Κρυπτογραφία μετά τους Diffie-Hellman

Κ. Α. Δραζιώτης

17 Οκτωβρίου 2020

Περιεχόμενα

2ο Μέρος

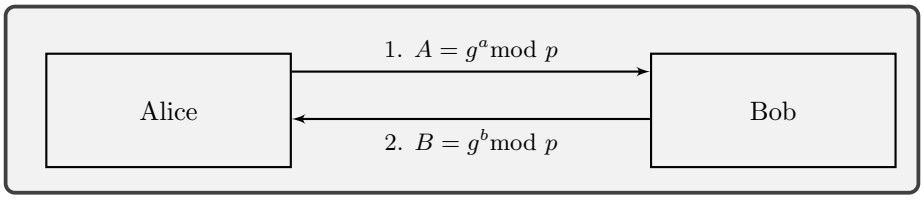
Κρυπτογραφία Δημοσίου Κλειδιού

Κεφάλαιο 1

Το πρόβλημα ανταλλαγής κλειδιού

1.1 Το Πρωτόκολλο Diffie-Hellman

[1] Ένα βασικό πρόβλημα, όταν χρησιμοποιούμε συμμετρική κρυπτογραφία, είναι το πρόβλημα ανταλλαγής του κλειδιού. Ας υποθέσουμε ότι η Alice και ο Bob δεν γνωρίζονται μεταξύ τους και θέλουν να ανταλλάξουν ένα $AES-$κλειδί, ώστε να μπορούν να κρυπτογραφούν με ασφάλεια μεταξύ τους. Επίσης, υποθέτουμε ότι η Εύα μπορεί να παρακολουθεί την μεταξύ τους επικοινωνία αλλά δεν μπορεί να εισάγει δεδομένα στην γραμμή επικοινωνίας. Η Alice και ο Bob συμφωνούν σε έναν πρώτο αριθμό p και έναν αριθμό g από το σύνολο $\{1, ..., p-1\}$. Κατόπιν η Alice διαλέγει ένα τυχαίο αριθμό a από το σύνολο $\{1, ..., p-1\}$ και ο Bob διαλέγει τον b από το ίδιο σύνολο. Οι αριθμοί a, b που επιλέξανε δεν δημοσιεύονται και κρατούνται ιδιωτικοί και είναι τα μυστικά κλειδιά. Κατόπιν γίνεται η παρακάτω "ανταλλαγή" :



Οι δυνάμεις g^a, g^b που υπολογίζουν οι Alice και Bob μπορούν να γίνουν αρκετά γρήγορα, όπως θα δούμε στην ενότητα 0. Εφόσον αντάλλαξαν τους αριθμούς A, B οι Alice και Bob μπορούν να καταλήξουν στον υπολογισμό ενός κοινού κλειδιού. Η Alice υπολογίζει,

$$B^a \bmod p = (g^b)^a \bmod p = g^{ba} \bmod p.$$

ενώ ο Bob

$$A^b \bmod p = (g^a)^b \bmod p = g^{ab} \bmod p.$$

Αλλά $g^{ba} = g^{ab} \bmod p$, επομένως καταλήγουν στο ίδιο κλειδί! Το πρωτόκολλο που μόλις περιγράψαμε ονομάζεται *πρωτόκολλο των Diffie-Hellman* [6] και έφερε επανάσταση στο χώρο της κρυπτογραφίας. Όπως είδαμε η Alice και ο Bob μπορούν να καταλήξουν γρήγορα στο ίδιο κλειδί (όπως θα δούμε στις παρακάτω

[1] ©2016-2020 K.Draziotis

ενότητες η ύψωση σε δύναμη μπορεί να γίνει σε πολυωνυμικό χρόνο, οπότε δικαιολογείται η λέξη *γρήγορα*). Γιατί όμως είναι ασφαλές το πρωτόκολλο; Η Εύα που παρακολουθεί την επικοινωνία των Alice και Bob γνωρίζει τους αριθμούς p, g, A και B. Τίθεται το ερώτημα αν μπορεί να υπολογίσει την συνάρτηση

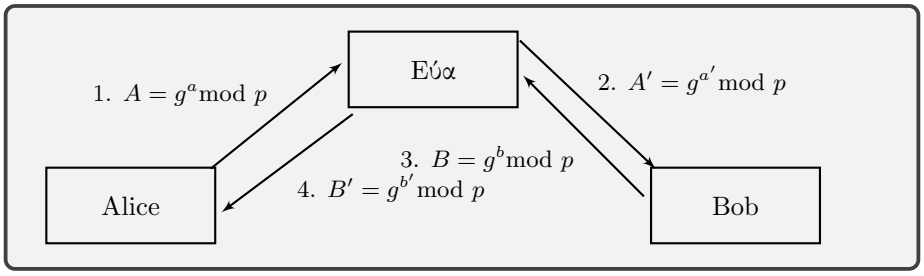
$$DH_p(g^a, g^b) = g^{ab} \bmod p.$$

Η συνάρτηση αυτή ονομάζεται *συνάρτηση των Diffie-Hellman* mod p και το πρόβλημα υπολογισμού της ονομάζεται *πρόβλημα των Diffie-Hellman*. Ο καλύτερος αλγόριθμος για τον υπολογισμό αυτής της συνάρτησης (μέχρι σήμερα) είναι ο GNFS : General Number Field Sieve και ο χρόνος υπολογισμού είναι :

$$\exp\left(\left(\sqrt[3]{\frac{64}{9}} + o(1)\right) n^{1/3} (\ln n)^{2/3}\right).$$

Είναι δηλαδή υποεκθετικού χρόνου. Ο αλγόριθμός αυτός υπολογίζει το x αν ως είσοδος δοθεί το $g^x (\bmod p)$, λύνει δηλαδή το πρόβλημα του διακριτού λογαρίθμου.

Αν για παράδειγμα, έχουμε έναν πρώτο αριθμό με 1024-bits, τότε για να υπολογίσουμε την DH_p χρειαζόμαστε περίπου χρόνο 2^{80} bits. Καταλήγουμε επομένως στο συμπέρασμα ότι, αν η Εύα απλά παρακολουθεί την επικοινωνία της Alice και του Bob , δεν μπορεί (στην πράξη) να βρει τα μυστικά τους κλειδιά a, b. Βέβαια ο πρώτος p πρέπει να είναι τουλάχιστον 1024 bits. Αν δώσουμε στην Εύα την δυνατότητα να μπορεί να παρεμβαίνει στην επικοινωνία, τότε θα δούμε ότι το πρωτόκολλο δεν είναι πλέον ασφαλές.



Η Alice στέλνει το $A = g^a \bmod p$ στον Bob. Το A το υποκλέπτει η Εύα και στην θέση του στέλνει το $A' = g^{a'} \bmod p$ για το a' που επέλεξε η Εύα. Ο Bob δεν μπορεί να καταλάβει ότι το A' είναι της Εύας, οπότε συνεχίζει στέλνοντας το δικό του $B = g^b \bmod p$ στην Alice. Πάλι η Εύα το υποκλέπτει και στην θέση του στέλνει το $B' = g^{b'} \bmod p$. Επίσης, η Alice δεν μπορεί να καταλάβει ότι το B' είναι της Εύας. Η Alice τώρα υπολογίζει το κλειδί της $K_A = g^{b'a} \bmod p$, ενώ ο Bob το $K_B = g^{a'b} \bmod p$. Ας υποθέσουμε ότι η Alice χρησιμοποιεί το κλειδί K_A ως κλειδί στον συμμετρικό αλγόριθμο AES για να κρυπτογραφήσει τον αριθμό της πιστωτικής της κάρτας για να τον στείλει στον Bob. Καθώς η Εύα γνωρίζει το K_A μπορεί να υποκλέψει το μήνυμα της Alice και να το αποκρυπτογραφήσει. Κατόπιν θα το ξανακρυπτογραφήσει με το κλειδί K_B του Bob και θα του το στείλει. Οπότε η Εύα έχει στα χέρια της τον αριθμό της πιστωτικής της Alice και ο Bob δεν έχει καταλάβει ότι έχει γίνει παραποίηση του μηνύματος της Alice. Αυτή

η επίθεση ονομάζεται Man in the Middle Attack. Ένας έξυπνος τρόπος για να αμυνθεί κάποιος σε αυτή την επίθεση είναι η χρήση των ψηφιακών πιστοποιητικών. Το πρωτόκολλο αυτό μαζί με την χρήση ψηφιακών πιστοποιητικών(για αυθεντικοποίηση) χρησιμοποιείται στο πρωτόκολλο SSL\TLS, που έχει ευρύτατη χρήση στο ηλεκτρονικό εμπόριο και τις ηλεκτρονικές συναλλαγές με τράπεζες. Το χαρακτηριστικό του είναι το γράμμα s που εμφανίζεται στο τέλος του http. Τέλος, το πρωτόκολλο Diffie-Hellman μπορεί να εφαρμοστεί, αν αντικαταστήσουμε την ομάδα $\mathbb{Z}_p^*$:*πολλαπλασιαστική ομάδα* mod p με μια άλλη κυκλική ομάδα G, όπως για παράδειγμα την ομάδα μίας ελλειπτικής καμπύλης πάνω στο $\mathbb{Z}_p$ ή την ομάδα της Ιακωβιανής μιας υπερελλειπτικής καμπύλης πάνω στο $\mathbb{Z}_p$ ή την ομάδα των κλάσεων των φανταστικών τετραγωνικών σωμάτων (Class Group of Imaginary quadratic number fileds).

1.2 Diffie-Hellman χωρίς αλληλεπίδραση

Για να γίνει η προηγούμενη ανταλλαγή κλειδιού, πρέπει η Alice και ο Bob να είναι και οι δύο online ταυτόχρονα. Το προηγούμενο πρωτόκολλο μπορεί να χρησιμοποιηθεί χωρίς την προηγούμενη απαίτηση. Η Alice διαλέγει ένα ζευγάρι (x, g^x) και δημοσιεύει τον αριθμό g^x. Ο Bob θέλει να κρυπτογραφήσει ένα μήνυμα m κάνοντας χρήση ενός συμμετρικού συστήματος κρυπτογράφησης που έχει συμφωνήσει με την την Alice. Ο Bob διαλέγει έναν τυχαίο αριθμό r και υπολογίζει τον αριθμό (όλα είναι mod p) $K = (g^x)^r = g^{xr}$. Στέλνει στην Alice $(g^r, E_{sym}(K, m) = c)$. Η Alice υπολογίζει τον αριθμό $(g^r)^x = g^{rx} = g^{xr} = K$. Τέλος, υπολογίζει $D_{sym}(K, c) = m$.

Κεφάλαιο 2

Πολυπλοκότητα & Μηχανές Turing

Hugo Cabret: So I figured, if the entire world was one big machine, I couldn't be an extra part. I had to be here for some reason...

Για να μελετήσουμε το πρωτόκολλο Diffi-Hellman, το σύστημα RSA, την ψηφιακή υπογραφή DSA κ.ά. συστήματα, πρέπει να έχουμε κάποιο βασικό μαθηματικό υπόβαθρο το οποίο αποτελείται κυρίως από θέματα της θεωρίας αριθμών : *πρώτοι αριθμοί*, *ισοδυναμίες*, πεπερασμένα σώματα, διακριτό λογάριθμο κ.α. Για μια εκτενή ανάλυση της θεωρίας αριθμών προτείνουμε από την Ελληνική βιβλιογραφία [1, 2, 3, 5]. Επίσης, θα παρουσιάσουμε ψευδοκώδικες από αρκετούς αλγορίθμους της θεωρίας αριθμών. Για την ανάλυση των αλγορίθμων μπορείτε να συμβουλευτείτε το κλασικό βιβλίο του Knuth [10]. Επομένως, θα χρειαστούμε κάποιους ορισμούς όσον αφορά την πολυπλοκότητα των αλγορίθμων.

2.1 Πολυπλοκότητα

Πολυπλοκότητα (complexity) θα ονομάζουμε το μέτρο της ποσότητας του χρόνου (ταχύτητα) ή/και του χώρου(μνήμη) που απαιτείται από έναν αλγόριθμο για είσοδο δεδομένων μήκους n. Μια κλάση πολυπλοκότητας είναι ένα σύνολο προβλημάτων που έχουν την ίδια πολυπλοκότητα (για να ορίσουμε με ακρίβεια μια κλάση πολυπλοκότητας χρειαζόμαστε ένα υπολογιστικό μοντέλο, όπως μία μηχανή Turing).[2] Στην κρυπτολογία παίζει μεγάλο ρόλο η θεωρία πολυπλοκότητας. Τα πρώτα κρυπτοσυστήματα δημόσιου κλειδιού βασίζονταν σε προβλήματα NP-πλήρη (NP:Non Determenistic Polynomial), όπως για παράδειγμα το πρόβλημα του γυλιού (subset sum problem).

[2] turingmaschine.klickagent.ch/einband Δίνεται μία προσομοίωση της μηχανής Turing, ανάκτηση 24-9-2015

Πρόβλημα του γυλιού - πρόβλημα απόφασης (NP-πλήρες)

Δίνεται ένα σύνολο Α από $n-$ θετικούς ακεραίους και ένας θετικός ακέραιος s. Υπάρχει υποσύνολο του Α που έχει άθροισμα s;

Πρόβλημα του γυλιού (NP-hard)

Δίνεται ένα σύνολο Α από $n-$ θετικούς ακεραίους και ένας θετικός ακέραιος s. Βρείτε (εφόσον υπάρχει) υποσύνολο του Α που έχει άθροισμα s.

Γρήγορα όμως έγινε κατανοητό ότι αυτό δεν ήταν αρκετό για την ασφάλεια των κρυπτοσυστημάτων δημόσιου κλειδιού. Το κρυπτοσύστημα του γυλιού *έσπασε* από τον Adi Shamir μετά από τρία χρόνια αφότου παρουσιάστηκε. Ο λόγος είναι ότι στα NP-hard προβλήματα μετράμε την πολυπλοκότητα της χειρότερης περίπτωσης (worst case complexity). Ενώ στα κρυπτοσυστήματα θέλουμε για όλα τα στιγμιότυπα του προβλήματος, το πρόβλημα να είναι δύσκολο και όχι μόνο για κάποιες συγκεκριμένες παραμέτρους. Αυτό στην πράξη είναι δύσκολο να μετρηθεί. Η νέα έννοια που προέκυψε για την ασφάλεια των κρυπτοσυστημάτων δημόσιου κλειδιού είναι η συνάρτηση μίας φοράς (one way function). Δηλαδή συναρτήσεις που είναι *εύκολο* να υπολογιστούν αλλά *δύσκολο* να αντιστραφούν. Εύκολο και δύσκολο με όρους της θεωρίας υπολογισμού σημαίνει ότι υπάρχει πιθανοτικός πολυωνυμικού χρόνου αλγόριθμος ή όχι αντίστοιχα. Η ύπαρξη των συναρτήσεων αυτών δεν έχει αποδειχθεί μέχρι σήμερα, αλλά αν υπάρχουν, τότε ισχύει $P \neq NP$. Στα ασφαλή κρυπτοσυστήματα ο νόμιμος χρήστης μπορεί να αποκρυπτογραφεί (γνωρίζοντας κάποια μυστική πληροφορία), ενώ ο μη νόμιμος δεν μπορεί να αποκρυπτογραφεί εύκολα (ή αποδοτικά), αν δεν γνωρίζει την μυστική πληροφορία. Η κλάση των NP-hard προβλημάτων *δεν είναι καλή για την κρυπτογραφία*, διότι αυτά τα προβλήματα είναι δύσκολα στην χειρότερη περίπτωση. Αλλά ακόμη και αν υπήρχαν προβλήματα σε αυτή την κλάση που ήταν δύσκολα κατά μέσο όρο (hard on average), τότε θα έπρεπε αυτά να είναι εύκολα αν είχαμε στην διάθεση μας μια βοηθητική πληροφορία. Διαφορετικά θα ήταν δύσκολα και για τον νόμιμο χρήστη να αποκρυπτογραφεί. Επομένως, τα ασφαλή κρυπτοσυστημάτα (δημόσιου κλειδιού) πρέπει να έχουν τις εξής δύο ιδιότητες :
(i). Είναι εύκολο να λύνω αυτά τα προβλήματα, αν γνωρίζω μία μυστική πληροφορία,
(ii). Είναι δύσκολο (κατά μέσο όρο) να λύνω αυτά τα προβλήματα, όταν δεν γνωρίζω την μυστική πληροφορία.

Το πρόβλημα του κατά πόσο μπορούμε να χρησιμοποιήσουμε NP-hard προβλήματα στην κρυπτογραφία τέθηκε πρώτη φορά από τους Diffie-Hellman. Αυτοί πρότειναν την έννοια της Trapdoor Function (TDF), χωρίς όμως να προτείνουν κάποιο συγκεκριμένο παράδειγμα. Η πρώτη κατασκευή TDF έγινε λίγο αργότερα με την κατασκεύη του RSA.

Το πρόβλημα της χρήσης προβλημάτων που είναι δύσκολα στην χειρότερη περίπτωση επανήλθε από τον Miklos Ajtai, ο οποίος πρότεινε χρήση προβλημάτων που είναι δύσκολα στην χειρότερη περίπτωση, κατασκευάζοντας όμως, μια αναγω-

γή από το πρόβλημα της χειρότερης περίπτωσης σε ένα πρόβλημα δύσκολο κατά μέσο όρο. Ήταν αυτή η εργασία που έστρεψε το βλέμμα των ειδικών στα πλέγματα και την χρήση αυτών στην κρυπτογραφία.

Θα δούμε παρακάτω ότι αυτές τις δύο προϋποθέσεις τις έχουν οι συναρτήσεις καταπακτής (Trapdoor Functions - TDF). Θα κάνουμε μια μικρή εισαγωγή στις κλάσεις πολυπλοκότητας P, NP, NP-complete, NP-hard και PSPACE, αφού δούμε κάποια βασικά στοιχεία για τις μηχανές Turing.

2.1.1 Μηχανές Turing

Computers are useless.They can only give you answers.
Pablo Picasso

Μία μηχανή Turing αποτελείται από μία επτάδα $(Q, S, \sqcup, A, q_0, F, \delta)$:

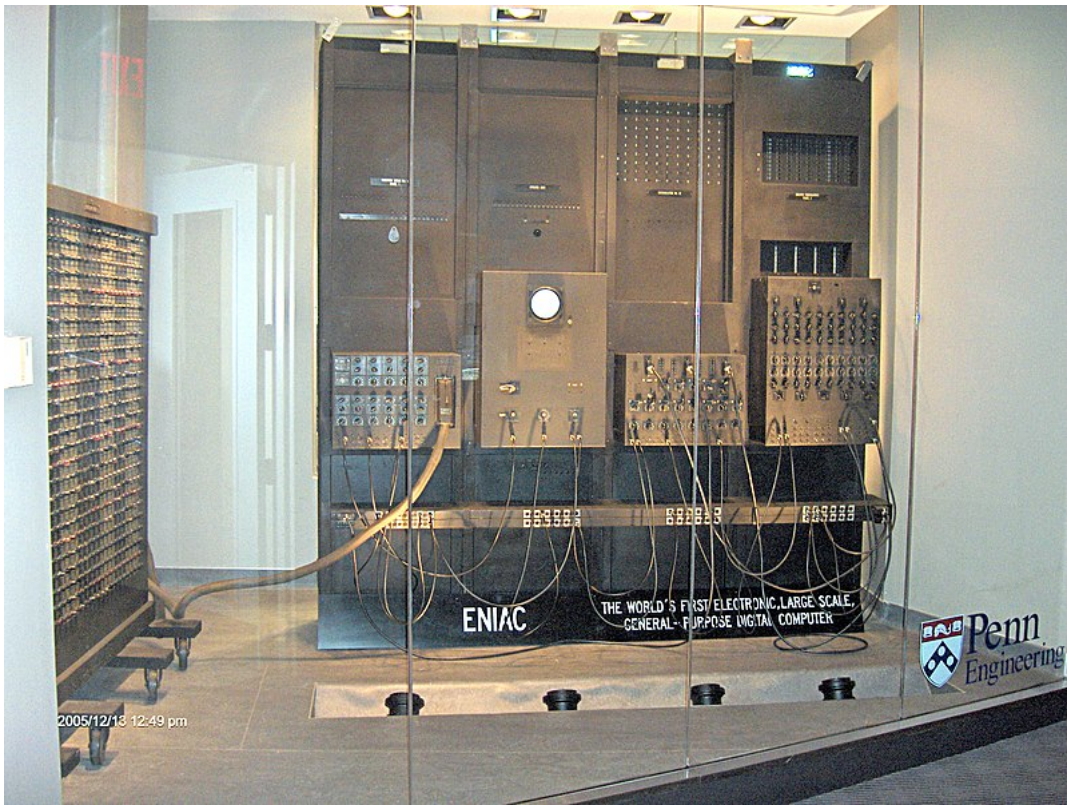
- Ένα πεπερασμένο σύνολο Q (το σύνολο των καταστάσεων)
- Ένα πεπερασμένο σύνολο S (αλφάβητο)
- Ένα στοιχείο $\sqcup \in S$ (το σύμβολο του κενού)
- $A \subset Q$ το οποίο ονομάζεται εξωτερικό αλφάβητο και $\sqcup \notin Q$
- Μία αρχική κατάσταση $q_0 \in Q$
- Από ένα σύνολο $F \neq \emptyset$ υποσύνολο του Q που αποτελείται από τις τελικές καταστάσεις (Halting states)
- Μία συνάρτηση μετάβασης $\delta : Q \backslash F \times S \to Q \times S \times \{-1, 0, 1\}$

Το παραπάνω σύνολο είναι το software μίας μηχανής Turing. Το Hardware αποτελείται από μία ταινία άπειρου μήκους από δεξιά που είναι χωρισμένη σε κελιά, όπου κάθε κελί μπορεί να περιέχει ένα στοιχείο από το αλφάβητο S. Επίσης, περιέχει μία κεφαλή που κινείται κατά μήκος της ταινίας (ή ισοδύναμα η κεφαλή είναι σταθερή και μετακινεί την ταινία) και μπορεί να γράφει και να σβήνει στα κελιά. Η συμπεριφορά της μηχανής ελέγχεται από ένα πεπερασμένο αυτόματο. Σε κάθε βήμα υπολογισμού το αυτόματο βρίσκεται σε μία κατάσταση $q \in Q$ και η δ καθορίζει την μετάβαση της μηχανής Turing (MT) στην νέα κατάσταση. Αν για παράδειγμα η κεφαλή βρίσκεται στο κελί που περιέχει το s, τότε $\delta(q, s) = (q', s', \Delta_p)$, όπου q' είναι η νέα κατάσταση s' το νέο σύμβολο και Δ_p υποδηλώνει την μετακίνηση της κεφαλής μία θέση δεξιά ή αριστερά αν έχει την τιμή 1 ή -1 αντίστοιχα. Η αρχική κατάσταση της MT δίνεται με ένα string του A, την αρχική κατάσταση q_0 και η κεφαλή είναι τέρμα αριστερά (στην αρχή της ταινίας). Προφανώς μία MT υλοποιεί έναν αλγόριθμο. Το αντίστροφο ονομάζεται θέση των Church-Turing.

Church-Turing Thesis: Κάθε αλγόριθμος υλοποιείται με μία MT.

Η θέση Church-Turing δεν είναι κάποιο θεώρημα, μπορεί όμως, να υποστηριχθεί από εμπειρικά αποτελέσματα.

Ας δούμε ένα παράδειγμα. Ας υποθέσουμε ότι πάνω στην ταινία μιας MT είναι τυπωμένη η ακολουθία $q = 101$. Θέλουμε η μηχανή μας να κάνει την πράξη $p + q$, όπου $p = (001)$, δηλαδή, να γίνει η πράξη $(101) + (001)$ (το αποτέλεσμα είναι (110)). Αυτή η MT χρειάζεται τρεις καταστάσεις, τις οποίες συμβολίζουμε a, b, c. Άρα $Q = \{a, b, c\}$. Επίσης, κάθε κατάσταση θα πρέπει να περιγράφει τι να κάνει η MT, αν διαβάσει η κεφαλή της ένα στοιχείο από το σύνολο $S = \{0, 1, \sqcup\}$. Η ιδέα του Turing να χρησιμοποιήσει ως αλφάβητο δυαδικά ψηφία, οφείλεται στον Shanon, που απέδειξε ότι τα (relay) circuits και η άλγεβρα του Boole (Boolean algebra) είναι ισοδύναμα [3] (π.χ. ο ENIAC (1943-1945) ήταν δεκαδικός υπολογιστής (decimal computer) και όχι δυαδικός (digital computer)) [4].



Σχήμα 1: Η μηχανή ENIAC. Άδεια CC BY-SA 3.0 (από Wikipedia) `https://commons.wikimedia.org/wiki/File:ENIAC_Penn1.jpg`.

Η κατάσταση a.
Περιγράφεται από τις επόμενες τρεις τετράδες: ($\sqcup$,Print $\sqcup$,$\Delta_p = -1$, b). Δηλαδή, αν η κεφαλή είναι πάνω σε ένα κενό κελί, τότε άφησε το κελί κενό (Print $\sqcup$). Κατόπιν μετακίνησε την κεφαλή αριστερά και τέλος πήγαινε στην κατάσταση b. Η

[3] A Symbolic Analysis of Relay and Switching Circuits, Master thesis, MIT (1937)
[4] The Annotated Turing, C.Petzold (Wiley)

συνάρτηση μετάβασης τότε δίνεται από τον τύπο $\delta(a, \sqcup) = (b, \sqcup, -1)$. Οι άλλες δύο τετράδες που περιγράφουν την συνάρτηση μετάβασης για την κατάσταση a είναι : (0,Print 0,$\Delta_p = 1$, a), (1,Print 1,$\Delta_p = 1$, a).

Η κατάσταση b περιγράφεται με τις τετράδες :
($\sqcup$,Print 1,$\Delta_p = 1$, c), (0,Print 1,$\Delta_p = -1$, c), (1,Print 0,$\Delta_p = -1$, b).

Η κατάσταση c περιγράφεται με τις τετράδες :
($\sqcup$,Print $\sqcup$,$\Delta_p = -1$, halt), (0,Print 0,$\Delta_p = 1$, c), (1,Print 1,$\Delta_p = 1$, c).

Η q_0 είναι η κατάσταση a. Επίσης, υποθέτουμε ότι η κεφαλή βρίσκεται ένα κελί δεξιά του αριθμού q (δηλαδή, σε προηγούμενο στάδιο έχει τυπώσει τον αριθμό και έχει σταματήσει ακριβώς ένα κελί δεξιά του αριθμού). Εφόσον, η MT ξεκινάει από την κατάσταση a και βρίσκεται σε κενό κελί, θα μετακινηθεί αριστερά και θα μεταβεί στην κατάσταση b. Το τρέχον κελί δεν είναι κενό και στην περίπτωση μας είναι 1, οπότε τυπώνει μηδέν και πάει μια θέση αριστερά και παραμένει στην κατάσταση b. Το τρέχον κελί είναι 0, οπότε τυπώνει 1, η κεφαλή μετακινείται μία θέση αριστερά και μεταβαίνει στην κατάσταση c. Το τρέχον κελί τώρα είναι 1 και βρισκόμαστε στην κατάσταση c. Οπότε 1 (δηλαδή δεν αλλάζει η τιμή του κελιού) και η κεφαλή κινείται μία θέση δεξιά και παραμένουμε στην κατάσταση c. Το τρέχον κελί τώρα έχει την τιμή 0, οπότε δεν αλλάζει η τιμή και μετακινείται η κεφαλή μία θέση δεξιά και καταλήγουμε σε κενό κελί, οπότε η κεφαλή μετακινείται μία θέση αριστερά και τερματίζει το πρόγραμμα. Τώρα στην ταινία βρίσκεται ο αριθμός (110).

Το αρχικό παράδειγμα του Turing ήταν μια MT που τυπώνει την ακολουθία 010101... με ένα κενό ανάμεσα στα σύμβολα $0, 1$. Αυτή η μηχανή Turing μπορεί να περιγραφεί με τον παρακάτω πίνακα.

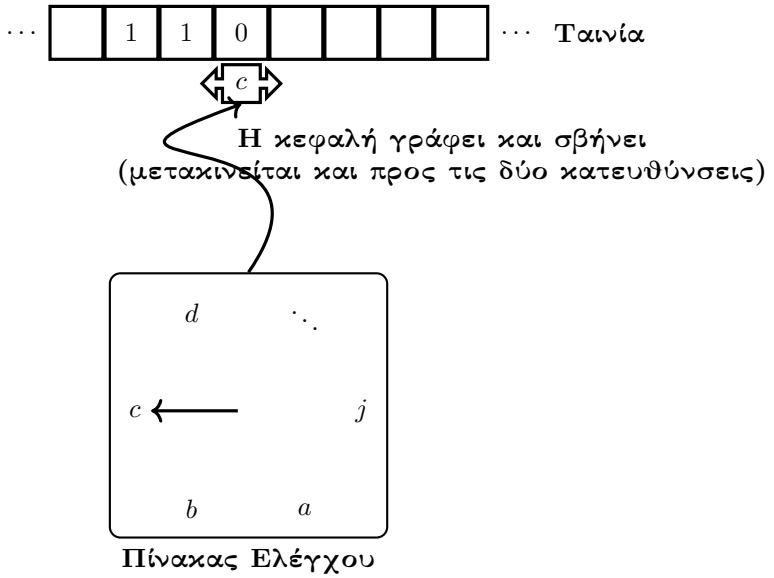
Κατάσταση	Σύμβολο	Πράξη στην ταινία	Τελική κατάσταση
b	$\sqcup$	Print 0, $\Delta_p = 1$	c
c	$\sqcup$	Print $\sqcup$, $\Delta_p = 1$	e
e	$\sqcup$	Print 1, $\Delta_p = 1$	f
f	$\sqcup$	Print $\sqcup$, $\Delta_p = 1$	b

Η μηχανή έχει αρχική κατάσταση b και βρίσκεται στην αρχή της ταινίας. Εύκολα μπορούμε να δούμε ότι η μηχανή θα τυπώσει την ακολουθία $0 \sqcup 1 \sqcup 0 \sqcup 1$...

Κλάσεις Πολυπλοκότητας

Η κλάση πολυπλοκότητας P αποτελείται από όλα τα προβλήματα απόφασης (δηλ. η απάντηση είναι ΝΑΙ ή ΟΧΙ) που μια MT μπορεί να τα απαντήσει σε πολυωνυμικό χρόνο. Για να ορίσουμε τα NP προβλήματα χρειάζεται να ορίσουμε τις μη-ντετερμινιστικές μηχανές Turing(NDTM). Πριν μιλήσουμε γι'αυτές, μπορούμε να δώσουμε έναν απλό ορισμό της κλάσης NP.

Ορισμός 2.1.1. *(απλός ορισμός) Ένα πρόβλημα απόφασης ανήκει στην κλάση*



Σχήμα 2: Μία μηχανή Turing έχει ένα πίνακα ελέγχου S (software) καθώς και μία ταινία άπειρου μήκους προς τα δεξιά με μία κεφαλή που έχει την δυνατότητα εγγραφής και σβησίματος (hardware), καθώς και μετακίνησής της επί της ταινίας. Το σχήμα είναι παραλλαγή του σχήματος του Sebastian Sardina και έχει άδεια CC. (http://www.texample.net/tikz/examples/turing-machine-2)

NP, αν υπάρχει πολυωνυμικός αλγόριθμος που ελέγχει αν μια δοθείσα πιθανή λύση είναι πράγματι λύση.

Σε αυτά τα προβλήματα μπορεί να μην είναι εύκολο να βρούμε έναν αλγόριθμο που τα λύνει, αλλά αν έχουμε μια λύση, μπορούμε αυτό να το επαληθεύσουμε σε πολυωνυμικό χρόνο. Αν οι δύο κλάσεις P και NP είναι ίσες, τότε σε όλα τα προβλήματα που μπορούμε γρήγορα να ελεγξουμε τις λύσεις, είναι και εξίσου εύκολο να υπολογίσουμε τις λύσεις. Το πρόβλημα αυτό μέχρι σήμερα είναι άλυτο, δηλ. δεν γνωρίζουμε αν ισχύει $P = NP$.

κλάση P : Μπορώ να βρω αποδοτικό αλγόριθμο που να λύνει τα προβλήματα-quick solvable.
κλάση NP : Μπορώ να βρω αποδοτικό αλγόριθμο που να επαληθεύει μια λύση-quick checkable.

NDTM

Οι NDTM σε μια συγκεκριμένη κατάσταση μπορεί να έχουν πολλές δυνατότητες επιλογών και όχι μόνο μία όπως συμβαίνει στις κλασικές (δηλ. ντετερμινιστικές) μηχανές Turing. Η κλάση NP αποτελείται από όλα τα προβλήματα απόφασης που

μπορούν να απαντηθούν σε πολυωνυμικό χρόνο από μία (NDTM). Μπορούμε να φανταστούμε ότι μια NDTM, αποτελείται από πολλές μηχανές Turing που είναι συνδεδεμένες παράλληλα. Σε μια NDTM, υπάρχει μια εντολή goto both line 1,2. Επομένως ο υπολογισμός διακλαδίζεται σε δύο παράλληλους υπολογισμούς οποτεδήποτε υπάρχει η εντολή goto both line 1,2. Αν η έξοδος της μηχανής έχει τουλάχιστον ένα μονοπάτι που δίνει απάντηση ΝΑΙ, τότε λέμε ότι το αρχικό μας πρόβλημα έχει απάντηση ΝΑΙ. Αν όλα τα μονοπάτια οδηγούν στην απάντηση ΌΧΙ, τότε η απάντηση στο αρχικό μας πρόβλημα είναι ΌΧΙ. Με αυτό τον τρόπο προβλήματα εκθετικού χρόνου, μια NDTM μπορεί να τα λύσει σε πολυωνυμικό χρόνο.

Ορισμός 2.1.2. *Ένα πρόβλημα λέγεται NP, αν υπάρχει ένα μη ντετερμινιστικό πρόγραμμα (δηλαδή, περιέχει την εντολή goto both line 1,2), που δίνει απάντηση ΝΑΙ αν και μόνο αν υπάρχει ένα μονοπάτι που δίνει απάντηση ΝΑΙ.*

Μπορούμε να προσομοιάσουμε το μη ντετερμινιστικό πρόγραμμα με ένα ντετερμινιστικού πρόγραμμα αλλά με ένα επιπλέον στοιχείο, το οποίο θα λέει πιο δρόμο να ακολουθήσει το πρόγραμμα κάθε φορά που συνάντα ένα goto both. Το επιπλέον στοιχείο ονομάζεται πιστοποιητικό ή μάρτυρας (certificate ή witness). Αν το ντετερμινιστικό πρόγραμμα $C(x, w)$ το ονομάσουμε πιστοποιητή (certifier ή prover) που δέχεται δύο εισόδους, ένα στιγμιότυπο του προβλήματος μας x που έχει απάντηση ΝΑΙ και το πιστοποιητικό w, τότε σε πολυωνυμικό χρόνο απαντά ΝΑΙ. Αντιστρόφως, αν υπάρχει ένα πιστοποιητικό w τέτοιο ώστε για το τυχαίο στιγμιότυπο x του προβλήματος να έχω $C(x, w) =$ΝΑΙ, τότε το x έχει απάντηση ΝΑΙ. Μπορούμε να διατυπώσουμε τον προηγούμενο ορισμό κάνοντας χρήση των πιστοποιητικών.

Ορισμός 2.1.3. *(2ος ορισμός) Ένα πρόβλημα καλείται NP αν υπάρχει πολυωνυμικού χρόνου πιστοποιητής.*

Ακόμη πιο φορμαλιστικά μπορούμε να γράψουμε:
$X \in NP$ αν υπάρχει πιστοποιητικό $w = poly(|x|)$ με $C(x, w) =$ΝΑΙ, όπου C είναι στην κλάση P.

Ένα παράδειγμα προβλήματος που ανήκει στην κλάση NP είναι το πρόβλημα εύρεσης ενός δρόμου σε ένα γράφο, που να περνάει μόνο μία φορά από κάθε κόμβο (Hamiltonian path problem).
Πιστοποιητικό. Μια μετάθεση των κόμβων.
Πιστοποιητής. Ελέγχει ότι η μετάθεση περιέχει κάθε κόμβο μία φορά και ότι υπάρχει ακμή μεταξύ δύο γειτονικών κόμβων.
Ο Πιστοποιητής είναι στην κλάση P, άρα το πρόβλημα είναι NP. Το πρόβλημα του ενός εκατομμυρίου δολαρίων είναι αν P=NP (Godel 1956 & Cook 1971).

Αν θεωρήσουμε τα προβλήματα x των οποίων η απάντηση είναι ΌΧΙ, και διατυπώσουμε τους προηγούμενους ορισμούς, έχουμε μια νέα κλάση που ονομάζεται co-NP. Πιστεύουμε ότι αυτή η κλάση έχει πολύ διαφορετικά προβλήματα από την κλάση NP (πράγμα που δεν ισχύει με την κλάση P, διότι co-P=P). Ας δούμε ένα παράδειγμα. Δοθέντος ενός γραφήματος G, θέλουμε να απαντήσουμε με ΝΑΙ/ΌΧΙ για το επόμενο πρόβλημα: *Το G δεν έχει Hamiltonian μονοπάτι.* Ε-

ύκολα μπορούμε να απαντήσουμε αυτή την ερώτηση, αλλά δεν ισχύει το ίδιο για το πρόβλημα: *Το G έχει Hamiltonian μονοπάτι*;[5] Πιστεύουμε ότι co-NP≠ NP.

Η κλάση NP καθώς και η co-NP περιέχει την κλάση P. Υπάρχουν προβλήματα στην τομή τους; Το πρόβλημα Primes, δηλαδή δοθέντος ενός αριθμού να απαντήσουμε, Ναι είναι πρώτος ή Όχι δεν είναι πρώτος, αποδεικνύεται ότι ανήκει στην τομή τους (Pratt's Theorem). Επίσης, με το θεώρημα AKS αποδείχτηκε ότι το πρόβλημα Primes ανήκει στην κλάση P. Επίσης και το επόμενο πρόβλημα είναι σημαντικό για την κρυπτογραφία.

FACTOR (πρόβλημα απόφασης)

Δίνονται δύο θετικοί ακέραιοι n, k $(n > k)$. Υπάρχει διαιρέτης του n που είναι μικρότερος του k;

Το πρόβλημα αυτό είναι NP διότι ένας θετικός ακέραιος $p < k$ με p διαιρέτης του k, είναι ένα πιστοποιητικό για τα ΝΑΙ-στιγμιότυπα. Επίσης είναι co-NP, διότι η παραγοντοποίηση του n σε πρώτους παράγοντες είναι ένα πιστοποιητικό για τα ΟΧΙ-στιγμιότυπα και υπάρχει πολυωνυμικός πιστοποιητής, ο αλγόριθμος AKS.

