη διαδρομή προς τον Μπομπ χωρίς οι ενδιάμεσοι δρομολογητές να το γνωρίζουν; Ένας γνωστός δρομολογητής κρεμμυδιού είναι το Tor (σύντομο για τον δρομολογητή κρεμμυδιού), ο οποίος λειτουργεί περίπου ως εξής. Το Tor αποτελείται από έναν αριθμό ενδιάμεσων δρομολογητών που είναι διαθέσιμοι στην Άλις. Αρχικά, η Άλις επιλέγει το σύνολο των ενδιάμεσων δρομολογητών που θα χρησιμοποιήσει. Αυτοί ονομάζονται Δρομολογητές Κρεμμυδιού (ORs). Υποθέστε ότι επιλέγει τρεις ORs: OR1,OR2,OR3. Η Αλις θέλει το μήνυμά της να ταξιδέψει μέσω της διαδρομής της Άλις →OR1→ OR2 → OR3 → Μπομπ. Μόνο η Άλις θα γνωρίζει αυτή τη διαδρομή. Το μήνυμα θα ταξιδεύει από δρομολογητή σε δρομολογητή, με στρώματα κρυπτογράφησης να αποκαλύπτονται σε κάθε δρομολογητή. Μπορείτε να δείτε μια απεικόνιση στο σχήμα 5.5, όπου υποθέτουμε ότι το Tor περιλαμβάνει OR_{n3} κρεμμύδι δρομολογητές τακτοποιημένοι ωραία ως μήτρα; η πραγματική τοπολογία του Tor φυσικά δεν είναι έτσι και αλλάζει καθώς προστίθενται ή αφαιρούνται δρομολογητές κρεμμυδιού με την πάροδο του χρόνου. Μπορεί να επιλέξει διαφορετικούς δρομολογητές την επόμενη φορά που θέλει να επικοινωνήσει με τον

Μπομπ, έτσι ώστε να μην είναι δυνατό να παρακολουθήσουν τις επικοινωνίες της με την πάροδο του χρόνου. Η Άλις αρχίζει επικοινωνώντας με OR1 χρησιμοποιώντας τον RSA και στέλνοντας οδηγίες για το πώς να ρυθμίσετε τη διαδρομή επικοινωνίας. Καθώς αυτό είναι ένα πακέτο με το οποίο η Άλις δίνει εντολές OR1 για να κάνουμε ορισμένα πράγματα, μπορούμε να το σκεφτούμε ως μια πακέτο εντολών. Περιέχει το μέρος της σε μια ανταλλαγή κλειδιών Di e-Hellman με OR1. Επιπλέον, περιέχει μια εντολή που λέει OR1 ότι θα σημειώνει τα πακέτα της με μια ειδική ταυτότητα που επιλέγει, η οποία ονομάζεται αναγνωριστικό κυκλώματος C1. Ας ονομάσουμε αυτό το πακέτο εντολών ένα πακέτο CreateHop(C1, g^{x1}), συντομεύοντας το μέρος Di e-Hellman στην σημειωτική μας.OR1 απαντά με το μέρος της ανταλλαγής κλειδιού Di e Hellman. Όλα τα μηνύματα που θα στέλνει η Άλις σε OR1 θα είναι κρυπτογραφημένο με το κλειδί που έχουν καθορίσει, λένε DH1.

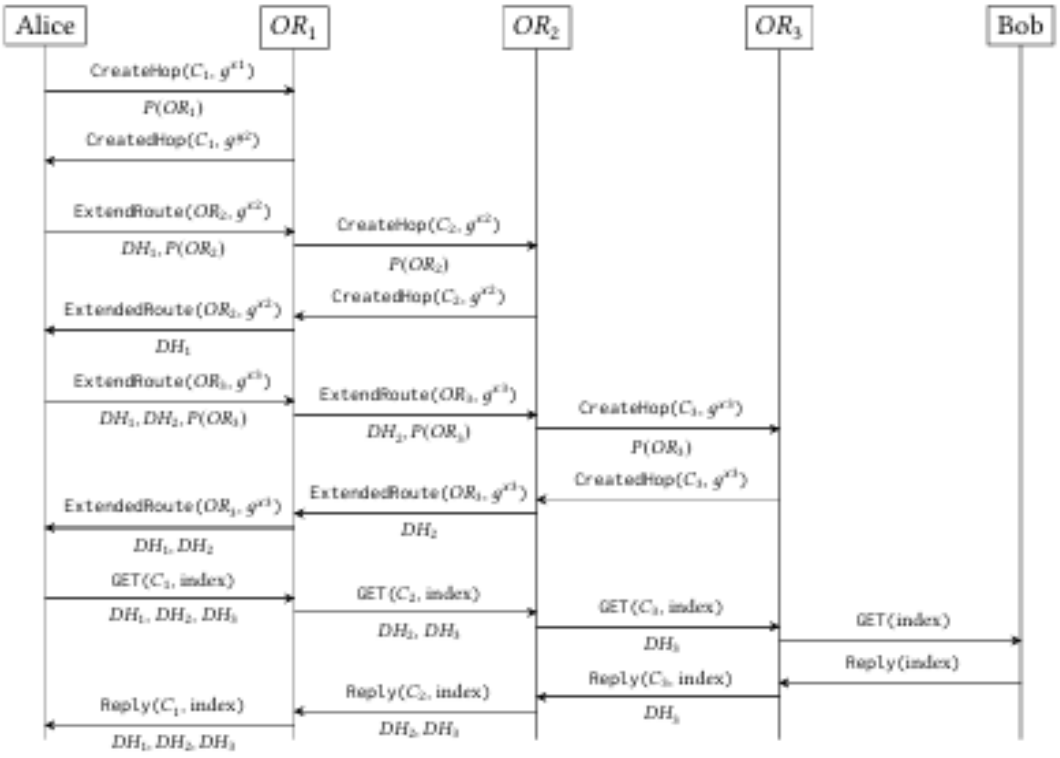


Εικόνα 12 Επιλογή δρομολογητών για την αποστολή μηνύματος. Πηγή [1]

Στη συνέχεια, η Άλις επικοινωνεί ξανά με το OR1 και του λέει ότι από εδώ και πέρα θέλει OR1 να προωθήσει όλα τα μηνύματα από αυτήν σε το OR2. Για να το κάνει αυτό, στέλνει ένα πακέτο εντολής στο OR1 με την εντολή να επεκτείνει τη διαδρομή και το κομμάτι

της νέας ανταλλαγής κλειδιών Di e-Hellman. Το μέρος Di e-Hellman είναι κρυπτογραφημένο με το δημόσιο κλειδί RSA του OR2. Το ολόκληρο πακέτο είναι κρυπτογραφημένο με DH1. Ας ονομάσουμε αυτό το πακέτο εντολής ExtendRoute (OR2, g^{x2}) πακετο. Όταν OR1 λαμβάνει το πακέτο, το αποκρυπτογραφεί. Στη συνέχεια δημιουργεί μια νέα CreateHop (C2, g^{x2}) πακέτο που στέλνει σε OR2 Η πακέτα εντολής περιέχει το μέρος Di e-Hellman από την Άλις στο OR2, και λέει OR2 Ότι θα επισημαίνει πακέτα με άλλο θα προσδιορίζει πακέτα με άλλο αναγνωριστικό κυκλώματος, C2. Λέει ότι προς OR2 χωρίς να του πείτε ότι τα μηνύματα θα έρχεται από την Άλις. OR1 καταγράφει το γεγονός ότι τα πακέτα που είναι επισημασμένα με C1 θα σταλούν σε και τα πακέτα που παραλαμβάνονται από OR2 επισημασμένα με C2 θα περάσουν πίσω στην Άλις.OR1 επιστρέφει την απάντηση Di e-Hellman που λαμβάνει από OR2 στην Άλις, οπότε Άλις και OR2 μοιραστείτε ένα κλειδί Di e-Hellman, DH2. Για να δημιουργήσει τη διαδρομή προς το OR3, η Άλις δημιουργεί ένα πακέτο εντολών ExtendRoute(OR3, g^{x3}) για να επεκτείνει τη διαδρομή από το OR2 έως το OR3. Το πακέτο περιέχει το μέρος της κλειδαριάς Di e-Hellman που θέλει να καθιερώσει με το OR3. Το μέρος Di e Hellman κρυπτογραφείται με το δημόσιο κλειδί RSA του OR3. Ολόκληρο το πακέτο κρυπτογραφείται με το DH2 και στη συνέχεια κρυπτογραφείται ξανά με το DH1. Η Άλις στέλνει το πακέτο στο DH1. Όταν το OR1 λαμβάνει το πακέτο, μπορεί να αποκρυπτογραφήσει μόνο το πρώτο επίπεδο. Το OR1 γνωρίζει ότι τα κύτταρα που επισημαίνονται με το C1 πρέπει να προωθηθούν στον προορισμό που σχετίζεται με το C2, το OR2, αλλά δεν γνωρίζει το περιεχόμενο του. Το OR1 επισημαίνει το πακέτο με το C2 και προωθεί το πακέτο με ένα επίπεδο αποκρυπτογραφημένο στο OR2. OR2 λαμβάνει το πακέτο από OR1 και το αποκρυπτογραφεί χρησιμοποιώντας OR2, ανακτώντας Extend Rout (OR3, g^{x3}). Ακολουθεί την ίδια σειρά βημάτων που έκανε OR1 προηγουμένως, καθώς OR2 έχει λάβει ένα πακέτο εντολής για την επέκταση της διαδρομής, το οποίο περιέχει το κομμάτι της Alice από την ανταλλαγή κλειδιών Di e-Hellman με OR3. Δημιουργεί και στέλνει ένα νέο πακέτο εντολής CreateHop(C3, g^{x3}) προς OR3. Το πακέτο εντολών περιέχει το μέρος Di e-Hellman από την Alice προς OR3 και το ενημερώνει ότι θα σφραγίζει πακέτα με ένα άλλο ID κυκλώματος, ας πούμε C3. OR2 καταγράφει το γεγονός ότι τα πακέτα που θα σφραγίζονται με C2 θα σταλούν προς OR3, και τα πακέτα που παραλαμβάνονται από 3 σφραγισμένα με 3 θα επιστραφούν πίσω στον OR1. OR2 επιστρέφει την απάντηση Di e-Hellman από OR3 στην Alice μέσω του OR1, έτσι ώστε η Alice και ο OR3 να μοιράζονται ένα Di

e-Hellman κλειδί, DH3. Τώρα η Άλις μπορεί να στείλει το μήνυμά της στον Μπομπ γνωρίζοντας ότι όχι μόνο το περιεχόμενο, αλλά και η διαδρομή θα παραμείνουν μυστικά. Για να στείλει ένα μήνυμα στον Μπομπ, η Άλις δημιουργεί ένα πακέτο με το μήνυμά της που απευθύνεται στον Μπομπ, κρυπτογραφημένο με DH3, που με τη σειρά του είναι κρυπτογραφημένο με DH2, που με τη σειρά του είναι κρυπτογραφημένο με DH1 και επισημασμένο με C1. Το πακέτο πηγαίνει πρώτα στον C1. Δεδομένου ότι το πακέτο είναι επισημασμένο με C1, ο OR1 γνωρίζει ότι πρέπει να το προωθήσει στον OR2. OR1 αφαιρεί το πρώτο επίπεδο χρησιμοποιώντας το C1 και το προωθεί στον C2, επισημασμένο με 2. Ο OR2 αφαιρεί το δεύτερο επίπεδο χρησιμοποιώντας το DH2. Γνωρίζει ότι τα πακέτα που είναι επισημασμένα με C2 πρέπει να προωθούνται στον OR3, οπότε το επισημαίνει με C3 και το στέλνει στον OR3. OR3 παίρνει το πακέτο από 2 και το αποκρυπτογραφεί χρησιμοποιώντας 3. Βλέπει ότι είναι ένα μήνυμα που απευθύνεται στον Μπομπ, οπότε το προωθεί εκεί. Η απάντηση από τον Μπομπ θα ακολουθήσει ακριβώς την αντίστροφη διαδρομή, Μπομπ →OR3→OR2→OR1→Αλις, κρυπτογραφημένο ξανά με DH1 τότε DH2 ,τότε DH3 κατευθυνόμενοι με τον ίδιο τρόπο χρησιμοποιώντας C3, C2, C1. Μπορείτε να δείτε όλη την αλληλεπίδραση στο σχήμα 5.6, διαβάζοντας από πάνω προς τα κάτω και από αριστερά προς τα δεξιά. Κάθε βέλος αντιστοιχεί σε ένα μήνυμα. Η ετικέτα πάνω από το βέλος αντιστοιχεί στο περιεχόμενο του μηνύματος και η ετικέτα κάτω από αυτό.