Το πρόβλημα της παραγοντοποίησης FACTOR, είναι στην τομή NP και co-NP. Δεν έχει αποδειχτεί μέχρι σήμερα αν είναι στην κλάση P.

Αν ίσχυε όμως $P = NP$, τότε θα υπήρχε αλγόριθμος πολυωνυμικού χρόνου που λύνει το FACTOR. Η επίδραση του στην οικονομία θα ήταν τεράστια. Βέβαια αν $P \neq NP$ αυτό δεν συνεπάγεται ότι δεν υπάρχει αυτός ο πολυωνυμικός αλγόριθμος.

Υπάρχουν όμως προβλήματα στην κλάση NP που είναι δύσκολα, τα ονομάζουμε NP-πλήρη (NP-complete), με την έννοια όλα τα προβλήματα της κλάσης NP ανάγονται πολυωνυμικά σε αυτό. Τα πρόβληματα τα οποία είναι τουλάχιστον τόσο δύσκολα όσο τα NP-complete ορίζουν μια νέα κλάση που την ονομάζουμε NP-hard. Σε αυτή την κλάση υπάρχουν και προβλήματα που δεν είναι προβλήματα απόφασης. Δηλαδή, υπάρχουν προβλήματα που δεν ανήκουν στην κλάση NP αλλά ανήκουν στην κλάση NP-hard.

Αν P≠NP τότε τα NP-hard προβλήματα δεν μπορούν να λυθούν σε πολυωνυμικό χρόνο από μία MT. Ένας ισοδύναμος ορισμός της κλάσης NP-hard είναι ο εξής: Ένα πρόβλημα Λ καλείται NP-hard αν έχω ένα μαντείο[6] (που απαντά σε πολυωνυμικό χρόνο) για το Λ, τότε με χρήση αυτού του μαντείου μπορώ να λύσω οποιοδήποτε πρόβλημα της κλάσης NP σε πολυωνυμικό χρόνο.

Τέλος, τα NP-complete βρίσκονται στην τομή NP-hard∩ NP. Πιο φορμαλιστικά μπορούμε να δώσουμε τον εξής ορισμό για τα NP-πλήρη προβλήματα. Χρειαζόμαστε τον παρακάτω ορισμό.

[5] Δηλαδή υπάρχει ένας απλός κύκλος στο γράφο, που επισκέπτεται κάθε κόμβο;

[6] με τον όρο *μαντείο* εννοούμε μια ντερμενιστική μηχανή. Ενώ ο όρος τυχαίο μαντείο (random oracle) είναι μια μηχανή που σε μία είσοδο απαντά με μια τυχαία ακολουθιά χαρακτήρων. Με την ίδια είσοδο πάντα απαντά με την ίδια έξοδο.

Ορισμός 2.1.4. *Ένα πρόβλημα L_1 λέμε ότι ανάγεται πολυωνυμικά στο πρόβλημα απόφασης L_2, αν υπάρχει αλγόριθμος που λύνει το L_1, ο οποίος χρησιμοποιεί ως υπορουτίνα έναν αλγόριθμο που λύνει το L_2 και είναι πολυωνυμικός αν ο αλγόριθμος που λύνει το L_2 είναι πολυωνυμικός. Γράφουμε $L_1 \leq_p L_2$.*

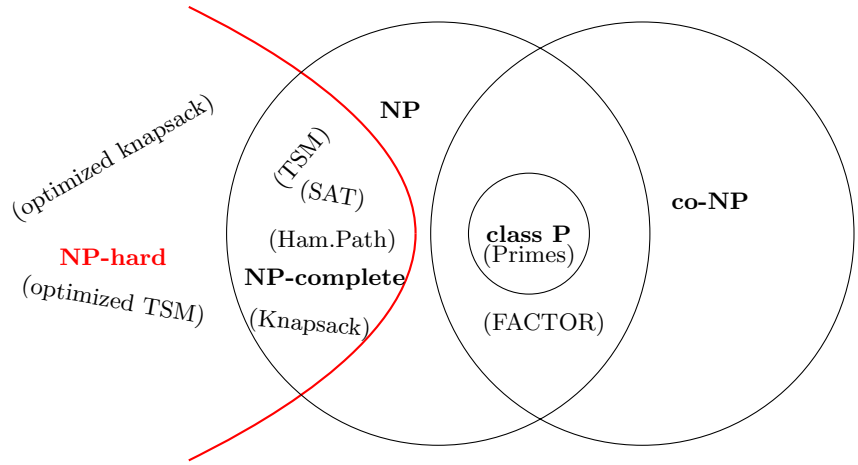
Ορισμός 2.1.5. *Ένα πρόβλημα απόφασης L λέγεται NP-πλήρες αν*
i. $L \in NP$
ii. $L_1 \leq_p L$ *για κάθε* $L_1 \in NP$.

Το πρόβλημα (απόφασης) του γυλιού (knapsack problem) είναι NP-πλήρες. Τώρα ένα πρόβλημα L_1 τέτοιο ώστε να υπάρχει L_2 ένα NP-πλήρες πρόβλημα με $L_2 \leq_p L_1$, λέμε ότι το L_1 είναι NP-hard. Τα NP-hard δεν είναι απαραίτητα προβλήματα απόφασης. Π.χ. το optimization knapsack, που είναι το πρόβλημα εύρεσης του υποσυνόλου που απαντάει και στη ερώτηση αν υπάρχει υποσύνολο που το άθροισμα του να ισούται με s, είναι NP-hard.

Στην περίπτωση του προβλήματος του γυλιού το πρόβλημα απόφασης και το αντίστοιχο πρόβλημα εύρεσης είναι ισοδύναμα, με την έννοια ότι αν έχουμε λύση στο ένα τότε έχουμε λύση και στο άλλο. Ας υποθέσουμε ότι έχουμε ένα μαντείο που δέχεται ως είσοδο ένα σύνολο $A = \{a_1, a_2, ..., a_n\}$ και ένα θετικό ακέραιο s, και δίνει ως έξοδο, true αν υπάρχει υποσύνολο του A που να έχει άθροισμα s, διαφορετικά false. Δηλαδή αυτό το μαντείο λύνει το πρόβλημα απόφασης. Τότε, αν κάνουμε n κλήσεις στο μαντείο μπορούμε να λύσουμε το optimization knapsack. Η πρώτη ερώτηση στο μαντείο-knapsack είναι το ζεύγος $(A - \{a_n\}, s - a_n)$. Αν το μαντείο απαντήσει *true*, τότε το $x_n = 1$ διαφορετικά το $x_n = 0$. Η δεύτερη ερώτηση είναι $(A - \{a_n, a_{n-1}\}, s - x_n a_n - a_{n-1})$. Αν το μαντείο απαντήσει true, τότε έχουμε $x_{n-1} = 1$, διαφορετικά $x_{n-1} = 0$. Άρα, με $n-$ερωτήσεις μπορούμε να λύσουμε το optimization knapsack. Τα προβλήματα που έχουν αυτήν την ιδιότητα ονομάζονται *αυτοανάγωγα* (self-reducible). Το προηγούμενο παράδειγμα δείχνει ότι κάποια NP-hard είναι το ίδιο δύσκολα με κάποια NP-πλήρη προβλήματα. Αυτό δεν ισχύει για την περίπτωση του προβλήματος του περιπλανώμενου πωλητή (TSM).

Να αναφέρουμε και την κλάση PSPACE που αποτελείται από όλα τα προβλήματα απόφασης που για να απαντηθούν από μία ντετερμινιστική μηχανή Turing, χρησιμοποιούν poly(n) κελιά από την ταινία, για είσοδο μήκους n. Για παράδειγμα $P \subseteq PSPACE$ διότι τα προβλήματα της κλάσης P καταναλώνουν πολυωνυμικό *χώρο* για να δώσουν την απάντηση στο πρόβλημα και αυτό διότι χρειάζονται πολυωνυμικό χρόνο για να εκτελεστούν.

Τέλος, να αναφέρουμε ότι υπάρχουν προβλήματα της κλάσης P που στην ουσία δεν μπορούμε να τα λύσουμε. Για παράδειγμα οι Kuratowski και Wagner, έδειξαν ότι η ιδιότητα Planar στους γράφους (γραφήματα που οι ακμές τους δεν τέμνονται) είναι στην κλάση P. Παρόλα αυτά δεν έχουμε κάποιον πρακτικό αλγόριθμο.



Σχήμα 3: Υποθέσαμε ότι P≠NP και NP≠co-NP

2.1.2 Πολυπλοκότητα Βασικών Πράξεων

Έστω a, b δύο $n-$bit ακέραιοι. Τότε μπορούμε να τους προσθέσουμε ή να τους αφαιρέσουμε σε γραμμικό χρόνο $O(n)$. Ο πολλαπλασιασμός των a, b απαιτεί χρόνο $O(n^2)$. Ο πολλαπλιασιασμός αυτός ήταν γνωστός στους αρχαίους πολιτισμούς (Αιγυπτιους, Βαβυλωνιους κλπ). Υπάρχουν και υποτετραγωνικου χρόνου αλγόριθμοι. Η ιδέα που χρησιμοποιούμε είναι να ανάγουμε τον πολλαπλασιαμό των ακεραίων σε πολλαπλασιασμό δύο ακέραιων πολύωνύμων. Αν εφαρμόσουμε τον πολλαπλασιασμό του Karatsuba(1960), τότε ο χρόνος μειώνεται σε $O(n^{1.585})$. Για να καταλάβουμε τον πολλαπλασιασμό του Karatsuba ας δούμε ένα παράδειγμα. Έστω $x = 1782,\ y = 1659$. Για κάθε $m \in \mathbb{Z}_{>0}$ με $m < n$, γράφουμε το $x = B^m \cdot x_2 + x_1,\ y = B^m \cdot y_2 + y_1$ και $x_1, y_1 < B^m$. Διαλέγουμε $B^m = 100$ (επίσης θα δούλευε αν το $B = 10$ και $m = n/2$). Οπότε έχουμε

$$1782 \times 1659 = (17 \cdot 100 + 82) \cdot (16 \cdot 100 + 59) = z_2 \cdot 100^2 + z_1 \cdot 100 + z_0,$$

όπου

$$z_0 = x_1 y_1,\ z_1 = x_1 y_2 + x_2 y_1,\ z_2 = x_2 y_2.$$

Έχουμε, $z_0 = 4838,\ z_1 = 2315,\ z_2 = 272$, άρα

$$1782 \times 1659 = 2720000 + 231500 + 4838 = 2956338.$$

Ο προηγούμενος κλασικός πολλαπλασιασμός απαιτεί τέσσερις πολλαπλασιασμούς. Οπότε, αν γενικεύσουμε το προηγούμενο έχουμε χρόνο $T(n)$ που ικανοποιεί την εξίσωση $T(n) = 4T(n/2) + O(n)$. Ο Karatsuba παρατήρησε ότι το $z_1 = (x_2 + x_1) \cdot (y_2 + y_1) - z_2 - z_0$. Δηλαδή συνολικά τρεις πολλαπλασιασμούς (με κάποιες επιπλέον προσθέσεις). Οπότε, έχουμε χρόνο $T(n) = 3T(n/2) + O(n)$. Η τελευταία εξίσωση αποδεικνύεται ότι δίνει $T(n) = \Theta(n^\alpha),\ \alpha = \log_2 3$ (δείτε την άσκηση 2.2). Έτσι,

το $z_1 = (17+82)\cdot(16+59) - z_2 - z_1 = 99\cdot 75 - 272 - 4838 = 2315$. Γενικά αν x, y έχουν ρ ψηφία και ρ άρτιος, μπορούμε να γράψουμε τον $x = 10^{\rho/2}x_2 + x_1,\ y = 10^{\rho/2}y_2 + y_1$. Αν n περιττός $x = 10^{(\rho+1)/2}x_2 + x_1,\ y = 10^{(\rho+1)/2}y_2 + y_1$.

Αλγόριθμος 2.1.1. : Πολλαπλασιασμός του Karatsuba
Είσοδος. a, b ακέραιοι
Έξοδος. $a \cdot b$

```
1  def karatsuba(a, b)
2  if a < 100 or b < 100 then
   |   return a · b
   end
3  m = max(log10(a), log10(b))
4  m2 = floor(m/2)
5  high(a) = take the first m2 decimal digits of a
6  low(a) = take the last m2 decimal digits of a
7  high(b) = take the first m2 decimal digits of b
8  low(b) = take the last m2 decimal digits of b
9  z0 = karatsuba(low(a), low(b))
10 z1 = karatsuba((low(a) + high(a)), (low(b) + high(b)))
11 z2 = karatsuba(high(a), high(b))
12 print (z2 · 10^(2m2) + (z1 − z2 − z0) · 10^m2 + z0
```

Μια βελτίωση του Karatsuba είναι ο αλγόριθμος των Toom-Cook και για "μεγάλους" ακέραιους ο αλγόριθμος των Schönhage-Strassen (1971) με πολυπλοκότητα $O(n\log_2 n\log_2\log_2 n)$, που ήταν ο καλύτερος αλγόριθμος πριν δοθεί ο αλγόριθμος του Furer (2007). Το 2019, οι David Harvey, Joris Van Der Hoeven παρουσίασαν αλγόριθμο που πολλαπλασιάζει δύο ακέραιους σε χρόνο $O(n\log_2 n)$. Οι περισσότερες κρυπτογραφικές ρουτίνες υλοποιούν τον πολλαπλασιασμό του Karatsuba. Όσον αφορά την διαίρεση με υπόλοιπο απαιτείται τετραγωνικός χρόνος $O(n^2)$. Γραμμικός χρόνος απαιτείται για την πρόσθεση και αφαίρεση.

Άσκηση 2.1 Να πολλαπλασιάσετε (με το χέρι) τους αριθμούς $1234, 5678$ με χρήση του πολλαπλασιασμού Karatsuba. Υλοποιήστε τον σε όποια γλώσσα προγραμματισμού θέλετε και κατόπιν πολλαπλασιάστε τους αριθμούς,

$$123456789,\ 987654321.$$

Άσκηση 2.2 Έστω, $a > b > 1$. Έστω η αναγωγική ακολουθία $T(n) = aT(n/b) + Cn$. Αποδείξτε επαγωγικά ότι υπάρχουν δύο σταθερές A, B που εξαρτώνται από το $T(1), C$ τέτοιες ώστε

$$An^{\log_b a} \leq T(n) \leq Bn^{\log_b a}$$

για κάθε $n \geq 1$. Για απλοποίηση υποθέστε ότι $n = k^b$.

2.1.3 Γρήγορη ύψωση σε δύναμη

Πολλές φορές στην κρυπτογραφία απαιτείται να μπορούμε να υπολογίζουμε δυνάμεις mod m γρήγορα, δηλ. σε πολυωνυμικό χρόνο. Ένας γρήγορος αλγόριθμός ύψωσης σε δύναμη, βασίζεται στην εξής παρατήρηση. Έστω ότι θέλουμε να υπολογίσουμε την δύναμη $3^7 \pmod{12}$. Αναλύουμε στο δυαδικό του ανάπτυγμα

τον εκθέτη $7 = (111)_2 = 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$. Κατόπιν γράφουμε την δύναμη $3^7 = 3^{2^2+2+1} = 3^4 \cdot 3^2 \cdot 3$. Τέλος υπολογίζουμε τις δυνάμεις $3^2 = 9, 3^4 = (3^2)^2 = 9^2 = 81 \equiv 9 (\mathrm{mod}\ 12)$. Το αποτέλεσμα είναι $9 \cdot 9 \cdot 3 \equiv 3 \bmod 12$. Μερικές φορές η μέθοδος αυτή ονομάζεται από τα *δεξιά προς τα αριστερά δυαδική μέθοδος*.

Αλγόριθμος 2.1.2. : Γρήγορη ύψωση σε δύναμη mod m
Είσοδος. b (*base*), e (*exponent*), m (*modulus*) ($b < m$)
Έξοδος. $b^e \bmod m$

```
1  B ← b
2  result ← 1
3  M ← m
4  E ← e
   while E > 0 do
5  |  if E mod 2 = 1 then
6  |  |  result ← result * B mod M
   |  end
7  |  E ← ⌊E/2⌋
8  |  B ← B^2 mod M
   end
```

Στην γραμμή 7, υπολογίζουμε το ακέραιο μέρος του $E/2$ το οποίο μας δίνει το πηλίκο της διαίρεσης του E με το 2 (Θεώρημα 3.1.1). Το δυαδικό ανάπτυγμα του εκθέτη e υπολογίζεται στην γραμμή 5. Αν είναι 0 τότε δεν υπολογίζουμε τίποτα, διαφορετικά υπολογίζουμε το $b^{2^i} \bmod m$, όπου i είναι η θέση ενός 1 στο δυαδικό ανάπτυγμα του e.

Παράδειγμα 2.1.1. Έστω ότι θέλουμε να υπολογίσουμε την δύναμη $2^{53} \bmod 157$ (όλες οι πράξεις στην δεύτερη και τρίτη στήλη είναι mod 157).

α/α	B	$result$	E
initalization	$b = 2$	1	$e = 53$
1	2^2	2	$\lfloor 53/2 \rfloor = 26$
2	$2^4 = 16$	2	$\lfloor 26/2 \rfloor = 13$
3	$2^8 = 99$	32	$\lfloor 13/2 \rfloor = 6$
4	$2^{16} = 99^2 = 67$	32	$\lfloor 6/2 \rfloor = 3$
5	$2^{32} = 67^2 = 93$	$2144 = 103$	$\lfloor 3/2 \rfloor = 1$
6		$9579 = 2$	$\lfloor 1/2 \rfloor = 0$

Επομένως, $2^{53} \equiv 2 \bmod 157$. Η πολυπλοκότητα για τον υπολογισμό του b^e είναι $O(\log_2 e \cdot (\log_2 m)^2)$. Ο εκθέτης 2 μπορεί να μειωθεί στο 1.5 αν χρησιμοποιήσουμε τον πολλαπλασιασμό του Karatsuba. Αν $n = \max\{b, e, m\}$, τότε $O((\log_2 n)^3)$ η bit-πολυπλοκότητα του αλγορίθμου. Επομένως ο αλγόριθμος είναι πολυωνυμικής bit-πολυπλοκότητας.

Άσκηση 2.3 Υλοποιήστε σε όποια γλώσσα προγραμματισμού θέλετε τον αλγόριθμο 2.1.2 και κατόπιν υπολογίστε τη δύναμη $5^{77} \bmod 19$.

Κεφάλαιο 3

Εισαγωγή στην Θεωρία αριθμών

Number theorists are like lotus-eaters – having tasted this food they can never give it up.
Leopold Kronecker

Η θεωρία αριθμών είναι ένας σημαντικός κλάδος των καθαρών μαθηματικών. Παρόλο που σε πολλούς κλάδους των καθαρών μαθηματικών δεν υπάρχει η ανάγκη υπολογισμών που απαιτούν Η/Υ, στην θεωρία αριθμών η ανάγκη αυτή αποδεδειγμένα υπάρχει. Για παράδειγμα η διάσημη εικασία των Birch και Swinnerton-Dyer που συνδέει την βαθμίδα μιας ελλειπτικής καμπύλης με την τάξη στο μηδέν συγκεκριμένης $L-$σειράς, βασίστηκε σε υπολογιστικά δεδομένα. Επίσης στην εύρεση αντιπαραδειγμάτων η θεωρία αριθμών βασίζεται στους Η/Υ. Π.χ. αν $\lambda(n) = (-1)^r$, όπου r το πλήθος των πρώτων διαιρετών του n και $\Lambda(x) = \sum_{n \leq x} \lambda(x)$, ο Polya διατύπωσε την εικασία ότι $\Lambda(x) \leq 0$. Ο Lehman το 1960 με την βοήθεια Η/Υ υπολόγισε ότι $L(906180359) = 1$. Επίσης ο Mertens διατύπωσε την εικασία ότι $|M(x)| = |\sum_{n \leq x} \mu(x)| \leq \sqrt{x}$, όπου $\mu(x)$ η συνάρτηση του Mobius. Οι Odlyzko και Riele το 1985 βρήκαν ένα αντιπαράδειγμα βασισμένοι σε πολύπλοκους υπολογισμούς που έγιναν σε Η/Υ.

Σε αυτό το κεφάλαιο θα ξεκινήσουμε με εισαγωγικές έννοιες, όπως η διαιρετότητα και ο μέγιστος κοινος διαιρετής και θα ολοκληρώσουμε με τεστ πιστοποίησης πρώτων αριθμών.

Στην κρυπτογραφία χρησιμοποιούμε προβλήματα της θεωρίας αριθμών που είναι δύσκολα να λυθούν κατά μέσο όρο. Δύο τέτοια προβλήματα είναι το πρόβλημα της παραγοντοποίησης και το πρόβλημα του διακριτού λογαρίθμου $\bmod p$, για p αρκετά μεγάλο πρώτο ($> 1024-$bits). Το τελευταίο είναι το πρόβλημα που χρησιμοποιείται στο πρωτόκολλο Diffi-Hellman. Τα προβλήματα αυτά θα τα δούμε πιο αναλυτικά στο επόμενο κεφάλαιο.

3.1 Διαιρετότητα

Ορισμός 3.1.1. *Λέμε ότι ο ακέραιος αριθμός a διαιρεί τον ακέραιο αριθμό b και γράφουμε $a|b$ αν υπάρχει ακέραιος αριθμός c τέτοιος ώστε $b = ac$. Διαφορετικά*

γράφουμε $a \not| b$.

Για παράδειγμα $2|8, 3|9, 22963|45926$, ενώ $17 \not| 18$.

Πρόταση 3.1.1. (i). $a|a$ *για κάθε ακέραιο* a.
(ii). *Αν* $a, b \neq 0$ *και* $a|b,\ b|c$ *τότε* $a|c$, *για κάθε ακέραιο* a, b, c.
(iii). *Αν* $a|b,\ a|c$ *τότε* $a|\lambda b + \mu c$, *για κάθε ακέραιο* a, b, c, λ *και* μ.
(iv). *Αν* $a, b \neq 0$ *και* $a|b$ *τότε*, $|a| \leq |b|$.
(v). *Αν* $b \neq 0$ *και* $a|b,\ b|a$ *τότε*, $|a| = |b|$.

Παρατήρηση 3.1.1. Πάντα το $a|0$ $(a \neq 0)$, $1|b$ για κάθε ακέραιο b. Ενώ αν $0|b$ τότε $b = 0$. Τέλος,

$$a|b \Leftrightarrow -a|b \Leftrightarrow a|-b \Leftrightarrow |a||b|.$$

Θεώρημα 3.1.1 (Ευκλείδεια διαίρεση). *Για κάθε ζεύγος ακεραίων* (a, b) *με* $a \geq b \geq 0$ *υπάρχει ακριβώς ένα ζευγάρι ακεραίων* (q, r) *τέτοιων ώστε* $a = bq + r,\ 0 \leq r < b$. *Ειδικότερα* $q = \lfloor \frac{a}{b} \rfloor$ *και ονομάζεται ακέραιο πηλίκο του* a *διά του* b.

Απόδειξη. Πρώτα θα αποδείξουμε ότι το $q = \lfloor \frac{a}{b} \rfloor$ ικανοποιεί την απαιτούμενη ιδιότητα και κατόπιν την μοναδικότητα.

Θυμίζουμε ότι το ακέραιο μέρος $\lfloor A \rfloor$ του A, είναι ο μεγαλύτερος ακέραιος μικρότερος ή ίσος του A, δηλαδή $\lfloor A \rfloor \leq A < \lfloor A \rfloor + 1$. Επομένως

$$q \leq \frac{a}{b} < q + 1.$$

Πολλαπλασιάζουμε και τα δύο μέλη με το $b > 0$ και έτσι έχουμε $bq \leq a < bq + b$. Επομένως υπάρχει $r \in \mathbb{Z}$ με $0 \leq r < b$ τέτοιο ώστε $a = bq + r$.

Αν υπήρχε ακόμη ένα ζευγάρι, (q', r') τέτοιο ώστε, $a = bq' + r'$ και $0 \leq r' < b$, τότε $bq + r = bq' + r'$. Άρα $b(q - q') = r' - r$ και επομένως $b||r - r'| \Rightarrow b \leq |r - r'|$. 'τοπο, διότι $0 \leq r, r' < b$ δηλ. $|r - r'| < b$. □

Το προηγούμενο θεώρημα ισχύει και γενικότερα.

Θεώρημα 3.1.2 *Έστω* $a, b \in \mathbb{Z}$ *με* $b \neq 0$. *Τότε υπάρχει μοναδικό ζευγάρι* (q, r) *με* $q \in \mathbb{Z}$ *και* $0 \leq r < |b|$, *τέτοιο ώστε* $a = bq + r$.

Παρατήρηση 3.1.2. Αν a ένας θετικός ακέραιος, τότε το δυαδικό του μήκος είναι $\text{len}_2(a) = \lfloor \log_2 a \rfloor + 1$. Τότε, $\text{len}_2(a + b) = O(\max\{\text{len}_2(a), \text{len}_2(b)\})$, $\text{len}_2(ab) = O(\text{len}_2(a)\text{len}_2(b))$ και η διάρεση του a δια του b με πηλίκο q κοστίζει $O(\text{len}_2(a)\text{len}_2(q))$. Όλες οι προηγούμενες πράξεις χρειάζονται $O(\text{len}_2(a)+\text{len}_2(b))$ χώρο.

Παρατήρηση 3.1.3. Στην κρυπτογραφία συνήθως η λέξη *αποδοτικός* ταυτίζεται με την λέξη *πολυωνυμικός*. Εξηγούμε τι ακριβώς εννοούμε μ'αυτό. Έστω ότι έχουμε έναν αλγόριθμο με είσοδο τους θετικούς ακέραιους, $a_1, ..., a_n$. Λέμε ότι ο αλγόριθμος είναι πολυωνυμικός αν εκτελείται σε

$$O(\text{len}_2(a_1)^{e_1} \cdot \text{len}_2(a_2)^{e_2} \cdots \text{len}_2(a_n)^{e_n}) \text{ βήματα}$$

για κάποια $e_1, ..., e_n$ μη αρνητικά.

Άσκηση 3.1 Ν.α.ο. αριθμοί της μορφής $4n + 3$ δεν είναι τέλεια τετράγωνα. Κατόπιν, ν.α.ο. κάνενας από τους αριθμούς

$$11, 111, ..., 111 \cdots 111, \ldots$$

δεν είναι τέλειο τετράγωνο.

Άσκηση 3.2 cross+roads=danger. Αν κάθε γράμμα αντιστοιχεί σε ένα μη μηδενικό ψηφίο, βρείτε την αριθμητική τιμή της λέξης danger.

Άσκηση 3.3 (**) Θεωρούμε την ακόλουθη διαδικασία. Έστω x ένας ακέραιος. Αν x άρτιος τότε διαίρεσε τον με το 2, διαφορετικά τον πολλαπλασιάζουμε με τον 3 και προσθέτουμε 1. Αν καταλήξουμε στον αριθμό 1, σταματάμε. Επαναλαμβάνουμε την διαδικασία. Π.χ. αν $x = 12$ τότε έχουμε την ακολουθία

$$12, 6, 3, 10, 5, 16, 8, 4, 2, 1.$$

Το πλήθος των επαναλήψεων (στο προηγούμενο παράδειγμα είναι 9) ονομάζεται total stopping time. Μπορείτε να βρείτε έναν ακέραιο που ο total stopping time είναι άπειρος;

Άσκηση 3.4 Ν.δ.ο. το 2^m $(m \geq 2)$ δεν είναι άθροισμα διαδοχικών θετικών ακεραίων.

Άσκηση 3.5 Να βρείτε έναν αποδοτικό αλγόριθμο που με είσοδο δύο λίστες $P = [p_1, ..., p_n]$ (p_i πρώτοι) και $e = [e_1, ..., e_n]$ (e_i θετικοί ακέραιοι) να εξάγει όλους τους διαιρέτες του $N = p_1^{e_1} \cdots p_n^{e_n}$. Να γίνει μελέτη όσον αφορά τις απαιτήσεις σε μνήμη.

Άσκηση 3.6 Ν.α.ο. $n^5 + 1 | n^{10} - 1$ για κάθε φυσικό n.

Άσκηση 3.7 Να αποδείξετε την Πρόταση 3.1.1.

Άσκηση 3.8 Ν.α.ο. το γινόμενο n διαδοχικών φυσικών αριθμών διαιρείται από το $n!$

Άσκηση 3.9 Να αποδείξετε ότι ο ακέραιος $n \geq 2$ διαιρεί ακριβώς έναν αριθμό, από ένα σύνολο n διαδοχικών ακεραίων.

Άσκηση 3.10 Βρείτε του διαιρέτες των αριθμών 220 και 284. Άς είναι $\sigma(220)$ και $\sigma(284)$ τα αθροίσματα αυτών. Παρατηρήστε ότι $\sigma(220) = \sigma(284)$. Βρείτε το επόμενο ζευγάρι αριθμών (n, m) ώστε $\sigma(n) = \sigma(m)$. Οι αριθμοί αυτοί ονομάζονται amicable nunmbers.

Άσκηση 3.11 Αν $H_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}$ (n θετικός ακέραιος), ν.α.ο. υπάρχει το

$$\lim_{n\to\infty} (H_n - \ln n).$$

Ο αριθμός αυτός ονομάζεται σταθερά των Euler-Mascheroni και συμβολίζεται με το ελληνικό γράμμα γ.

Άσκηση 3.12 Να κάνετε τα γραφήματα των ακολουθιών[7] $a_n = \ln n + 2\gamma - 1$ και $b_n = \frac{1}{n}\sum_{r=1}^{n} \tau(r)$. Όπου $\gamma \approx 0.577$ η σταθερά των Euler-Mascheroni[8] και $\tau(n)$ το πλήθος των θετικών διαιρετών του n, π.χ. $\tau(12) = 6$. Ο Lejeune Dirichlet το 1838 απόδειξε ότι ο μέσος όρος των διαιρετών του n πλησιάζει στο $\ln n + 2\gamma - 1$.

Άσκηση 3.13 (**) Αν $\mathcal{A} \subset \mathbb{N}$ και

$$\sum_{x\in\mathcal{A}} \frac{1}{x} = \infty,$$

τότε υπάρχουν $a, c \in \mathbb{N}$ τέτοια ώστε,

$$\{a + c, a + 2c, ..., a + kc\} \subset \mathcal{A},$$

για αυθαίρετο φυσικό k.

Άσκηση 3.14 (**) (Erdős-Borwein) Ν.α.ο. ο αριθμός

$$\sum_{n=1}^{\infty} \frac{1}{2^n - 1}$$

είναι ρητός.

3.1.1 Πρώτοι αριθμοί

God may not play dice with the universe, but something strange is going on with the prime numbers.
Paul Erdős

Οι πρώτοι αριθμοί αρχικά μελετήθηκαν από τον Ευκλείδη. Ένας φυσικός αριθμός μεγαλύτερος του 1, ονομάζεται **πρώτος** αν διαιρείται μόνο από την μονάδα και τον εαυτό του. Διαφορετικά ονομάζεται σύνθετος. Το σύνολο των πρώτων αριθμών είναι:

$$\{2, 3, 5, 7, 11, 13..., 101, ..., 571, ..., 6997, ..., 2^{57885161} - 1, ..., 2^{82589933} - 1, ...\}. \tag{3.1.1}$$

Κάθε αριθμός > 1 έχει έναν πρώτο διαιρέτη.

[7] Π.χ. μπορείτε να χρησιμοποιήσετε το Sagemath.
Η εντολή `list_plot( [ln(n) + 0.154 for n in range(1, 100) ])` θα σας έδινε το πρώτο γράφημα.

[8] Julian Havil, γ : Exploring Euler's constant, 2003, Princeton University Press.

Λήμμα 3.1.1. *Κάθε ακέραιος $n > 1$, έχει έναν πρώτο διαιρέτη.*

Απόδειξη. Θα χρησιμοποιήσουμε (ισχυρή) επαγωγή. Υποθέτουμε ότι όλοι οι ακέραιοι k με $2 \leq k < n$ έχουν έναν πρώτο διαιρέτη. Θα αποδείξουμε ότι και ο n έχει έναν πρώτο διαιρέτη. Έστω ότι ο n δεν είναι πρώτος. Αν είναι δεν έχουμε να αποδείξουμε κάτι. Εφόσον ο n είναι σύνθετος, θα γράφεται $n = ab$ με $a, b > 1$. Αλλά, $a < n$, οπότε από την υπόθεση της επαγωγής υπάρχει πρώτος διαιρέτης p του a. Δηλ. $p|a$, επομένως $p|ax$ για κάθε ακέραιο x. Για $x = b$, προκύπτει $p|ab = n$. Επομένως ισχύει και για $k = n$. □

Αμέσως τίθεται το ερώτημα που σταματάει το σύνολο των πρώτων.

Θεώρημα 3.1.3 (Ευκλείδης) Υπάρχουν άπειροι πρώτοι αριθμοί.

Απόδειξη. Υποθέτουμε ότι υπάρχουν πεπερασμένοι πλήθους πρώτοι αριθμοί

$$p_1, p_2, ..., p_n.$$

Σχηματίζουμε τον αριθμό :

$$m = p_1 p_2 \cdots p_n + 1.$$

Ο αριθμός m είναι μεγαλύτερος από όλους τους πρώτους αριθμούς, άρα δεν μπορεί να είναι πρώτος. Επομένως είναι σύνθετος αριθμός. Τότε όμως θα έχει ένα πρώτο διαιρέτη. Επειδή όλοι οι πρώτοι είναι οι $p_1, ..., p_n$, ο πρώτος διαιρέτης θα είναι κάποιος από τους αριθμούς $p_1, ..., p_n$. Ας είναι ο p_i για κάποιο $i \in \{1, 2, ..., n\}$. Αν διαιρέσουμε τον m με το p_i έχουμε υπόλοιπο ένα, ενώ θα έπρεπε το υπόλοιπο να είναι μηδέν. Άτοπο. Άρα υπάρχουν άπειροι το πλήθος πρώτοι. □

Παρατήρηση 3.1.4. Εύκολα προκύπτει από την απόδειξη ότι ο ακέραιος $k = \prod_{i=1}^{n} p_i + 1$ $(p_1 < p_2 < \cdots < p_n)$ δεν διαιρείται από κανέναν πρώτο στο διάστημα $[1, p_n]$. Έστω q πρώτος διαιρέτης του k. Τότε $q = p_{n+j}$ για $j > 0$. Επομένως, $p_n < p_{n+j} \leq \prod_{i=1}^{n} p_i + 1$.

Παρατήρηση 3.1.5. Ένας σύνθετος θετικός ακέραιος n, πάντα έχει πρώτο διαιρέτη $p < \sqrt{n}$. Έστω $n = ab$ με $1 < a \leq b$. Αν $a > \sqrt{n}$, τότε $ab = n > n$. Άτοπο.

Στο προηγούμενο σύνολο (3.1.1), συμπεριλάβαμε τον αριθμό 101. Μπορεί κάποιος δοκιμάζοντας να βρει διαιρέτες του 101, να διαπιστώσει σχετικά γρήγορα ότι δεν υπάρχουν άλλοι εκτός του 1 και του 101. Π.χ. ελέγχουμε έναν-έναν τους αριθμούς

$$2, 3, 4, \ldots, \lfloor \sqrt{101} \rfloor,$$

αν είναι διαιρέτες του 101. Αν κανείς από αυτούς δεν είναι, τότε είναι πρώτος. Τι γίνεται όμως με τον 6997; Επίσης πως θα ελέγχαμε ότι ο αριθμός 14220533 είναι πράγματι πρώτος; Ακόμη χειρότερα πως θα μπορούσαμε να αποδείξουμε ότι ο αριθμός $2^{82589933} - 1$ είναι πρώτος; (ο συγκεκριμένος αριθμός το 2018 αποδείχθηκε ότι είναι ένας πρώτος αριθμός του Mersenne). Αυτό είναι το πρόβλημα της *πιστοποίησης των πρώτων αριθμών* και έχει λυθεί με την έννοια ότι υπάρχει

πολυωνυμικός ντετερμινιστικός αλγόριθμος που να αποφαίνεται αν ένας αριθμός είναι πρώτος ή όχι. Φυσικά για τον πρώτο αριθμό του Mersenne που γράψαμε η υπολογιστική ισχύς που απαιτείται για να τερματίσει ο αλγόριθμος πιστοποίησης είναι τεράστια. Γι'αυτό το λόγο υπάρχει το κατανεμημένο δίκτυο "Great Internet Mersenne Prime Search" (GIMPS). Το επόμενο πρόβλημα στην θεωρία των πρώτων αριθμών είναι το πρόβλημα της παραγοντοποίησης ενός αριθμού στους πρώτους παράγοντες του. Ένα σημαντικό θεώρημα της θεωρίας αριθμών είναι το παρακάτω.

Θεώρημα 3.1.4 (Θεμελιώδες θεώρημα της αριθμητικής). Κάθε μη μηδενικός φυσικός αριθμός n αναλύεται σε γινόμενο πρώτων παραγόντων με μοναδικό τρόπο,

$$n = p_1^{a_1} \cdots p_r^{a_r},$$

με a_i θετικούς ακέραιους και $p_1 < p_2 < \cdots < p_r$ πρώτους.

Η απόδειξη του θεωρήματος δεν μας δίνει κάποιο τρόπο να υπολογίζουμε τους πρώτους παράγοντες του φυσικού αριθμού και οι αλγόριθμοι που υπάρχουν δεν είναι πολυωνυμικού χρόνου. Ακριβώς σε αυτή την δυσκολία πιστεύουμε ότι βασίζεται η ασφάλεια του RSA. Για την μελέτη των πρώτων αριθμών συστήνουμε το βιβλίο των Pomerance και Grandall [9]. Βέβαια υπάρχουν και συστήματα κρυπτογράφησης που δεν βασίζονται στην παραγοντοποίηση ή στον διακριτό λογάριθμο. Ένα τέτοιο παράδειγμα είναι τα συστήματα που βασίζονται σε πλέγματα ή σε κώδικες. Ειδικότερα η τάση που επικρατεί είναι να μετακινηθούμε από το RSA προς την κρυπτογραφία που βασίζεται στα πλέγματα επειδή το πρόβλημα της παραγοντοποίησης και του διακριτού λογαρίθμου έχει αποδειχτεί ότι λύνεται σε πολυωνυμικό χρόνο αν έχουμε έναν κβαντικό υπολογιστή (με αρκετά μεγάλη μνήμη). Βέβαια μέχρι σήμερα δεν έχουμε κατασκευάσει κβαντικούς υπολογιστές με αρκετά μεγάλη μνήμη, αλλά πολλοί ειδικοί πιστεύουν ότι αυτό θα συμβεί αργά ή γρήγορα.

Παρατήρηση 3.1.6. Ένας λόγος που ο 1 δεν θεωρείται πρώτος, είναι για να έχω την μοναδικότητα του θεωρήματος 3.1.4. Αν $p_1 = 1$, τότε δεν υπάρχει μοναδικός ακέραιος a_1.