Εικόνα 13 Δημιουργία κρυπτογραφημένου μονοπατιού δρομολογητών. Πηγή [1]

αντιστοιχεί στην κρυπτογράφηση που εφαρμόζεται, είτε το Di e-Hellman είτε το RSA με το δημόσιο κλειδί του παραλήπτη. Η Άλιντ ζητά από τον διακομιστή του Μπομπ να της στείλει την αρχική σελίδα του, και ο Μπομπ απαντά με αυτή τη σελίδα. Για να ζητήσει την αρχική σελίδα, πρέπει να στείλει ένα αίτημα GET στον διακομιστή του Μπομπ χρησιμοποιώντας το Πρωτόκολλο Μεταφοράς Υπερκειμένου (HTTP), αλλά οι λεπτομέρειες δεν μας αφορούν εδώ. Το Tor επιτρέπει επίσης τη δημιουργία των κλειδιών Di e-Hellman χρησιμοποιώντας ένα νεότερο πρωτόκολλο που δεν χρησιμοποιεί RSA; το νεότερο πρωτόκολλο είναι ταχύτερο και καλύτερο αλλά πιο δύσκολο να περιγραφεί. Επίσης, όλη η επικοινωνία μεταξύ δύο σημείων στο Tor (Άλιντ, Onion Routers, και Μπομπ) χρησιμοποιεί πρωτόκολλα αυθεντικοποίησης και κρυπτογράφησης. .Συνολικά, το Tor είναι στιβαρό και αποτελεί βασικό εργαλείο για τη διατήρηση της ανωνυμίας στο Διαδίκτυο.

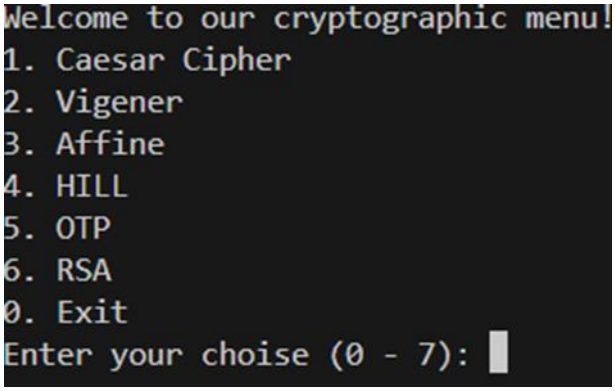
7 Κρυπτογραφική εφαρμογή

Το πρακτικό κομμάτι της εργασίας μας αποτελείται από την υλοποίηση διάφορων κρυπτογραφικών συναρτήσεων και αλγορίθμων στην γλώσσα python. Έχουμε δημιουργήσει δυο modules: Τα main program.py και cryptoLib.py.

7.1 mainprogram.py

Στο module αυτό έχουμε υλοποιήσει το κυρίως menu το όποιό εμφανίζεται στον χρήστη κατά την εκτέλεση του προγράμματος. Ο χρήστης μπορεί να αποφασίσει ποιον αλγόριθμο κρυπτογράφησης θέλει να εκτελεστεί πληκτρολογώντας τον κατάλληλο κωδικό από αυτούς που του προτείνει το μενού. Ανάλογα το κωδικό που πληκτρολογεί εμφανίζεται προτρεπτικό μήνυμα που καλεί τον χρήστη να δώσει τα απαραίτητα δεδομένα και επιλογές προκειμένου να εκτελεστεί ο αλγόριθμος που έχει επιλέξει. Σε περίπτωση που ο χρήστης δεν δώσει τα καταλληλά δεδομένα εμφανίζεται μήνυμα λάθους και επίσης καλείται ο χρήστης να πληκτρολόγηση ξανά τα σωστά δεδομένα. Αυτό ισχύει για κάθε περίπτωση αλγορίθμου που έχουμε υλοποιήσει. Ο χρήστης μπορεί να επιλέξει μεταξύ 6 γνωστών αλγορίθμων κρυπτογράφησης: Caesar Cipher, Vigener, Affine, HILL, OTP, RSA. Μπορεί επίσης να επιλέξει το 0 ώστε να σταματήσει η εκτέλεση του προγράμματος.

Η αρχική εικόνα που εμφανίζεται όταν εκτελούμε το πρόγραμμα μας:



Εικόνα 14 Το αρχικό menu της εφαρμογής μας

7.2 cryptoLib.py

Αυτό το module λειτουργεί σαν βιβλιοθήκη των αλγορίθμων που έχουμε υλοποιήσει. Κάθε ένας αλγόριθμος έχει υλοποιηθεί σαν συνάρτηση η οποία παίρνεις τα ορίσματα της το αρχικό μήνυμα καθώς και το κλειδί που χρειάζεται για την κρυπτογράφηση και επιστρέφει το αρχικό μήνυμα σε κρυπτογραφημένη μορφή. Οι συναντήσεις που αντιστοιχούν στους αλγορίθμους που έχουμε υλοποιήσει είναι οι: Caesar_encrypt, encrypt_vigenere, affine_encrypt, Hill_encrypt, OTP_encryption, encryptRSA. Επίσης σε αυτό το module υπάρχουν και κάποιες βοηθητικές συναρτήσεις που χρειάζονται οι κύριοι αλγόριθμοι για την λειτουργία τους.

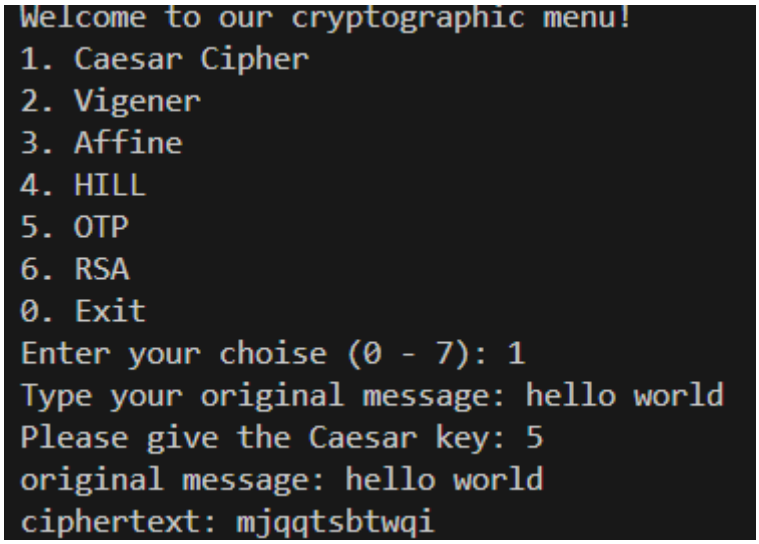
7.3 Περιπτώσεις εκτέλεσης αλγορίθμων

Παρακάτω ακολουθούν εικόνες εκτέλεσης της εφαρμογής μας για κάθε ένα αλγόριθμο ξεχωριστά. Οι εικόνες εκτέλεσης αναφέρονται σε ορθή περίπτωση χρήσης των αλγορίθμων καθώς και σε περιπτώσεις που δίνουμε λανθασμένα δεδομένα όπου φαίνεται η αντίδραση της εφαρμογής μας με καταλληλά διορθωτικά μηνύματα .

Παρακάτω ακολουθούν ενδεικτικές εικόνες εκτέλεσης του προγράμματος μας:

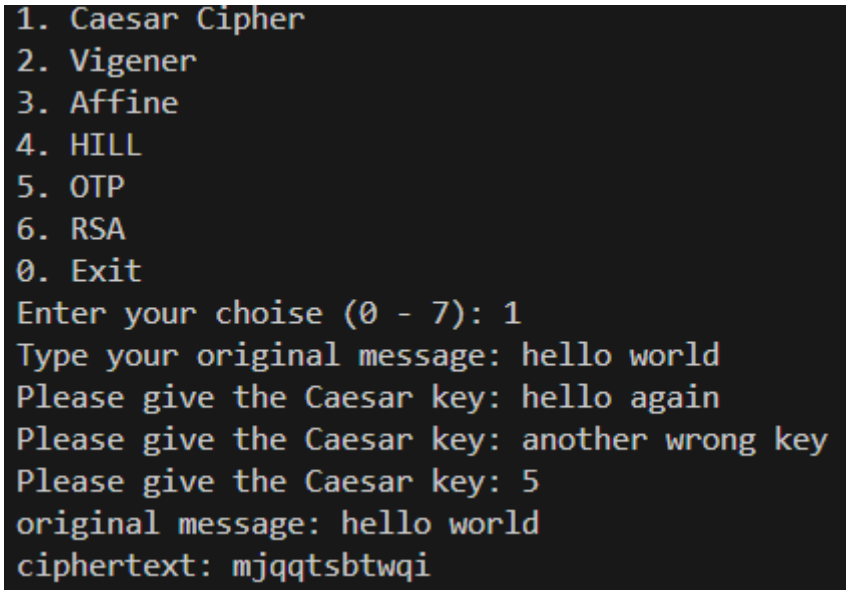
Περίπτωση αλγορίθμου Καίσαρα

Ο χρήστης έδωσε την επιλογή 1. Το πρόγραμμα του ζήτησε να πληκτρολόγηση το κειμένου προς κρυπτογράφηση. Ο χρήστης πληκτρολόγησε το hello world και κατόπιν το πρόγραμμα του ζήτησε να πληκτρολογήσει το κλειδί κρυπτογραφήσεις για τον αλγόριθμο του Καίσαρα. Ο χρήστης πληκτρολόγησε το 5 και κατόπιν το πρόγραμμα του εμφάνισε το αρχικό του κείμενο και το αντίστοιχο σε κρυπτογραφημένη μορφή σύμφωνα με τον αλγόριθμο που έχει επιλέξει.



Εικόνα 15 Παράδειγμα σωστής εκτέλεσης αλγορίθμου του Καίσαρα

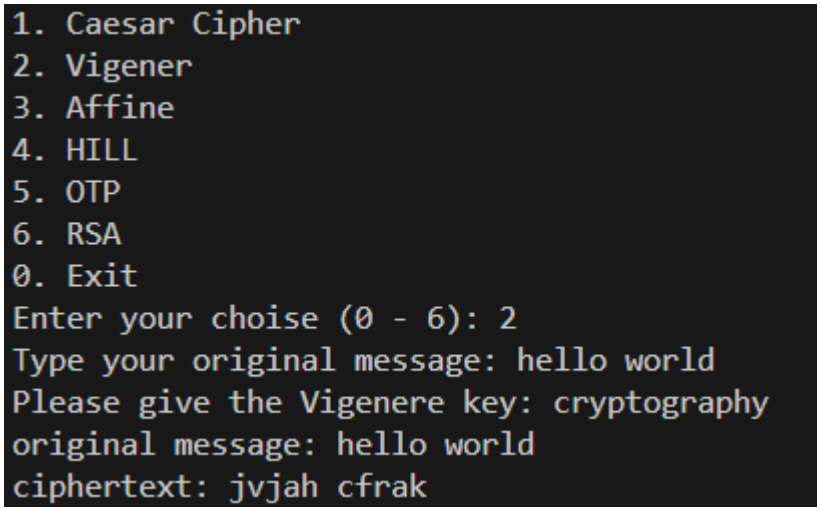
Στην παρακάτω εικόνα ο χρήστης έχει επιλέξει να κάνει κρυπτογράφηση με τον αλγόριθμο του Καίσαρα. Ωστόσο δίνει λάθος κλειδί, δηλαδή αντί να δώσει αριθμό μετατόπισης δίνει κείμενο. Το πρόγραμμα του ζητάει ξανά να δώσει κλειδί. Ωστόσο ο χρήστης πληκτρολογεί πάλι κείμενο. Το πρόγραμμα ξαναζητά από τον χρήστη σωστό κλειδί και τελικά ο χρήστης αναγκάζεται να δώσει το 5.Το πρόγραμμα του εμφανίζει το κανονικό και το κρυπτογραφημένο κείμενο.