Με χρήση του θεωρήματος 3.1.4, μπορούμε να δώσουμε ακόμη μια απόδειξη ότι υπάρχουν άπειροι πρώτοι αριθμοί.

Λήμμα 3.1.2 (2η απόδειξη ότι υπάρχουν άπειροι πρώτοι)**.** *Το σύνολο των πρώτων είναι άπειρο.*

Απόδειξη. Ας υποθέσουμε ότι υπάρχουν πεπερασμένοι πρώτοι αριθμοί, $p_1, p_2, ..., p_r$. Διαλέγουμε αυθαίρετα ένα $n \geq 2$. Ας είναι ένας ακέραιος s με $1 \leq s \leq n$. Από την υπόθεση μας ότι υπάρχουν πεπερασμένοι πρώτοι, ο αριθμός s θα έχει μια παραγοντοποίηση της μορφής,

$$s = p_1^{a_1} p_2^{a_2} \cdots p_r^{a_r},$$

όπου a_i μη αρνητικοί ακέραιοι αριθμοί. Θεωρούμε την συνάρτηση,

$$\Psi : \mathbb{Z} \to \mathbb{Z}^r, \ \Psi(s) = (a_1, ..., a_r).$$

Η Ψ είναι 1-1. Επομένως το πλήθος των διαφορετικών $\Psi(s)$ είναι ακριβώς n.

Ισχύει $p_i^{a_i} \leq n$, επομένως, $a_i \leq \log_{p_i} n \leq \log_2 n$. Εφόσον ο $a_i \geq 0$ παίρνει το πολύ $1 + \log_2 n < 2\log_2 n$ τιμές. Άρα, συνολικά το $\Psi(s)$ παίρνει το πολύ $2^r(\log_2 n)^r$ τιμές. Αλλά όπως είδαμε, το πληθος των s είναι n επομένως και το πλήθος των διαφορετικων τιμών του $\Psi(s)$ είναι n. Καταλήγουμε ότι,

$$2^r(\log_2 n)^r > n.$$

Αυτή η ανισότητα δεν ισχύει για μεγάλο n. Άτοπο. □

Άσκηση 3.15 Να βρείτε 10 πρώτους αριθμούς που όταν αντιστρέψετε τα ψηφία τους να προκύπτουν πάλι πρώτοι. Π.χ. ο 13 διότι ο 31 είναι πρώτος.

Άσκηση 3.16 Ν.α.ο. ο $n-$οστός πρώτος p_n είναι μεγαλύτερος ή ίσος από το $n+1$.

Άσκηση 3.17 Να αποδείξετε ότι

$$\prod_{p \leq x} p \leq 4^{x-1}$$

για κάθε πραγματικό $x \geq 2$ και p πρώτος.

Άσκηση 3.18 Να αποδείξετε ότι ο $n-$οστός πρώτος p_n ικανοποιεί την ανισότητα

$$p_n < 2^{2^n}.$$

Υποδ. Χρησιμοποιήστε επαγωγή σε συνδυασμό με την παρατήρηση 3.1.4.

Άσκηση 3.19 Ο αριθμός $H_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots \frac{1}{n}$ ονομάζεται $n-$οστός αρμονικός αριθμός. Ν.α.ο. $\lim_{n\to\infty} H_n = \infty$. Η ακολουθία αυτή πλησιάζει το άπειρο πολύ *αργά*. Αν περιορίσουμε το n να παίρνει τιμές στους πρώτους τότε έχουμε την σειρά,

$$\sum_{p:\text{prime}} \frac{1}{p}.$$

Τι μπορείτε να πείτε για την σύγκλιση αυτής της σειράς;
Υπόδειξη.

$$H_n = 1 + \frac{1}{2} + (\frac{1}{3} + \frac{1}{4}) + (\frac{1}{5} + \cdots + \frac{1}{8}) + \cdots >$$

$$1 + \frac{1}{2} + \frac{1}{2} + \frac{1}{2} + \cdots$$

Ένας άλλος τρόπος θα ήταν να υπολογιστεί το ολοκλήρωμα

$$\int_{-\infty}^{0} \frac{e^x}{1-e^x}\, dx = \cdots = \lim_{n\to\infty} H_n.$$

Άσκηση 3.20 Με την βοήθεια Η/Υ μπορείτε να επαληθεύσετε (για αρκετά μεγάλα x και p πρώτος)

$$\sum_{p\leq x}\frac{1}{p}\geq \ln\ln x-\ln 2\ (x\geq 2);$$

Άσκηση 3.21 (Euler's formula). Ν.α.ο.

$$\sum_{n=1}^{\infty}\frac{1}{n}=\prod_{p:\text{prime}}\frac{1}{1-p^{-1}}.$$

Αυτή η ισότητα αποδυκνύει ότι υπάρχουν άπειροι πρώτοι. Γιατί;

Άσκηση 3.22 Έστω p πρώτος. Με την βοήθεια Η/Υ μπορείτε να επαληθεύσετε (για αρκετά μεγάλα x)

$$\sum_{p\leq x}\frac{1}{p^n}<1\ \ (n>1)$$

Επίσης, να δώσετε και μία μαθηματική απόδειξη.
Υπόδειξη.

$$\sum_{n=1}^{\infty}\frac{1}{n^2}=\frac{\pi^2}{6}.$$

Αυτή η σταθέρα ισουται με $\zeta(2)$, όπου $\zeta(s)$ η συνάρτηση του Riemann (δείτε και άσκηση 3.47).

Άσκηση 3.23 Αν p πρώτος ν.α.ο. ο $\sqrt{p}$ είναι άρρητος.

3.2 Μέγιστος Κοινός Διαιρέτης

Έστω x,y δύο ακέραιοι αριθμοί (ένας τουλάχιστον μη μηδενικός). Κάθε ακέραιος που διαιρεί τους x,y λέγεται κοινός διαιρέτης των x,y. Αν $L(x,y)$ το σύνολο όλων των κοινών διαιρετών των x,y, το μέγιστο στοιχείο του L λέγεται μέγιστος κοινός διαιρέτης (μ.κ.δ.) των x,y και συμβολίζεται $\gcd(x,y)$ (gcd : **g**reatest **c**ommon **d**ivisor). Το σύνολο $L(x,y)$ είναι πεπερασμένο, διότι ένας από τους x,y δεν είναι μηδέν, και είναι και μη κενό, διότι το $1\in L(x,y)$. Για παράδειγμα $\gcd(2,4)=2$. Επίσης $\gcd(-2,4)=2$. Δηλαδή ο μ.κ.δ. είναι ανεξάρτητος των προσήμων. Αν $x=0$, τότε $\gcd(0,y)=|y|$. Αν $x=1$, τότε $\gcd(x,y)=\gcd(1,y)=1$. Τέλος, αν $\gcd(x,y)=1$ λέμε ότι οι x,y είναι πρώτοι μεταξύ τους. Σε αυτή την περίπτωση οι x,y δεν έχουν κοινούς διαιρέτες. Π.χ. οι $101,102$ είναι πρώτοι μεταξύ τους.

Ο υπολογισμός του $\gcd(x,y)$ ανάγεται στα αρχαία χρόνια. Συνδέεται στενά με τον υπολογισμό του $x^{-1}(\mathrm{mod}\ N)$ (εφόσον υπάρχει). Η σύνδεση αυτή θα γίνει πιο ξεκάθαρη στο επόμενο βασικό θεώρημα.

Θεώρημα 3.2.1 (Ταυτότητα Bézout). Αν x,y ακέραιοι αριθμοί, όχι και ο δύο μηδέν και $d=\gcd(x,y)$, τότε υπάρχουν ακέραιοι αριθμοί a,b τέτοιοι ώστε

$$d=ax+by.$$

Απόδειξη. Έστω $L_{x,y} = \{n \in \mathbb{N} - \{0\} : n = ax + by \text{ για κάποια } a, b \in \mathbb{Z}\}$. Αρχικά παρατηρούμε ότι $L_{x,y} \neq \emptyset$. Δηλ. υπάρχει ακέραιος θετικός αριθμός που γράφεται στην μορφή $ax+by$ για κάποια $a, b \in \mathbb{Z}$. Για παράδειγμα $x^2+y^2 = x\cdot x+y\cdot y$. Εφόσον $L_{x,y} \subset \mathbb{N}$ το σύνολο έχει ελάχιστο στοιχείο. Ισχυριζόμαστε ότι ο ελάχιστος θετικός ακέραιος g που γράφεται στη μορφή $ax + by$ για κάποια $a, b \in \mathbb{Z}$, είναι τελικά ο $\gcd(x, y)$.

Κάθε διαιρέτης των x, y θα διαιρεί και τον g. Επομένως, $d|g$. Άρα,

$$d \leq g. \tag{3.2.1}$$

Θα αποδείξουμε ότι $g \leq d$. Υποθέτουμε ότι $x \neq 0$. Αν $x = 0$, από την υπόθεση του θεωρήματος το $y \neq 0$, οπότε $\gcd(0, y) = |y| = g$. Έστω $x \neq 0$. Τότε, το $|x|$ γράφεται ως $\pm x + 0y$ (ανάλογα αν είναι θετικό ή αρνητικό), άρα είναι της μορφής $ax + by$, επομένως $|x| \in L_{x,y}$. Αλλά ο g είναι ο ελάχιστος φυσικός του συνόλου $L_{x,y}$ άρα $|x| \geq g \geq 0$. Από το θέώρημα 3.1.1, υπάρχει (t, r) έτσι ώστε, $|x| = tg+r$, $0 \leq r < g$. Χωρίς βλάβη της γενικότητας υποθέτουμε ότι $x > 0$. Άρα,

$$x = t(ax + by) + r \Rightarrow r = (1 - ta)x - tby.$$

Αν $r \neq 0$, τότε, $r \in L_{x,y}$ και $r < g$. Αυτό αντίκειται στην επιλογή του g. Άτοπο. Επομένως, $r = 0$ και $g|x$. Το ίδιο αν $x < 0$. Με τον ίδιο ακριβώς τρόπο μπορούμε να δείξουμε ότι $g|y$. Επομένως, ο g είναι ένας κοινός διαιρέτης των x, y. Αναγκαστικά, $g \leq d$. Αλλά ισχύει και $d \leq g$ από την σχέση (3.2.1). Δηλ. $g = d$. □

Παρατήρηση 3.2.1. Αν $d = 1$ τότε γράφεται $ax + by = 1$. Μερικές φορές την ονομάζουμε ταυτότητα Bachet.

Παρατήρηση 3.2.2. Οι αριθμοί a, b του προηγούμενου θεωρήματος δεν είναι μοναδικοί. Ας είναι (a, b) τέτοια ώστε $ax+by = d$. Ισχύει $d|x, d|y$ άρα $x = dx_0$, $y = dy_0$. Τότε οι $a' = a + ty_0$, $b' = b - tx_0$ για τυχαίο ακέραιο t, ικανοποιούν την εξίσωση $a'x + b'y = d$.

Πρόταση 3.2.1. *Αν $a|x, b|x$ και $\gcd(a, b) = 1$, τότε $ab|x$.*

Απόδειξη. $a|x$ σημαίνει ότι υπάρχει ακέραιος y_1 τέτοιος ώστε $x = ay_1$. Παρόμοια, $x = by_2$. Επομένως, $ay_1 = by_2$. Επίσης, υπάρχουν k_1, k_2, τέτοια ώστε $ak_1 + bk_2 = 1$. Πολλαπλασιάζουμε και τα δύο μέλη με y_2 οπότε έχουμε, $ak_1y_2 + (by_2)k_2 = y_2$ ισοδύναμα $a(k_1y_2 + y_1k_2) = y_2$ ισοδύναμα $az_2 = y_2$. Από την ισότητα $x = by_2$, έχουμε $x = abz_2$ δηλ. $ab|x$. □

Από την απόδειξη της ταυτότητας Bézout έχουμε το εξής, *ένας κοινός διαιρέτης των a, b διαιρεί τον μέγιστο κοινό διαιρέτη των a, b.* Μπορούμε να διατυπώσουμε και τον παρακάτω ισοδύναμο ορισμό του gcd κάνοντας χρήση του προηγούμενου θεωρήματος.

Ορισμός 3.2.1. *Έστω a, b ακέραιοι αριθμοί. Ο θετικός ακέραιος d ονομάζεται μέγιστος κοινός διαιρέτης των a, b αν-ν*
i. $d|a, d|b$
ii. *Αν $d'|a, d'|b$, τότε $d'|d$.*

Έστω ένας ακέραιος x και y θετικός ακέραιος. Με $x \bmod y$ συμβολίζουμε το υπόλοιπο της διαίρεσης του x διά του y. Αυτό υπάρχει από το θεώρημα 3.1.2.

Λήμμα 3.2.1. *Για $x \geq y > 0$, ισχύει $\gcd(x, y) = \gcd(y, x \bmod y)$.*

Απόδειξη. Ισχύει $x = qy + r,\ r = x \bmod y$. Ας είναι

$$d = \gcd(x, y), d' = \gcd(y, x \bmod y),$$

τότε

$$d|x, d|y \Rightarrow d|x - qy = r$$

άρα $d|d'$. Επίσης

$$d'|y, d'|r = x - qy \Rightarrow d'|x \Rightarrow d'|d.$$

□

Αλγόριθμος 3.2.1. : Αλγόριθμος του Ευκλείδη για τον μέγιστο κοινό διαιρέτη
Είσοδος. Ακέραιοι x, y με $x \geq y > 0$
Έξοδος. $\gcd(x, y)$

while $y \neq 0$ **do**
  $t \leftarrow y$
  $y \leftarrow x \bmod t$
  $x \leftarrow t$
end
return x

Ο αλγόριθμός αυτός παρουσιάστηκε πρώτη φορά στο βιβλίο *Στοιχεία* του Ευκλείδη το 300 π .χ. (βιβλίο VII). Ίσως είναι ο παλαιότερος αλγόριθμος που επιβίωσε μέχρι την σημερινή εποχή. Παρόλο την απλότητα του αλγορίθμου, δεν είναι καθόλου εύκολο να υπολογίσουμε μετά από πόσες επαναλήψεις τερματίζει (δηλ. την πολυπλοκότητα του). Έστω n το πλήθος των επαναλήψεων του αλγορίθμου (3.2.1).

Πρόταση 3.2.2 (Reynaud, 1811)**.** $n < y$.

Πρόταση 3.2.3 (Reynaud, 1821)**.** $n < y/2$.

Πρόταση 3.2.4 (Finck, 1841)**.** $n < 2 \log_2 y + 1$.

Απόδειξη. Θα αποδείξουμε ότι $x \bmod y < x/2$. Διακρίνουμε δύο περιπτώσεις :
• Αν $y \leq x/2$, τότε το υπόλοιπο της διαίρεσης του x διά του y είναι $x \bmod y < y \leq x/2$.
• Αν $y > x/2$, τότε $x = 1 \cdot y + (x - y)$, λόγω της μοναδικότητας του πηλίκου και του υπολοίπου έχουμε, $x \bmod y = x - y < x/2$.
Ο αλγόριθμος του Ευκλείδη, ξεκινάει με μια είσοδο $(x, y) = (x_0, y_0)$ και στον επόμενο γύρο (1η επανάληψη) υπολογίζει
◦ $(x_1, y_1) = (y_0, x_0 \bmod y_0)$, και όπως αποδείξαμε, ισχύει $y_1 < \frac{x_0}{2}$.
Στην επόμενη επανάληψη,
◦ $(x_2, y_2) = (y_1, x_1 \bmod y_1)$ και $y_2 < \frac{x_1}{2}$.
Γενικότερα,

∘ $(x_i, y_i) = (y_{i-1}, x_{i-1} \bmod y_{i-1})$ και $y_i < \frac{x_{i-1}}{2}$.
Δηλαδή,

$$x_i = y_{i-1}, y_i < \frac{x_{i-1}}{2} = \frac{y_{i-2}}{2}.$$

Αργά η γρήγορα ο αλγόριθμος θα τερματίσει. Ειδικότερα, μπορούμε να υποθέσουμε χωρίς βλάβη της γενικότητας ότι ο i είναι άρτιος:

$$y_i < \frac{y_{i-2}}{2} < \frac{y_0}{2^{i/2}} = \frac{y}{2^{i/2}}.$$

Ο αλγόριθμος θα τερματίσει όταν

$$\frac{y}{2^{i/2}} < 2.$$

Εύκολα προκύπτει

$$i > 2\log_2 y - 2.$$

Για την περίπτωση που i είναι περιττός προκύπτει

$$i > 2\log_2 y.$$

Άρα μετά από $2\log_2 y + 1$ επαναλήψεις στη χειρότερη περίπτωση, ο αλγόριθμος θα τερματίσει. □

Επομένως, η συνολική πολυπλοκότητα σε bit-operations είναι $O(\log_2 x(\log_2 y)^2)$. Αν $N = \max\{x, y\}$ έχουμε $O((\log_2 N)^3)$. Πράγματι έχω $O(\log_2 N)$ επαναλήψεις και $O((\log_2 N)^2)$ πολλαπλασιασμούς σε κάθε βήμα.

Πόρισμα 3.2.1. *Η bit πολυπλοκότητα του αλγορίθμου (3.2.1) είναι* $O((\log_2 N)^3)$.

Επίσης, βελτιώθηκε το προηγούμενο φράγμα για το n,

Πρόταση 3.2.5 (Finck, 1844)**.** $n < 5\log_{10} y$.

Τέλος,

Θεώρημα 3.2.2 (Lamé, Dixon, Heilbronn). Αν $x > y$ ακέραιοι από το διάστημα $[1, N]$. Τότε ο αριθμός των βημάτων που απαιτείται στον ευκλείδειο αλγόριθμο δεν ξεπερνάει τον αριθμό

$$\left\lceil \ln(N\sqrt{5})/\ln\frac{(1+\sqrt{5})}{2} \right\rceil - 2,$$

όπου $\lceil a \rceil$ είναι ο μικρότερος ακέραιος μεγαλύτερος από τον πραγματικό a. Επίσης, ο μέσος όρος των βημάτων (καθώς x, y τρέχουν στους ακεραίους) είναι,

$$\frac{12\ln 2}{\pi^2}\ln N.$$

Με μια πιο προσεκτική ανάλυση του Ευκλείδειου αλγορίθμου αποδεικνύεται ότι η bit-complexity είναι $O((\log_2 N)^2)$ [9, Άσκηση 2.6].

Παράδειγμα 3.2.1. Έστω ότι θέλω να υπολογίσω τον μκδ των αριθμών $78, 221$. Διαιρώ το 221 με το 78 και έχω $221 = 78 \cdot 2 + 65$. Κατόπιν διαιρώ τον διαιρέτη, δηλαδή το 78 με το υπόλοιπο, δηλαδή με το 65 και έχουμε $78 = 65 \cdot 1 + 13$. Τέλος, συνεχίζοντας όπως προηγουμένως έχω $65 = 5 \cdot 13 + 0$. Το τελευταίο μη μηδενικό υπόλοιπο, δηλαδή το 13, είναι ο μκδ των αριθμών.

Αν επιπλέον ζητάμε και τους συντελεστές Bézout εφαρμόζουμε την παρακάτω διαδικασία που ονομάζεται Ευκλείδειος Αλγόριθμος (ή επεκτεταμένος Ευκλείδειος Αλγόριθμος : Egcd). Ως συνήθως $x > y \geq 0$.

$$
\begin{aligned}
&x = yq_1 + r_2, && 0 \leq r_2 < x \\
&y = r_2q_2 + r_3, && 0 \leq r_3 < r_2 \\
&r_2 = r_3q_3 + r_4, && 0 \leq r_4 < r_3 \\
&\cdots \\
&r_{N-2} = r_{N-1}q_{N-1} + r_N, && 0 \leq r_N < r_{N-1} \\
&r_{N-1} = r_Nq_N + 0, && r_{N+1} = 0.
\end{aligned}
$$

Από τα προηγούμενα προκύπτει $\gcd(x, y) = r_N$. Ακολουθώντας την αντίστροφη πορεία μπορούμε να βρούμε και τους συντελεστές Bézout. Έτσι στο παράδειγμα 3.2.1 έχουμε

$$13 = 78 - 65 = 78 - (221 - 78 \cdot 2) = -221 + 3 \cdot 78.$$

Κάθε φορά αντικαθιστώ τα προηγούμενα υπόλοιπα.

Αλγόριθμος 3.2.2. : Επεκτεταμένος Αλγόριθμος του Ευκλείδη-Υπολογισμός συντελεστών Bézout
Είσοδος. Ακέραιοι x, y με $x \geq y \geq 0$ και $x > 0$
Έξοδος. $(a, b, d = \gcd(x, y))$ έτσι ώστε $ax + by = d$

Initialization
1 $(a, b, u, v, g, w) \leftarrow (1, 0, 0, 1, x, y)$
Extended Euclidean Loop
while $w > 0$ **do**
2 $q = \lfloor g/w \rfloor$
3 $t \leftarrow w$
4 $w \leftarrow g \pmod t$
5 $g \leftarrow t$
6 **if** $w = 0$ **then**
break
end
7 $U \leftarrow a - qu$
8 $V \leftarrow b - qv$
9 $a \leftarrow u$
10 $b \leftarrow v$
11 $u \leftarrow U$
12 $v \leftarrow V$
end
13 return U, V, g

Παρατηρήστε ότι οι γραμμές $3-5$ είναι ο Αλγόριθμος 3.2.1 και μας εγγυάται ότι τελικά το $g = \gcd(x, y)$. Αν $x = 1769, y = 551$ έχουμε :
initialization: $(a, b, u, v, g, w) = (1, 0, 0, 1, 1769, 551)$.
Κατόπιν έχουμε $q = 3, w = 1769 \pmod{551} = 116, U = a - qu = 1 - 3 \cdot 0 = 1, V = b - qv = 0 - 3 \cdot 1 = -3, (a, b, u, v) = (0, 1, 1, -3)$ κ.ο.κ. Συνοψίζουμε στον παρακάτω πίνακα.

α/α	a	b	u	v	g	w	U	V	q
initalization	1	0	0	1	1769	551			
1	0	1	1	-3	551	**116**	**1**	**-3**	3
2	1	-3	-4	13	116	**87**	**-4**	**13**	4
3	-4	13	5	-16	87	**29**	**5**	**-16**	1
4	-4	13	5	-16	**29**	0			

Εύκολα επαληθεύουμε ότι $5x - 16y = 29$.

Υπάρχει και άλλη μέθοδος υπολογισμού των συντελεστών του Bézout που βασίζεται στο παρακάτω θεώρημα.

Θεώρημα 3.2.3 Ας είναι $x \geq y > 0$ δύο ακέραιοι. Θέτουμε

$$r_0 = x,\ r_1 = y,\ s_{-1} = s_0 = 1.$$

Έστω οι ακολουθίες

$$q_i = \lfloor r_{i-2}/r_{i-1} \rfloor\ (i \geq 2)$$

$$r_{i-1} = r_i q_{i+1} + r_{i+1},\ s_i = s_{i-2} - s_{i-1} q_{n-i+2},\ i = 1, 2, \ldots, n$$

όπου n τέτοιο ώστε $r_{n+1} = 0$. Τότε

$$\gcd(x, y) = d = x s_{n-1} + y s_n.$$

Απόδειξη. Οι ακέραιοι r_i είναι τα υπόλοιπα της διαίρεσης r_{i-2} διά του r_{i-1} και ισχύει $r_0 \geq r_1 > r_2 > \cdots$ οπότε αργά ή γρήγορα για κάποιο δείκτη, έστω $n+1$, θα έχουμε $r_{n+1} = 0$. Ειδικότερα το n φράσσεται όπως είδαμε στο Θεώρημα 3.2.2 από το $\ln(x\sqrt{5})$. Θα αποδείξουμε επαγωγικά ότι

$$r_n = s_i r_{n-i+1} + s_{i-1} r_{n-i}.$$

Τότε, το θεώρημα μας προκύπτει αν θέσουμε $i = n$. Παρατηρούμε ότι για $i = 0$ ισχύει. Έστω ότι ισχύει για κάποιο δείκτη $i \in \{1, 2, ..., n\}$. Θα αποδείξουμε ότι ισχύει για $i + 1$. Δηλαδή, θα αποδείξουμε ότι ισχύει

$$r_n = s_{i+1} r_{n-i} + s_i r_{n-i-1}. \tag{3.2.2}$$

Αντικαθιστούμε στο δεξιό μέλος της εξίσωσης (3.2.2), τις σχέσεις

$$s_{i+1} = s_{i-1} - s_i q_{n-i+1},\ r_{n-i-1} = r_{n-i} q_{n-i+1} + r_{n-i+1}.$$

Τότε θα έχουμε

$$s_{i+1} r_{n-i} + s_i r_{n-i-1} = (s_{i-1} - s_i q_{n-i+1}) r_{n-i} + s_i (r_{n-i} q_{n-i+1} + r_{n-i+1}) =$$

$$s_{i-1} r_{n-1} + s_i r_{n-i+2}$$

το οποίο σύμφωνα με την υπόθεση της επαγωγής ισούται με $r_n = d$. □

Αλγόριθμος 3.2.3. : Επεκτεταμένος Αλγόριθμος του Ευκλείδη (2η εκδοχή)-Υπολογισμός συντελεστών Bézout
Ορίζουμε τις συναρτήσεις $r(n,x,y)$, $q(n,x,y)$ οι οποίες επιστρέφουν αντίστοιχα τις τιμές των ακολουθιών r_n, q_n, κατόπιν ορίζουμε την συνάρτηση $N(x,y)$ η οποία επιστρέφει το πλήθος των επαναλήψεων που χρειάζεται ο ευκλείδειος αλγόριθμος για να τερματίσει. Αυτό θα χρησιμοποιηθεί στην ρουτίνα για τον υπολογισμό της s_i που δέχεται ως εισόδους (n,x,y,N). Τέλος ορίζουμε την συνάρτηση $egcd(x,y,N)$ η οποία μας επιστρέφει τις ζητούμενες τιμές μαζί με τον μέγιστο κοινό διαιρέτη.
Είσοδος. Ακέραιοι x,y με $x \geq y \geq 0$ και $x > 0$
Έξοδος. $\big(a,b,d = \gcd(x,y)\big)$ έτσι ώστε $ax + by = d$

Computation of r_i
def $r(n,x,y)$
if $n = 0$ **then**
| return x
end
if $n = 1$ **then**
| return y
else
| r(n-2,x,y)mod r(n-1,x,y)
end
Computation of N
def N(x,y)
$j = 0$
while $j < \lfloor \sqrt{5}\ln(x) \rfloor$ **do**
| **if** $r(j,x,y) = 0$ **then**
| | print $j-1$
| | break
| **else**
| | $j = j+1$
| **end**
end
Computation of q_i
def q(n,x,y)
return $\lfloor r(n-2,x,y)/r(n-1,x,y) \rfloor$
Computation of s_i
def s(n,x,y,N)
if $n = -1$ **then**
| return 1
end
if $n = 0$ **then**
| return 1
else
| return $s(n-2,x,y,N) - s(n-1,x,y,N) \cdot q(N-n+2,x,y)$
end
def egcd(x,y,N)
return $s(N-1,x,y,N), s(N,x,y,N), x \cdot s(N-1,x,y,N) + y \cdot s(N,x,y,N)$

Για την περίπτωση $x = 7168, y = 917$ αφού εκτελέσαμε την $N(x,y)$ ο αλγόριθμος επέστρεψε 6 και κατόπιν η συνάρτηση $egcd(x,y,N)$ έδωσε $(-71, 555, 7)$. Ο αλγόριθμος 3.2.2 για τα ίδια x,y επέστρεψε την τριάδα $(60, -469, 7)$ (δείτε και την αναφορά [10]). Ένα ερώτημα που προκύπτει είναι αν υπάρχει αλγόριθμος που να βρίσκει ισοσταθμισμένους συντελεστές Bézout. Το ερώτημα αυτό έχει θετική απάντηση. Τέλος, ο Egcd είναι ένα χρήσιμο κρυπταναλυτικό εργαλείο. Μέχρι σήμερα, μοντέρνοι κρυπτοαλγόριθμοι μπορούν να υποστούν επιθέσεις από τον Ευκλείδειο αλγόριθμο.

Λήμμα 3.2.2 (Λήμμα του Ευκλείδη). *Αν p πρώτος και $p|ab$, τότε $p|a$ ή $p|b$.*

Απόδειξη. Χωρίς βλάβη της γενικότητας $\gcd(p,a)=1$. Διαφορετικά, $p|a$. Αρκεί να απόδειξουμε ότι $p|b$. Από την ταυτότητα του Bézout (λήμμα (3.2.1)), υπάρχουν ακέραιοι x,y τέτοιοι ώστε $1=xp+ya$. Οπότε, $b=bxp+bya$. Αλλά, $p|bxp$ και από την υπόθεση $p|y(ab)$. Επομένως, $p|bxp+bya$ δηλ. $p|b$. □

Υπάρχει η εξής γενίκευση.

Λήμμα 3.2.3. *Αν $n|ab$ και $\gcd(n,a)=1$, τότε $n|b$.*

Άσκηση 3.24 Να αποδείξετε το λήμμα 3.2.3.

Άσκηση 3.25 Έστω N,a,b,c θετικοί ακέραιοι με $\gcd(a,b)=\gcd(b,c)=\gcd(a,c)=1$. Έστω

$$n=|\{x\in\mathbb{Z}: a\not|x, b\not|x, c\not|x, 1\le x\le N\}|.$$

Ν.α.ο. ο αριθμός

$$m=N(1-1/a)(1-1/b)(1-1/c)$$

είναι μια καλή προσέγγιση του n.

Άσκηση 3.26 Να βρείτε το μκδ των αριθμών $540, 315$. Κατόπιν αν d ο είναι ο μ.κ.δ., βρείτε ακεραίους α,β τέτοιους ώστε, $540\alpha+315\beta=d$ (κάντε τους υπολογισμούς με το χέρι).

Άσκηση 3.27 Να αποδείξετε ότι οι αριθμοί $1456, 1717$ είναι πρώτοι μεταξύ τους.

Άσκηση 3.28 Αν $\gcd(a,b)=1$ τότε
(i). για κάθε ακέραιο c ισχύει $\gcd(ac,b)=\gcd(c,b)$.
(ii). $\gcd(a+b,a-b)\in\{1,2\}$. Ειδικότερα ν.α.ο. αν a,b περιττοί, τότε $\gcd(a+b,a-b)=2$.
(iii). $\gcd(2^a-1,2^b-1)=1$.
(iv). $\gcd(M_p,M_q)=1$, όπου $M_p=2^p-1, M_q=2^q-1$ Mersenne ακέραιοι (p,q πρώτοι με $p\neq q$).

Άσκηση 3.29 (**) Αν $\sigma(n)$ το άθροισμα των διαιρετών του n και

$$H_n=1+\frac{1}{2}+\cdots+\frac{1}{n},$$

ν.α.ο.

$$\sigma(n)\le H_n+e^{H_n}\ln H_n.$$

Άσκηση 3.30 Να υλοποιήσετε τον Ευκλείδειο αλγόριθμο και κατόπιν μετρήστε πόσα βήματα απαιτούνται για τον υπολογισμό του $\gcd(121393, 75025)$.

Άσκηση 3.31 Έστω $x>y>0$ ακέραιοι αριθμοί και έστω ότι ο $\gcd(x,y)$ υπολογίζεται μετά από $n-$βήματα. Τότε να αποδείξετε ότι $x\ge F_{n+2}$, $y=F_{n+1}$. Όπου F_n η ακολουθία Fibonacci, $F_0=0$, $F_1=1$, $F_n=F_{n-1}+F_{n-2}$ $(n\ge 2)$. (Υποδ. Χρησιμοποιήστε επαγωγή.)

Άσκηση 3.32 Να αποδείξετε ότι το ελάχιστο πλήθος διαιρετών του B είναι 12 αν το πλήθος διαιρετών του AB είναι 105 και του A είναι 24.
(Υποδ. Το πλήθος των διαιρετών ενός ακεραίου $n = p_1^{a_1} p_2^{a_2} \cdots p_n^{a_n}$ είναι $\sigma(n) = (a_1 + 1)(a_2 + 1) \cdots (a_n + 1)$. Γράψτε κώδικα στο Sagemath για όλες τις δυνατές περιπτώσεις, βάζοντας τους κατάλληλους περιορισμούς.)

Άσκηση 3.33 Έστω

$$a = \prod_{p \text{ πρώτος}} p^{a_p}, \quad b = \prod_{p \text{ πρώτος}} p^{b_p}.$$

η ανάλυση σε πρώτους παράγοντες των ακεραίων a, b. Να αποδείξετε ότι

$$\gcd(a, b) = \prod_{p \text{ πρώτος}} p^{\min(a_p, b_p)}.$$

Άσκηση 3.34 Έστω p πρώτος αριθμός και

$$\sum_{1 \le x \le p-1} \frac{1}{x} = \frac{A_p}{B_p},$$

όπου $\gcd(A_p, B_p) = 1$. Ν.α.ο. $p^2 | A_p$.

3.3 Ισοτιμίες (ή Ισοδυναμίες)

Η έννοια της ισοτιμίας οφείλεται στον Gauss. Αν m ακέραιος > 1 καλούμε τους ακέραιους x, y ισότιμους modulo m αν η διαφορά $x - y$ διαιρείται από το m. Δηλ. $m | x - y$, και γράφουμε $x \equiv y \pmod m$. Ισχύει το εξής.

Λήμμα 3.3.1. *$m | x - y \Leftrightarrow$ οι x, y έχουν ίδιο υπόλοιπο διαιρούμενοι με τον m.*

Απόδειξη. ($\Rightarrow$) Έστω $0 \le \upsilon_1, \upsilon_2 < m$ τα υπόλοιπα των διαιρέσεων x, y διά του m, αντίστοιχα. Τότε $x = mq_1 + \upsilon_1$, $y = mq_2 + \upsilon_2$. Θα δείξουμε ότι $\upsilon_1 = \upsilon_2$. Έχουμε,

$$x - y = m(q_1 - q_2) + \upsilon_1 - \upsilon_2. \tag{3.3.1}$$

Άρα,

$$|\upsilon_1 - \upsilon_2| = |m(q_2 - q_1) + x - y|.$$

Αλλά, $0 \le \upsilon_1, \upsilon_2 < m$ οπότε $|\upsilon_1 - \upsilon_2| < m$. Επομένως,

$$0 \le |\upsilon_1 - \upsilon_2| = |m(q_2 - q_1) + x - y| < m.$$

Από την υπόθεση υπάρχει $q \in \mathbb{Z}$ τέτοιο ώστε,

$$x - y = mq, \tag{3.3.2}$$

άρα

$$0 \le |\upsilon_1 - \upsilon_2| = m|q_2 - q_1 + q| < m.$$

Εφόσον $|q_2 - q_1 + q|$ είναι μη-αρνητικός ακέραιος, αναγκαστικά, $|q_2 - q_1 + q| = 0$. Οπότε, $|\upsilon_1 - \upsilon_2| = 0$ ισοδύναμα, $\upsilon_1 = \upsilon_2$.
($\Leftarrow$) Αν $\upsilon_1 = \upsilon_2$, τότε $x - y = m(q_1 - q_2)$, άρα $m | a - b$. □

Το σύνολο των ακέραιων αριθμών που είναι ισότιμοι modulo m με έναν ακέραιο a ονομάζεται *κλάση* του a και συμβολίζεται με $\bar{a}$. Δηλαδή,

$$\bar{a} = \{x \in \mathbb{Z} : x \equiv a (\text{mod } m)\}.$$

Επομένως, από το προηγούμενο λήμμα, κάθε κλάση modulo m αναπαρίσταται με ένα στοιχείο του συνόλου $\{\bar{0}, \bar{1}, ..., \overline{m-1}\}$. Το σύνολο αυτό ονομάζεται *σύνολο κλάσεων υπολοίπων modulo* m και συμβολίζεται με $\mathbb{Z}_m$. Αυτό το σύνολο είναι ένας δακτύλιος με μοναδιαίο στοιχείο (ring with unity). Με απλά λόγια έχει τις ιδιότητες που έχει και το σύνολο των ακεραίων. Βέβαια, ενώ οι ακέραιοι είναι ακέραια περιοχή (integral domain), το σύνολο $\mathbb{Z}_m$ γενικά δεν είναι ακέραια περιοχή. Δηλ. υπάρχουν διαιρέτες του μηδενός. Π.χ. αν $m = 6$ τότε $\overline{2} \cdot \overline{3} = \overline{0}$. Αν ο $m = p$ είναι πρώτος τότε μερικές φορές συμβολίζεται και ως $\mathbf{F}_p$. Σε αυτή την περίπτωση είναι ακεραία περιοχή (ειδικότερα είναι σώμα (field)). Επιπλέον ισχύουν οι εξής βασικές ιδιότητες.

Λήμμα 3.3.2. (i). $a \equiv a$ (mod m).
(ii). *Αν* $a \equiv b$ (mod m), *τότε* $b \equiv a$ (mod m).
(iii). *Αν* $a \equiv b$ (mod m) *και* $b \equiv c$ (mod m) *τότε* $a \equiv c$ (mod m).

Αυτό το λήμμα μας λέει ότι η σχέση $\equiv$, είναι μια σχέση ισοδυναμίας επί του συνόλου $\mathbb{Z}$. Επίσης ισχύει η εξής βασική πρόταση.

Πρόταση 3.3.1. (i). *Έστω* $a \equiv b$ (mod m), $c \equiv d$ (mod m), *τότε* $a \pm c \equiv b \pm d$ (mod m).
(ii). *Έστω* $a \equiv b$ (mod m), *τότε* $a^n \equiv b^n$(mod m), n *θετικός ακέραιος.*
(iii). *Έστω* $a \equiv b$ (mod m), *τότε* $na \equiv nb$ (mod m), n *ακέραιος.*
(iv). *Έστω* $a \equiv b$ (mod m), *και* $d|m$, *τότε* $a \equiv b$ (mod d).
(v). *Έστω* d *κοινός διαιρέτης των* a, b *με* $a \equiv b$ (mod m). *Αν* $\gcd(d, m) = 1$, *τότε*

$$a/d \equiv b/d (\text{mod } m).$$

(vi). *Έστω* $a \equiv b$ (mod m) *και* $a \equiv b$ (mod n). *Αν* $\gcd(m, n) = 1$, *τότε*

$$a \equiv b (\text{mod } mn).$$

Απόδειξη. (vi). Ισοδύναμα, $m|a - b, n|a - b$. Επειδή $\gcd(m, n) = 1$, έχουμε από την πρόταση 3.2.1 ότι, $mn|a - b$. Το ζητούμενο έπεται.