Εικόνα 16 Παράδειγμα εισαγωγής λάθος δεδομένων

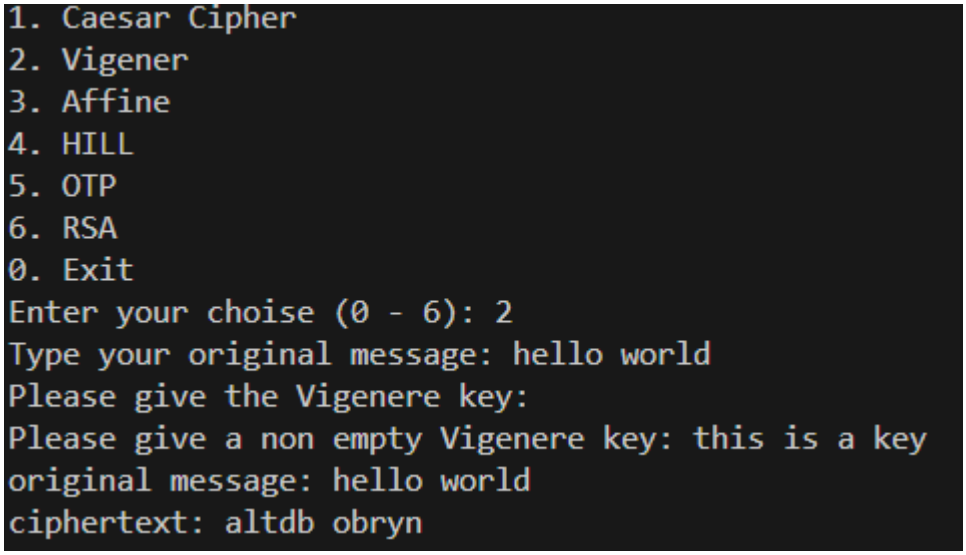
Περίπτωση αλγορίθμου Vigener:

Στην παρακάτω εικόνα εικόνα ο χρήστης έχει επιλέξει να κάνει κρυπτογράφηση με τον αλγόριθμο Vigener για αυτό έχει πληκτρολογήσει την επιλογή 2. Στην συνέχεια δίνει το κείμενο προς κρυπτογράφηση και κατόπιν επιλεγεί το κλειδί cryptography. Το πρόγραμμα εμφανίζει το πρότυπο μήνυμα και το κρυπτογραφημένο του.



Εικόνα 17 Παράδειγμα αλγορίθμου Vigener

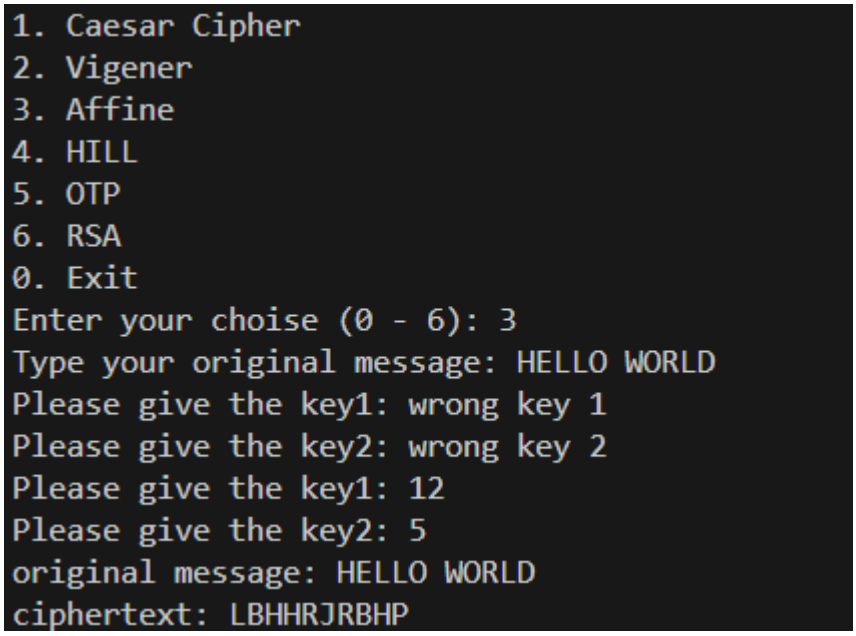
Στην παρακάτω εικόνα ο χρήστης αντί να δώσει κλειδί πατάει enter με αποτέλεσμα το πρόγραμμα να μην δεχτεί το κενό κλειδί και να του ζητήσει να δώσει κλειδί εκ νέου.



Εικόνα 18 Παράδειγμα λάθος δεδομένων στο Vigener

Περίπτωση αλγορίθμου Affine:

Στην παρακάτω εικόνα ο χρήστης επέλεξε τον αλγόριθμο του Affine. Ωστόσο πρέπει να δώσει 2 κλειδιά που να αποτελούνται από αριθμούς ώστε ο αλγόριθμος να μπορέσει να κρυπτογραφήσει το μήνυμα που έστειλε αλλιώς θα συνεχίσει να ζητάει να του δοθούν αυτά τα δυο κλειδιά. Ο χρήστης αναγκάζεται να δώσει τα σωστά κλειδιά ώστε να κρυπτογραφηθεί το κείμενο του.



Εικόνα 19 Παράδειγμα αλγορίθμου Affine

Περίπτωση αλγορίθμου HILL:

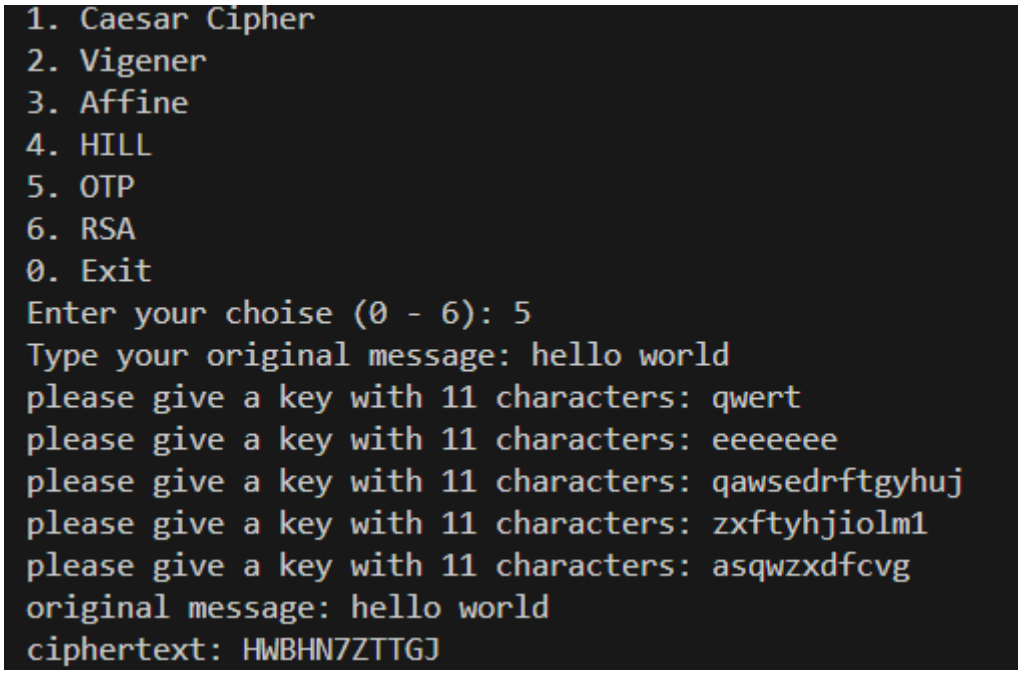
Στην παρακάτω εικόνα ο χρήστης επέλεξε τον αλγόριθμο του HILL. Ωστόσο αντί να δώσει κλειδί με 9 χαρακτήρες έδωσε κλειδί με λιγότερους από εννέα χαρακτήρες. Το πρόγραμμα του ζήτησε να δώσει κλειδί με εννέα χαρακτήρες και αφού τους έδωσε του εμφάνισε το κρυπτογραφημένο κείμενο.

```
1. Caesar Cipher
2. Vigener
3. Affine
4. HILL
5. OTP
6. RSA
0. Exit
Enter your choise (0 - 7): 4
Type your original message: HELLO WORLD
Please give a nine characters Hill key: AESDF
Please give a nine characters Hill key: ASDQWEZXC
original message: HELLO WORLD
ciphertext: B K D B K D B K D B K D
```

Εικόνα 20 Παράδειγμα αλγορίθμου Hill

Περίπτωση αλγορίθμου OTP:

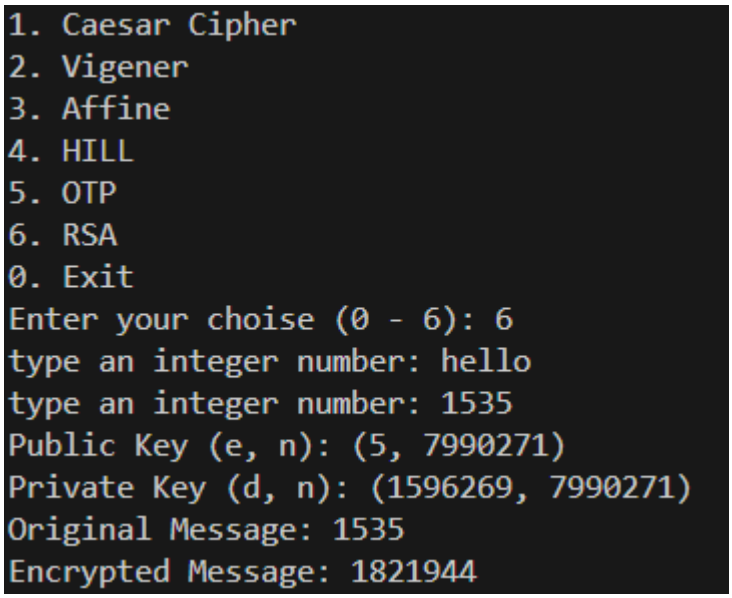
Στην παρακάτω περίπτωση ο χρήστης χρησιμοποιεί τον αλγόριθμο OTP ο οποίος ζητάει από την χρήστη να του δώσει το μήνυμα. Δίνοντας του το μήνυμα ζητάει να δώσει ένα κλειδί που θα αποτελείται από 11 χαρακτήρες. Βλέπουμε όμως ότι όσες φορές δίνει ο χρήστης το κλειδί που δεν αποτελείται από 11 χαρακτήρες να ξαναζητιέται το κλειδί μέχρι να το δώσει ο χρήστης το κατάλληλο για να μας δοθεί το κρυπτογραφημένο μήνυμα.



Εικόνα 21 Παράδειγμα αλγορίθμου OTP

Περίπτωση αλγορίθμου RSA:

Στην παρακάτω εικόνα ο χρήστης επέλεξε τον αλγόριθμο RSA ο οποίος κρυπτογραφεί έναν ακέραιο αριθμό. Ωστόσο ο χρήστης έδωσε κείμενο αντί για αριθμό και το πρόγραμμα του ξαναζητάει έναν αριθμό προς κρυπτογράφηση. Ο χρήστης δίνει τον αριθμό 1535 και το πρόγραμμα του εμφανίζει τα ζεύγη των δημοσίων και ιδιωτικών κλειδιών που αντιστοιχούν στην κρυπτογράφηση και αποκρυπτογράφηση αυτού του αριθμού. Επίσης το εμφανίζει και το αποτέλεσμα της κρυπτογράφησης.



Εικόνα 22 Παράδειγμα αλγορίθμου RSA

Βιβλιογραφία

[1] Panos Louridas, « Review of Real-World Algorithms: A Beginners Guide», MIT Press, 2017.

[2] Alfred j. Menezes, Paul C. van Oorschot, Scott A.Vanstone, « Handbook of Applied Cryptography», 1st edition, CRC Press, 2018.

[3] Burmester Mike, Γκρίτζαλης Στέφανος, Κάτσικας Σωκράτης, Χρυσικόπουλος Βασίλειος, « Σύγχρονη Κρυπτογραφία: Θεωρία και Εφαρμογές», Εκδόσεις Παπασωτηρίου, 2011.