□

Η αντίστροφη πρόταση της (iii) είναι η (v). Δηλαδή μπορώ να διαγράψω το n από την ισοδυναμία $na \equiv nb$ (mod m), μόνο αν $\gcd(n, m) = 1$. Π.χ. $30 \equiv 12$ (mod 2), ενώ (μετά την απλοποίηση του 6) $5 \not\equiv 2$ (mod 2). Μπορούμε να προσθέτουμε και να πολλαπλασιάζουμε στοιχεία του $\mathbb{Z}_m$ ως εξής:

$$\bar{a} + \bar{b} = \overline{a + b}, \quad \bar{a} \cdot b = \overline{ab}.$$

Παρατηρούμε ότι το στοιχείο $\bar{1}$ έχει την ιδιότητα $\bar{1}\cdot\bar{a}=\bar{a}$ το οποίο ονομάζεται μοναδιαίο στοιχείο του $\mathbb{Z}_m$. Αν δύο στοιχεία του $\mathbb{Z}_m$ έχουν γινόμενο $\bar{1}$, λέμε ότι είναι αντιστρέψιμα. Π.χ. $\bar{5}\cdot\bar{5}=\overline{25}=\bar{1}$, επομένως το $\bar{5}$ αντιστρέφεται και έχει αντίστροφο τον εαυτό του στο $\mathbb{Z}_6$. Επίσης, παρατηρούμε ότι το γινόμενο $\bar{3}\cdot\bar{4}=\overline{12}=\bar{0}$, δηλαδή τα στοιχεία $\bar{3},\bar{4}$ δεν είναι αντιστρέψιμα στο $\mathbb{Z}_{12}$. Το σύνολο των αντιστρέψιμων στοιχείων συμβολίζεται με $\mathbb{Z}_m^*$. Το μοναδιαίο στοιχείο πάντα αντιστρέφεται, δηλ. $\bar{1}\in\mathbb{Z}_m^*$, επομένως $\mathbb{Z}_m^*\neq\emptyset$. Ειδικότερα ισχύει

$$\mathbb{Z}_m^*=\{x\in\mathbb{Z}_m : \text{υπάρχει } a\in\mathbb{Z}_m \text{ με } ax\equiv 1(\text{mod } m)\}.$$

Στην περίπτωση όπου $m=6$ έχουμε $\mathbb{Z}_6^*=\{\bar{1},\bar{5}\}$. Το πλήθος των αντιστρέψιμων στοιχείων του $\mathbb{Z}_m$ συμβολίζεται με $\phi(m)$. Άρα $\phi(6)=2$. Επίσης $\phi(m)\geq 1$ διότι $\bar{1}\in\mathbb{Z}_m^*$. Πολλές φορές παραλείπουμε την ''μπάρα'' πάνω από τους αριθμούς, και γράφουμε $1\in\mathbb{Z}_m^*$. Δηλ. αντί του $\bar{1}$ θεωρούμε το στοιχείο 1 της κλάσης $\bar{1}$. Μία μέθοδος υπολογισμού του αντιστρόφου ενός στοιχείου mod m δίνεται από τον επεκτεταμένο Ευκλείδειο αλγόριθμο.

Παράδειγμα 3.3.1. Έστω ότι θέλουμε να υπολογίσουμε το 4^{-1} (mod 15). Εφαρμόζουμε τον επεκτεταμένο Ευκλείδειο αλγόριθμο. Έχουμε, $15=4\cdot 3+3$ και $4=3\cdot 1+1$, επομένως $1=4-3\cdot 1=4-(15-4\cdot 3)=4-15+4\cdot 3=4\cdot 4-15$ δηλαδή $1=4\cdot 4-15$. Εφαρμόζουμε mod 15 και στα δύο μέλη και έχουμε $1\equiv 4\cdot 4$ (mod 15) $\Rightarrow 4^{-1}\equiv 4$ (mod 15).

Στο προηγούμενο παράδειγμα χρησιμοποιήσαμε το εξής,

Λήμμα 3.3.3. $\gcd(x,y)=1$ *αν και μόνο αν υπάρχουν ακέραιοι* a,b *με*

$$ax+by=1.$$

Απόδειξη. Η μία φορά ($\Rightarrow$) προκύπτει από το Θεώρημα του Bézout. Το αντίστροφο προκύπτει ως εξής. Αν $d=\gcd(x,y)$ και $ax+by=1$ τότε

$$d|x, d|y \Rightarrow d|ax+by=1\Rightarrow d=1.$$

□

Εφόσον ο χρόνος υπολογισμού του Ευκλείδειου αλγόριθμου είναι $O((\log_2 N)^3)$ με $N=\max\{x,y\}$, το ίδιο ισχύει και για τον χρόνο υπολογισμού του αντίστροφου στοιχείου mod N. Επομένως η εύρεση αντιστρόφου μπορεί να γίνει σε πολυωνυμικό χρόνο (bit complexity).

Παράδειγμα 3.3.2. Να υπολογίσετε το $\phi(24)$.

Αρκεί να βρούμε τα αντιστρέψιμα στοιχεία του $\mathbb{Z}_{24}$ και κατόπιν να τα μετρήσουμε. Από τα προηγούμενα προκύπτει ότι αρκεί να βρούμε τα στοιχεία του $\mathbb{Z}_{24}$ που δεν έχουν κοινό διαιρέτη με το 24. Εύκολα βλέπουμε ότι $\phi(24)=8$.

Δίνουμε χωρίς απόδειξη το παρακάτω θεώρημα, που μας βοηθάει στον υπολογισμό του $\phi(n)$.

Θεώρημα 3.3.1 Αν $n = p_1^{a_1} \cdots p_k^{a_k}$ η πρωτογενή ανάλυση του φυσικου n σε πρώτους παράγοντες, τότε

$$\phi(n) = n\Big(1 - \frac{1}{p_1}\Big) \cdots \Big(1 - \frac{1}{p_k}\Big).$$

Παρατήρηση 3.3.1. Αν γνωρίζουμε όλους τους διαιρέτες του n και όχι την πρωτογενή του ανάλυση, επίσης μπορούμε να χρησιμοποιήσουμε το προηγούμενο θεώρημα για να υπολογίσουμε το $\phi(n)$. Δηλ. να γνωρίζουμε του πρώτους διαιρέτες του n, αλλά όχι τους εκθέτες a_i.

Πόρισμα 3.3.1. *Αν* $\gcd(m, n) = 1$, *τότε* $\phi(mn) = \phi(m)\phi(n)$.

Απόδειξη. Άσκηση για τον αναγνώστη. □

Παρατήρηση 3.3.2. Παρατηρήστε ότι αν δεν γνωρίζω την παραγοντοποίηση του n, τότε δεν γνωρίζω κάποιο τρόπο υπολογισμού του $\phi(n)$. Αυτή η παρατήρηση είναι σημαντική διότι το κρυπτοσύστημα RSA βασίζει την ασφάλεια του στην δυσκολία υπολογισμού του $\phi(n)$, όταν δεν γνωρίζουμε την παραγοντοποίηση του n.

Άσκηση 3.35 Να αποδείξετε ότι αν $a \equiv 1 \pmod 5$, τότε $a^2 \equiv 1 \pmod 5$.

Άσκηση 3.36 Να αποδείξετε ότι $a^3 \equiv 0$ ή 1 ή $8 \pmod 9$.

Άσκηση 3.37 Αν p πρώτος, να αποδείξετε ότι $(a + b)^p \equiv a^p + b^p \pmod p$.

Άσκηση 3.38 (*) Ένας φυσικός αριθμός n ονομάζεται congruent number αν υπάρχει ορθογώνιο τρίγωνο εμβαδού n που έχει πλευρές ρητού μήκους. Π.χ. αν $n = 1131$ τότε

$$a = 104, b = \frac{87}{4}, c = \frac{425}{4}.$$

ικανοποιούν $a^2 + b^2 = c^2$. Επομένως ο 1131 είναι congruent number. Ν.α.ο. επίσης ο $n = 469409$ είναι congruent number.

Το πρόβλημα αυτό εμφανίστηκε πρώτη φορά (σύμφωνα με την ιστορία του Dickson) το 962 μ.χ. στα χειρόγραφα ενός ανώνυμου άραβα. Ενώ φαίνεται απλό, είναι αρκετά σύνθετο και εμπλέκει θεωρία των ελλειπτικών καμπυλών.

Άσκηση 3.39 Ν.α.ο. **δεν** υπάρχει θετικός ακέραιος n τέτοιος ώστε

$$(2^n + 1) \equiv 0 \pmod 7.$$

Άσκηση 3.40 Να αποδείξετε ότι αν $a \equiv 1 \pmod 2$, τότε $a^4 \equiv 1 \pmod{16}$.

Άσκηση 3.41 Να αποδείξετε το λήμμα 3.3.2.

Άσκηση 3.42 Να υπολογίσετε το $\phi(1134)$ και το $\phi(2457)$.

Άσκηση 3.43 Αν $\gcd(a, m) = 1$ και $n_1 \equiv n_2 \pmod{\phi(m)}$, τότε $a^{n_1} \equiv a^{n_2} \pmod m$.
Υποδ. Εφαρμογή του Θεωρήματος του Euler.

3.3.1 Παραγοντοποίηση και υπολογισμός της $\phi(n)$

Το πρόβλημα της παραγοντοποίησης και το πρόβλημα υπολογισμού της $\phi(n)$ είναι **ισοδύναμα** προβλήματα (δείτε [12, Ενότητα 10.4]). Δηλ. αν έχω ένα πολυωνυμικό αλγόριθμο που λύνει το ένα πρόβλημα τότε μπορούμε να λύσουμε σε πολυωνυμικό χρόνο και το άλλο. Αν έχουμε την παραγοντοποιήση του n από το θέωρημα 3.3.1 μπορούμε να υπολογίσουμε το $\phi(n)$.

Για την περίπτωση RSA-ακεραίων είναι αρκετά απλό να δείξουμε αυτήν την ισοδυναμία. RSA-ακεραίους, ονομάζουμε ακέραιους της μορφής pq όπου p, q πρώτοι. Έστω $n = pq$, όπου p, q πρώτοι αριθμοί. Αν γνωρίζουμε το $\phi(n)$, τότε μπορούμε να υπολογίσουμε τα p, q. Πράγματι, $\phi(n) = (p-1)(q-1) = n - (p+q) + 1$. Επομένως, $p + q = n - \phi(n) + 1$. Αλλά $n = pq$. Επομένως, λύνουμε το σύστημα και βρίσκουμε τους p, q. Τώρα, αν γνωρίζουμε την παραγοντοποίηση του n μπορούμε να υπολογίσουμε το $\phi(n) = (p-1)(q-1)$.

3.4 Πρώτοι Αριθμοί

Ίσως το πιο διάσημο πρόβλημα των μαθηματικών *η εικασία του Riemann* μας απαντάει στο πρόβλημα της κατανομής των πρώτων αριθμών. Οι πρώτοι αριθμοί είναι οι φυσικοί > 1 που διαιρούνται μόνο με την μονάδα και τον εαυτό τους. Το σύνολο των πρώτων $\{2, 3, 5, 7, 11, ..., 101, ...\}$ είναι άπειρο. Ενώ οι πρώτοι αριθμοί είναι διεσπαρμένοι μέσα στους φυσικούς με ακανόνιστο τρόπο. Όσο μελετούνται, παρατηρούμε ότι εμφανίζουν και μία κανονικότητα. Για παράδειγμα (μία εκδοχή) του θεωρήματος των πρώτων αριθμών μας λέει ότι κατά μέσο όρο ένας πρώτος αριθμός $p \leq N$ εμφανίζεται με πιθανότητα $1/\ln N$. Έτσι αν αναζητούμε έναν πρώτο αριθμό με 2048 bits θα χρειαστούν κατά μέσο όρο $\ln 2^{2048}/2 = 1024$ προσπάθειες (διαιρέσαμε με το 2 διότι οι άρτιοι δεν είναι πρώτοι). Ειδικότερα αν $\pi(X)$ το πλήθος των πρώτων $\leq X$ τότε ισχύει η εξής ανισότητα για μεγάλα X,

$$\frac{X}{\ln(X)} < \pi(X) < \int_2^X \frac{dt}{\ln t}. \tag{3.4.1}$$

Η τελευταία συνάρτηση στην ανισότητα (3.4.1), $\int_2^X \frac{dt}{\ln t}$ $(X \geq 2)$ συμβολίζεται με $Li(X)$ (offset Logarithmic integral). Το θέωρημα των πρώτων αριθμών (PNT : Prime Number Theorem) λέει ότι,

$$\pi(x) \sim \frac{x}{\ln x}.$$

Το διατύπωσε πρώτη φορά ο Gauss το 1792, όταν ήταν 15 χρονών. Με το σύμβολο $\sim$ εννοούμε ότι ο λόγος των $\pi(x)$ και $\frac{x}{\ln x}$ για x μεγάλο, τείνει στο 1,

$$\lim_{x\to\infty} \frac{\pi(x)}{\frac{x}{\ln x}} = 1.$$

Πρώτη φόρα το PNT αποδείχτηκε από τον Hadamard και ανεξάρτητα από τον La Vallee de Pousin το 1896. Επίσης ο Tchebychev το 1852 απέδειξε ότι υπάρχουν δύο σταθερές a, b τέτοιες ώστε,

$$ax/\ln x < \pi(x) < bx/\ln x, \ x \geq 2.$$

Ο Havil το 2003 απέδειξε ότι

$$0.922 < \frac{\pi(n)}{\frac{n}{\ln n}} < 1.105.$$

Η εικασία του Riemann είναι ισοδύναμη με την πρόταση,

$$|\pi(X) - Li(X)| = O(\sqrt{X}\ln X).$$

Επίσης, το θεώρημα των πρώτων αριθμών ισοδύναμα γράφεται,

$$\pi(x) \sim Li(x).$$

Επομένως η εικασία του Riemann μας δίνει βελτιωμένο error term.

Άσκηση 3.44 Να αποδείξετε ότι $\pi(x) \geq \ln x - 1$.
Υποδ. Αρχικά ν.δ.ο. για $n \leq x < n+1$,

$$\ln x \leq 1 + \frac{1}{2} + \cdots + \frac{1}{n} \leq \sum \frac{1}{m}$$

όπου το m έχει όλους τους πρώτους διαιρέτες του $\leq x$. Άρα,

$$\sum \frac{1}{m} = \prod_{p:\ p\leq x} \sum_{k\geq 0} \frac{1}{p^k}.$$

Αλλά

$$\sum_{k\geq 0} \frac{1}{p^k} = \frac{1}{1-1/p}$$

κλ.π.

Άσκηση 3.45 Να υπολογίσετε για $2 < x < 1000$ τις τιμές των συναρτήσεων

$$\pi(x), \frac{x}{\ln x} \text{ και } Li(x).$$

Φτιάξτε ένα σχεδιάγραμμα που να απεικονίζει αυτές τις τιμες.

Άσκηση 3.46 Με το θεώρημα των πρώτων αριθμών, ν.α.ο. $p_n \sim n\ln n$, όπου p_n ο $n-$οστός πρώτος.

Άσκηση 3.47 Η συνάρτηση ζήτα του Riemann είναι

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s},\ s \in \mathbb{C},\ \Re(s) > 1.$$

Ν.α.ο.

$$\prod_{p:\text{prime}} \left(1 - p^{-s}\right)^{-1} = \zeta(s).$$

Η εικασία του Riemann λέει ότι, η αναλυτική επέκταση της συνάρτησης ζ έχει όλες τις (μη τετριμμένες) ρίζες του στον άξονα $\Re(s) = 1/2$ (οι τετριμμένες ρίζες είναι $-2, -4, ...$).

Στα συστήματα δημόσιου κλειδιού, που βασίζονται στην παραγοντοποίηση ή τον διακριτό λογάριθμο, απαιτείται η κατασκευή μεγάλων πρώτων (>1024 bits). Έχοντας την συνάρτηση $\pi(x)$ μπορούμε να αποδείξουμε το παρακάτω λήμμα.

Λήμμα 3.4.1.

$$Pr = Pr(x \in \mathbb{Z}_{>0} \text{ και περιττός}: \ \text{len}_2(x) = R, \text{ και } x \text{ πρώτος}) \approx \frac{2\log_2(e)}{R}$$

Απόδειξη. Η πιθανότητα να διαλέξω τυχαία έναν πρώτο με $R-$bits είναι,

$$\frac{\pi(2^R) - \pi(2^{R-1})}{2^R - 2^{R-1}} \approx \frac{\frac{2^R}{\ln 2^R} - \frac{2^{R-1}}{\ln 2^{R-1}}}{2^{R-1}} =$$

$$\frac{2^{R-1}}{\ln 2}\frac{R-2}{R(R-1)2^{R-1}} \approx \frac{1}{(\ln 2)R} = \frac{\log_2(e)}{R}.$$

Επειδή οι άρτιοι δεν είναι πρώτοι προκύπτει $Pr \approx \frac{2\log_2 e}{R}$. □

Σύμφωνα με τα προηγούμενα για να κατασκευάσω ένα πρώτο αριθμό με 2048 bits διαλέγω τυχαία ακέραιους αριθμούς με 2048 bits και εφαρμόζω κατόπιν ένα *τεστ πιστοποίησης πρώτων αριθμών*. Μετά από περίπου 1024 προσπάθειες (κατά μέσο όρο) με μεγάλη πιθανότητα μπορώ να βρω ένα πρώτο αριθμό με 2048 bits.

Επιπλέον θυμίζουμε ότι, κάθε ακέραιος αριθμός γράφεται με μοναδικό τρόπο σε γινόμενο πρώτων παραγόντων. Μέχρι σήμερα δεν γνωρίζουμε έναν αποτελεσματικό αλγόριθμο που να παραγοντοποιεί γρήγορα. Ενώ το πρόβλημα της πιστοποίησης πρώτου, δηλαδή αν έχω έναν ακέραιο αριθμό, να μπορώ να αποφανθώ αν είναι πρώτος ή όχι, έχει λυθεί σε πολυωνυμικό χρόνο με τον ντετερμινιστικό αλγόριθμο AKS [1, 3]. Για το πρόβλημα της παραγοντοποίησης έχουμε μέχρι στιγμής μόνο υποεκθετικού χρόνου αλγόριθμους [13]. Να σημειώσουμε εδώ ότι το πρόβλημα της παραγοντοποίησης δεν έχει αποδειχτεί ότι είναι NP-hard, ενώ αποδείχτηκε ότι είναι πολυωνυμικού χρόνου σε κβαντικούς υπολογιστές. Μέχρι σήμερα δεν έχουμε κατασκευάσει τέτοιους υπολογιστές με αρκετά μεγάλη μνήμη.

Άσκηση 3.48 Ν.α.ο. δεν υπάρχει πρώτος στο διάστημα $[n!+2, n!+n]$ για $n \geq 2$.

Άσκηση 3.49 Ν.α.ο. αν n σύνθετος, τότε υπάρχει πρώτος $p|n$ με $p \leq \sqrt{n}$.

Άσκηση 3.50 Έστω $R(p,n) = \max\{r : p^r \big| \binom{2n}{n}\}$. Ν.α.ο.
i. Για κάθε πρώτο p, $p^{R(p,n)} \leq 2n$.
ii. αν p περιττός πρώτος και $2n/3 < p \leq n$, τότε $R(p,n) = 0$.

Άσκηση 3.51 (*)(Bertrand's Postulate) Ν.α.ο. για κάθε $x > 0$ υπάρχει ένας πρώτος στο διάστημα $(x, 2x)$. Διατυπώθηκε πρώτη φόρα το 1845 από τον Joseph Bertrand και αποδείχτηκε από τον Chebyshev το 1850. Κατόπιν αρκετές αποδείξεις παρουσιάστηκαν από τους Ramanujan, Erdős κ.α.

Άσκηση 3.52 Να αποδείξετε ότι, αν ο αριθμός $M_m = 2^m - 1$ είναι πρώτος, τότε ο m είναι πρώτος. Βρείτε όλους τους πρώτους αριθμούς M_p για $p < 100$. Οι αριθμοι αυτοί ονομάζονται πρώτοι αριθμοί του Mersenne (ανακαλύφθηκαν το 1640 από τον Mersenne (1588-1648)). Επίσης, είναι ανοιχτό πρόβλημα αν υπάρχουν άπειροι πρώτοι αριθμοί του Mersenne. Οι αριθμοί της μορφής $F_n = 2^{2^n} + 1$ ονομάζονται αριθμοί του Fermat. Επίσης, να αποδείξετε ότι αν ο αριθμός $2^m + 1$ είναι πρώτος, τότε ο m είναι δύναμη του 2.

Άσκηση 3.53 Ν.α.ο. αν $p|F_n = 2^{2^n} + 1$ $(n \geq 2)$ όπου p πρώτος, τότε $p = 2^{n+1}k + 1$ για κάποιο φυσικό k. Ο Euler το 1732 έδειξε ότι $641|F_5$ ενώ ο Laundry το 1880 ότι $274177|F_6$. Βρείτε ένα πρώτο διαιρέτη του F_7. Το 2004 βρέθηκε ένας πρώτος διαιρέτης του $F_{3329780}$.

Άσκηση 3.54 (*) Βρείτε ένα πρώτο διαιρέτη του $F_9 = 2^{512} + 1$. Ο Western το 1903 βρήκε έναν πρώτο διαιρέτη του.

Άσκηση 3.55 Ν.α.ο. αν $q|M_p$ όπου p, q πρώτοι, τότε $q \equiv 1 \pmod{p}$. Ο Fermat χρησιμοποίησε το προηγούμενο για να αποδείξει ότι ο $2^{23} - 1$ δεν είναι πρώτος. Δώστε και εσείς μια απόδειξη αυτού.

Άσκηση 3.56 Οι πρώτοι αριθμοί του Mersenne εμπλέκονται και με τους τέλειους αριθμούς. Έστω n ένας ακέραιος και $\sigma(n)$ το πλήθος των θετικών διαιρετών του. Ένας ακέραιος καλείται τέλειος αν και μόνο αν $\sigma(n) = 2n$. Π.χ. $6 = 1 + 2 + 3 + 6 = 2 \cdot 6$. Να αποδείξετε ότι ένας άρτιος αριθμός n είναι τέλειος αν-ν $n = 2^{q-1}M_q$, για κάποιο πρώτο q. Το αποτέλεσμα αυτό αποδείχθηκε από τον Euler. Ο Ευκλείδης απόδειξε ότι αν M_q είναι πρώτος, τότε ο αριθμός $2^{q-1}M_q$ είναι τέλειος. Για $q = 7$ προκύπτει τέλειος αριθμός;

Άσκηση 3.57 Έστω $p > 3$ πρώτος. Να αποδείξετε ότι υπάρχει μόνο μια τριάδα $(p, p + 2, p + 4)$ όπου και οι τρεις αριθμοί είναι πρώτοι.

Άσκηση 3.58 Προσεγγίστε την σταθερά Landau-Ramanujan,

$$L = \frac{1}{\sqrt{2}} \prod_{p \equiv 3 \bmod 4} \left(1 - \frac{1}{p^2}\right)^{-1/2}.$$

Αυτή η σταθέρα συνδέεται με την αρρητότητα του αριθμού $\sum_{n=0}^{\infty} \frac{1}{2^{n^2}}$.

Άσκηση 3.59 Επαληθεύστε ότι για αρκετά μεγάλα $n \geq 1$,

$$\sum_{p \leq n} \ln p \leq 2n \ln 2.$$

Μπορείτε να το αποδείξετε;

Υποδ. Αρκεί $\prod_{p \leq n} p \leq 4^n$.

Άσκηση 3.60 Ν.α.ο. αν p περιττός πρώτος, τότε $(p-1)! \equiv -1 \pmod{p}$. Επίσης ισχύει και το αντίστροφο. Αυτή η πρόταση ονομάζεται θεώρημα του Wilson. Πρώτη φορά παρατηρήθηκε από τον Πέρση al-Haytham (964-1040 μ.χ.).

Άσκηση 3.61 Ν.α.ο. αν $n \neq m$, τότε $\gcd(F_n, F_m) = 1$. Κατόπιν αποδείξτε ότι υπάρχουν άπειροι πρώτοι (με F_n εννοούμε τον $n-$οστό αριθμό του Fermat $2^{2^n}+1$).

3.5 Τεστ Πιστοποίησης πρώτων αριθμών

3.5.1 Τεστ Πιστοποίησης του Fermat

Ο Euler έχει αποδείξει ότι

Θεώρημα 3.5.1 (Euler) Για κάθε θετικό ακέραιο m πρώτο προς τον ακέραιο a, ισχύει

$$a^{\phi(m)} \equiv 1 \pmod{m}.$$

Έστω ότι ζητάμε το $a^{-1} \pmod{m}$. Μπορούμε να χρησιμοποιήσουμε το θεώρημα του Euler ως εξής,

$$a^{-1} \equiv a^{\phi(m)-1} \pmod{m}.$$

Με αυτόν τον τρόπο έχουμε μία μέθοδο υπολογισμού αντίστροφων στοιχείων mod m. Ειδικότερα για $m = p$ πρώτο και $p \nmid a$ προκύπτει

$$a^{p-1} \equiv 1 \pmod{p} \text{ (Μικρό Θεώρημα του Fermat).}$$

Ισοδύναμα γράφεται

$$a^p \equiv a \pmod{p}, \tag{3.5.1}$$

για κάθε ακέραιο a και πρώτο p.

Απόδειξη του Μικρού θεωρήματος του Fermat. Παρατηρούμε ότι το σύνολο $\mathcal{A} = \{ja \in \mathbb{Z}_p : j = 1, 2, ..., p-1\}$ αποτελείται από διακριτά στοιχεία. Πράγματι, αν $ja \equiv ia \pmod{p}$ προκύπτει $j \equiv i \pmod{p}$. Αλλά, $i, j < p$ όποτε, $p|i-j$ και $|i-j| < p$. Επομένως, $i = j$. Εφόσον η ομάδα $\mathbb{Z}_p^*$ έχει $p-1$ στοιχεία,

$$\prod_{i=1}^{p-1} ia \equiv \prod_{i=1}^{p-1} i \pmod{p}$$

ισοδύναμα

$$a^{p-1} \prod_{i=1}^{p-1} i \equiv \prod_{i=1}^{p-1} i \pmod{p}$$

και επειδή $\gcd\left(\prod_{i=1}^{p-1} i,\ p\right) = 1$, προκύπτει

$$a^{p-1} \equiv 1 \pmod{p}.$$

□

Άσκηση 3.62 Το μικρό θεώρημα του Fermat μας λέει ότι $p|2^{p-1}-1$ (για p περιττό πρώτο). Μερικές φορές το $p^2|2^{p-1}-1$. Οι πρώτοι αριθμοί με αυτή την ιδιότητα ονομάζονται Wieferich primes. Να βρείτε όλους τους Wieferich πρώτους που είναι < 2000. Είναι ανοιχτό πρόβλημα στην θεωρία αριθμών, το αν υπάρχουν άπειροι Wieferich πρώτοι αριθμοί.

Το θεώρημα του Euler μας επιτρέπει να δώσουμε τον παρακάτω ορισμό.

Ορισμός 3.5.1. *Αν* $\gcd(a,m)=1$ *ονομάζουμε τάξη του* $a \bmod m$ *τον ελάχιστο φυσικό αριθμό* n *που ικανοποιεί την ισοτιμία*

$$a^n \equiv 1 \pmod m.$$

Την τάξη του στοιχείου $a \pmod m$ *την συμβολίζουμε* $ord_m(a)$.

Επομένως, $ord_m(a)|\phi(m)$. Πράγματι, αν θέσουμε $d=ord_m(a)$ έχουμε,

$$a^d \equiv 1 \pmod m, a^{\phi(m)} \equiv 1 \pmod m.$$

Από Bézout υπάρχουν x,y τέτοια ώστε $dx+\phi(m)y=\gcd(d,\phi(m))$. Τότε,

$$a^{dx+\phi(m)y} = a^{\gcd(d,\phi(m))} \equiv 1 \pmod m.$$

Λόγο της επιλογής του d προκύπτει

$$\gcd(d,\phi(m)) \geq d.$$

Αλλά, $\gcd(d,\phi(m))|d$, επομένως $\gcd(d,\phi(m))=d$. Άρα, $d|\phi(m)$. Για $m=p$ πρώτο, $ord_p(a)|p-1$.

Άσκηση 3.63 Ν.α.ο. για θετικούς ακέραιους n,a ισχύει $n|\phi(a^n-1)$.
Υποδ. Αρκεί $\mathrm{ord}_{a^n-1}(a)=n$.

Άσκηση 3.64 Ν.α.ο. για πρώτο p και φυσικό $n \not| (a-1)$ και $a^p \equiv 1 \pmod n$, τότε $ord_n(a)=p$.
Υποδ. $ord_n(a)=1$ ή p ...

Άσκηση 3.65 Έστω p πρώτος. Ν.α.ο. $ord_p(ab)=ord_p(a)+ord_p(b)$ για όλους τους ακέραιους a,b.

Άσκηση 3.66 Έστω p πρώτος με $p|M_n$, $M_n=2^n-1$. Ν.α.ο. $ord_p(M_{kn}) = ord_p(M_n)+ord_p(k)$.

Άσκηση 3.67 Ν.α.ο. $\gcd(M_n,M_m)=M_{\gcd(n,m)}$. $(M_n=2^n-1)$.

Άσκηση 3.68 Ν.α.ο. $a^x \equiv a^y \pmod n \Rightarrow x \equiv y \pmod{ord_n(a)}$.

Άσκηση 3.69 Ν.α.ο. δεν υπάρχει ακέραιος $n \geq 2$ με $n|2^n-1$.

Άσκηση 3.70 (**) Ν.α.ο. για κάθε θετικό ακέραιο n υπάρχει ακέραιος $m > 0$ με $m \neq n$ τέτοιος ώστε $\phi(n) = \phi(m)$.

Κάνοντας χρήση του θεωρήματος του Fermat μπορούμε να έχουμε ένα αποτελεσματικό κριτήριο για το αν ένας ακέραιος είναι σύνθετος, εκτελώντας τον παρακάτω πιθανοτικό αλγόριθμο. Ο αλγόριθμος αυτός είναι Monte Carlo, δηλαδή εξάγει True με πιθανότητα 1, αλλά αν εξάγει False υπάρχει μια πιθανότητα το πραγματικό αποτέλεσμα να είναι true. Αν όμως τον εκτελέσουμε αρκετές φορές, θα δείξουμε ότι η πιθανότητα αποτυχίας μειώνεται σημαντικά. Οι αριθμοί που εξάγονται στην περίπτωση αυτή ονομάζονται *πιθανοί πρώτοι* (*probable primes*).

Αλγόριθμος 3.5.1. : Τεστ του Fermat
Είσοδος. Τυχαίο ακέραιο n
Έξοδος. Αν είναι σύνθετος εξάγει True, διαφορετικά Probable prime with respect a

```
1 a ←R {2, ..., n − 1}
2 if gcd(a, n) > 1 then
3 |   return True
  end
4 if a^{n−1} ≡ 1 (mod n) then
5 |   return Probable prime with respect a:n πιθανός πρώτος
  else
6 |   return True
  end
```

Στην γραμμή 2, εφόσον ο $a < n$ και $\gcd(a, n) > 1$ προκύπτει ότι ο n είναι σύνθετος. Στην γραμμή 4 του αλγορίθμου υπάρχει μια πιθανότητα το αποτέλεσμα να είναι λάθος, δηλαδή ο αριθμός να είναι σύνθετος. Γι′αυτό το λόγο τους σύνθετους αριθμούς που ικανοποιούν την ισοτιμία (3.5.1) τους ονομάζουμε *ψευδοπρώτους αριθμούς του Fermat* ως προς την βάση a (*Fermat pseudoprimes with respect* a). Γενικά η έξοδος του προηγούμενου αλγορίθμου είναι ένας *πιθανός πρώτος* (probable prime).

Ορισμός 3.5.2. *Ενας σύνθετος αριθμός n που ικανοποιεί την ισοτιμία*

$$a^n \equiv a \pmod{n} \tag{3.5.2}$$

για κάποιο a, $2 \leq a \leq n-1$, ονομάζεται ψευδοπρώτος αριθμός του Fermat ως προς την βάση a.

Ορισμός 3.5.3. *Ορίζουμε την συνάρτηση του Carmichael $\lambda(m)$, να είναι ο ελάχιστος φυσικός αριθμός n τέτοιος ώστε, $a^n \equiv 1 \pmod{m}$ για κάθε ακέραιο a πρώτο προς τον m.*

Εξ ορισμού $ord_m(a)|\lambda(m)$. Ο $\lambda(m)$ είναι η μεγαλύτερη τάξη mod m που μπορούμε να έχουμε. Αν $n = p_1 p_2 \cdots p_r$, τότε $\lambda(n) = \text{lcm}(p_1 - 1, p_2 - 1, ..., p_r - 1)$.

Άσκηση 3.71 Να υπολογιστούν τα $\phi(65520)$, $\lambda(65520)$.

Για παράδειγμα ο 91 είναι ψευδοπρώτος ως προς την βάση 3, διότι $3^{91} \equiv 3 (\text{mod } 91)$ και $91 = 7 \cdot 13$, σύνθετος. Επίσης, ο αριθμός 341 είναι ψευδοπρώτος

αριθμός του Fermat ως προς την βάση 2. Γενικά υπάρχουν άπειροι ακέραιοι που είναι ψευδοπρώτοι αριθμοί του Fermat ως προς μια βάση a (για $a \geq 2$)[9]. Η περίπτωση $a = 1$ είναι αδιάφορη, διότι όλοι οι θετικοί ακέραιοι είναι ψευδοπρώτοι ως προς την βάση $a = 1$. Αν $\gcd(a, n) = 1$ τότε μπορούμε να απλοποιήσουμε το a στην σχέση (3.5.2). Τώρα θα αποδείξουμε ότι υπάρχουν άπειροι ψευδοπρώτοι αριθμοί του Fermat ως προς μια βάση $a \geq 2$.

Θεώρημα 3.5.2 Για κάθε ακέραιο $a \geq 2$ υπάρχουν άπειροι ψευδοπρώτοι ως προς την βάση a.

Χρειαζόμαστε ένα βοηθητικό λήμμα.

Λήμμα 3.5.1. (i). *Έστω $n, m \in \mathbb{Z}_{>0}$. Αν $n|m$, τότε $a^n - 1|a^m - 1$.*
(ii). *Ο αριθμός $\frac{a^{2p}-a^2}{a^2-1}$ για κάθε περιττό πρώτο p και ακέραιο $a \geq 2$, είναι άρτιος ακέραιος αριθμός.*
(iii). *Αν p περιττός πρώτος με $p \not| a^2 - 1$ (για κάποιο $a \geq 2$) τότε $p|\frac{a^{2p}-a^2}{a^2-1}$.*

Απόδειξη. (i). Εφόσον $n|m$ υπάρχει ακέραιος k τέτοιος ώστε $m = nk$. Επειδή, n, m θετικοί ακέραιοι, και ο k θα είναι θετικός ακέραιος. Αναπτύσσοντας την ταυτότητα $a^m - 1 = (a^n)^k - 1$ προκύπτει το ζητούμενο.
(ii). Θέτουμε $N = \frac{a^{2p}-a^2}{a^2-1}$. Ισχύει,

$$N = \frac{a^2}{a^2-1}(a^{2p-2} - 1) =$$

$$\frac{a^2}{a^2-1}(a^2 - 1)(a^{2(p-2)} + a^{2(p-3)} + \cdots + 1) =$$

$$a^{2p-2} + a^{2p-4} + \cdots + a^2.$$

Άρα ο αριθμός N είναι ακέραιος και άρτιος, διότι είναι άθροισμα άρτιου πλήθους ακέραιων αριθμών.
(iii). Από το μικρό θεώρημα του Fermat ισχύει, $a^p \equiv a \pmod p$ άρα $a^{2p} \equiv a^2 \pmod p$. Επομένως,

$$p|a^{2p} - a^2.$$

Αλλά $p \not| a^2 - 1$.Το ζητούμενο έπεται. □

Απόδειξη του θεωρήματος 3.5.2. Έστω $a \geq 2$. Ας είναι p περιττός πρώτος τέτοιος ώστε $p \not| a^2 - 1$. Θέτουμε

$$n = \frac{a^{2p} - 1}{a^2 - 1}.$$

Από το προηγούμενο λήμμα (i), ο $n \in \mathbb{Z}$. Αρχικά παρατηρούμε ότι ο n είναι σύνθετος. Ισχύει,

$$n = A \cdot B,$$

[9] https://de.wikibooks.org/wiki/Pseudoprimzahlen:_Tabelle_Fermatsche_Pseudoprimzahlen

όπου

$$A = \frac{a^p - 1}{a - 1} \in \mathbb{Z}, \ B = \frac{a^p + 1}{a + 1} \in \mathbb{Z}.$$

Επίσης, $a+1|a^p+1$. Αρκεί να αποδείξουμε ότι $a^{n-1} \equiv 1 \pmod{n}$. Παρατηρούμε ότι

$$n - 1 = \frac{a^{2p} - a^2}{a^2 - 1}.$$

Από το λήμμα 3.5.1 (ii)

$$2|\frac{a^{2p} - a^2}{a^2 - 1}.$$

Από λήμμα 3.5.1 (iii) επίσης έχουμε,

$$p|\frac{a^{2p} - a^2}{a^2 - 1}.$$

Επομένως, από την Πρόταση 3.2.1, $2p|n-1$. Από το (i) του ίδιου λήμματος, προκύπτει $a^{2p}-1|a^{n-1}-1$. Αλλά, εξ ορισμού του n έχουμε $n|a^{2p}-1$, επομένως $n|a^{n-1}-1$. Άρα,

$$a^{n-1} \equiv 1 \pmod{n}.$$

□

Επομένως, πρέπει να εκτελέσουμε τον αλγόριθμο 3.5.1 αρκετές φορές, ώστε να είμαστε σίγουροι ότι ο αριθμός μας είναι πρώτος. Για την περίπτωση που είναι σύνθετος, η έξοδος του αλγορίθμου είναι πάντα σωστή.

Θα μελετήσουμε την πιθανότητα αποτυχίας του Test.