[4] William Stallings, « Κρυπτογραφία και Ασφάλεια Δικτύων. Αρχές και Εφαρμογές.», Εκδοτικός όμιλος Ιων, 2011.

[5] Βασίλειος Α. Κάτος, Γ. Χ. Στεφανίδης, « Τεχνικές Κρυπτογραφίας και Κρυπτανάλυσης», Εκδόσεις Ζυγός, 2003.

Παράρτημα Α

mainprogram.py

```
import cryptoLib as crypto

# Προτρέπει το χρήστη να δώσει το μήνυμα για κρυπτογράφηση
def get_message():
    msg = input("Type your original message: ")
    while msg == "":
        msg = input("Type your original message: ")
    return msg

# Επιστρέφει true εάν το num είναι αριθμός αλλιώς false
def is_number(num):
    return num.isnumeric()

def print result(msg, cipher):
    print("original message:", msg)
    print("ciphertext:", cipher)

# Αν ο χρήστης έχει δώσει την επιλογή 1(αλγόριθμος Καισαρα)
def option_1():
    msg = get_message ()
    key = input ("Please give the Caesar key: ")
    while is_number(key) == False:
        key = input ("Please give the Caesar key: ")
    key = int(key)
```

```
    cipher = crypto. Caesar_encrypt (msg, key)
    print result (msg, cipher)

def option_2():
    msg = get_message ()
    key = input ("Please give the Vigenere key: ")
    while key == "":
        key = input ("Please give a non-empty Vigenere key: ")
    cipher = crypto. Encrypt_vigenere (msg, key)
    print result (msg, cipher)

def option_3():
    msg = get_message ()
    key1 = input ("Please give the key1")
    key2 = input ("Please give the key2")
    while is_number(key1) == False or is_number(key2) == False:
        key1 = input ("Please give the key1: ")
        key2 = input ("Please give the key2: ")
    cipher = crypto.affine_encrypt(msg, [int(key1), int(key2)])
    print_result(msg, cipher)

def option_4():
    msg = get_message()
    key = input("Please give a nine characters Hill key: ")
    while len(key) != 9:
        key = input("Please give a nine characters Hill key: ")
    cipher = crypto.Hill_encrypt(msg, key)
    print_result(msg, cipher)
```

```
def option_5():
    msg = get_message()
    msg_input = "please give a key with " + str(len(msg)) + " characters: "
    key = input(msg_input)
    while len(key) != len(msg):
        key = input(msg_input)
    cipher = crypto.OTP_encryption(msg, key)
    print_result(msg, cipher)

def option_6():
    msg = input("type an integer number: ")
    while msg.isdigit() == False:
        msg = input("type an integer number: ")

    msg = int(msg)
    crypto.RSA_encrypt(msg)




def print_menu():
    print("1. Caesar Cipher")
    print("2. Vigener")
    print("3. Affine")
    print("4. HILL")
    print("5. OTP")
    print ("6. RSA")
    print ("0. Exit")
```

```
def main_menu():
    print("Welcome to our cryptographic menu!")



    while True:
        print_menu()
        choice = input("Enter your choise (0 - 7): ")

        if choice == "1":
            option_1()
        elif choice == "2":
            option_2()
        elif choice == "3":
            option_3()
        elif choice == "4":
            option_4()
        elif choice == "5":
            option_5()
        elif choice == "6":
            option_6()
        elif choice == "0":
            print("Program is terminated...")
            break
        else:
            print("Invalid choice. Please try again...")

if __name__ == "__main__":
    main_menu()
```