Θεώρημα 3.5.3 Έστω $n \geq 2$ και υπάρχει b με $\gcd(b, n) = 1$ και $b^{n-1} \not\equiv 1 \pmod{n}$. Τότε,

$$\mathcal{P} = Pr\Big(x \xleftarrow{R} \{1, 2, ..., n-1\} : x^{n-1} \equiv 1 \pmod{n}\Big) \leq \frac{1}{2}.$$

Απόδειξη. Έστω

$$\mathcal{A} = \{1 \leq x \leq n-1 : x^{n-1} \equiv 1 \pmod{n}\}$$

$$\mathcal{B} = \{1 \leq x \leq n-1 : \gcd(x, n) = 1, \ x^{n-1} \not\equiv 1 \pmod{n}\}$$

$$\mathcal{C} = \{1 \leq x \leq n-1 : x^{n-1} \not\equiv 1 \pmod{n}, \ \gcd(x, n) > 1\}.$$

Από τη υπόθεση έχουμε ότι $\mathcal{B} \neq \emptyset$, δηλαδή υποθέσαμε ότι ο n είναι σύνθετος (από την υπόθεση $b^{n-1} \not\equiv 1 \pmod{n}$). Τα σύνολα αυτά είναι ξένα ανα δύο μεταξύ τους και ισχύει

$$|\mathcal{A}| + |\mathcal{B}| + |\mathcal{C}| = n - 1.$$

Επίσης, $|\mathcal{A}| \leq |\mathcal{B}|$. Πράγματι, η απεικόνιση $\Phi_b : \mathcal{A} \to \mathcal{B}$, $\Phi_b(x) = bx \bmod n$ είναι $1-1$ (εφόσον $\gcd(b,n) = 1$). Επομένως,

$$n - 1 = |\mathcal{A}| + |\mathcal{B}| + |\mathcal{C}| \geq |\mathcal{A}| + |\mathcal{A}| = 2|\mathcal{A}|,$$

άρα, $|\mathcal{A}| \leq \frac{n-1}{2}$, και το ζητούμενο έπεται, διότι

$$\mathcal{P} = \frac{|\mathcal{A}|}{n-1} \leq \frac{(n-1)/2}{n-1} = \frac{1}{2}.$$

□

Επομένως μετά από k εκτελέσεις η πιθανότητα ο αλγόριθμος 3.5.1 στο βήμα 3 να επιστρέψει έναν probable prime, ενώ ο αριθμός είναι σύνθετος είναι, $< \frac{1}{2^k}$. Μπορούμε να μετατρέψουμε τον αλγόριθμο 3.5.1 σ΄ ένα Monte Carlo αλγόριθμο, ως εξής.

Αλγόριθμος 3.5.2. : Τεστ του Fermat-2
Είσοδος. Ο φυσικός αριθμός n που θέλουμε να παραγοντοποιήσουμε και το πλήθος των επαναλήψεων k.
Έξοδος. Σύνθετος αν ο n είναι σύνθετος, διαφορετικά Πρώτος με πιθανότητα $1 - 2^{-k}$

```
1  i = 1
2  j = 0
3  while i ≤ k do
4  |  a ←R {2, ..., n − 1}
5  |  if gcd(a, n) > 1 then
6  |  |  return n : Σύνθετος
   |  end
7  |  if a^{n−1} ≡ 1 (mod n) then
8  |  |  j = j + 1
9  |  |  if j = k then
10 |  |  |  return n : Πρώτος με πιθανότητα 1 − 2^{−k}
   |  |  end
   |  end
11 |  i ← i + 1
   end
12 return n : Σύνθετος
```

Παρατήρηση 3.5.1. Με λίγη θεωρία ομάδων, επίσης μπορούμε να αποδείξουμε το προηγούμενο θεώρημα. Έστω $\mathbb{Z}_n^*$ η ομάδα όλων των αντιστρέψιμων στοιχείων mod n. Ας είναι $G = \{x \in \mathbb{Z}_n^* : x^{n-1} \equiv 1 \bmod n\}$. Η G είναι υποομάδα της $\mathbb{Z}_n^*$. Από το Θεώρημα του Lagrange $|G| \big| |\mathbb{Z}_n^*|$ (η τάξη της υποομάδας G διαιρεί την τάξη της ομάδας $\mathbb{Z}_n^*$). Εφόσον ο n είναι σύνθετος, $|G| < |\mathbb{Z}_n^*|$, επομένως, $|G| \leq |\mathbb{Z}_n^*|/2 = \phi(n)/2$. Άρα,

$$Pr\Big(x \xleftarrow{R} \mathbb{Z}_n^* : x \in G\Big) \leq \frac{\phi(n)/2}{\phi(n)} = \frac{1}{2}.$$

Υπάρχει η περίπτωση ο προηγούμενος αλγόριθμος να δώσει έξοδο Πρώτος, αλλά ο n να είναι σύνθετος, όσο και αν αυξήσουμε το k (επαναλήψεις). Δηλαδή,

υπάρχει περίπτωση ένας αριθμός n να είναι ψευδοπρώτος αριθμός του Fermat ως προς όλες τις βάσεις $a \in \{2, 3, ..., n-1\}$.

Ορισμός 3.5.4. *(Carmichael). Ένας σύνθετος αριθμός $n > 1$ που ικανοποιεί την ισοτιμία*

$$a^n \equiv a \pmod{n}$$

για κάθε θετικό ακέραιο a, ονομάζεται αριθμός του Carmichael.

Αν $\gcd(a, n) = 1$ τότε $a^{n-1} \equiv 1 \pmod{n}$. Δηλ. οι αριθμοί Carmichael είναι εκείνοι οι σύνθετοι, για τους οποίους ισχύει $a^{n-1} \equiv 1 \pmod{n}$ για κάθε θετικό ακέραιο a με $\gcd(a, n) = 1$ και $1 < a < n$. Οι αριθμοί Carmichael είναι περιττοί (άσκησης 3.74). Με αλλά λόγια οι αριθμοί Carmichael είναι οι περιττοί σύνθετοι αριθμοί που πετυχαίνουν στο Τεστ του Fermat. Δεν είναι προφανές αν υπάρχουν αριθμοί Carmichael. Τον μικρότερο αριθμό του Carmichael, **561**, τον ανακάλυψε ο Carmichael το 1910, απ΄ όπου και πήραν το όνομα τους αυτοί οι αριθμοί. Αυτός είναι ο πρώτος αριθμός Carmichael που βρέθηκε. Μερικοί ακόμη αριθμοί του Carmichael είναι οι $\{1105, 1729, 2465, 2821, 6601, 8911\}$[10]. Ο αριθμός 1729 είναι γνωστός και ως taxicub number ή αριθμός του Ramanujan. Υπάρχει περίπτωση ο αλγόριθμος 3.5.2 να δώσει ως έξοδο σύνθετος, για την περίπτωση ενός Carmichael, στην περίπτωση όπου $\gcd(a, n) > 1$.

Άσκηση 3.72 Να αποδείξετε ότι ο μικρότερος φυσικός αριθμός που γράφεται ως $x^3 + y^3$ $(x, y > 0)$ με **δύο** διαφορετικούς τρόπους, είναι ο 1729. Αν δεν μπορείτε θεωρητικά, χρησιμοποιήστε Η/Υ για να ελέγξετε όλες τις δυνατές περιπτώσεις. Με χρήση του προγράμματός σας βρείτε και τον μικρότερο φυσικό αριθμό που γράφεται ως άθροισμα δύο θετικών κύβων με **τρεις** διαφορετικούς τρόπους. Γενικά, υπάρχει η ακολουθία $T(n)$ που είναι ο ελάχιστος φυσικός αριθμός που γράφεται ως άθροισμα δύο κύβων με $n-$ διαφορετικούς τρόπους[11]. Π.χ. $T(1) = 2$ και $T(2) = 1729$.

Οι Alford, Granville και Pomerance το 1994 απέδειξαν ότι υπάρχουν άπειροι αριθμοί του Carmichael. Ειδικότερα απέδειξαν το παρακάτω,

Θεώρημα 3.5.4 Υπάρχουν άπειροι αριθμοί του Carmichael. Ειδικότερα, για μεγάλο x αν $C(x)$ το πλήθος των αριθμών Carmichael που είναι μικρότεροι του x, τότε $C(x) > x^{2/7}$.

Απόδειξη. [9, Θεώρημα 3.4.7] □

Οπότε στην περίπτωση που η είσοδος είναι αριθμός του Carmichael το Test Fermat θα αποτύχει όσες φορές και αν εκτελέσουμε τον αλγόριθμο (δηλ. θα αποτύχει για κάθε $a \in \{2, ..., n-1\}$).

Παρατήρηση 3.5.2. Ο Erdős το 1956 απέδειξε ότι για μεγάλα X ισχύει

$$C(X) > X \exp\Big(-\frac{O(1)\ln X \ln\ln\ln X}{\ln\ln X}\Big).$$

[10] https://oeis.org/A002997
[11] https://oeis.org/A011541

Παρατήρηση 3.5.3. Ο εκθέτης 2/7 βελτιώθηκε το 2008 στο 1/3, από τον Harman. Πειραματικά βρέθηκε ότι $C(10^{15}) = 105212$ (Pinch, 1993) και $C(10^{21}) = 20138200$ (Pinch, 2007). Έτσι αν διαλέξουμε τυχαία έναν αριθμό το πολύ 70-bits, η πιθανότητα να είναι αριθμός του Carmichael είναι $\approx 2^{-45}$.

Εφόσον οι αριθμοί Carmichael είναι σχετικά σπάνιοι το τεστ του Fermat γίνεται ένα αποτελεσματικό τεστ πιστοποίησης πρώτων αριθμών με πιθανότητα επιτυχίας $> 1 - \frac{1}{2^k}$. Η bit-πολυπλοκότητα του αλγορίθμου είναι $O((\log_2 n)^3)$ για $k \ll \log_2 n$. Να συμπληρώσουμε ότι δεν έχει αποδειχτεί ότι οι αριθμοί Carmichael είναι σπάνιοι, αλλά πειράματα που έχουν γίνει μας επιτρέπουν να ισχυριστούμε ότι είναι σπάνιοι [12].

Άσκηση 3.73 (*)(Korselt, 1899). Ν.α.ο. ο $N > 2$ είναι ένας αριθμός του Carmichael αν και μόνο αν ο N είναι σύνθετος, ελεύθερος τετραγώνων[13] και για κάθε πρώτο $p|N$, ο $p - 1|N - 1$.

Άσκηση 3.74 (*) Έστω Λ ένας θετικός ακέραιος και

$$\mathcal{P}_\Lambda = \{p : p \text{ πρώτος } p - 1|\Lambda,\ p \not| \Lambda\}.$$

Να βρείτε μια εκτίμηση του $|\mathcal{P}_\Lambda|$.

Άσκηση 3.75 Ν.α.ο. ο 341 είναι ψευδοπρώτος αριθμός του Fermat ως προς την βάση $a = 2$. Επίσης, ν.α.ο. δεν είναι Carmichael. Ο αριθμός 3914864773 είναι ψευδοπρώτος αριθμός του Fermat ως προς την βάση 2;

Άσκηση 3.76 Ν.α.ο. οι αριθμοί 9999109081, 6553130926752006031481761 είναι αριθμοί του Carmichael. Μπορείτε να βρείτε κάποιον μεγαλύτερο;

Άσκηση 3.77 Ν.α.ο. ο 561 είναι ο μικρότερος αριθμός Carmichael με τρεις πρώτους παράγοντες. Βρείτε τον μικρότερο Carmichael με τέσσερις πρώτους παράγοντες.

Άσκηση 3.78 Με το κριτήριο του Korselt να αποδείξετε ότι αν N είναι ένας αριθμός του Carmichael, τότε είναι περιττός.

Άσκηση 3.79 Με το κριτήριο του Korselt να αποδείξετε ότι αν N είναι ένας αριθμός του Carmichael, τότε έχει τουλάχιστον τρεις πρώτους διαιρέτες.

Άσκηση 3.80 Με το κριτήριο του Korselt να αποδείξετε ότι αν N είναι ένας αριθμός του Carmichael, τότε όλοι οι πρώτοι του παράγοντες είναι $< \sqrt{N}$.

Άσκηση 3.81 Βρείτε έναν ψευδοπρώτο ως προς την βάση 10 που είναι μεγαλύτερος από το 10000.

[12] http://www.cecm.sfu.ca/Pseudoprimes/index-2-to-64.html περιέχει όλους τους Carmichael αριθμούς που είναι $< 2^{64}$

[13] δηλ. είναι της μορφής $N = p_1 p_2 \cdots p_r$ όπου p_i πρώτοι, όλοι διαφορετικοί μεταξύ τους.

Άσκηση 3.82 Έστω το σύνολο $\Sigma(X) = \{n \in \mathbb{Z} : 0 < n \leq X,\ n^2+1 \text{ πρώτος}\}$. Για διάφορες τιμές του X (π.χ. $X = 100, 200, ..., 10000$) να βρείτε το $|\Sigma(X)|$ με χρήση του Τεστ του Fermat. Η θεωρία λέει ότι $|\Sigma(X)| = O(X/\ln(X))$.

Άσκηση 3.83 Βρείτε δύο πρώτους αριθμούς της μορφής $12m+1$ και $24m+1$. Κατόπιν να δείξετε ότι ο $n = (12m+1)(24m+1)$ είναι ψευδοπρώτος ως προς την βάση 2 και 3.

Άσκηση 3.84 Να δείξετε ότι ο αριθμός 3215 είναι Carmichael και ισχυρός ψευδοπρώτος ως προς την βάση 7.

Άσκηση 3.85 Ικανοποιούν το Τεστ του Fermat οι αριθμοί

$$557717, 6203837, 1234567892;$$

Το ίδιο για τον αριθμό που αρχίζει από το 82 και μειώνεται κατά 1, μέχρι να φτάσουμε στο 1, δηλ.

$$8281 \cdots 504948 \cdots 10987654321.$$

Άσκηση 3.86 Ικανοποιούν το Τεστ του Fermat οι αριθμοί

$$835335 \cdot 2^{39014} \pm 1;$$

Ζευγάρια πρώτων αριθμών της μορφής $(p, p+2)$ ονομάζονται δίδυμοι πρώτοι. Είναι ανοιχτό πρόβλημα αν υπάρχουν άπειροι δίδυμοι πρώτοι. Έχει αποδειχτεί από το μεγάλο Κινέζο μαθηματικό Jing R. Chen (1933-1996), ότι υπάρχουν άπειρα ζευγάρια της μορφής $(a, a+2)$ όπου το a και το $a+2$ έχουν το πολύ δύο πρώτους παράγοντες.

Το πρόβλημα των δίδυμων πρώτων ισοδύναμα γράφεται

$$\liminf_{n\to\infty}(p_{n+1} - p_n) = 2.$$

Ο Zhang απέδειξε ότι,

$$\liminf_{n\to\infty}(p_{n+1} - p_n) < 7 \times 10^7.$$

Αργότερα ο Maynard βελτίωσε το άνω φράγμα στο 600 και το polymath project[14] στο 246.

Άσκηση 3.87 Ο Brun απόδειξε ότι η σειρά,

$$B = \sum_{p\in\mathcal{P}_2}\Big(\frac{1}{p} + \frac{1}{p+2}\Big),$$

συγκλίνει, όπου $\mathcal{P}_2$ το σύνολο των δίδυμων πρώτων. Βρείτε μια προσέγγιση αυτής της σειράς (είναι περίπου 1.902).

Άσκηση 3.88 Το σύνολο, $\{7, 37, 67, 97, 127, 157\}$, είναι μια αριθμητική πρόοδος πρώτων αριθμών. Μπορείτε να βρείτε άλλες τέτοιες ακολουθίες με μεγαλύτερο μήκος; Το θεώρημα του Dirichlet μας εγγυάται ότι υπάρχουν.

[14] DHJ Polymath,Variants of the Selberg sieve, and bounded intervals containing many primes

Άσκηση 3.89 Αποδείξτε (υπολογιστικά) ότι όλοι οι πρώτοι μεταξύ 1000 και 2000 είναι άθροισμα τριών πρώτων (όχι κατ'ανάγκη διαφορετικών). Ο Goldbach σ' ένα γράμμα του προς τον Euler (1742) ισχυρίστηκε ότι κάθε αριθμός μεγαλύτερος του 5 είναι άθροισμα τριών πρώτων. Ο Euler απάντησε ότι αυτό έπεται από τον ισχυρισμό ότι κάθε άρτιος μεγαλύτερος του 2, είναι άθροισμα δύο πρώτων (το τελευταίο είναι γνωστό ως η εικασία του Goldbach). Έχει αποδειχτεί ότι όλοι οι περιττοί αριθμοί $> 2 \cdot 10^{1346}$ μπορούν να γραφούν ως άθροισμα τριών πρώτων. Πειραματικά έχει υπολογιστεί ότι όλοι οι άρτιοι n με $2 < n \leq 4 \cdot 10^{18}$ είναι άθροισμα δύο πρώτων.

Άσκηση 3.90 Υπολογίστε τις τιμές του πολυωνύμου $f(x) = x^2 + x + 41$ για $x = 0, 1, ..., 40$. Εφαρμόστε το Τεστ του Fermat για να ελέγξετε αν οι τιμές που προκύπτουν είναι πρώτοι. Τι παρατηρείτε; Αν αντί 41 έχουμε -1354363 τι παρατηρείτε;

Άσκηση 3.91 Χρησιμοποιώντας αριθμούς της μορφής

$$(2 \cdot 3 \cdots p)^2 + 1,$$

αποδείξτε ότι υπάρχουν άπειροι πρώτοι $\equiv 1 \pmod 4$.

Άσκηση 3.92 Έστω a, n φυσικοί αριθμοί με $a \geq 2$. Έστω $N = a^n - 1$. Αφού δείξετε ότι η τάξη του $a \pmod N$ είναι n, να συμπεράνετε ότι $n|\phi(N)$. Κατόπιν, αν ο n είναι πρώτος αριθμός, να δείξετε ότι υπάρχουν άπειροι πρώτοι αριθμοί $\equiv 1 (\mathrm{mod}\ n)$.

Άσκηση 3.93 Ν.α.ο.

$$\lim_{x \to \infty} \frac{\pi(x)}{x} = 0.$$

Άσκηση 3.94 Όταν ήταν 14 χρονών ο Gauss, διατύπωσε την εικασία

$$\frac{x}{2 \ln x} < \pi(x) < \frac{2x}{\ln x},$$

δηλ. το θεώρημα των πρώτων αριθμών. Επαληθεύστε υπολογιστικά για $x = 2^{12}$.

Άσκηση 3.95 Βρείτε 5 πρώτους της μορφής $a^2 + b^4$. Έχει αποδειχτεί ότι υπάρχουν άπειροι πρώτοι αυτής της μορφής (Friedlander & Iwaniec, 1998).

Άσκηση 3.96 Αποδείξτε ότι αν $a|b$, τότε $\lambda(a)|\lambda(b)$ (όπου $\lambda(n)$ η συνάρτηση του Carmichael, ορισμός 3.5.3).

Άσκηση 3.97 Έστω p πρώτος με $p \equiv 1 \pmod 6$. Τότε, είναι γνωστό ότι $p = x^2 + 3y^2$. Μπορείτε να το αποδείξετε; Βρείτε 100 αριθμούς της μορφής $p = x^2 + 3y^2$ που να ικανοποιούν το τεστ του Fermat.

Άσκηση 3.98 (Lucas-Lehmer Test) Έστω η αναγωγική ακολουθία

$$S_1 = 4, S_{n+1} = S_n^2 - 2, \quad n \geq 2.$$

Έστω p ένας περιττός πρώτος. Ν.α.ο. $M_p = 2^p - 1$ είναι πρώτος αν-ν $S_{p-1} \equiv 0 \pmod{M_p}$. Το 1877 ο Lucas απέδειξε ότι ο

$$M_{127} = 2^{127} - 1 = 170141183460469231731687303715884105727$$

είναι πρώτος. Να το αποδείξετε και εσείς με χρήση του κριτηρίου (χωρίς την χρήση υπολογιστή).

3.5.2 Τεστ πιστοποίησης των Miller-Rabin

Ο Miller αρχικά έδωσε ένα ντετερμινιστικό κριτήριο πιστοποίησης, υποθέτοντας την Επεκτεταμένη Υπόθεση του Riemann (GRH : Generalized Riemann Hypothesis). Την εικασία αυτή, πολλές φορές, την θεωρούμε αληθή (το πιθανότερο να είναι). Κατόπιν ο Rabin αφαίρεσε αυτή την υπόθεση και έδωσε ένα πιθανοτικό τεστ πιστοποίησης πρώτων αριθμών, αλλά δεν υπάρχει το πρόβλημα με τους αριθμούς του Carmichael και η πιθανότητα να αποτύχει είναι 4^{-k} αντί, 2^{-k} που είναι στο τεστ του Fermat. Το αντιθετοαντίστροφο του παρακάτω θεώρηματος, μας δίνει μια συνθήκη για φυσικούς n που ικανοποιούν την ισοτιμία $a^n \equiv a \pmod{n}$, να είναι σύνθετοι.

Θεώρημα 3.5.5 Έστω ότι ο n είναι ένας περιττός πρώτος και $n-1 = 2^s t$, όπου t περιττός και s θετικός ακέραιος. Έστω ένας ακέραιος $a \in \{1, 2, ..., n-1\}$ με $a \not| n$. Θέτουμε $b \equiv a^t \pmod{n}$. Τότε ισχύει το παρακάτω:

$$b \equiv 1 \pmod{n} \text{ ή } b^{2^i} \equiv -1 \pmod{n} \tag{3.5.3}$$

για κάποιο i με $0 \leq i \leq s-1$.

Ορισμός 3.5.5. *Ένας περιττός ακέραιος αριθμός $n > 3$ λέγεται ισχυρός πιθανός πρώτος ως προς την βάση a με $1 < a < n-1$ (strong probable prime), αν ισχύει η συνθήκη (3.5.3). Αν είναι σύνθετος και ικανοποιεί την (3.5.3) καλείται ισχυρώς ψευδοπρώτος αριθμός του Fermat ως προς την βάση a (strong Fermat pseudoprime to base a).*

Για παράδειγμα ο 341 που είναι ψευδοπρώτος ως προς την βάση 2, δεν είναι ισχυρός ψευδοπρώτος ως προς την βάση 2. Άρα ένας αλγόριθμος που χρησιμοποιεί τους *ισχυρούς ψευδοπρώτους* και όχι *ψευδοπρώτους αριθμούς του Fermat*, στην περίπτωση του 341 θα εξάγει: *σύνθετος για την βάση $a = 2$*. Ο ακέραιος 2047

είναι ισχυρός ψευδοπρώτος ως προς την βάση 2.

Αλγόριθμος 3.5.3. Τεστ των Miller-Rabin
Είσοδος. Τυχαίο περιττό ακέραιο $n > 3$ και k θετικό ακέραιο
Έξοδος. Σύνθετος αν ο n είναι σύνθετος, διαφορετικά, Πρώτος με πιθανότητα $1 - 4^{-k}$

```
1  i = 1
2  j = 0
3  Find s, t such that n − 1 = 2^s t
4  while i ≤ k do
5  |  a ←R {2, ..., n − 1}
6  |  b ← a^t (mod n)
7  |  if b ≡ 1 (mod n) then
8  |  |  j = j + 1
9  |  |  if j = k then
10 |  |  |  return n : strong probable prime
   |  |  end
   |  end
11 |  for 0 ≤ r ≤ s − 1 do
12 |  |  if b ≡ −1 (mod n) then
13 |  |  |  j = j + 1
14 |  |  |  if j = k then
15 |  |  |  |  return n : strong probable prime
   |  |  |  end
   |  |  end
16 |  |  b ← b^2 (mod n)
   |  end
17 |  i ← i + 1
   end
18 return n : Σύνθετος
```

Στις γραμμές 7 και 12, ο αλγόριθμος ελέγχει τις υποθέσεις του προηγούμενου θεωρήματος. Αν ο αλγόριθμος εισέλθει στην γραμμή 8, τότε δεν θα εισέλθει στην γραμμή 13. Π.χ. αν $a^t \equiv 1 \pmod n$ τότε, $(a^t)^{2^i} \equiv 1 \pmod n$ και όχι -1. Μόλις συμπληρωθούν k επιτυχίες συνολικά, δηλαδή αν ικανοποιηθεί η σχέση (3.5.3) $k-$φορές, τότε ο αλγόριθμος εξάγει strong probable prime. Για να αναλύσουμε την πιθανότητα επιτυχίας του αλγορίθμου χρειαζόμαστε το παρακάτω θεώρημα.

Θεώρημα 3.5.6 (Monier και Rabin). Το πλήθος των $a \in \{1, 2, ..., n-1\}$ για τα οποία ένας περιττός σύνθετος $n > 9$ είναι ισχυρός πιθανός πρώτος ως προς την βάση a είναι $\leq \frac{\phi(n)}{4}$.

Επομένως με πιθανότητα $> 1 - \frac{1}{4^k}$ το αποτέλεσμα της γραμμής 10 ή 15 είναι σωστό. Π.χ. για $k = 40$ η πιθανότητα αποτυχίας του τεστ είναι $\approx 2^{-80}$.

Άσκηση 3.99 Βρείτε τον μικρότερο ισχυρό ψευδοπρώτο ως προς την βάση 32.

Άσκηση 3.100 Να εξετάσετε αν ο αριθμός Fibonacci F_{104911} είναι ισχυρός πιθανός πρώτος (δείτε άσκηση 3.31).
(Υποδ. Ο αριθμός αυτός έχει 21925 ψηφία. Θα χρειαστείτε έναν κατάλληλο αλγόριθμο που να μπορεί να υπολογίζει δυνάμεις mod p για πολύ μεγάλους ακέραιους.)

Άσκηση 3.101 (Pepin's Test). Ν.α.ο. ο αριθμός του Fermat $n = 2^{2^k} + 1$ $(k \geq 2)$ είναι πρώτος αν και μόνο αν $5^{(n-1)/2} \equiv -1 \pmod{n}$. Επομένως για του αριθμούς του Fermat υπάρχει ντετερμενιστικός πολυωνυμικός αλγόριθμος για το αν είναι πρώτος ή όχι.

3.5.3 Κατασκευή μεγάλων πρώτων

Η πιθανότητα να διαλέξω ένα περιττό θετικό ακέραιο με 2048-bits και αυτός να είναι πρώτος, είναι περίπου 1/1024 (δείτε το λήμμα 3.4.1). Άρα, αρκετά γρήγορα, μετά από περίπου 1024 επαναλήψεις ο αλγόριθμος θα εντοπίσει έναν πρώτο αριθμό με μεγάλη πιθανότητα. Γενικά, στο σύστημα RSA χρειάζεται να βρούμε μεγάλους πρώτους αριθμούς (τουλάχιστον 1024 bits). Το τεστ του Fermat έχει την αδυναμία ότι μπορεί η έξοδος να είναι ένας αριθμός Carmichael και αυτό δεν μπορεί να διορθωθεί όποια επιλογή του a και αν κάνουμε. Το τεστ των Miller-Rabin είναι πιο αξιόπιστο και αυτό είναι που συνήθως χρησιμοποιούμε. Μερικές φορές οι πρώτοι (ή ακριβέστερα οι πιθανοί πρώτοι) που προκύπτουν από αυτό το τεστ ονομάζονται industrial-grade primes (ονομασία που οφείλεται στον Henri Cohen και χρησιμοποιείται και από τους Pomerance-Grandall).

3.6 Η κυκλική ομάδα $\mathbb{Z}_p^*$

Ένα εύλογο ερώτημα που προκύπτει είναι πως υπολογίζουμε την τάξη ενός ακεραίου mod n (δείτε τον ορισμό 3.5.1). Ειδικότερα αν έχω την ομάδα $\mathbb{Z}_p^*$ που έχει $p-1$ στοιχεία, ένας ακέραιος τάξης $p-1$ θα παράγει όλα τα στοιχεία του $\mathbb{Z}_p^*$. Δηλαδή, για κάθε στοιχείο $a \in \mathbb{Z}_p^*$ υπάρχει ένας ακέραιος, έστω g τάξης $p-1$ με $g^i = a$ για κάποιον φυσικό αριθμό i. Ισχύει η εξής πρόταση.

Πρόταση 3.6.1. *Έστω p ένας πρώτος. Τότε υπάρχει ένα στοιχείο $g \in \mathbb{Z}_p^*$ με $ord_p(g) = p-1$. Ισοδύναμα, $\mathbb{Z}_p^*$ είναι μια κυκλική ομάδα.*

Για παράδειγμα ας θεωρήσουμε τώρα την ομάδα $\mathbb{Z}_{19}^*$. Για να βρούμε το g μπορούμε να πάρουμε ένα στοιχείο της ομάδας (όχι το 1) τέτοιο ώστε $g^{18} \equiv 1 (\text{mod } p)$ και να το υψώνουμε σε δυνάμεις. Αν παράγει όλους τους αριθμούς της ομάδας, τότε είναι το ζητούμενο. Ας δοκιμάσουμε το 2. Τότε $\{2^i : i = 1, 2, ..., 18\} = \{1, 2, 3, ..., 18\}$. Αν αντί του 19 είχαμε έναν πρώτο 100− bits, τότε η προηγούμενη διαδικασία θα ήταν πολύ χρονοβόρα. Δίνουμε χωρίς απόδειξη το παρακάτω λήμμα.

Λήμμα 3.6.1. *Το g έχει τάξη n αν και μόνο αν $g^n \equiv 1 (\text{mod } p)$ και $g^{n/p} \not\equiv 1 (\text{mod } p)$ για κάθε πρώτο p διαιρέτη του n.*

3.7 e-οστές ρίζες mod p

Έστω p πρώτος αριθμός.

Ορισμός 3.7.1. *Το $x \in \mathbb{Z}_p$ τέτοιο ώστε $x^e = c$ στο $\mathbb{Z}_p$ καλείται $e-$οστή ρίζα του c modulo p.*

Για παράδειγμα το $7^{1/3} = 6$ στο $\mathbb{Z}_{11}$ διότι $6^3 = 7 \bmod 11$. Αλλά η τετραγωνική ρίζα του 2 στο σώμα $\mathbb{Z}_{11}$ δεν υπάρχει. Τίθεται αμέσως το ερώτημα πότε υπάρχει μια $e-$οστή ρίζα σε ένα σώμα $\mathbb{Z}_p$.

Λήμμα 3.7.1. *Αν* $\gcd(e, p-1) = 1$, *τότε υπάρχει η* $e-$*οστή ρίζα* $c^{1/e}$ *στο* $\mathbb{Z}_p$, *για κάθε* $c \in \mathbb{Z}_p^*$ *και υπολογίζεται σε πολυωνυμικό χρόνο.*

Απόδειξη. Έστω $d \equiv e^{-1} \pmod{p-1}$. Τότε υπάρχει ακέραιος k τέτοιος ώστε $de = k(p-1)+1$. Επίσης από το θεώρημα του Fermat έχουμε ότι $c^{p-1} \equiv 1 \pmod{p}$. Παρατηρούμε ότι $(c^d)^e = c^{de} = c^{k(p-1)+1} = c \cdot [c^{p-1}]^k = c \cdot 1^k = c$ στο $\mathbb{Z}_p$. Άρα το $x = c^d$ είναι $e-$οστή ρίζα $\bmod p$. Τέλος, εφόσον ο υπολογισμός του $e^{-1} \pmod{p}$ γίνεται σε χρόνο $O((\log_2 p)^2)$, ο υπολογισμός σε bit operations της $e-$οστής ρίζας $\bmod p$ γίνεται σε πολυωνυμικό χρόνο. □

Παρατηρήστε ότι για $e = 2$ και p περιττό πρώτο πάντα θα ισχύει $\gcd(e, p-1) > 1$. Άρα δεν μπορούμε να εφαρμόσουμε το προηγούμενο λήμμα για τον υπολογισμό τετραγωνικών ριζών στο $\mathbb{Z}_p$.

Ορισμός 3.7.2. *Το* $x \in \mathbb{Z}_p^*$ *ονομάζεται τετραγωνικό υπόλοιπο (Τ.Υ.)* $\bmod p$ *αν και μόνο αν έχει τετραγωνική ρίζα στο* $\mathbb{Z}_p$.

Αυτός ο ορισμός γενικεύεται και για μη πρώτους p ως εξής.

Ορισμός 3.7.3. *Έστω* n *ένας θετικός ακέραιος. Το* $x \in \mathbb{Z}_n$ *ονομάζεται τετραγωνικό υπόλοιπο (Τ.Υ.)* $\bmod n$ *αν και μόνο αν έχει τετραγωνική ρίζα στο* $\mathbb{Z}_n$ *και* $\gcd(a, n) = 1$.

Σύμφωνα με τον προηγούμενο ορισμό το 0 δεν είναι ούτε τετραγωνικό υπόλοιπο ούτε μη-τετραγωνικό υπόλοιπο $\bmod n$.

Καταρχάς παρατηρούμε ότι αν το x είναι Τ.Υ. υπάρχουν δύο στοιχεία του $\mathbb{Z}_p$ που είναι τετραγωνικές ρίζες του x. Αν για παράδειγμα y η μία τετραγωνική ρίζα τότε και το $-y$ είναι τετραγωνική ρίζα του x. Επομένως εύκολα προκύπτει το επόμενο λήμμα.

Λήμμα 3.7.2. $\#\{T.\Upsilon. \bmod p\} = \frac{p-1}{2} + 1$.

Το πλήθος των Τ.Υ. $\bmod 2p$ είναι,

$$\#\{y \in \mathbb{Z}_p^* : x^2 \equiv y (\bmod\ 2p)\}$$

το οποίο ισούται με $\#\{y : x^2 \equiv y (\bmod\ p)\} \times 2 - 1 = p$ (δύο εξισώσεις είναι ίδιες γι'αυτό αφαιρέσαμε το 1). Το θεώρημα του Euler μας δίνει ένα κριτήριο για να είναι ένα στοιχείο του $\mathbb{Z}_p^*$ Τ.Υ.

Θεώρημα 3.7.1 $x \in \mathbb{Z}_p^*$ είναι Τ.Υ. αν και μόνο αν $x^{\frac{p-1}{2}} = 1$ στο $\mathbb{Z}_p$ (p περιττός πρώτος αριθμός).

Τον αριθμό $x^{\frac{p-1}{2}}$ τον καλούμε και *σύμβολο του Legendre*. Παρατηρούμε ότι $x^{\frac{p-1}{2}} \in \{-1, 1\}$. Η απόδειξη του θεωρήματος του Euler δεν είναι κατασκευαστική, επομένως δεν μας δίνει ένα αλγόριθμο υπολογισμού των τετραγωνικών ριζών. Για την περίπτωση όπου $p \equiv 1 \pmod{4}$ μπορούμε να αποδείξουμε το παρακάτω λήμμα.

Λήμμα 3.7.3. *Αν $c \in \mathbb{Z}_p^*$ με $p \equiv 1 \pmod 4$ είναι Τ.Υ. τότε, $\sqrt{c} = c^{(p+1)/4}$ στο $\mathbb{Z}_p$.*

Απόδειξη. $[c^{(p+1)/4}]^2 = c^{(p+1)/2} = c^{(p-1)/2} \cdot c = 1 \cdot c = c$ (όλες οι πράξεις μέσα στην ομάδα $\mathbb{Z}_p^*$). □

Για την περίπτωση $p \equiv 3 (\text{mod } 4)$ μπορούμε να δουλέψουμε ως εξής. Έστω $2k = p-1$, με $k = 2^s t$ όπου t περιττός και $s \geq 0$. Τότε από το θεώρημα 3.5.1

$$c^{(p-1)/2} = c^k = c^{2^s t} \equiv 1 (\text{mod } p), \tag{3.7.1}$$

Ας είναι N ένα μη-τετραγωνικό υπόλοιπο mod p, δηλαδή, $N^{(p-1)/2} = N^{2k} = N^{2^s t} \equiv -1 (\text{mod } p)$. Εξάγουμε τετραγωνικές ρίζες από τα δύο μέλη της ισοτιμίας (3.7.1). Οπότε προκύπτει

$$c^{2^{s-1}t} \equiv \pm 1 (\text{mod } p).$$

Διακρίνουμε δύο περιπτώσεις. Αν $c^{2^{s-1}t} \equiv 1 (\text{mod } p)$ συνεχίζουμε και εξάγουμε μία τετραγωνική ρίζα, διαφορετικά πολλαπλασιάζουμε με το $N^{2^{s-1}t}$ και κατόπιν εξάγουμε μία τετραγωνική ρίζα. Συνεχίζοντας καταλήγουμε σε μια ισοτιμία της μορφής $c^t N^{2\ell} \equiv 1 (\text{mod } p)$ άρα $x \equiv \pm c^{(t+1)/2} N^{\ell} (\text{mod } p)$ (οπότε $x^2 \equiv c (\text{mod } p)$ και x είναι μία τετραγωνικά ρίζα mod p του c).

Γενικά, μπορούμε να χρησιμοποιήσουμε τον πιθανοτικό αλγόριθμο (πολυωνυμικού χρόνου) του Tonelli [9, Algorithm 2.3.8] που είναι πιο αποδοτικός από τον προηγούμενο. Δεν γνωρίζουμε κάποιον ντετερμινιστικό αλγόριθμο για τον υπολογισμό τετραγωνικών ριζών mod p, αλλά αν υποθέσουμε την Επεκτεταμένη υπόθεση του Riemann τότε αποδεικνύεται ότι υπάρχει ντετερμινιστικός αλγόριθμος.

Μέχρι σήμερα όλοι οι αλγόριθμοι υπολογισμού $e-$οστών ριζών απαιτούν να γνωρίζουμε την παραγοντοποίηση του N. Το πρόβλημα RSA (RSA problem) είναι η εύρεση $e-$οστών ριζών mod N ($e > 2$), όταν δεν γνωρίζουμε την παραγοντοποίηση του N. Είναι ανοιχτό πρόβλημα, αν το RSA-problem είναι **ισοδύναμο** με το πρόβλημα της παραγοντοποίησης.

Άσκηση 3.102 Αν a τετραγωνικό υπόλοιπο mod p, τότε ν.α.ο. η εξίσωση $x^2 \equiv a (\text{mod } p)$ έχει δύο λύσεις.
Υπόδειξη.
Αν x, y λύσεις της εξίσωσης, ν.α.ο. $x \equiv \pm y \pmod p$.

Άσκηση 3.103 Έστω q πρώτος με $q \equiv 5 (\text{mod } 8)$. Ας είναι $x \in \mathbb{Z}_q$. Ν.α.ο. αν $x^{(q-1)/4} \equiv 1 (\text{mod } q)$ τότε μια τετραγωνική του ρίζα είναι η $x^{(q+3)/8} \text{mod } q$. Ενώ αν $x^{(q-1)/4} \equiv -1 (\text{mod } q)$ τότε μια τετραγωνική του ρίζα είναι η

$$2^{-1}(4x)^{(q+3)/8} \text{mod } q.$$

3.8 Κινέζικο Θεώρημα Υπολοίπων (CRT)

Το κινέζικο θεώρημα υπολοίπων (CRT : **C**hinese **R**emainder **T**heorem) ήταν γνωστό στον Κινέζο μαθηματικό και αστρονόμο Sun-Zi τον πρώτο αιώνα μ.Χ. Στο έργο του *The Mathematical Classic of Sunzi*, περιέχεται το πρώτο ιστορικά γνωστό παράδειγμα επίλυσης συστήματος γραμμικών ισοδυναμιών.

Θεώρημα 3.8.1 Έστω $m_1, m_2, ..., m_r$ θετικοί ακέραιοι > 1, πρώτοι μεταξύ τους ανά δύο, με γινόμενο $M = m_1 \cdot m_2 \cdots m_r$. Έστω ότι δίνονται τα r αντίστοιχα υπόλοιπα $n_i \bmod m_i$. Τότε το σύστημα ισοδυναμιών και η ανισότητα

$$x \equiv n_i (\bmod\ m_i), \quad 0 \leq x < M$$

έχει μοναδική λύση. Επιπλέον, αυτή η λύση δίνεται με το ελάχιστο θετικό υπόλοιπο $\bmod M$,

$$\sum_{i=1}^{r} n_i v_i M_i,$$

όπου $M_i = M/m_i$ και τα v_i ορίζονται από τις ισοδυναμίες $v_i M_i \equiv 1 \pmod{m_i}$.