cryptoLib.py

```
# Συνάρτηση κρυπτογράφησης για τον αλγόριθμο του ΚΑΙΣΑΡΑ

# Συνάρτηση κρυπτογράφησης του αλγορίθμου του καίσαρα
def Caesar_encrypt (text, s):
    result = ""

    # Για κάθε ένα χαρακτήρα του μηνύματος τον κρυπτογραφούμε
    for i in range(len(text)):
        char = text[i]


        #  Κρυπτογράφηση κεφαλαίων χαρακτήρων
        if (char.isupper()):
            result += chr((ord(char) + s-65) % 26 + 65)


        # Κρυπτογράφηση πεζών χαρακτήρων
        else:
            result += chr((ord(char) + s - 97) % 26 + 97)


    return result


#Συναρτήσεις για κρυπτογράφηση αποκρυπτογράφηση VIGENERE


# Αν το κλειδί που έδωσε ο χρήστης έχει ιδιο μήκος με το μήνυμα τότε απλώς το επι-
στρέφουμε
# Διαφορετικά αναπαράγουμε το κλειδί μέχρι να έχει μήκος όσο το μήνυμα
def generate key(msg, key):
    key = list(key)
    if len(msg) == len(key):
```

```
        return key
    else:
        for i in range(len(msg) - len(key)):
            key. Append (key[i % len(key)])
    return "".join(key)

# Κρυπτογράφηση κειμένου με την μέθοδο Vigenere
def encrypt_vigenere(msg, key):
    encrypted_text = []
    key = generate_key(msg, key)
    for i in range(len(msg)):
        char = msg[i]
        if char.isupper():
            encrypted_char = chr((ord(char) + ord(key[i]) - 2 * ord('A')) % 26 + ord('A'))
        elif char.islower():
            encrypted_char = chr((ord(char) + ord(key[i]) - 2 * ord('a')) % 26 + ord('a'))
        else:
            encrypted_char = char
        encrypted_text.append(encrypted_char)
    return "".join(encrypted_text)



# Υλοποιηση του αλγορίθμου AFFINE CIPHER

# Επεκταμένος αλγόριθμος του Ευκλείδη για να βρίσκουμε το πολλαπλασιαστικό αντί-
στροφο ενός αριθμού
# eg: modinv(7, 26) = 15
def egcd(a, b):
    x,y, u,v = 0,1, 1,0
    while a != 0:
```

```
        q, r = b//a, b%a
        m, n = x-u*q, y-v*q
        b,a, x,y, u,v = a,r, u,v, m,n
    gcd = b
    return gcd, x, y

def modinv (a, m):
    gcd, x, y = egcd (a, m)
    if gcd != 1:
        return None  # Δεν υπάρχει πολλαπλασιαστικό αντίστροφο
    else:
        return x % m

# Συνάρτηση κρυπτογράφησης του Affine
def affine_encrypt(text, key):
    '''
    C = (a*P + b) % 26
    '''
    return ''. join ([ chr (((key [0] *(ord(t) - ord('A')) + key[1] ) % 26)
                + ord('A')) for t in text.upper().replace(' ', '') ])

# Συναρτήσεις αλγορίθμου του HILL

def getKeyMatrix(key, keyMatrix):
    n = 3
    k = 0
    for i in range(n):
        for j in range(n):
            keyMatrix[i][j] = ord(key[k]) % 65
```

```
            k += 1

    return keyMatrix

# Following function encrypts the message
def encrypt(messageVector, cipherMatrix, keyMatrix):
    n = 3
    for i in range(n):
        for j in range(1):
            cipherMatrix[i][j] = 0
            for x in range(n):
                cipherMatrix[i][j] += (keyMatrix[i][x] * messageVector[x][j])
            cipherMatrix[i][j] = cipherMatrix[i][j] % 26

def Hill_encrypt(message, key):

    n = 3

    keyMatrix = [[0] * n for i in range(n)]

    # Δημιουργεί πίνακα για το μπλοκ μηνύματος
    messageVector = [[0] for i in range(n)]

    # Δημιουργεί πίνακα για το cipher των μπλοκς μηνύματος
    cipherMatrix = [[0] for i in range(n)]

    # Καλούμε την συνάρτηση για να δημιουργήσει βάση του κλειδιού
    # Τον κατάλληλο πίνακα κλειδιού
    keyMatrix = getKeyMatrix(key, keyMatrix)
```

```
    # Θα πρέπει το μήνυμα να αποτελείται από τριάδες χαρακτήρων
    # Όσο το μήκος μηνύματος δεν είναι πολλαπλάσιο του 3 προσθέτουμε
    # στο τέλος χαρακτήρες # πχ hello θα γίνει hello#
    while len(message) % n != 0:
        message += '#'

    # Σπάμε το μήνυμα προς κρυπτογράφηση σε μπλοκ των τριών χαρακτήρων
    # και το βάζουμε στην λιστα.πχ hello# θα γίνει [[hel], [lo#]]
    message_blocks = [message[i:i+n] for i in range(0, len(message), n)]

    cipherText = []

    for mblock in message_blocks:
        # Γεμίζουμε πίνακα για το μπλοκ μηνύματος
        for i in range(n):
            messageVector[i][0] = ord(message[i]) % 65

        # Δημιουργούμε το αντίστοιχο κρυπτογραφημένο μπλοκ
        encrypt (messageVector, cipherMatrix, keyMatrix)

        # Προσθέτουμε στην λίστα του κρυπτογραφημένους χαρακτήρες του μπλοκ
        for i in range(n):
            cipherText.append(chr(cipherMatrix[i][0] + 65))

    # Επιστρέφουμε το κρυπτογραφημένο μήνυμα
    return ' '.join(cipherText)

# Υλοποιηση του αλγορίθμου κρυπτογράφησης ONE TIME PAD
```

```
# Συνάρτηση για κρυπτογράφηση ONE TIME PAD
def OTP_encryption (text, key):

    # μετατρέπουμε το κείμενο και το κλειδί σε κεφαλαίους χαρακτήρες
    text = text.upper()
    key = key.upper()
    # αρχικοποίηση του string με το κρυπτογραφημένο μήνυμα
    cipherText = ""

    # Αρχικοποίει την λίστα cipher με τα αθροίσματα που προκύπτουν από
    # τους χαρακτήρες του μηνύματος και του κλειδιού
    cipher = []
    for i in range(len(key)):
        cipher.append(ord(text[i]) - ord('A') + ord(key[i])-ord('A'))

    # αν κάποια θέση του cipher έχει αριθμο μεγαλύτερο του 25 τότε αφαιρούμε
    # από αυτό τον αριθμο το 26 και αποθηκεύουμε ξανά
    for i in range(len(key)):
        if cipher[i] > 25:
            cipher[i] = cipher[i] - 26

    # γεμίζουμε τον πίνακα ciphers με τους κρυπτοχαρακτηρες που προκύπτουν από
    # την άθροιση των αριθμών του ciphers με το 65
    for i in range(len(key)):
        x = cipher[i] + ord('A')
        cipherText += chr(x)

    # Επιστρέφουμε το κρυπτοκείμενο
```

```
    return cipherText

# Κωδικας για την υλοποιηση του RSA Algorithm

# υπολογιζει και επιστρεφει τον αριθμο (base^expo) mod m
def power(base, expo, m):
    res = 1
    base = base % m
    while expo > 0:
        if expo & 1:
            res = (res * base) % m
        base = (base * base) % m
        expo = expo // 2
    return res

# υπολογίζει το πολλαπλασιαστικό αντίστροφο ενός αριθμού
def modInverse(e, phi):
    for d in range(2, phi):
        if (e * d) % phi == 1:
            return d
    return -1

# δημιουργεί τα κλειδιά για τον αλγόριθμο
def generateKeys():
    p = 7919
    q = 1009

    n = p * q
    phi = (p - 1) * (q - 1)
```

```
    # επιλεγεί κατάλληλο e, όπου 1 < e < phi(n) και gcd(e, phi(n)) == 1
    e = 0
    for e in range(2, phi):
        if gcd(e, phi) == 1:
            break

    # υπολογίζει το d τέτοιο ώστε e * d ≡ 1 (mod phi(n))
    d = modInverse(e, phi)

    return e, d, n

# υπολογίζει το μέγιστο κοινό διαιρετή δυο αριθμών a, b
def gcd(a, b):
    while b != 0:
        a, b = b, a % b
    return a

# κρυπτογραφεί το μήνυμα με κλειδί (e, n)
def encryptRSA(m, e, n):
    return power(m, e, n)


def RSA_encrypt(m):
    e, d, n = generateKeys()

    print(f"Public Key (e, n): ({e}, {n})")
    print(f"Private Key (d, n): ({d}, {n})")
```

```
# Message
print(f"Original Message: {m}")

# Encrypt the message
C = encryptRSA(m, e, n)
print(f"Encrypted Message: {C}")
```