Παράδειγμα 3.8.1. Να λυθεί το σύστημα γραμμικών ισοδυναμιών,

$$\begin{cases} x \equiv 2 \pmod 5 \\ x \equiv 1 \pmod 6 \\ x \equiv 3 \pmod 7 \end{cases}$$

Θέτουμε $m_1 = 5, m_2 = 6, m_3 = 7$. Ισχύει $\gcd(m_i, m_j) = 1$, επομένως μπορούμε να εφαρμόσουμε το CRT. Υπολογίζουμε διαδοχικά $M = 5 \cdot 6 \cdot 7 = 210,\ M_1 = M/m_1 = 42,\ M_2 = 35,\ M_3 = 30$. Κατόπιν υπολογίζουμε τα αντίστροφα στοιχεία $v_i \equiv M_i^{-1} \pmod{m_i}$. Με τον Ευκλείδειο αλγόριθμο βρίσκουμε $v_1 = 3, v_2 = 5, v_3 = 4$. Επομένως,

$$x = n_1 v_1 M_1 + n_2 v_2 M_2 + n_3 v_3 M_3 =$$

$$2 \cdot 42 \cdot 3 + 1 \cdot 35 \cdot 5 + 3 \cdot 30 \cdot 4 = 787 \equiv 157 \bmod 210.$$

Παρατήρηση 3.8.1. Δίνεται $N = n_1 n_2 \cdots n_k$, (n_i, n_j) πρώτοι μεταξύ τους για $i \neq j$. Έστω η απεικόνιση

$$\Phi : \mathbb{Z}_N \to \mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} \times \cdots \times \mathbb{Z}_{n_k}$$

$$\Phi(x) = (x \bmod n_1, x \bmod n_2, \ldots, x \bmod n_k).$$

Το CRT μας εγγυάται ότι η απεικόνιση είναι επί. Επίσης η απεικόνιση είναι 1-1, πάλι από το CRT. Επομένως, είναι ένας ισομορφισμός δακτυλίων.

$$\mathbb{Z}_N \cong \mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} \times \cdots \times \mathbb{Z}_{n_k}.$$

Για παράδειγμα αν $N = 35 = 5 \times 7$, ο αριθμός 17 μπορεί να αναπαρασταθεί ως $(2 = 17 \bmod 5, 3 = 17 \bmod 7)$. Στο RSA μπορούμε να χρησιμοποιήσουμε το CRT για πιο γρήγορη αποκρυπτογράφηση (δείτε την ενότητα 5.1.1).

Πόρισμα 3.8.1. *Έστω $m_1, m_2, ..., m_r$ θετικοί ακέραιοι > 1, πρώτοι μεταξύ τους ανά δύο, με γινόμενο $M = m_1 \cdot m_2 \cdots m_r$. Τότε,*

$$a \equiv b \pmod{M} \Leftrightarrow a \equiv b \pmod{m_i}, \text{ για κάθε } i \in \{1, ..., r\}.$$

Απόδειξη. ($\Rightarrow$) Έστω $z = b \sum_{i=1}^{r} \upsilon_i M_i$. Εφόσον το a είναι μια λύση του συστήματος $x \equiv b \pmod{m_i}$, $1 \leq i \leq r$, από CRT προκύπτει $z \equiv a \pmod{M}$. Επομένως, $a \equiv b \sum_{i=1}^{r} \upsilon_i M_i \pmod{M}$. Άρα, από τον ορισμό των υ_i, M_i προκύπτει $a \equiv b \pmod{m_i}$. Πράγματι, όλοι αριθμοί $M_1, ..., M_{i-1}, M_{i+1}, ..., M_r$ διαιρούνται από το m_i, επομένως $M_j \equiv 0 \pmod{m_i}$, $j \neq i$. Ενώ, $\upsilon_i M_i \equiv 1 \pmod{m_i}$. Επομένως, $a \equiv b \pmod{m_i}$ για $i = 1, 2, ..., r$.
($\Leftarrow$) Προκύπτει επαγωγικά από την πρόταση 3.3.1 (vi). □

Παρατήρηση 3.8.2. Έστω $m = p_1^{a_1} \cdots p_r^{a_r}$ με θετικά a_i και $p_1 < p_2 < \cdots < p_r$. Τότε υπάρχουν 2^r διαφορετικές τετραγωνικές ρίζες της μονάδας modulo m. Πράγματι, από το προηγούμενο πόρισμα έχουμε ότι η ισοδυναμία $x^2 \equiv 1 \pmod{m}$ είναι ισοδύναμη με το σύστημα $x^2 \equiv 1 \pmod{p_i^{a_i}}$ για $1 \leq i \leq r$. Κάθε μία από αυτές τις εξισώσεις έχει δύο λύσεις, επομένως έχουμε συνολικά 2^r γραμμικά συστήματα ισοτιμιών όπου κάθε ένα από αυτά έχει μοναδική λύση. Άρα έχουμε συνολικά 2^r λύσεις mod m. Δηλαδή έχουμε 2^r τετραγωνικές ρίζες της μονάδας mod m.

Αν με κάποιο τρόπο είχαμε δύο λύσεις διαφορετικές από την $\pm 1 \pmod{m}$ π.χ. $a^2 \equiv 1 \pmod{m}$ και $a \not\equiv \pm 1 \pmod{m}$, τότε ο $\gcd(a - 1, m)$ είναι ένας μη τετριμμένος παράγοντας του m. Αυτή είναι μια βασική ιδέα που χρησιμοποιεί ο αλγόριθμος παραγοντοποίησης του Dixon στην ενότητα 4.1.4.

Άσκηση 3.104 Λύστε το πρόβλημα του Sun Zi.

$$\begin{cases} x \equiv 2 \pmod{3} \\ x \equiv 3 \pmod{5} \\ x \equiv 2 \pmod{7} \end{cases}$$

Η λύση που πρέπει να βρείτε είναι 23.

Κεφάλαιο 4

Παραγοντοποίηση & Διακριτός λογάριθμος

Erdős : A mathematician is a machine for turning coffee into theorems... and A programmer is a device for turning caffeine into code.

4.1 Παραγοντοποίηση

Υπάρχουν αρκετοί αλγόριθμοι παραγοντοποίησης. Μπορούμε να τους χωρίσουμε σε εκθετικούς και υποεκθετικούς. Για παράδειγμα οι ακόλουθοι αλγόριθμοι δοκιμαστική διαίρεση (trial division), Fermat, Legendre, $p-1$ Pollard, Pollard ρ, Coppersmith είναι εκθετικού χρόνου. Αυτοί οι αλγόριθμοι αναπτύσσονται στο [3]. Ενώ οι αλγόριθμοι των συνεχών κλασμάτων των Morrison-Brillhart, του Dixon[15], του Pomerance Quadratic Sieve (QS)[16], Number Field Sieve (NFS) είναι υποεκθετικού χρόνου (μερικοί από αυτούς έχουν αποδεδειγμένα υπο-εκθετικό χρόνο εκτέλεσης και άλλοι υπό την παραδοχή κάποιων ευρετικών υποθέσεων πετυχαίνουν υπο-εκθετική πολυπλοκότητα). Ο αλγόριθμος NFS προήλθε από μια ιδέα του John Pollard το 1988 που προτάθηκε για την επίλυση του διακριτού λογάριθμου. Δηλ. ο NFS δουλεύει και για τα δύο προβλήματα : παραγοντοποίηση και διακριτό λογάριθμο. Θα εξηγήσουμε αναλυτικά το τι σημαίνει υπο-εκθετική πολυπλοκότητα λίγο παρακάτω. Στην επόμενη ενότητα παρουσιάζουμε τον αλγόριθμο του Fermat και κατόπιν τον αλγόριθμο του Dixon. Ο αλγόριθμος που είναι πιο αποδοτικός σήμερα είναι ο GNFS : General Number Field Sieve και είναι μια παραλλαγή του NFS.

Στην επόμενη ενότητα ξεκινάμε με τον πιο απλό αλγόριθμο παραγοντοποίησης (trial division), συνεχίζουμε με την μέθοδο του Fermat που αποτελεί την βάση για πολλούς σύγχρονους αλγόριθμους παραγοντοποίησης. Κλείνουμε με τον πολύ βασικό αλγόριθμο QS, υποεκθετικού χρόνου. Τέλος, για την μελέτη αλγορίθμων

[15] J. D. Dixon (1981). Asymptotically fast factorization of integers, Math. Comp. 36 (153), p. 255–260. `https://tinyurl.com/y7hlsnyf`

[16] C. Pomerance, The Quadratic Sieve Algorithm, `https://math.dartmouth.edu/~carlp/PDF/paper52.pdf`

παραγοντοποίησης συστήνουμε το βιβλίο των Grandall & Pomerance : Primes [9].

4.1.1 Δοκιμαστική διαίρεση (trial division)

Είναι ο πιο απλός τρόπος να παραγοντοποιήσουμε έναν θετικό ακέραιο. Δόθηκε από τον Fibonacci περί το 1200. Ο αλγόριθμος ξεκινάει από τον φυσικό αριθμό 2 και ελέγχει αν το 2 διαιρεί τον φυσικό αριθμό n. Αν τον διαιρεί, τότε υπολογίζει το πηλίκο $n/2$ και θέτει ως νέα τιμή του n το πηλίκο $n/2$, και ξεκινάει από την αρχή. Αν δεν διαιρείται με το 2, τότε δοκιμάζει τον επόμενο φυσικό το 3. Αν διαιρείται με το 3 τότε κατά τα γνωστά ανανεώνει την τιμή του n με το πηλίκο και ξεκινάει από την αρχή (αυτή την φορά από το 3 και όχι από το 2). Κατόπιν αυξάνει την τιμή του 3 στο 5 (δηλαδή κατά 2) κ.ο.κ. Μια υλοποίηση είναι η παρακάτω.

Αλγόριθμος 4.1.1. : trial division
Είσοδος. n φυσικός
Έξοδος. όλοι οι πρώτοι παράγοντες του n

```
1  L = [ ] // is the list that will add all the divisors of n
2  while n ≡ 0 (mod 2) do
3  |   Append L with 2
4  |   n ← n/2
   end
5  f ← 3 // a possible divisor of n, also called trial divisor.
6  while f^2 ≤ n do
7  |   if n ≡ 0 (mod f) then
8  |   |   Append L with f
9  |   |   n ← n/f
   |   else
10 |   |   f ← f + 2
   |   end
   end
11 if n ≠ 1 then
12 |   Append L with n
   end
13 return L
```

Η πολυπλοκότητα του αλγορίθμου με είσοδο έναν φυσικό n είναι $O(\sqrt{n})$ αριθμητικές πράξεις. Αναλυτικότερα, αν ℓ το δυαδικό μήκος του n τότε χρειαζόμαστε,

$$\pi(2^{\ell/2}) \approx \frac{2^{\ell/2}}{\frac{\ell}{2}\ln 2}$$

πλήθος διαιρέσεων. Αυτός ο αλγόριθμος έχει εκθετική bit πολυπλοκότητα.

Στην πράξη αυτός ο αλγόριθμος είναι αργός όταν η είσοδος είναι πρώτος αριθμός ή γενικα αν ο μικρότερος πρώτος διαιρέτης του n είναι αρκετά μεγαλός (για τους σημερίνους υπολογιστές, ας πούμε μεγαλύτερος από 100 bits). Οπότε ο αλγόριθμος αυτός είναι ακατάλληλος για RSA modulus.

Άσκηση 4.1 Να υλοποιήσετε (σε όποια γλώσσα προγραμματισμού επιθυμείτε) τον

αλγόριθμο *δοκιμαστικής διαίρεσης* και να παραγοντοποιήσετε τους αριθμούς $2^{61}-1$ και $2^{62}-1$.

Άσκηση 4.2 Με τον αλγόριθμο της *δοκιμαστικής διαίρεσης* να παραγοντοποιήσετε τους αριθμούς $10!, 6!, 7!$. Επαληθεύστε ότι $10! = 6!7!$. Μπορείτε να βρείτε άλλες λύσεις της εξίσωσης (στους φυσικούς) $n! = a!b!$;

4.1.2 Η μέθοδος του Fermat για παραγοντοποίηση

Η μέθοδος αυτή είναι εκθετικού χρόνου. Ο αλγόριθμος αυτός είναι η βάση για πολλούς μοντέρνους αλγόριθμους παραγοντοποίησης. Η ιδέα είναι να εκφράσουμε τον ακέραιο n του οποίου ζητάμε την παραγοντοποίηση, ως διαφορά δύο μη διαδοχικών τετραγώνων. Ισχύει ότι, κάθε περιττός φυσικός γράφεται ως διαφορά δύο τετραγώνων. Αν n ένας περιττός φυσικός με $n = AB$, τότε ο n γράφεται $n = a^2 - b^2$ όπου,

$$a = \frac{1}{2}(A+B), \quad b = \frac{1}{2}(A-B).$$

Για το πρόβλημα της παραγοντοποίησης αρκεί να μελετήσουμε περιττούς ακέραιους. Διότι αν ο n είναι άρτιος, γράφεται $n = 2^k n'$ (k θετικός ακέραιος), όπου n' είναι περιττός. Οπότε χώρις βλάβη της γενικότητας υποθέτουμε ότι ο n είναι άρτιος. Τότε, $n = a^2 - b^2$, όπου a, b θετικοί ακέραιοι. Αν $a - b > 1$ έχουμε μία παραγοντοποίηση (όχι υποχρεωτικά στους πρώτους παράγοντες) του $n = (a-b)(a+b)$. Επομένως, η μέθοδος του Fermat δουλεύει ως εξής:
1. Αρχικά θέτει ως $x = \lceil \sqrt{n} \rceil$.
2. Υπολογίζουμε όλα τα $a_1 = (x+1)^2 - n$, $a_2 = (x+2)^2 - n$, ...,$a_k = (x+k)^2 - n$, όπου το $k = \frac{A+B}{2} - \lceil \sqrt{n} \rceil$ στην χειρότερη περίπτωση. Σταματάμε μόλις βρούμε κάποιο a_i ($i \in \{1, 2, ..., k\}$) γίνει τετράγωνο. Δηλ. $a_i = b^2$. Τότε, $(x+i)^2 - n = b^2$ άρα $n = (x+i-b)(x+i+b)$. Αν $x+i-b$ ή $x+i+b \neq 1$ βρήκαμε κάποιους μη τετριμμένους διαιρέτες του n.

Μια εκτίμηση για το k είναι $k < n - \lceil \sqrt{n} \rceil$ όποτε στην χειρότερη περίπτωση απαιτούνται $O(n)$ τιμές των a_i. Δηλαδή, η bit πολυπλοκότητα του αλγορίθμου είναι εκθετική. Η μέθοδος αυτή είναι αποδοτική (δηλ. βρίσκει *γρήγορα* έναν διαιρέτη) αν ο n έχει κάποιον διαιρέτη κοντά στο $\sqrt{n}$. Πράγματι, έστω ότι ο διαιρέτης A είναι κοντά στο $\sqrt{n}$. Επειδή $AB = n$, αναγκαστικά και ο B είναι κοντά στο $\sqrt{n}$. Δηλ. οι A και B είναι κοντά, επομένως η διαφορά $A - B$ είναι κοντά στο μηδεν. Σε αυτή την περίπτωση ο ακέραιος $b = \frac{A-B}{2}$ είναι επίσης κοντά στο μηδέν. Οπότε γρήγορα κάποια από τα a_i που υπολογίζουμε αρχικά θα γίνουν ίσα με το b^2 (παρατηρήστε ότι η ακολουθία a_i είναι γνήσια αύξουσα).

Αν το n είναι της μορφής pq με p, q πρώτους, τότε ο αλγόριθμος του Fermat είναι αποτελεσματικός όταν οι p και q είναι πολύ κοντά.

Άσκηση 4.3 Να αποδείξετε ότι αν $n = pq$ όπου p, q πρώτοι, η μέθοδος του Fermat χρειάζεται στη χειρότερη περίπτωση $\frac{(p-q)^2}{8\lfloor\sqrt{n}\rfloor} + 1$ επαναλήψεις για να τερματίσει. Πόσο κοντά πρέπει να είναι τα p, q ώστε ο αλγόριθμος του Fermat να χρειαστεί μόνο μία επανάληψη.

Αλγόριθμος 4.1.2. : Η μέθοδος του Fermat
Είσοδος. n θετικός περιττός ακέραιος
Έξοδος. Ένας μη τετριμμένος διαιρέτης του n

```
for ⌈√n⌉ ≤ a ≤ ⌊(n+9)/6⌋ do
    b ← √(a² − n)
    if b είναι ακέραιος then
        return gcd(a − b, n)
    end
end
```

Ένας άλλος τρόπος να περιγράψουμε την μέθοδο του Fermat είναι να θεωρήσουμε το πολυώνυμο $Q(x) = (x+a)^2 - n$ και αναζητούμε λύσεις της $Q(x) = z^2$ για $a = \lceil \sqrt{n} \rceil$ και $x > 0$. Μπορούμε να γενικεύσουμε αυτή την μέθοδο, αν θεωρήσουμε το πολυώνυμο $Q_a(x, y) = (x + ay)^2 - ny^2$, και αναζητούμε λύσεις της διοφαντικής εξίσωσης $Q_a(x, y) = z^2$ για $x > 0$ και κατόπιν υπολογίζουμε τον $\gcd(x + ay - z, n)$.

4.1.3 Οι ιδέες του Maurice Kraitchik

Ίσως η πιο καταλυτική ιδέα που βασίστηκε στη μέθοδο του Fermat και έδωσε τεράστια ώθηση στους αλγορίθμους παραγοντοποίησης είναι αυτή του βέλγου μαθηματικού Maurice Kraitchik (1882-1957). Η ιδέα να χρησιμοποιήσουμε ισοτιμίες $x^2 \equiv y^2 \bmod n$ αντί την ισότητα $n = x^2 - y^2$ είναι του Kraitchik[17] και παρουσιάστηκε το 1920. Ο αλγόριθμος του Kraitchik έθεσε τις βάσεις για τον σύγχρονο (υποεκθετικό) αλγόριθμο quadratic sieve. Γενικά η ιδέα εύρεσης δύο ακεραίων (x, y) με $x^2 \equiv y^2 \bmod n$ κυριαρχεί σε όλους τους σύγχρονους αλγορίθμους παραγοντοποίησης. Βέβαια, στην περίπτωση που βρούμε δύο ακέραιους x, y με την προηγούμενη επιθυμητή ιδιότητα, μπορεί να μην καταφέρουμε να παραγοντοποιήσουμε το n. Αυτή η ανεπιθύμητη περίπτωση θα συμβεί όταν $x \equiv \pm y \pmod n$. Επομένως στην περίπτωση που $x \not\equiv y \pmod n$, ο $\gcd(n, x - y)$ είναι ένας μη τετριμμένος διαιρέτης του n. Έστω,

$$\mathcal{A}_n = \{(x, y) \in \mathbb{Z}^2 : \gcd(xy, n) = 1,\ x^2 \equiv y^2 \pmod n\},$$

και

$$\mathcal{B}_n = \{(x, y) \in \mathcal{A}_n : x \not\equiv y \pmod n\}.$$

Αν n περιττός με τουλάχιστον δύο πρώτους διαιρέτες, τότε $|\mathcal{B}_n| \geq |\mathcal{A}_n|/2$. Αν ο n είναι RSA modulus, τότε $|\mathcal{B}_n| = |\mathcal{A}_n|/2$ (δείτε την άσκηση 4.5). Επομένως τουλάχιστον οι μισοί αριθμοί που ικανοποιούν την $x^2 \equiv y^2 \pmod n$, μας οδηγούν στην παραγοντοποίηση του n.

Η δεύτερη ιδέα του Kraitchik. Βρήκε ένα έξυπνο τρόπο να ψάχνει για αριθμούς (x, y) με $x^2 \equiv y^2 \pmod n$. Η μέθοδος του Fermat ξεκινάει από τον μικρότερο ακέραιο k που είναι μεγαλύτερος από τον $\sqrt{n}$ και προσπαθεί να βρεί τετράγωνα της μορφής $k^2 - n$. Ο Kraitchik υπολόγιζε πολλούς τέτοιους αριθμούς

[17] οι ρίζες της ιδέας αυτής πάνε πίσω στον Gauss και τον Seelhoff

a_i με $a_i = k_i^2 - n$ με τελικό στόχο να βρει ένα γινόμενο $a_{i_1} \cdots a_{i_r} \equiv U^2 \pmod{n}$. Αλλά,

$$U^2 \equiv a_{i_1} \cdots a_{i_r} \equiv (x_{i_1}^2 - n) \cdots (x_{i_r}^2 - n) \equiv (x_{i_1} \cdots x_{i_r})^2 \equiv V^2 \pmod{n}.$$

Αυτές οι δύο ιδέες αναπτύχθηκαν από τους Lehmer, Powers, Morrison, Brillhart, Dixon, Pomerance, Lenstra, Coppersmith, Pollard, Odlyzko και άλλους.

Άσκηση 4.4 Να αποδείξετε ότι αν p πρώτος αριθμός, τότε $|\mathcal{B}_p| = 0$.

Άσκηση 4.5 Να αποδείξετε ότι αν n περιττός με τουλάχιστον δύο (διαφορετικούς) πρώτους διαιρέτες, τότε $|\mathcal{B}_n| \geq |\mathcal{A}_n|/2$. Στην περίπτωση που το n είναι ένα RSA modulus, τότε $|\mathcal{B}_n| = |\mathcal{A}_n|/2$. Ειδικότερα ν.α.ο. $|\mathcal{A}_n| = 4\phi(n)$.

Άσκηση 4.6 (*) Να αποδείξετε ότι αν $n = p_1 p_2 \cdots p_r$, τότε $|\mathcal{A}_n| = 2^r \phi(n)$ και $|\mathcal{B}_n| = (2^r - 2)\phi(n)$.

Άσκηση 4.7 Με την ιδέα των ισοτιμιών του Kraitchik να παραγοντοποιήσετε τον ακέραιο $\frac{10^{17}-1}{9}$ (η παραγοντοποίηση αυτού του αριθμού έγινε από τον V. Šimerka το 1858).

Άλλες γενικεύσεις. Μια άλλη γενίκευση έχει δοθεί από τον Lehman το 1974[18], όπου αναζητούσε λύσεις της μορφής $x^2 - y^2 = 4kn$ αντί $x^2 - y^2 = n$, για κάποιον k μικρό ακέραιο (ειδικότερα $k = O(n^{1/3})$). Το πλήθος επαναλήψεων που απαιτεί αυτός ο αλγόριθμος είναι $O(n^{1/3})$.

Ο Shanks (2004) θεώρησε την εξής γενίκευση που ονομάστηκε SQUFOF : SQUare FOrms Factorization. Πρότεινε έναν νέο τρόπο για να ψάχνει για ακεραίους x, y με $x^2 \equiv y^2 \bmod n$ που βασίζεται σε συνεχή κλάσματα (αλλά μπορεί να παρουσιαστεί και με χρήση τετραγωνικών μορφών). Η ευρετική πολυπλοκότητα του είναι $O(n^{1/4})$.

Επίσης υπάρχει ο ντετερμινιστικός αλγόριθμος του Coppersmith που βασίζεται σε πλέγματα με πολυπλοκότητα $O(n^{1/4})$.

4.1.4 Αλγόριθμος του Dixon/Quadratic Sieve

Έστω $c \in [0, 1]$ και d θετικός πραγματικός αριθμός. Θέτουμε

$$L_n[c, d] = \exp\left(d(\ln n)^c (\ln(\ln n))^{1-c}\right).$$

Αλγόριθμοι με πολυπλοκότητα $O(L_n[c, d])$ και $0 < c < 1$ καλούνται υποεκθετικοί. Αν $c = 0$ έχουμε πολυωνυμική bit πολυπλοκότητα, $L_n[0, d] = (\ln n)^d$. Ενώ αν $c = 1$ δηλ. $L_n[1, d] = n^d$, έχουμε εκθετική πολυπλοκότητα. Ο πρώτος αλγόριθμος παραγοντοποίησης υποεκθετικής πολυπλοκότητας βασίζεται σε συνεχή κλάσματα και παρουσιάστηκε στις αρχές του 1970 (Morrison-Brillhart). Οι αρχικές ιδέες αυτού του αλγορίθμου πρώτη φορά δόθηκαν από τους Lehmer και Powers το 1931.

[18] Factoring Large integers, Math. Comp. 28 (126), 1974

Ο αλγόριθμος αυτός βρήκε την παραγοντοποίηση του αριθμού $F_7 = 2^{2^7} + 1 = 2^{128} + 1$. Η bit πολυπλοκότητα του είναι

$$L_n[1/2, \sqrt{2}].$$

Πριν από αυτόν τον αλγόριθμο, δηλ. πριν την δεκαετία του 1970 δεν μπορούσαν να παραγοντοποιήσουν αριθμούς μεγαλύτερους από 20 ψηφία. Ο αλγόριθμος των Morrisson-Brillhart μπορεί να παραγοντοποιήσει (πρακτικά) ακέραιους αριθμούς μέχρι 50 δεκαδικά ψηφία.

Ο αλγόριθμος του Dixon παρουσιάστηκε το 1981 από τον John D. Dixon και αποτελεί μια βελτίωση της μεθόδου του Fermat. Είναι ο πρώτος που αναλύθηκε και έχει αποδεδειγμένα υπόεκθετική πολυπλοκότητα,

$$L_n[1/2, 2\sqrt{2}].$$

Η βελτίωση του αλγορίθμου αυτού είναι η μέθοδος του τετραγωνικού κόσκινου (QS : Quadratic Sieve) που έχει bit πολυπλοκότητα

$$L_n[1/2, 1].$$

Παρουσιάστηκε από τον Pomerance το 1981 και αρκετά αργότερα από τους Lenstra-Pommerance δόθηκε (αποδεδειγμένα) η πολυπλοκότητα του[19]. Ο NFS : Number Field Sieve παρουσιάστηκε το 1993 και είναι ο καλύτερος αλγόριθμος που έχουμε σήμερα. Η αρχική ιδέα αυτού του αλγορίθμου προέρχεται από τον Pollard (Special NFS) και βελτιώθηκε από τους Lenstra, Pomerance και Coppersmith. Αν και για μεγάλους αριθμούς είναι καλύτερος από τον QS, για αριθμούς μέχρι 100 δεκ. ψηφία είναι πιο αργός από τον QS.

Υπάρχει και αποδοτικότερος αλγόριθμος, ο GNFS : General Number Field Sieve (είναι μια παραλλαγή του NFS) με (ευρετική) πολυπλοκότητα

$$L_n[1/3, 1.923].$$

Αυτός ο αλγόριθμος είναι ο καλύτερος μέχρι σήμερα [9, Ενότητα 6.2.3. σελ.287]). Γενικά ένα RSA-modulus 512 bit δεν αποτελεί ασφαλή επιλογή, διότι κοινή πρακτική στην κρυπτογραφία αποτελεί το φράγμα 2^{80} (ή και 2^{90}). Για να δούμε το τελευταίο αρκεί να γράψουμε την

$$L_n[1/3, 1.923] = \exp\left(1.923(\ln n)^{1/3}(\ln(\ln n)\right)^{2/3} = e^{2.774(\ln n \cdot (\ln(\ln n))^2)^{1/3}}.$$

Οπότε αν θέσουμε $n \approx 2^{512}$ έχουμε $e^{2.774(512\cdot(\ln(512)^2)^{1/3}} \approx 2^{64}$.

Πρώτη φορά έγινε παραγοντοποίηση ενός αριθμού 512-bit το 1999, με χρήση εκατοντάδων υπολογιστών και σε χρόνο περίπου 7 μηνών. Αργότερα, το 2015, κατάφεραν να παραγοντοποιήσουν 512-bit RSA modulus σε 4-ώρες, με χρήση υπηρεσιών της Amazon και υλοποίηση του GNFS. Οπότε, είναι απαγορευτική η

[19] H. W. Lenstra, Jr. and C. Pomerance. A rigorous time bound for factoring integers. Journal of the AMS, 4, p. 483–516, 1992.

χρήση τέτοιων αριθμών στη κρυπτογραφία. Σήμερα οι πρώτοι αριθμοί που χρησιμοποιούνται στο RSA είναι μήκους 1024 ή 2048 bits, που δίνουν RSA modulus με 2048 ή 4096 bits, αντίστοιχα.

Για να καταλάβουμε πως δουλεύει ο αλγόριθμος του Dixon ας δούμε το επόμενο παράδειγμα. Έστω $n = 1649$. Ξεκινάμε από το $N = \lceil \sqrt{n} \rceil = 41$ και δίνουμε τιμές στην μεταβλητή $x = N, N+1, ...$ και υπολογίζουμε τα $x^2 \text{mod } n$. Έτσι έχουμε,

$$41^2 \equiv 32 \text{mod } n, 42^2 \equiv 115 \text{mod } n, 43^2 \equiv 200 \text{mod } n, 44^2 \equiv 287 \text{mod } n,$$

$$45^2 \equiv 376 \text{mod } n, 46^2 \equiv 467 \text{mod } n, 47^2 \equiv 560 \text{mod } n, 48^2 \equiv 655 \text{mod } n,$$

$$49^2 \equiv 752 \text{mod } n, 50^2 \equiv 851 \text{mod } n.$$

Παρατηρούμε ότι $32 \cdot 376 \cdot 752 \equiv 1 \text{mod } n$. Επομένως, αν $x = 41 \cdot 45 \cdot 49$ προκύπτει ότι $x^2 \equiv 1 \text{mod } n$ και $\gcd(x-1, n) = 97$. Βρήκαμε έναν διαιρέτη του n (που τυχαίνει να είναι και πρώτος). Παρατηρούμε ότι οι αριθμοί $32, 376, 752$ έχουν όλοι πρώτους παράγοντες < 48. Αυτοί οι αριθμοί λέγονται και $48-$ομαλοί. Είναι προτιμότερο να αναζητούμε αριθμούς που είναι $B-$ομαλοί για B σχετικά μικρό, παρά να ψάχνουμε στη τύχη.

Επίσης παρατηρούμε ότι $752 \cdot 655 \cdot 467 \cdot 115 \equiv 4 \text{ mod } n$ και αν θέσουμε $y = 49 \cdot 48 \cdot 46 \cdot 42$ έχουμε $y^2 \equiv 4 \text{ mod } n$ και $\gcd(y-2, n) = 97$. Γενικά, αναζητούμε συνδυασμούς ώστε το γινόμενο να μας δίνει τετράγωνο mod n (αυτή είναι η *δεύτερη* ιδέα του Kraitchik). Ο Dixon πρότεινε να χρησιμοποιήσουμε μια Factor basis. Δηλαδή, αντί να ψάχνουμε για τετράγωνα να ψάχνουμε για αριθμούς που έχουν πρώτους παράγοντες $\leq B$, για κάποιο κατάλληλο αριθμό $B > 0$. Κατόπιν, μια έξυπνη ιδέα από την γραμμική άλγεβρα μας επιτρέπει να εντοπίσουμε τετράγωνα mod n και να συνεχίσουμε όπως στο παράδειγμα (είναι μια ιδέα των Brillhart-Morrison). Η βελτίωση του αλγορίθμου του Dixon ως προς την επιλογή του B και τον τρόπο που κατασκευάζουμε $B-$ομαλούς ακέραιους, ονομάστηκε Quadratic Sieve algorithm.

Άσκηση 4.8 Παραγοντοποιήστε τον αριθμό 8051 το πολύ σε 5 λεπτά, χωρίς υπολογιστή.

B-smooth numbers

Ορισμός 4.1.1. *Καλούμε έναν ακέραιο $B-$ομαλό (B-smooth) αν δεν έχει πρώτους παράγοντες μεγαλύτερους του B.*

Για παράδειγμα ο 20 είναι 5-ομαλός ενώ ο -2^k, για k φυσικό, είναι $2-$ομαλός. Αν θέσουμε

$$S(x, B) = \{1 \leq n \leq x : n \text{ είναι } B\text{-ομαλός}\}$$

και

$$\psi(x, B) = \#S(x, B),$$

τότε ο Dickman το 1930 απόδειξε ότι

$$\psi(x, x^{1/u}) \sim \rho(u)x, (u \to \infty) \tag{4.1.1}$$

Από τον Hildebrand το 1986 αποδείχτηκε ότι

$$\psi(x, x^{1/u}) = \rho(u)x\left(1 + O\left(\frac{\ln(u+1)}{\ln y}\right)\right) \tag{4.1.2}$$

για

$$1 \leq u \leq \exp\left(\left(\frac{\ln x}{u}\right)^{3/5-\varepsilon}\right).$$

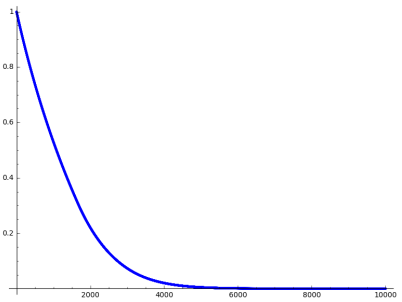
Δηλ. μπορούμε να χρησιμοποιήσουμε την εκτίμηση (4.1.1) όχι μόνο για μεγάλα u αλλά και σε μικρότερα διαστήματα. Επίσης από τον Hildebrand το 1986 αποδείχτηκε ότι η υπόθεση του Riemann είναι ισοδύναμη με την ύπαρξη της εκτίμησης (4.1.2) για το διάστημα

$$1 \leq u \leq x^{1/(2u)-\varepsilon}.$$

Μια ασυμπτωτική εκτίμηση για την συνάρτηση Dickman (αλλά όχι ιδιαίτερα καλή) είναι

$$\rho(u) \sim u^{-u+o(u)}. \tag{4.1.3}$$

Αυτή η εκτίμηση ισχύει για μεγάλα u.



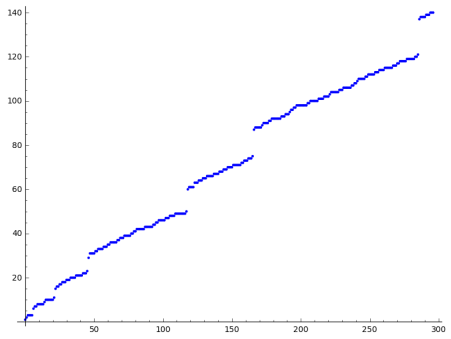
Σχήμα 4: Η συνάρτηση του Dickman $\rho(u)$. Παρατηρήστε ότι $\rho(u) \leq 1$ και τείνει στο μηδέν πολύ γρήγορα.

Π.χ. $S(10,5) = \{1,2,3,4,5,6,8,9,10\}$, $S(4,10) = \{1,2,3\}$ και $S(12,4) = \{1,2,3,4,6,8,9,12\}$ και

y	10	20	30	40	50	60	70	80	90
$\psi(y,\sqrt{y})$	7	10	18	21	31	34	37	40	43

Θέτουμε $L(n) = L[1/2,1] = e^{\sqrt{\ln n \cdot \ln(\ln n)}}$, από την (4.1.1) και (4.1.3) έχουμε,

$$\Pr\left(x : x \text{ είναι } L(n)^{1/2}\text{-ομαλός και } x \xleftarrow{\$} \{1,2,....,L(n)\}\right) \sim \frac{\rho(2)L(n)}{L(n)} \approx 0.3. \tag{4.1.4}$$



Σχήμα 5: Στον κάθετο άξονα είναι οι τιμές της συνάρτησης $\psi(y,\sqrt{y})$ και στον οριζόντιο άξονα το y

Άσκηση 4.9 Έστω x θετικός ακέραιος και $B=\sqrt[3]{x}$. Να υπολογίσετε πόσοι $B-$ομαλοί ακέραιοι υπάρχουν στο διάστημα $[1,x]$ για $x=100,200,....,1000$. Υπολογίστε το ποσοστό αυτών και βρείτε τον μέσο όρο.

Άσκηση 4.10 Να επεκτείνετε τον προηγούμενο πίνακα για $y=1000,1100,...,2000$.

Άσκηση 4.11 Για μεγάλα x ισχύει $\psi(x,4)\sim c\big(\ln x\big)^2$. Κατασκευάστε ένα γράφημα με τις δύο συναρτήσεις $f(x)=\psi(x,4)$ και $g(x)=c(\ln x)^2$ για κατάλληλη σταθερά c που πρέπει να υπολογίσετε πειραματικά. Η γενίκευση αυτού του αποτελέσματος είναι

$$\psi(x,B)\sim c(B)\big(\ln x\big)^{\pi(B)}.$$

Θυμίζουμε ότι $f(n)\sim g(n)$ αν και μόνο αν $\lim_{n\to\infty}\frac{f(n)}{g(n)}=1$.

Υπάρχουν και μη ασυμπτωτικά κάτω φράγματα,

Πρόταση 4.1.1. *(Pomerance, Konyagin, 1997). Έστω $u\in\mathbb{R}$. Για όλα τα $x\geq 4$ και $2\leq x^{1/u}\leq x$ έχουμε*

$$\psi(x,x^{1/u})\geq\frac{x}{(\ln x)^u}.$$

Ο λόγος που χρησιμοποιούμε $B-$ομαλούς ακέραιους είναι οι εξής.
(i). Έχουν απλή πολλαπλασιαστική δομή (το γινόμενο τους είναι επίσης $B-$ομαλός).
(ii). Δεν είναι σπάνιοι (δείτε την (4.1.4)).
(iii). Είναι εύκολο να τους αναγνωρίσουμε. Για παράδειγμα με δοκιμαστική διαίρεση μπορούμε γρήγορα να αναγνωρίσουμε αν ένας ακέραιος είναι $B-$ομαλός.

Ακολουθούμε την επόμενη διαδικασία, που είναι μια βελτίωση του Dixon και ονομάζεται Quadratic Sieve.

Ο Αλγόριθμος Quadratic Sieve

Είσοδος : $n > 2$ περιττός σύνθετος ακέραιος που δεν είναι τέλεια δύναμη.
Έξοδος : Ένας μη τετριμμένος διαιρέτης του n.

[Initialization Step]
1. $B \leftarrow \lfloor L(n)^{1/2} \rfloor = \lfloor L[1/2, 1/2] \rfloor$, $S \leftarrow \{-1, p_1 = 2, p_2, p_3, ..., p_t\}$ όπου $p_2 < \cdots < p_t$ είναι $B-$ομαλοί και το n είναι τετραγωνικό υπόλοιπο $\bmod p_i$, για $i > 1$.

[Sieving Step]
2. Για κάθε αριθμό x_i στο σύνολο $\{\lceil \sqrt{n} \rceil, \lceil \sqrt{n} \rceil \pm 1, ...\}$ υπολογίζουμε τα $c_i = x_i^2 - n$ ώστε όλοι οι διαιρέτες του να είναι στην βάση παραγόντων S. Έστω S_1 αυτό το σύνολο. Σταματάμε όταν $|S_1| = t + 2$.
3. Για κάθε $y \in S_1$ υπολογίζουμε το διάνυσμα $\mathbf{e}(y)$ ως εξής. Έστω η συνάρτηση

$$\mathbf{e} : S_1 \to \{0, 1\}^{t+1},$$

$$\mathbf{e}((-1)^{b_0} 2^{b_1} p_2^{b_2} \cdots p_K^{b_t}) = (b_0, b_1 \bmod 2, b_2 \bmod 2, ..., b_t \bmod 2),$$

με $b_i \geq 0$ για $i \geq 1$ και $b_0 \in \{0, 1\}$. Έστω $S_2 = \{\mathbf{e}(y) : y \in S_1\}$. Ισχύει $|S_2| = t + 2$.

[Linear Algebra step & Factorization step]
4. Σχηματίζουμε τον δυαδικό πίνακα διαστάσεων $(t+1) \times (t+2)$ πίνακα με στήλες τα $\mathbf{e}(y)$ του συνόλου S_2. Έστω M ο πίνακας αυτός.
5. Βρίσκουμε μια μη-μηδενική λύση του ομογενούς συστήματος $M\mathbf{x} = \mathbf{0} \pmod 2$. Η λύση αυτού του συστήματος θα μας δώσει ένα γραμμικό εξαρτημένο υποσύνολο του S_2. Π.χ. αν $\mathbf{e}(y_1) + \mathbf{e}(y_3) = \mathbf{0} \pmod 2$, τότε η 1η και 3η στήλη είναι γραμμικά εξαρτημένες.
6. Αν $\mathbf{e}(c_{i_1}) + \cdots + \mathbf{e}(c_{i_r}) = \mathbf{0} \pmod 2$ θέτουμε $x^2 = c_{i_1} \cdots c_{i_r} \bmod n$ και

$$z = x_{i_1} \cdots x_{i_r} \bmod n.$$

Τότε, $x^2 \equiv z^2 \bmod n$.
7. $d \leftarrow \gcd(x - z, n)$.
8. Αν $d > 1$, επιστρέφουμε τον αριθμό $d = \gcd(x - z, n)$ διαφορετικά δηλ. αν $d = 1$ επέστρεψε *αποτυχία*.

Μερικές παρατηρήσεις.

Βήμα 1.
• Απαιτούμε ο αριθμός n να μην είναι δύναμη πρώτου, ώστε να διαιρείται από τουλάχιστον δύο πρώτους.
• Από την σχέση (4.1.4) αν διαλέξουμε το B όπως στο βήμα 1, με πιθανότητα περίπου 0.3 θα έχουμε ένα $L(n)^{1/2}-$ ομαλό ακέραιο αν τον διαλέξουμε από το σύνολο $\{1, 2, ..., L(n)\}$.
• Στο βήμα **1**, στο σύνολο S δεν βάζουμε πρώτους p που το n δεν είναι τετραγωνικό υπόλοιπο $\bmod p$. Αυτό διότι, αν $p | x_i^2 - n$, τότε $x_i^2 \equiv n \bmod p$.

Δηλαδή, αν το p είναι διαιρέτης του $x_i^2 - n$, τότε το n είναι τετραγωνικό υπόλοιπο mod p. Επομένως, τους πρώτους για τους οποίους το n δεν είναι τετραγωνικά υπόλοιπό mod p δεν τους βάζουμε στην factor base.

Επίσης, ο λόγος που σταματάμε στο $t+2$ στο βήμα **3** είναι το επόμενο λήμμα.

Λήμμα 4.1.1. *Αν $L = \{m_1, m_2, ..., m_t\}$ είναι θετικοί $B-$ομαλοί ακέραιοι, και αν $t > \pi(B)$ τότε υπάρχει υποσύνολο $B \subset L$ τέτοιο ώστε $\prod_{x\in B} x$ να είναι τετράγωνο.*

Απόδειξη. Έστω m ένας $B-$ομαλός θετικός ακέραιος. Τότε,

$$m = p_1^{e_1} \cdots p_K^{e_K}, \quad K = \pi(B),$$

όπου p_i είναι ο $i-$οστός πρώτος αριθμός. Έστω $\mathbf{e}(m) = (e_1, ..., e_K) \in \mathbb{F}_2^K$. Τώρα θεωρούμε ένα σύνολο από $B-$ομαλούς ακέραιους, έστω $m_1, m_2, ..., m_r$ για κάποιο r. Το γινόμενο $m_1 m_2 \cdots m_r$ είναι τετράγωνο αν και μόνο αν

$$\mathbf{e}(m_1) + \mathbf{e}(e_2) + \cdots + \mathbf{e}(e_r) \equiv 0 \pmod 2$$

δηλ. αν και μόνο αν τα διανύσματα $\mathbf{e}(m_1), \mathbf{e}(m_2), \ldots, \mathbf{e}(m_r)$ είναι γρ. εξαρτημένα στον χώρο $\mathbb{F}_2^K$. Αλλά ο διανυσματικός χώρος $\mathbb{F}_2^K$ έχει διάσταση $K = \pi(B)$. Άλλα από την υπόθεση έχουμε $r = t$ και $t > \pi(B)$, επομένως τα διανύσματα $\mathbf{e}(m_1), \mathbf{e}(m_2), ..., \mathbf{e}(m_t)$ είναι γρ. εξαρτημένα. Άρα, υπάρχει υπακόλουθία της $m_1, ..., m_t$ που το γινόμενο είναι τετράγωνο. □

• Μέσα στην $B-$ βάση παραγόντων (B-Factor Base) S υπάρχει και ο αριθμός -1. Ο λόγος είναι διότι στο βήμα **2** μπορεί κάποιο c_i να επιλεχθεί αρνητικό ώστε να είναι $B-$ομαλός.

• Για να υπολογίσουμε την συνάρτηση $\mathbf{e}(y)$ χρειαζόμαστε την παραγοντοποίηση του y. Αλλά το $y \in S_1$ είναι $B-$ομαλός, οπότε για σχετικά μικρό B μπορούμε να βρούμε την παραγοντοποίηση του y. Επομένως ο πίνακας του βήματος **4** μπορεί να υπολογιστεί (εδώ θα χρειαστεί και αρκετή μνήμη).

• Στο βήμα **6**, εφόσον

$$\mathbf{e}(c_{i_1}) + \cdots + \mathbf{e}(c_{i_r}) = \mathbf{0} \pmod 2$$

προκύπτει ότι το γινόμενο $c_{i_1} c_{i_2} \cdots c_{i_r}$ είναι γινόμενο αριθμών του S_1 όπου κάθε πρώτος εμφανίζεται σε *άρτιο* εκθέτη. Οπότε

$$\prod_{j=1}^{r} c_{i_j} \equiv x^2 \pmod n.$$

Επίσης τα $\{c_{i_j}\}_j$ είναι στοιχεία του S_1 άρα από κατασκευής είναι τετράγωνα mod n. Οπότε υπάρχουν x_{i_j} τέτοια, ώστε $x_{i_j}^2 \equiv c_{i_j} \pmod n$. Επομένως αν θέσουμε $z = x_{i_1} x_{i_2} \cdots x_{i_r} \pmod n$, τότε $z^2 \equiv x^2 \pmod n$.

• Υπάρχει ψευδοκώδικας στο βιβλίο των A. Menezes, P. van Orschot και S. Vanstone [11, Αλγόριθμος 3.21, σελ. 96]

4.2 Διακριτός Λογάριθμος

Έστω p πρώτος αριθμός. Έχουμε δει ότι το σύνολο των αντιστρέψιμων στοιχείων mod p, $\mathbb{Z}_p^*$ έχει $\phi(p) = p-1$ στοιχεία. Επίσης, αποδεικνύεται ότι υπάρχει ένα στοιχείο του g, που παράγει όλα τα υπόλοιπα. Δηλαδή, για κάθε στοιχείο $b \in \mathbb{Z}_p^*$ υπάρχει ένας φυσικός αριθμός n τέτοιος ώστε $g^n = b$. Κάθε φυσικός αριθμός $m \equiv n \pmod{(p-1)}$ έχει την ιδιότητα $a^m = b$. Επομένως, μπορούμε να διαλέξουμε τον $n : 0 \leq n \leq p-2$. Το ερώτημα που θα συζητήσουμε αφορά στην επίλυση της εξίσωσης $a^x = b$ στην ομάδα $\mathbb{Z}_p^*$. Το x=dlog$_a(b)$, ονομάζεται διακριτός λογάριθμος του b με βάση a. Το πρόβλημα αυτό ονομάζεται *πρόβλημα διακριτού λογαρίθμου* mod p (DLP : **D**iscrete **L**ogarithm **P**roblem). Ας πάρουμε για παράδειγμα την εξίσωση $2^x = 7$ στην $\mathbb{Z}_{13}^*$. Σχηματίζουμε τον παρακάτω πίνακα.

x	1	2	3	4	5	6	7	8	9	10	11
$2^x \bmod p$	2	4	8	3	6	12	11	9	5	10	7

Άρα $\text{dlog}_2 7 = 11$. Ο διακριτός λογάριθμος προσδιορίζεται modulo την τάξη της ομάδας $\langle 2 \rangle$.

Μερικές φορές καλούμε το διακριτό λογάριθμο και δείκτη mod p (index και γράφουμε $ind_g(a) = x$).

Ο διακριτός λογάριθμος εμφανίστηκε στην κρυπτογραφία από τους Diffie-Hellman. Πρότειναν τον διακριτό λογάριθμο ως υποψήφια συνάρτηση μίας φοράς (one way function). Στο ομώνυμο σύστημα ανταλλαγής κλειδιών απαιτείται η λύση στο εξής πρόβλημα :

$$\text{Δοθέντος } x, y \text{ και } g \in G \text{ βρείτε το } g^{xy} \text{ όπου } a = g^x,\ b = g^y.$$

Αν μπορούμε να λύσουμε αποδοτικά τον DLP τότε και το προηγούμενο πρόβλημα λύνεται. Το αντίστροφο δεν είναι γνωστό αν ισχύει.

Το πρόβλημα αυτό μπορεί να δοθεί για οποιαδήποτε πεπερασμένη κυκλική ομάδα G και όχι υποχρεωτικά για την $\mathbb{Z}_p^*$. Έστω G μία ομάδα (την γράφουμε πολλαπλασιαστικά) και $g \in G$. Ας είναι $\langle g \rangle$ η κυκλική υποομάδα της G που παράγεται από το στοιχείο g. Τότε το DLP είναι το πρόβλημα της εύρεσης του x δοθέντος του $a = g^x \in \langle g \rangle$. Το DLP δεν είναι γενικά δύσκολο πρόβλημα. Λέμε ότι είναι δύσκολο για την ομάδα G, αν δεν υπάρχει αποδοτικός αλγόριθμος που να λύνει το DLP στην ομάδα G. Αναλυτικότερα δίνουμε το εξής ορισμό.

Ορισμός 4.2.1. *Το DLP είναι δύσκολο στην ομάδα G (κυκλική ομάδα τάξης q) αν για κάθε αποδοτικό αλγόριθμο A η πιθανότητα*

$$Pr_{g \xleftarrow{R} G,\ x \xleftarrow{R} \mathbb{Z}_q}[A(G, g, g^x) = x] \text{ είναι αμελητέα.} \tag{4.2.1}$$

Για παράδειγμα αν η G είναι η ομάδα μίας ελλειπτικής καμπύλης επί ενός πεπερασμένου σώματος (που είναι κυκλική), τότε το DLP πιστεύουμε ότι είναι δύσκολο. Δεν έχει αποδειχτεί ότι είναι δύσκολο, αλλά μέχρι σήμερα δεν εχει βρεθεί κάποιος αποδοτικός αλγόριθμος που να το λύνει. Παρόμοια υπόθεση γίνεται και για το κρυπτοσύστημα RSA (δείτε την ανάλογη σχέση (5.1.1)). Επίσης, επί της ομάδας

$\mathbb{Z}_p^*$ για πρώτο $p \geq 2^{1023}$. Ειδικότερα το DLP επί της ομάδας μιας ελλειπτικής $\bmod p$ είναι δυσκολότερο από ότι στην $\mathbb{Z}_p^*$, και για αυτό το λόγο χρησιμοποιούμε μικρότερες παραμέτρους.

Παρατήρηση 4.2.1. Για να οριστεί το πρόβλημα του διακριτού λογαρίθμου δεν είναι απαραίτητο η βάση g να είναι γεννήτορας της ομάδας G. Στην κρυπτογραφία, χρησιμοποιούμε το πρόβλημα του διακριτού λογαρίθμου ως προς μία βάση που δεν είναι γεννήτορας. Για παράδειγμα στην ψηφιακή υπογραφή DSA ξεκινάμε από την ομάδα $G = \mathbb{Z}_p^*$ με p τουλάχιστον 2048 bits και το $p-1$ έχει ένα πρώτο διαιρέτη q με 160 bits. Κατόπιν διαλέγουμε ένα στοιχείο h της ομάδας με τάξη q. Π.χ. $h = g^{(p-1)/r}$. Για να σπάσουμε την ψηφιακή υπογραφή πρέπει να λύσουμε το πρόβλημα του διακριτού λογαρίθμου στην κυκλική ομάδα $\langle h \rangle$ ή στην αρχική ομάδα $\langle g \rangle = G$ (όπου τελικά θα μας δώσει και την λύση του διακριτού λογαρίθμου στην $\langle h \rangle$).

Παρατήρηση 4.2.2. Ο Peter Shor απόδειξε ότι υπάρχει πολυωνυμικός αλγόριθμος που λύνει το πρόβλημα του διακριτού λογαρίθμου σε κβαντικό υπολογιστή. Επομένως, συστήματα που βασίζονται σε αυτό το πρόβλημα μπορούν ξαφνικά να γίνουν μη ασφαλή αν κατασκευαστεί κβαντικός υπολογιστής (με αρκετά μεγάλη μνήμη).

4.2.1 Ο αλγόριθμος του Shanks

Ο αλγόριθμος του Shanks έχει πολυπλοκότητα $O(n^{1/2} \log_2 n)$ και απαίτηση για μνήμη $O(n^{1/2})$ όπου $n = |G|$ η τάξη της ομάδας. Παρουσιάζουμε τον ψευδοκώδικα για την ομάδα $G = \mathbb{Z}_p^*$ (p πρώτος). Ο αλγόριθμος αυτός είναι ντετερμινιστικός (εκθετικού χρόνου) και δουλεύει για όλες τις πεπερασμένες κυκλικές ομάδες.

Αλγόριθμος 4.2.1. : Ο αλγόριθμος του Shanks
Είσοδος. G η κυκλική ομάδα $\mathbb{Z}_p^*$, g : γεννήτορας της G, $y = g^x$.
Έξοδος. $\mathrm{dlog}_g(y)$

```
1  L_1 = [ ]; L_2 = [ ]; A = ⌊√p⌋;
2  for i = 0 to A + 1 do
3  |   L_1 ← g^{A·i} (mod p) // giant steps
4  |   L_2 ← y · g^{-i} (mod p) // baby steps
5  |   if y · g^{-i} = g^{Ai} then
6  |   |   return k = A · (i + 1)
   |   end
7  |   if L_1 ∩ L_2 ≠ ∅ then
8  |   |   B = L_1 ∩ L_2
9  |   |   quotient ← L_1.index(A_1)
   |   |   // returns the position of the number B in the list L_1
10 |   |   reminder ← L_2.index(B)
11 |   |   k ← A · quotient + reminder
   |   |   return k
   |   end
   end
```

Στην περίπτωση μας $|G| = p-1$. Ας είναι $y \in G$ και $y = g^x$ με $x \in \{0, 1, ..., p-2\}$. Θέτουμε $A = \lfloor \sqrt{p} \rfloor$. Τότε $x = Ai_0 + j_0$ με $0 \leq i_0, j_0 < A$. Επομένως, $y = g^x$

ισοδύναμα γράφεται $yg^{-j_0} = g^{Ai_0}$. Προσπαθούμε να εντοπίσουμε σημεία της λίστας

$$L_1 = \{(g^{Ai}, i) : i = 1, 2, \ldots, A = \lfloor \sqrt{p} \rfloor\}$$

και

$$L_2 = \{(yg^{-j}, j) : j = 1, 2, \ldots, A\}$$

που συμφωνούν στην πρώτη συντεταγμένη. Έστω $g^{Ai_0} = yg^{-j_0}$. Τότε, ο διακριτός λογάριθμος είναι $k = Ai_0 + j_0$. Πράγματι, $g^k = g^{Ai_0+j_0} = g^{Ai_0}g^{j_0} = yg^{-j_0}g^{j_0} = y$. Το ερώτημα που προκύπτει είναι αν πάντα υπάρχει ένα τέτοιο σημείο τομής. Ο διακριτός λογάριθμος $x \in [0, p-2]$ είναι τέτοιος ώστε, $g^x = y$ και αν διαιρεθεί με το A μπορεί να γραφεί ως $x = Ax_1 + x_2$, όπου το υπόλοιπο $0 \leq x_2 < A$ και το πηλίκο x_1 είναι θετικό και $< A$. Αν $x_1 \geq A$ τότε $x \geq p$, άτοπο.

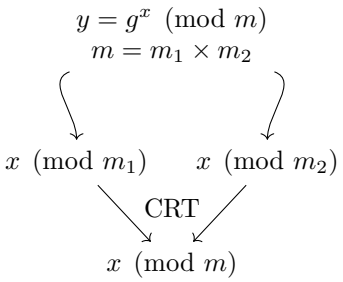
Η απαιτούμενη μνήμη, όπως φαίνεται στις γραμμές 3,4 είναι $2\sqrt{p}$ στοιχεία της ομάδας G. Η χρονική πολυπλοκότητα δίνεται από τον υπολογισμό της χρονικής πολυπλοκότητας εύρεσης της τομής των συνόλων L_1, L_2. Αυτό μπορεί να γίνει σε χρόνο $O(\sqrt{p}\log_2 p)$. Για να το δούμε αυτό ταξινομούμε μία από τις δύο λίστες π.χ. την L_1, αυτό με χρήση του αλγορίθμου mergesort (δείτε παρατήρηση 6.2.2) μπορεί να γίνει σε $O(\sqrt{n}\log_2 n)$. Κατόπιν, εφαρμόζουμε δυαδική αναζήτηση (χρόνο $O(\log_2 p)$) αν κάποιο στοιχείο $x \in L_2$ ανήκει στο L_1. Αυτό γίνεται για όλα τα στοιχεία του συνόλου L_2. Οπότε συνολικά έχουμε $O(\sqrt{p}\log_2 p)$ πράξεις μέσα στην ομάδα.

Το θετικό με τον αλγόριθμο του Shanks είναι η γενικότητα του (Generic algorithm). Δηλαδή, εφαρμόζεται σε οποιαδήποτε κυκλική ομάδα G. Στην περίπτωση αυτή στις γραμμές 3,4, οι πράξεις εννοούνται μέσα στην ομάδα (δηλ. όχι απαραίτητα mod p). Επίσης, ο αλγόριθμος αυτός θα δούλευε ακόμη και αν δεν γνωρίζαμε ακριβώς την τάξη της ομάδας G, αλλά μόνο ένα άνω φράγμα αυτής. Το μειονέκτημα του είναι η μεγάλη απαίτηση του σε μνήμη. Για παράδειγμα αν $|G| \approx 2^{160}$ (περίπτωση της ψηφιακής υπογραφής DSA) ο αλγόριθμος του Shanks, θα βρει τον διακριτό λογάριθμο με απαίτηση μνήμης $O(2^{80})$. Ο αλγόριθμος του Pollard θεραπεύει αυτήν ακριβώς την αδυναμία.

Παρατήρηση 4.2.3. Έστω ότι έχουμε το πρόβλημα εύρεσης του x με $g^x = w$ $(0 < x < m)$ σε μια κυκλική ομάδα G τάξης m. Αν γνωρίζουμε έναν διαιρέτη της τάξης της G τότε το πρόβλημα του διακριτού λογαρίθμου ανάγεται σε επιμέρους ευκολότερα προβλήματα. Ας είναι $m = m_1m_2$ με $\gcd(m_1, m_2) = 1$ και $a = g^{m_2}$, $b = g^{m_1}$. Θέτουμε $G_1 = \langle a \rangle$, $G_2 = \langle b \rangle$. Τότε, $|G_1| = m_1, |G_2| = m_2$. Επίσης η G είναι το ευθύ άθροισμα των G_1, G_2. Έστω ότι μπορούμε να υπολογίσουμε τους διακριτούς λογάριθμους $\log_a(w^{m_2})$, $\log_b(w^{m_1})$. Καταρχάς $w^{m_2} \in G_1$, $w^{m_1} \in G_2$, όποτε έχουν νόημα οι διακριτοί λογάριθμοι. Έστω $x_1 = \log_a(w^{m_2})$. Τότε έχουμε (όλες οι πράξεις είναι στην ομάδα G_1):

$$a^{x_1} = w^{m_2} \text{ άρα } a^{x_1} = (g^x)^{m_2} = (g^{m_2})^x = a^x, \text{ επομένως } x_1 \equiv x (\mathrm{mod}\ m_1).$$

Παρόμοια (για την ομάδα G_2), $x_2 \equiv x (\mathrm{mod}\ m_2)$. Εφαρμόζουμε CRT(δείτε υποενότητα 3.8) και βρίσκουμε το x. Οι πράξεις που απαιτούνται συνολικά είναι $O(\sqrt{m_1}\log_2 m_1 + \sqrt{m_2}\log_2 m_2)$. Επομένως, χωρίς βλάβη της γενικότητας μπορούμε να υποθέσουμε ότι η τάξη της ομάδας είναι μία δύναμη πρώτου (ή πρώτος αριθμός).

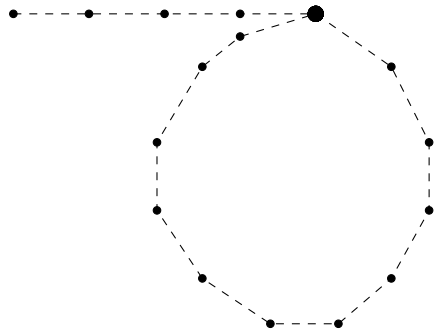


4.2.2 Μέθοδος Pollard-ρ

Το 1978 ο John Pollard, ανακάλυψε δύο μεθόδους τύπου Monte Carlo[20], που λύνουν (ευρετικά) το πρόβλημα του διακριτού λογαρίθμου. Η πρώτη μέθοδος την οποία και θα παρουσιάσουμε σε αυτήν την ενότητα, ονομάζεται Pollard-ρ μέθοδος ενώ η δεύτερη, λ-μέθοδος του Pollard (ή μέθοδος Καγκουρό). Οι μέθοδοι του Pollard έχουν την ίδια πολυπλοκότητα με τον προηγούμενο αλγόριθμο (δεν είναι όμως ντετερμινιστικοί) και απαιτούν πολύ λιγότερη μνήμη. Ένα άλλο πλεονέκτημα τους είναι ότι μπορούν να εκτελεστούν παράλληλα. Η ιδέα της ρ-μεθόδου είναι η εύρεση διενέξεων (collisions) που μπορούν να προκύψουν από την επανάληψη μιας συγκεκριμένης συνάρτησης. Π.χ. αν έχουμε μια συνάρτηση $F(x)$ και ένα σταθερό σημείο X_0 στο πεδίο ορισμού της F, παράγουμε μια ακολουθία σημείων $X_{i+1} = F(X_i)$. Ο στόχος είναι να βρούμε δύο $i, j (i \neq j)$ τέτοια ώστε $X_i = X_j$. Είναι σημαντικό εδώ να αναφέρουμε ότι η συνάρτηση F πρέπει να συμπεριφέρεται τυχαία. Αυτήν την υπόθεση την χρησιμοποιούμε για να εφαρμόσουμε το παράδοξο των γενεθλίων. Χωρίς να μπούμε σε λεπτομέρειες, μπορούμε να υποθέσουμε ότι, η F πρέπει μετά από κάποιο χρόνο να ξαναπαίρνει μια τιμή του παρελθόντος. Αυτό βασίζεται στην επόμενη παρατήρηση :
Όταν χρησιμοποιούμε μια γεννήτρια τυχαίων bits (PRG) που ξεκινάει με είσοδο ℓ − bits(seed), τότε αναμένουμε μετά από $2^{\ell/2}$ βήματα να επαναλαμβάνονται τα προηγούμενα bits.

Αυτό σχηματικά παρουσιάζεται όπως στο σχήμα 6. Η ευθεία ονομάζεται *ουρά* (tail) ενώ το δεύτερο κομμάτι του σχήματος *κύκλος* (cycle). Το πλήθος των σημείων του κύκλου, έστω ℓ, είναι η περίοδος της ακολουθίας $(X_i)_i$ και έστω s να είναι το πλήθος των σημείων που βρίσκονται στην ουρά της ακολουθίας $(X_i)_i$. Ένας τρόπος για να βρει κάποιος μια διένεξη της (τυχαίας) ακολουθίας $(X_i)_i$ είναι να αποθηκεύει κάθε τιμή X_i, και κάθε φορά που υπολογίζει μια νέα τιμή, πριν την αποθηκεύσει να εκτελεί δυαδική αναζήτηση στις προηγούμενες τιμές, και αν δεν έχει διένεξη τότε να την αποθηκεύει. Όταν φτάσουμε στην τιμή $X_{s+\ell}$ ο αλγόριθμος θα σταματήσει διότι θα έχει βρει μια διένεξη. Επομένως, χρειαζόμαστε μνήμη $O(s + \ell)$. Η πολυπλοκότητα του αλγορίθμου που υπολογίζει τα

[20]Ενας αλγόριθμος τύπου Monte Carlo είναι ένα πιθανοτικός πολυωνυμικός αλγόριθμος που η έξοδος του είναι σωστή με κάποια πιθανότητα. Σε αντίθεση με τους αλγόριθμους τύπου Las Vegas που η έξοδος είναι πάντα σωστή αλλά ο χρόνος μπορεί να είναι εκθετικός, αλλά κατά μέσο όρο πολυωνυμικός.



Σχήμα 6: ρ-Pollard.

$X_0, X_1, .., X_s, ..., X_{s+\ell} = X_s$, είναι $O((s+\ell)\log_2(s+\ell))$. Ένας τρόπος να αποφύγουμε την προηγούμενη υπερβολική μνήμη που χρειαζόμαστε είναι να εντοπίζουμε τις διενέξεις με τον κυκλικό αλγόριθμο του Floyd (ή τον πιο καλό αλγόριθμο του Brend). Οι αλγόριθμοι αυτοί, δεν εξάγουν τα s, ℓ όπως ο προηγούμενος, αλλά εντοπίζουν διενέξεις της ακολουθίας $(X_i)_i$, λίγο αργότερα στον κύκλο.

Έστω G μία πολλαπλασιαστική κυκλική ομάδα τάξης m. Αναζητούμε έναν θετικό ακέραιο $x < m$, τέτοιον ώστε $g^x = h$, για κάποιο $g \in G$. Η ιδέα του αλγορίθμου είναι η εύρεση δύο ζευγαριών $(a_i, b_i), (a_j, b_j)$ με $b_i \not\equiv b_j (\text{mod } m)$ τέτοια ώστε $g^{a_i} h^{b_i} = g^{a_j} h^{b_j}$. Τότε,

$$\log_g(h) = x \equiv \frac{a_j - a_i}{b_i - b_j} (\text{mod } m).$$

Αν η μετάβαση από την τιμή X_i στην X_{i+1} μοιάζει τυχαία, τότε αναμένουμε $X_i = X_j$ για $j \approx \sqrt{\frac{\pi m}{2}}$. Για να το δούμε αυτό χρειαζόμαστε την παρακάτω πρόταση (παράδοξο των γενεθλίων).

Πρόταση 4.2.1. *Έστω S ένα σύνολο με N στοιχεία. Αν κάνουμε ομοιόμορφη δειγματοληψία από το S, ο αναμενόμενος αριθμός δειγμάτων που πρέπει να έχουμε, πριν κάποιο στοιχείο εμφανιστεί δύο φορές, είναι $\sqrt{\pi N/2}$.*

Στην περίπτωση μας έχουμε μια συνάρτηση $f : G \to G$, που συμπεριφέρεται τυχαία και $X_i = f(X_{i-1})$, $i \geq 0$,(με X_0 κάποιο συγκεκριμένο σημείο της G) τότε, από την προηγούμενη πρόταση αναμένουμε μια διένεξη $X_i = X_j$, μετά από περίπου $\sqrt{\pi N/2}$, εφαρμογές της f στο X_0. Επίσης, συμπεραίνουμε ότι το συνολικό πλήθος σημείων, ουράς και κύκλου, είναι περίπου $\sqrt{\pi N/2}$. Αποδεικνύεται επίσης ότι[21] το μήκος του κύκλου (δηλ. το πλήθος των σημείων επί του κύκλου), αλλά και της ουράς, είναι ασυμπτωτικά περίπου $\sqrt{\pi N/8}$.

Περιγράφουμε αναλυτικά την μέθοδο. Έστω $G = G_1 \cup G_2 \cup G_3$ ξένα μεταξύ τους υποσύνολα της G με περίπου ίδιο μέγεθος ($|G_1| \approx |G_2| \approx |G_3|$). Για παράδειγμα αν $G = \mathbb{Z}_p^*$, τότε μπορούμε να διαλέξουμε

$$G_i = \{x \in G : (i-1)p/3 \leq x < ip/3\}, \ i = 1, 2, 3.$$

[21] Fajolet,Odlyzko, Random mapping Statistics

Θεωρούμε την συνάρτηση

$$F(X) = \begin{cases} gX, & X \in G_1 \\ X^2, & X \in G_2 \\ hX, & X \in G_3 \end{cases}.$$

Ορίζουμε την ακολουθία $X_{i+1} = F(X_i), X_0 = 1$. Αν $1 \in G_2$, τότε διαλέγουμε κάποιο άλλο X_0. Η ακολουθία αυτή μας δίνει στοιχεία της μορφής $g^{a_i}h^{b_i}$. Η μέθοδος απαιτεί να γνωρίζουμε σε κάθε βήμα τα ζευγάρια (a_i, b_i). Αν για παράδειγμα $X_i \in G_1$, τότε $F(X_i) = gX_i = g^{a_i+1}h^{b_i}$. Επομένως, $a_{i+1} = a_i + 1,\ b_{i+1} = b_i$. Με αυτόν τον τρόπο προκύπτει

$$a_{i+1} = \begin{cases} a_i + 1, & X_i \in G_1 \\ 2a_i, & X_i \in G_2 \\ a_i, & X_i \in G_3 \end{cases}$$

και

$$b_{i+1} = \begin{cases} b_i, & X_i \in G_1 \\ 2b_i, & X_i \in G_2 \\ b_i + 1, & X_i \in G_3 \end{cases}$$

με $a_0 = b_0 = 0$.

Ο Pollard για να εντοπίσει τις διενέξεις χρησιμοποίησε τον κυκλικό αλγόριθμο του Floyd για την προηγούμενη συνάρτηση F. Ο λόγος που χρησιμοποίησε αυτόν τον αλγόριθμο είναι γιατί δεν έχει απαιτήσεις σε μνήμη. Με αυτόν τον τρόπο πετυχαίνουμε την πολυπλοκότητα του αλγορίθμου Shanks αλλά με μνήμη $O(1)$. Βέβαια, ο αλγόριθμος του Pollard δεν είναι ντετερμινιστικός.

Αλγόριθμος 4.2.2. : Ο κυκλικός αλγόριθμος του Floyd
Είσοδος. Συνάρτηση F, αρχική τιμή X_0, πλήθος επαναλήψεων M.
Έξοδος. Μια διένεξη μεταξύ του i και $i+1$

```
1 X ← X_0
2 Y ← X_0
3 for i = 1 to M do
4 |   X ← F(X)
5 |   Y ← F(F(X))
6 |   if X = Y then
7 |   |   return διένεξη μεταξύ i και i + 1
8 |   |   Exit
  |   end
  end
9 return FAIL
```

Ο αλγόριθμος αυτός ψάχνει το μικρότερο t τέτοιο ώστε $X_t = Y_t = X_{2t}$. Η μνήμη που χρησιμοποιεί ο αλγόριθμος είναι πολύ λίγη. Αποθηκεύει σε κάθε κύκλο τα ζευγάρια (X, Y). Η πολυπλοκότητα είναι $O(t)$. Εφόσον $X_t = X_{2t}$ το $t|\ell$ όπου ℓ η περίοδος της $(X_i)_i$. Επειδή το X_t είναι επί του κύκλου της ακολουθίας $(X_i)_i$ προκύπτει ότι $t > s$, όπου s το μήκος της ουράς της ακολουθίας $(X_i)_i$. Τέλος, επειδή ο t είναι ο μικρότερος δυνατός θετικός ακέραιος με την ιδιότητα $X_t = X_{2t}$,

έπεται ότι $t = \lceil \frac{s}{\ell} \rceil \ell$. Επειδή, $s + \ell = O(\sqrt{\pi m/2})$ προκύπτει ότι η μέθοδος ρ έχει πολυπλοκότητα $O(\sqrt{\pi |G|/2})$ και ασήμαντη απαίτηση σε μνήμη.

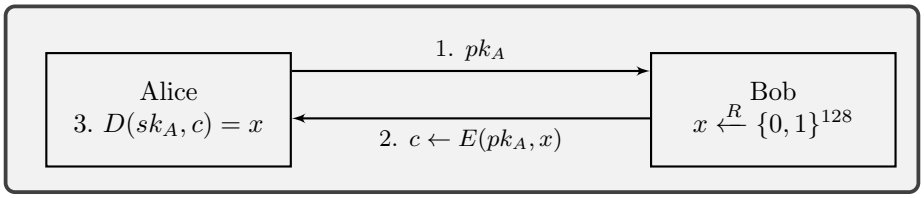
Κεφάλαιο 5

Trapdoor functions (TDF)

Αρχικά, δίνουμε το ορισμό ενός κρυπτοσυστήματος δημόσιου κλειδιού.

Ορισμός 5.0.1. *Μια τριάδα αλγορίθμων (G, E, D) ονομάζεται κρυπτοσύστημα δημόσιου κλειδιού αν*
(i). G : *πιθανοτικός αλγόριθμος για την δημιουργία δημόσιου κλειδιού (pk. : public key) και ιδιωτικού κλειδιού (sk. : secret key).*
(ii). $E(pk, m)$: *πιθανοτικός αλγόριθμος που δέχεται ως είσοδο το μήνυμα m και το δημόσιο κλειδί pk και δίνει έξοδο το κρυπτογραφημένο μήνυμα c.*
(iii). $D(sk, c)$: *Ντετερμινιστικός αλγόριθμος που δέχεται ως είσοδο το κρυπτογραφημένο μήνυμα c και το ιδιωτικό κλειδί sk και δίνει έξοδο το μήνυμα m.*
Για κάθε (pk, sk) που παράγεται από τον G, ισχύει $D(sk, E(pk, m)) = m$ για κάθε m.

Για παράδειγμα στο παρακάτω σχήμα η Alice κάνοντας χρήση του αλγορίθμου G παράγει ένα ζευγάρι κλειδιών (pk_A, sk_A). Κατόπιν στέλνει στον Bob το δημόσιο κλειδί της pk_A. Μετά ο Bob στέλνει ένα 128-bits AES-key στην Alice χρησιμοποιώντας το δημόσιο κλειδί της.



Όπως σε κάθε κρυπτοσύστημα ορίζουμε και μια έννοια της ασφάλειας, έτσι και εδώ θα ορίσουμε τι σημαίνει ασφαλές κρυπτοσύστημα δημόσιου κλειδιού $\Delta = \{G, E, D\}$, για παθητικές επιθέσεις, δηλαδή η Εύα έχει τον ρόλο του ωτακουστή (eavedropper). Για να ορίσουμε την ασφάλεια ενός συστήματος δημόσιου κλειδιού σε τέτοιου τύπου επιθέσεις, χρησιμοποιούμε το παρακάτω παίγνιο.
Υπόθεση. Η Εύα χρησιμοποιεί για τους υπολογισμούς της, έναν πιθανοτικό πολυωνυμικού χρόνου αλγόριθμο.

1. Η Εύα παράγει δύο (διαφορετικά) μηνύματα m_0, m_1 ίδιου μήκους και τα στέλνει στην Alice (στην οποία δίνουμε τον ρόλο ενός μαντείου).
2. Η Alice διαλέγει ένα από τα μηνύματα που έστειλε η Εύα, ρίχνοντας ένα δίκαιο νόμισμα, κατόπιν κρυπτογραφεί με το δημόσιο κλειδί της και στέλνει το αποτέλεσμα στην Εύα. Αν κρυπτογραφήσει το μήνυμα m_0, τότε λέμε ότι εκτέλεσε το

πείραμα EXP(0), διαφορετικά λέμε ότι εκτέλεσε το πείραμα EXP(1).

Τώρα η Εύα πρέπει να αποφασίσει ποιο πείραμα εκτέλεσε η Alice. Αν η Εύα δεν μπορεί να ξεχωρίσει τα δύο πειράματα με πιθανότητα $\gg 1/2$, δηλ. δεν γνωρίζει ποιο από τα δύο μηνύματα έχει κρυπτογραφήσει η Alice , τότε λέμε ότι το σύστημα μας είναι *Σημασιολογικά Ασφαλές* (semantically secure (SS)). Για να δώσουμε ένα φορμαλιστικό ορισμό δουλεύουμε ως εξής. Ας είναι $W_b,\ b \xleftarrow{R} \{0,1\}$ τα ενδεχόμενα

$$W_b = \{\text{Η Εύα εξάγει το bit 1 όταν εκτελείται το EXP(b)}\}.$$

Ορίζουμε την συνάρτηση (η οποία καλείται Semantically Secure Advantage Function)

$$\text{Adv}_{\text{SS}}(\Delta) = \big|Pr(W_0) - Pr(W_1)\big|.$$

Ισχύει $\text{Adv}_{\text{SS}}(\Delta) \in [0,1]$. Αν η Εύα μπορεί να ξεχωρίσει τα δύο πειράματα τότε $\text{Adv}_{\text{SS}}(A,\Delta) = 1$, διαφορετικά οι δύο πιθανότητες θα είναι πολύ κοντά οπότε $\text{Adv}_{\text{SS}}(\Delta) \to 0$.

Ορισμός 5.0.2. *Το Δ λέμε ότι είναι σημασιολογικά ασφαλές (SS) αν*

$$\text{Adv}_{SS}(\Delta) \to 0.$$

Παράδειγμα 5.0.1. Ένα *ντετερμινιστικό* σύστημα κρυπτογράφησης δεν είναι semantically secure. Πράγματι η Εύα μπορεί να ξεχωρίσει τα κρυπτογραφημμένα μηνύματα που προέρχονται από ίδια μηνύματα, από αυτά που προέρχονται από διαφορετικά μηνύματα. Η $Pr(W_0) = 0,\ Pr(W_1) = 1$ άρα $\text{Adv}_{\text{SS}}(\Delta) = 1$.

Παρατήρηση 5.0.1. Ένα SS σύστημα δημόσιου κλειδιού έχει τις παρακάτω ιδιότητες:
• Κατά την κρυπτογράφηση δεν διαρρέει κάποια πληροφορία.
• Η γνώση μόνο του αρχικού κειμένου δεν είναι ικανή να αποκαλύψει κάποια πληροφορία.
• Είναι αδύνατο να βρει κάποιος δύο μηνύματα που οι απόκρυπτογραφήσεις τους να μπορούν να βρεθούν.

Αν δώσουμε στην Εύα πιο ενεργό ρόλο, τότε ή Alice μπορεί να δεχτεί επιθέσεις επιλεγμένου κρυπτογραφημένου κειμένου (CCA-Chosen Ciphertext Attack). Όπως παραπάνω πρέπει να ορίσουμε και μία νέα έννοια της ασφάλειας για τις CCA επιθέσεις. Σε αυτή την περίπτωση θα ορίσουμε ένα διαφορετικό παίγνιο και θα απαιτήσουμε πάλι η συνάρτηση $\text{Adv}_{\text{CCA}}(\Delta) \to 0$. Το παίγνιο ορίζεται ως εξής.
1. Η Εύα ζητάει από την Alice να αποκρυπτογραφήσει τα $c_i, i = 1, 2, ..., n$ μηνύματα.
2. Η Alice απαντάει με τα $m_i = D(sk, c_i)$.
3. Η Εύα παράγει δύο μηνύματα m_0, m_1 ίδιου μήκους και τα στέλνει στην Alice.
4. Η Alice διαλέγει ένα από τα μηνύματα που έστειλε η Εύα, ρίχνοντας ένα δίκαιο νόμισμα, κατόπιν κρυπτογραφεί με το δημόσιο κλειδί της και στέλνει το αποτέλεσμα

στην Εύα, έστω $c = E(pk, m_b)$. Αν κρυπτογραφήσει το μήνυμα m_0 τότε λέμε ότι εκτέλεσε το πείραμα EXP(0), διαφορετικά λέμε ότι εκτέλεσε το πείραμα EXP(1).
5. Η Εύα ζητάει από την Alice να αποκρυπτογραφήσει τα $c_i' \neq c, i = 1, 2, ..., n$ μηνύματα.

Αν στο τέλος η Εύα δεν μπορεί να προσδιορίσει ποιο πείραμα εκτελέστηκε, δηλαδή $\text{Adv}_{CCA}(\Delta) \to 0$ λέμε τότε ότι το σύστημα μας είναι ασφαλές υπό επιθέσεις επιλεγμένων κρυπτογραφημένων κειμένων (CCA-secure).

Ορισμός 5.0.3. *Μια τριάδα αλγορίθμων (G, F, F^{-1}) ονομάζεται TDF όταν*
(i). *G : πιθανοτικός αλγόριθμος για την δημιουργία δ.κ.:δημόσιου κλειδιού (pk. : public key) και ι.κ.:ιδιωτικού κλειδιού (sk. : secret key). Επίσης το pk ορίζει μία συνάρτηση $F(pk, *) : X \to Y$.*
(ii). *$F(pk, *)$: αποδοτικός ντετερμινιστικός αλγόριθμος που ορίζει μία συνάρτηση $F(pk, *) : X \to Y$*
(iii). *υπάρχει $F^{-1}(sk, *)$, που ορίζει μια συνάρτηση από $Y \to X$ και είναι αντίστροφη της F. Δηλαδή, για κάθε ζεύγος (pk, sk) που παράγεται από τον G, ισχύει*

$$F^{-1}(sk, F(pk, x)) = x$$

για κάθε $x \in X$. Επίσης, ο υπολογισμός της $F^{-1}(sk, *)$ είναι αποδοτικός.

Ως συνήθως, πρέπει και σε αυτή την περίπτωση να ορίσουμε τη σημαίνει ασφαλής TDF.

Ορισμός 5.0.4. *Μία συνάρτηση TDF, (G, F, F^{-1}), ονομάζεται ασφαλής αν η F είναι συνάρτηση μίας φοράς.*

Δηλαδή, μπορεί να υπολογιστεί αποδοτικά αλλά όχι να αντιστραφεί, εκτός και αν γνωρίζουμε το κλειδί sk (trapdoor information). Αν $X = Y$, τότε η TDF ονομάζεται και TDP : Tradoor Permutation.

Οι Diffie-Hellman έδωσαν τον ορισμό της TrapDoor Function-TDF για να υλοποιήσουν κρυπτοσυστήματα δημόσιου κλειδιού. Τα πρώτα πραγματικά παραδείγματα δόθηκαν από τους Rivest-Shamir-Adleman[22] με την ανακάλυψη της RSA-TDF και του Rabin με την Rabin-TDF. Το εύλογο ερώτημα που προκύπτει, αν υποθέσουμε ότι έχουμε μία ασφαλή TDF, είναι πως μπορούμε να κατασκευάσουμε ένα ασφαλές κρυπτοσύστημα; Μία ιδέα είναι να εφαρμόσουμε απευθείας την TDF στο μήνυμα m. Δηλαδή η κρυπτογράφηση να είναι $c = E(pk, m) = F(pk, m)$ και η αποκρυπτογράφηση $D(sk, c) = D(sk, E(pk, m)) = m$. Αυτό το κρυπτοσύστημα έχει πολλά προβλήματα! Για παράδειγμα δεν είναι σημασιολογικά ασφαλές, διότι έχουμε μια ντετερμινιστική κρυπτογράφηση (παράδειγμα 5.0.1). Επίσης, θα δούμε στην επόμενη ενότητα μια επίθεση στο RSA αν η TDF-RSA χρησιμοποιηθεί όπως προηγουμένως, στην περίπτωση που τα μηνύματα είναι σχετικά μικρού μήκους. Οι πρακτικές που χρησιμοποιούνται για να μετασχηματίσουμε το μήνυμα ονομάζονται

[22]και οι τρεις, Ron Rivest, Adi Shamir και Leonard Adleman έχουν βασικό πτυχίο (BSc) μαθηματικού. Ο Rivest ήταν μαθητής του Floyd και ο Adleman ήταν μαθητής του Manuel Blum. Ο Adi Shamir ήταν μαθητής του Zohar Manna. Ήταν αποδέκτες του Turing Medal το 2002.

padding. Για το RSA το πιο γνωστό είναι το PKCS#1[23]. Ο σκοπός είναι αυτές οι πρακτικές να μετατρέψουν το σύστημα μας σε SS-CCA, σημασιολογικά ασφαλές υπό επιθέσεις κρυπτογτραφημένου κειμένου.

Δεν πρέπει ποτέ να υλοποιούμε ένα κρυπτοσύστημα δημόσιου κλειδιού εφαρμόζοντας απευθείας την TDF(δείτε την ενότητα 6.2.1).

Παρακάτω θα ορίσουμε ένα κρυπτοσύστημα δημόσιου κλειδιού (το οποίο το ονομάζουμε κρυπτοσύστημα ISO) που αποδεικνύεται ότι είναι CCA ασφαλές.

Διαλέγουμε
• μία TDF, (G, E, D) που να είναι ασφαλής,
• υποθέτω ότι έχω μια συνάρτηση κατακερματισμού $H(x)$ που συμπεριφέρεται σαν τυχαίο μαντείο-random oracle και
• έχω ένα ασφαλές συμμετρικό κρυπτοσύστημα (E_s, D_s) ορισμένο επί της τριάδας (K, M, C).

Τότε η κρυπτογράφηση $E(pk, m)$ γίνεται ως εξής.
Είσοδος: pk, m
$x \xleftarrow{R} X$
$y \leftarrow F(pk, x),\ k \leftarrow H(x)$
$c \leftarrow E_s(k, m)$
Έξοδος: (y, c)

Η αποκρυπτογράφηση $D(sk, (y, c))$ γίνεται ως εξής.
Είσοδος: $sk, C = (y, c)$
$x \leftarrow F^{-1}(sk, y)$
$k \leftarrow H(x)$
$m \leftarrow D_s(k, c)$
Έξοδος: m

5.1 RSA TDF

Η TDF-RSA είναι στην ουσία το πρώτο κρυπτοσύστημα δημόσιου κλειδιού που δουλεύει χωρίς την χρήση ενός συμμετρικού κρυπτοσυστήματος.

Ας είναι e, N φυσικοί αριθμοί με $e < N$. Ορίζουμε την συνάρτηση

$$RSA_e : \mathbb{Z}_N^* \to \mathbb{Z}_N^*$$

με

$$RSA_e(x) = x^e (\text{mod } N).$$

Εφόσον $x \in \mathbb{Z}_N^*$, έχουμε $\gcd(x, N) = 1$. Η TDF-RSA ορίζεται από την τριάδα (G, F, F^{-1}) ως εξής:

[23] Public Key Cryptographic Standards

• G : πιθανοτικός αλγόριθμος που παράγει δύο πρώτους αριθμούς p, q ίδιου μήκους (και τουλάχιστον 1024 bits). Θέτουμε $N = pq$. Κατόπιν η G παράγει δύο φυσικούς αριθμούς e, d τέτοιους, ώστε $ed \equiv 1 (\text{mod } \phi(N))$. Τότε, το δημόσιο κλειδί είναι $pk = (e, N)$ και το ιδιωτικό $sk = (d, N)$.
• $y = F(pk, x)$, ορίζεται $y = RSA_e(x)$.
• η $F^{-1}(sk, y) = y^d (\text{mod } N)$. Πράγματι, από το θεώρημα του Euler (Θεώρημα 3.5.1)

$$y^d = [(RSA_e(x)]^d \equiv x^{ed} \equiv x^{1+k\phi(n)} \equiv x \cdot \left(x^{\phi(N)}\right)^k \equiv x \cdot 1^k \equiv x \ (\text{mod } N).$$

Παρατήρηση 5.1.1. Ορίσαμε την συνάρτηση $RSA_e(x)$ να έχει πεδίο ορισμού το $\mathbb{Z}_n^*$. Σε πολλά βιβλία (αλλά όχι όλα!) που παρουσιάζουν τον RSA θα δείτε το πεδίο ορισμού να είναι το $\mathbb{Z}_n$. Αυτό διότι ισχύει $x^{ed} \equiv x \ (\text{mod } N)$ για κάθε $x \in \mathbb{Z}_N$ (δείτε την επόμενη άσκηση). Παρόλα αυτά αν επιτρέψουμε $x \in \mathbb{Z}_n$ τότε αν $x = p$ ή q εύκολα βρίσκουμε την παραγοντοποίηση του N. Π.χ. αν έχουμε το κρυπτογραφημένο μήνυμα $c = p^e \ (\text{mod } N)$, τότε $\gcd(c, N) = p$. Για να αποφύγουμε αυτή την προβληματική κατάσταση περιορίσαμε το πεδίο ορισμού της $RSA_e(x)$ να είναι το $\mathbb{Z}_n^*$.

Άσκηση 5.1 Έστω p, q δύο (διαφορετικοί) πρώτοι και $N = pq$. Επίσης e, d δύο ακέραιοι με $ed \equiv 1 \ (\text{mod } N)$. Ν.α.ο. για κάθε ακέραιο x ισχύει $x^{ed} \equiv x \ (\text{mod } N)$.

Η υπόθεση που κάνουμε είναι ότι η TDF-RSA είναι μίας φοράς. Δηλαδή, για κάθε αποδοτικό πιθανοτικό αλγόριθμο $\mathcal{A}$ υποθέτουμε ότι ισχύει

$$Pr[y \xleftarrow{R} \mathbb{Z}_N^* : \mathcal{A}(N, e, y) = y^{1/e} (\text{mod } N)] \ll 1, \tag{5.1.1}$$

όπου $N = pq$ και p, q τυχαίοι πρώτοι αριθμοί ίδιου μήκους.

Αυτή η υπόθεση δεν έχει αποδειχτεί. Δηλαδή, δεν υπάρχουν κάτω φράγματα για την πολυπλοκότητα επίλυσης αυτού του προβλήματος. Αυτό το φαινόμενο συμβαίνει συχνά στην κρυπτογραφία δημόσιου κλειδιού. Συνήθως, οι υποθέσεις που κάνουμε δεν έχουν αποδειχτεί, αλλά βασίζονται στο γεγονός ότι, μετά από μια μεγάλη χρονική περίοδο δεν έχουν βρεθεί αποδοτικοί αλγόριθμοι.

Οι δύο υποθέσεις που βασίζεται η σύγχρονη κρυπτογραφία δίνονται από τις σχέσεις 4.2.1,5.1.1.

Η εύρεση $e-$οστών ριζών, όταν δεν γνωρίζουμε την παραγοντοποίηση του N είναι μια τέτοια περίπτωση. Επίσης, η υπόθεση ότι δεν μπορούμε να παραγοντοποιήσουμε αποδοτικά είναι μια άλλη υπόθεση που κάνουμε στο RSA. Το σύστημα RSA δεν έχει αποδειχθεί ότι είναι ισοδύναμο με το πρόβλημα της παραγοντοποίησης. Δηλαδή, αν η Εύα μπορεί και αποκρυπτογραφεί μηνύματα που έχουν κρυπτογραφηθεί με το RSA, δεν σημαίνει ότι μπορεί να παραγοντοποιήσει το N (φυσικά, το αντίστροφο ισχύει). Να αναφέρουμε εδώ ότι υπάρχουν συστήματα που είναι ισοδύναμα με το πρόβλημα της παραγοντοποίσης, αλλά αυτά δεν είναι πρακτικά.

Παρατήρηση 5.1.2. (**i**). Αποδείξαμε ότι, $y^d \equiv x \pmod N$. Αυτό μπορεί να ισχύει ακόμη και αν $ed \not\equiv 1 (\mathrm{mod}\ \phi(N))$. Για παράδειγμα αν $N = 133$, $e = 5$ και $d = 11$, τότε $ed \equiv 55 \pmod{\phi(N)}$, αλλά $y^d = (x^e)^d = x^{55} \equiv x \pmod N$ για όλα x. Αυτό το παράδειγμα μας δείχνει ότι μπορεί να έχουμε κλειδιά RSA (e, d) τέτοια ώστε, $ed \not\equiv 1 \pmod{\phi(N)}$. Η εξήγηση για αυτό το φαινόμενο είναι ότι ικανή και αναγκαία συνθήκη για να ισχύει

$$x^{ed} \equiv x \pmod N$$

για κάθε x πρώτο προς το N, είναι

$$ed \equiv 1 \pmod{\lambda(N)},$$

όπου $\lambda(N) = \mathrm{lcm}(p-1, q-1)$ η συνάρτηση του Carmichael 3.5.3 για την περίπτωση $N = pq$. Παρατηρήστε ότι $ed \equiv 1 \pmod{\lambda(N)}$. Πράγματι, $\lambda(133) = 18$ και $55 \equiv 1 \pmod{18}$.
(**ii**). Η ασφάλεια της RSA-TDF βασίζεται στην δυσκολία εύρεσης $e-$οστών ριζών mod N όταν δεν γνωρίζουμε την παραγοντοποίηση του N. Το πρόβλημα αυτό ονομάζεται RSA problem. Μέχρι σήμερα οι αλγόριθμοι που έχουμε απαιτούν την παραγοντοποίηση του N. Δεν είναι γνωστό αν υπάρχουν αλγόριθμοι που να υπολογίζουν $e-$οστές ρίζες χωρίς την παραγοντοποίηση. Με άλλα λόγια δεν γνωρίζουμε αν το πρόβλημα RSA είναι ισοδύναμο με το πρόβλημα της παραγοντοποίησης.
(**iii**). Το πρόβλημα της παραγοντοποίησης είναι στην τομή NP∩co-NP. Είναι απίθανο να είναι NP-complete.
(**iv**). Ποτέ δεν χρησιμοποιούμε την TDF-RSA απευθείας για κρυπτογράφηση. Δηλαδή, αν m ένα μήνυμα ποτέ δεν στέλνουμε $c = RSA_e(m)$. Στην ενότητα 6.2.3 εξηγούμε γιατί δεν πρέπει ο RSA να χρησιμοποιείται χωρίς κατάλληλο σχήμα pad (επέκταση του μηνύματος που πρέπει να κρυπτογραφηθεί). Το σχήμα που χρησιμοποιούμε σήμερα (για κρυπτογράφηση και όχι για υπογραφή) είναι το OAEP : **O**ptimal **A**symmetric **E**ncryption **P**adding. Ο τρόπος με τον οποίο χρησιμοποιούμε σήμερα την TDF-RSA είναι κυρίως για ψηφιακή υπογραφή μηνυμάτων και ανταλλαγή κλειδιών. Όταν η κρυπτογράφηση γίνεται απευθείας λέμε ότι χρησιμοποιούμε το κρυπτοσύστημα textbook RSA (ή plain RSA). Σε αυτή την περίπτωση το N το ονομάζουμε RSA-modulus το e δημόσιο κλειδί και το d ιδιωτικό (ή μυστικό) κλειδί.
(**v**). Αν γνωρίζουμε το d μπορούμε να βρούμε την παραγοντοποίηση του N (το αντίστροφο ισχύει διότι, αν γνωρίζουμε την παραγοντοποίηση του N έχουμε το $\phi(N)$ άρα από την σχέση $ed \equiv 1 \pmod{\phi(N)}$ βρίσκουμε εύκολα το d). Θα δούμε αρχικά ένα ντετερμινιστικό αλγόριθμο που δουλεύει για $e = O(\ln n)$.

Αρχικά, παρατηρούμε ότι το $\phi(N)$ είναι άρτιος και $\phi(N) | ed - 1$. Θέτουμε $k = ed - 1$. Τότε, $k = \phi(N)r$, για κάποιο $r \in \mathbb{Z}$. Άρα $k/r = \phi(N)$. Εφόσον, $d < \phi(N)$, τότε

$$k = r\phi(N) = ed - 1 < e\phi(N) - 1 < e\phi(N),$$

επομένως $r < e = O(\ln n)$. Δηλαδή το $k = ed - 1$ έχει έναν διαιρέτη αρκετά μικρό (αρκετά μικρό σε σχέση με τον N), τον r, τέτοιον ώστε $k/r = \phi(N)$. Έχουμε τον παρακάτω αλγόριθμο.

1. Βρίσκω όλους τους διαιρέτες του $ed-1$ που είναι μικρότεροι του $O(\ln N)$.
2. Για κάθε διαιρέτη από το βήμα 1, θέτω $x=(ed-1)/r$ (το υποψήφιο $\phi(N)$), και αν είναι άρτιος λύνω το σύστημα:

$$p+q=N-x+1,\ pq=N,$$

με αγνώστους τα p,q.
Διαφορετικά, επαναλαμβάνω το βήμα 1 για κάποιο νέο διαιρέτη r.
3. Αν τα $p,q\in\mathbb{Z}$ είναι διαιρέτες του N σταματάω, διαφορετικά συνεχίζω στον επόμενο διαιρέτη που έχω από το βήμα 1.

Ο αλγόριθμος αυτός είναι ντετερμινιστικός που χρειάζεται στην χειρότερη περίπτωση $O(\ln N)$ επαναλήψεις. Το βήμα 1, γίνεται σε $O(\ln^2 N)$ bit complexity (η διαίρεση κοστίζει $O(\ln^2 N)$). Το βήμα 2, που είναι υπολογισμός ριζών μιας δευτεροβάθμιας εξίσωσης γίνεται σε $O(\ln^2 N)$ bit πολυπλοκότητα και τέλος το βήμα 3 κοστίζει $O(\ln N)$. Άρα συνολικά σε $O(\ln(N)\cdot(\ln^2(N)+\ln^2(N)+\ln N))=O(\ln^3 N)$ πράξεις bit στην χειρότερη περίπτωση.

Στην πράξη το e που χρησιμοποιούμε σήμερα είναι το $2^{16}+1$. Επομένως, μπορούμε να χρησιμοποιήσουμε τον προηγούμενο αλγόριθμο. Να αναφέρουμε ότι υπάρχει και ντετερμινιστικός αλγόριθμος που δουλεύει για μεγάλα e. Ο αλγόριθμος αυτός δίνεται με χρήση πλεγμάτων (lattices).

Ο A. May απέδειξε ότι αν $ed<n^2$ τότε υπάρχει ντετερμενιστικός αλγόριθμος πολυωνυμικού χρόνου που μας δίνει την παραγοντοποίηση του n [24].

Στην περίπτωση που το e είναι μεγάλο υπάρχει και πιθανοτικός αλγόριθμος που βρίσκει τα p,q. Ιστορικά, πρώτος ο Miller έδωσε έναν αλγόριθμο που παραγοντοποιεί το N αν γνωρίζουμε ένα πολλαπλάσιο του $\phi(N)$. Ο αλγόριθμος του χρησιμοποιούσε ως υπόθεση ότι ισχύει η επεκτεταμένη υπόθεση του Riemann.

Η περίπτωση για οποιοδήποτε e
Έστω g τυχαίο στοιχείο του συνόλου $\mathbb{Z}_N^*$. Από το θεώρημα του Euler έχουμε $g^{\phi(N)}\equiv 1\pmod N$. Εφόσον $k=\phi(N)r$ για κάποιο θετικό ακέραιο r, έχουμε $g^k\equiv 1\pmod N$. Άρα το $x=g^{k/2}$ είναι μία τετραγωνική ρίζα της μονάδας mod N (αφού $x^2\equiv 1\pmod N$).

Υποθέτουμε ότι το $x=g^{k/2}$ δεν είναι $\equiv\pm1\pmod N$. Τότε, $x\not\equiv\pm1\pmod p$, $x\not\equiv\pm1\pmod q$. Άρα $\gcd(x-1,N)=p$ ή $\gcd(x+1,N)=p$. Με αυτό τον τρόπο θα αποκαλύψουμε έναν πρώτο διαιρέτη του N.

Αν $g^{k/2}\equiv 1\pmod N$, τότε ο αριθμός $x=g^{k/4}$ είναι μία τετραγωνική ρίζα της μονάδας mod N. Αν επίσης $g^{k/4}=1$ τότε το $x=g^{k/8}\pmod N$ είναι μία τετραγωνική ρίζα της μονάδας mod N. Αργά ή γρήγορα θα σταματήσουμε στην χειρότερη περίπτωση στο $g^{k/2^l}$, $l=O(\ln N)$. Η πιθανότητα να διαλέξω ένα στοιχείο g ώστε ένα από τα στοιχεία $g^{k/2^i}\not\equiv\pm1\pmod N$ για κάποιο i, είναι $\geq 1/2$. Άρα έχουμε ένα αλγόριθμο Monte Carlo με πιθανότητα επιτυχίας τουλάχιστον $1/2$ και χρονική πολυπλοκότητα (σε bit) $O((\ln N)^3)$.

[24] Alexander May. Computing the rsa secret key is deterministic polynomial time equivalent to factoring.

(**vi**). Όσον αφορά το μήκος του δημόσιου κλειδιού N αυτό πρέπει να είναι τουλάχιστον 2048 bits. Μικρά N π.χ. της τάξης των 512 bits μπορούν να παραγοντοποιηθούν εύκολα (πρακτικά σε μερικές ώρες, αλλά όχι σε ένα απλό PC) με ένα κόστος της τάξης των 100 δολαρίων, με χρήση του cloud της amazon EC2[25].

Άσκηση 5.2 Έστω ότι στο textbook RSA (δείτε την παρατήρηση 5.1.2 (iv)) έχουμε ένα modulus $N = pq$ και δημόσιο κλειδί e. Έστω το μήνυμα $m = ap$ για κάποιον ακέραιο a. Είναι εύκολο να παραβιάσουμε την ασφάλεια του συστηματός αυτού με τις δοθείσες παραμέτρους;

5.1.1 CRT και RSA

Στο κλασικό RSA έχουμε δημόσιο κλειδί (e, N) και ιδιωτικό κλειδί (d, N), όπου $N = pq$. Υποθέτουμε ότι $\gcd(p-1, e) = \gcd(q-1, e) = 1$. Αντί του ιδιωτικού κλειδιού, μπορούμε να χρησιμοποιήσουμε (p, q, d_p, d_q, i_q), όπου

$$d_p = e^{-1} \bmod (p-1), d_q = e^{-1} \bmod (q-1), i_q = q^{-1} \bmod p.$$

Αλγόριθμος 5.1.1. CRT-RSA (αποκρυπτογράφηση)
Είσοδος. Τυχαίο ακέραιο c και (p, q, d_p, d_q, i_q).
Έξοδος. $c^d \in \mathbb{Z}_N$

```
1 S_p = c^{d_p} mod p
2 S_q = c^{d_q} mod q
3 S = S_q + q(i_q(S_p − S_q) mod p)
4 return S
```

Παρατηρούμε ότι το κλειδί στο CRT-RSA έχει $\approx (\frac{1}{2}\log_2 N)$ bits, δηλ. περίπου τα μισά bits απ' ότι στο κλασικό. Γι' αυτό το λόγο χρησιμοποιείται σε smart cards. Επίσης, είναι περίπου τέσσερις φορές πιο γρήγορο από το κλασικό.

5.2 Rabin TDF

Η TDF-Rabin(1979) βασίζει την ασφάλεια της στην δυσκολία της παραγοντοποίησης ενός ακέραιου αριθμού.
• G : πιθανοτικός αλγόριθμος που παράγει δύο πρώτους αριθμούς p, q ίδιου μήκους (και τουλάχιστον 1024 bits). Θέτουμε $N = pq$. Τότε $pk = N,\ sk = (p, q)$.
• $y = F(pk, x)$ ορίζεται $y = Rabin(x) = x^2 (\bmod N)$.
• η $F^{-1}(sk, y)$: η αντίστροφη συνάρτηση υπολογίζεται εφόσον γνωρίζουμε τους πρώτους παράγοντες p, q του N, με χρήση του Κινέζικου Θεωρήματος υπολοίπων (CRT). Η Διαδικασία είναι η εξής.

Από την ισοδυναμία $x^2 \equiv y (\bmod N)$ έχουμε $x \equiv \pm y_p (\bmod p)$ και $x \equiv \pm y_q (\bmod q)$. Έτσι προκύπτουν τέσσερα συστήματα τα οποία μπορούμε εύκολα να λύσουμε με το CRT και δίνουν τέσσερις λύσεις mod N, έστω $\{x_1, x_2, x_3, x_4\}$. Μία από αυτές είναι η ζητούμενη. Σε αυτό το σημείο υπάρχει μια προσυμφωνημένη

[25] https://github.com/eniac/faas

Σχήμα 7: Michael Rabin (1931-) Turing Medal 1979. Ήταν μαθητής του Alonzo Church.
(the photo has CC BY-SA 2.0 de licence.)

διαδικασία που μας επιτρέπει να επιλέξουμε το σωστό x_i. Για παράδειγμα αν υπάρχει πλεονασμός (αν το κείμενο είναι γραμμένο σε φυσική γλώσσα) τότε εύκολα θα μπορούσαμε να διαλέξουμε την λύση.

Παράδειγμα 5.2.1. Έστω $p = 7, q = 11,\ N = 77,\ x \in \{0, 1, ..., 76\}$. Έστω $x = 20$. Τότε
$F(pk, x) = 20^2 \pmod{77} = 15$.
$F^{-1}(sk, y) : x^2 \equiv 15 \pmod{77}$. Από το πόρισμα 3.8.1 προκύπτει

$$\begin{cases} x^2 \equiv 15 \pmod 7 = 1 \\ x^2 \equiv 15 \pmod{11} = 4 \end{cases}$$

Μια τετραγωνική ρίζα του 1(mod 7) είναι το 1 ενώ μία τετραγωνική ρίζα του 4(mod 11) είναι το 2. Λύνω το σύστημα $\begin{cases} x \equiv 1 \pmod 7 \\ x \equiv 2 \pmod{11} \end{cases}$. Εφαρμόζουμε CRT και σύμφωνα με την ορολογία της ενότητας 3.8 έχουμε

$$m = 77, m_1 = p = 7, m_2 = q = 11, M_1 = m/m_1 = 11, M_2 = m/m_2 = 7,$$

$$\upsilon_1 \equiv M_1^{-1} \pmod{m_1} = 2 \pmod 7, \upsilon_2 \equiv M_2^{-1} \pmod{m_2} = 8 \pmod{11}.$$

Το $x \equiv (1 \cdot M_1 \cdot \upsilon_1 + 2 \cdot M_2 \cdot \upsilon_2) \equiv 57 \bmod 77$. Οι υπόλοιπες λύσεις προκύπτουν παίρνοντας όλα τα δυνατά πρόσημα στην παράσταση

$$(\pm 1 \cdot M_1 \cdot \upsilon_1 \pm 2 \cdot M_2 \cdot \upsilon_2) \bmod 77 = 64, 13, \mathbf{20}.$$

Ασφάλεια
Ο Rabin παρατήρησε ότι αν μπορούμε να υπολογίζουμε τετραγωνικές ρίζες mod N, τότε μπορούμε να βρούμε την παραγοντοποίηση του N σε πολυωνημικό χρόνο.

Διαλέγουμε τυχαία ένα ακέραιο m από το σύνολο $\mathbb{Z}_N^*$ και υπολογίζουμε το τετράγωνο του mod N. Έστω, $y = m^2 \pmod N$. Έφοσον μπορούμε να υπολογίζουμε τετραγωνικές ρίζες, λύνουμε την εξίσωση, $x^2 \equiv y \pmod N$. Από CRT η ισοδύναμία $x^2 \equiv y \pmod N$ έχει τέσσερις διαφορετικές λύσεις mod N, τις $\{m, -m, a, -a\}$ όπου $a \not\equiv \pm m \pmod N$. Με πιθανότητα $1/2$ το $x \in \{a, -a\}$. Σε αυτή την περίπτωση $a^2 \equiv y \equiv m^2 \pmod N$. Οπότε, $N|a^2 - m^2 = (a-m)(a+m)$. Αν $\gcd(N, a-m) = 1$ τότε $N|a+m$ άτοπο, διότι $a \not\equiv -m \pmod N$. Επίσης, $\gcd(N, a-m) \neq N$ για τον ίδιο λόγο. Άρα $\gcd(N, a-m) = p$ ή q. Επομένως, με πιθανότητα $1/2$ μπορούμε να παραγοντοποιήσουμε το N σε πολυωνυμικό χρόνο.

Επομένως το πλεονέκτημα του Rabin κρυπτοσυστήματος είναι ότι έχει *απόδειξη* ασφάλειας, ενώ το RSA όχι. Επίσης τη προηγούμενη απόδειξη μπορούμε να την χρησιμοποιήσουμε για να δείξουμε ότι δεν είναι IND-CCA (όπως και το textbook-RSA δεν είναι IND-CCA). Έχει προταθεί το SAEP padding scheme για το σύστημα του Rabin ώστε να γίνει IND-CCA secure[26].

Αλγόριθμος 5.2.1. : Παραγοντοποίηση του N έχοντας μια ρουτίνα υπολογισμού τετραγωνικών ριζών mod N, όπου N είναι της μορφής pq.
Είσοδος. N
Έξοδος. Την παραγοντοποίηση του N

```
1 m ←$ Z*_N
2 y ← m^2 (mod N)
3 x ← √y (mod N)
4 if gcd(N, x − m) is between 2 and N − 1
5     return gcd(N, x − m)
6 else choose a new m
```

Με $\sqrt{y}$ εννοούμε έναν ακέραιο x τέτοιον ώστε $x^2 \equiv y \pmod N$.

Το κρυπτοσύστημα του Rabin έχει το πλεονέκτημα ότι το πρόβλημα στο οποίο βασίζεται έχει αποδειχτεί ότι είναι τόσο δύσκολο όσο η παραγοντοποίηση, πράγμα το οποίο δεν είναι γνωστό αν ισχύει για το RSA problem. Με άλλα λόγια η αντιστροφή της συνάρτησης του Rabin είναι ισοδύναμη με την παραγοντοποίηση.

5.3 ElGamal

Το κρυπτοσύστημα ElGamal είναι μια τροποποίηση του συστήματος ανταλλαγής κλειδιών Diffie-Hellman ώστε να κρυπτογραφούμε μηνύματα. Το κρυπτοσύστημα αυτό χρησιμοποιεί έναν πρώτο αριθμό p τέτοιο ώστε $q|p-1$, για κάποιο πρώτο q. Έστω, $G = \langle g \rangle$ η κυκλική ομάδα τάξης q με $G \subset \mathbb{Z}_p^*$. Το δημόσιο κλειδί του συστήματος είναι η τριάδα (p, q, g). Έστω k τυχαίο στοιχείο της ομάδας G. Υπολογίζουμε το στοιχείο $g^k = h$, μέσα στην ομάδα $\mathbb{Z}_p^*$. Το μυστικό κλειδί είναι το k.

[26] Dan Boneh, Simplified OAEP for the RSA and Rabin Functions

Η κρυπτογράφηση γίνεται με τον υπολογισμό του ζευγαριού

$$(r, s) = (g^\ell, mh^\ell) \in \mathbb{Z}_p^* \times \mathbb{Z}_p,$$

όπου ℓ τυχαίο στοιχείο της ομάδας G.

Η αποκρυπτογράφηση γίνεται σε δύο βήματα. Υπολογίζουμε, $r^k = g^{k\ell} = h^\ell$ και κατόπιν, $s(r^k)^{-1} = sh^{-\ell} = m$.

Το πλεονέκτημα αυτού του συστήματος έναντι του RSA είναι ότι μπορεί να χρησιμοποιηθεί με οποιαδήποτε κυκλική ομάδα αντί της $\mathbb{Z}_p^*$. Ειδικότερα, η αντικατάσταση της ομάδας με την ομάδα μια ελλειπτικής καμπύλης επί ενός πεπερασμένου σώματος, είναι η βάση της κρυπτογραφίας με ελλειπτικές καμπύλες.

5.3.1 Κρυπτογραφία με Ελλειπτικές καμπύλες

Χωρίς να επεκταθούμε σε τεχνικές λεπτομέρειες θα πούμε δύο λόγια για την πολύ σημαντική εφαρμογή των ελλειπτικών καμπύλων στην κρυπτογραφία. Πρώτοι, το 1986, ο Koblitz και Miller πρότειναν την εφαρμογή της ομάδας των σημείων μιας ελλειπτικής καμπύλης στα συστήματα Diffie-Hellman και ElGamal. Για να μπορέσει όμως να εφαρμοστεί αυτή η ιδέα στην κρυπτογραφία έπρεπε να υπολογιστεί η τάξη της ομάδας μιας ελλειπτικής καμπύλης. Ο Schoof κατέληξε σε έναν αλγόριθμο το 1985, πολυωνυμικής πολυπλοκότητας για το προηγούμενο πρόβλημα. Αυτό ακριβώς επέτρεψε τους Koblitz και Miller να υλοποιήσουν την ιδέα τους.

Η χρήση της κρυπτογραφίας ελλειπτικών καμπύλων (ECC : Elliptic Curve Cryptography) είναι σήμερα αρκετά διαδεδομένη, άλλα όχι στο βαθμό που είναι το RSA. Για παράδειγμα το ψηφιακό νόμισμα Bitcoin χρησιμοποιεί ψηφιακές υπογραφές που βασίζονται σε ελλειπτικές καμπύλες. Ένα μεγάλο πλεονέκτημα της ECC είναι τα μικρά ιδιωτικά κλειδιά που είναι μήκους 160−bits, ενώ του RSA τουλάχιστον 2048−bits. Επίσης, έχει πλεονέκτημα και έναντι των κλασικών συστημάτων πάνω στο $\mathbb{Z}_p^*$, όπου για την εύρεση του διακριτού λογαρίθμου έχουμε υποεκθετικούς αλγόριθμους, ενώ μέχρι σήμερα ο καλύτερος (πρακτικός) αλγόριθμος για την εύρεση του διακριτού λογαρίθμου επί ελλειπτικών καμπύλων είναι εκθετικός (Pollard-ρ).

Τελευταία, έχουν δοθεί ενδείξεις ότι μπορεί και να υπάρχουν υποεκθετικού χρόνου αλγόριθμοι και για ελλειπτικές καμπύλες. Αυτές οφείλονται στον Semaev που ανάγει όλο το πρόβλημα σε ένα τετραγωνικό σύστημα επί ενός πεπερασμένου σώματος (δηλ. σύστημα που οι εξισώσεις του είναι δευτέρου βαθμού).

Φυσικά, όπως και στο RSA, χρειάζεται ιδιαίτερη προσοχή στο πως θα υλοποιήσουμε αυτά τα συστήματα. Προτείνονται συγκεκριμένες ελλειπτικές καμπύλες π.χ. από το NIST, για υλοποίηση αυτών των πρωτοκόλλων. Η δική μας γνώμη είναι να χρησιμοποιείτε τις οικογένειές ελλειπτικών καμπύλων που δόθηκαν από τον Bernstein[27] ή τις ελλειπτικές καμπύλες του Edwards.

5.3.2 Post Quantum ECC

Ο Couveignes και οι Rostovtsev, Stolbunov το 2006 πρότειναν ένα πρωτόκολλο τύπου Diffie-Hellman βασιζόμενοι στο πρόβλημα εύρεσης ισογένειας μεταξύ δύο

[27] https://safecurves.cr.yp.to/

ελλειπτικών κανονικών καμπύλων (isogeny key exchange protocols over ordinary curves). Είναι σημαντικό να παρατηρήσουμε ότι αυτό το πρόβλημα πιστεύουμε (ελπίζουμε) ότι είναι κβαντικά ασφαλές, με την έννοια ότι δεν ανάγεται σε ένα πρόβλημα παραγοντοποίησης ή διακριτού λογαρίθμου. Επομένως, είναι υποψήφιο για την εποχή μετά τους κβαντικούς υπολογιστές (Post Quantum era). Έχει προταθεί βελτιωμένη έκδοση του προηγούμενου πρωτοκόλλου στον διαγωνισμό της NIST για Post Quantum Cryptosystems.

Κεφάλαιο 6

Επιθέσεις στο Κρυπτοσύστημα RSA

6.1 Η επίθεση του Wiener

Ένα απλό συνεχές κλάσμα, είναι μια έκφραση της μορφής

$$x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{a_4 + \cdots}}}}$$

το οποίο το συμβολίζουμε και $x = [a_0; a_1, a_2, ...]$. Οι όροι $a_0, a_2, ..$ είναι ακέραιοι και ονομάζονται *μερικά πηλίκα*. Τα μερικά πηλίκα μπορεί να είναι είτε πεπερασμένα είτε άπειρα το πλήθος. Όλοι οι πραγματικοί αριθμοί έχουν ένα τέτοιο ανάπτυγμα και μάλιστα είναι μοναδικό. Για να βρούμε τους όρους a_i μπορούμε να χρησιμοποιήσουμε τον παρακάτω αλγόριθμο.

Αλγόριθμος 6.1.1. : Συνεχές κλάσμα πραγματικού αριθμού
Είσοδος. $x \in \mathbb{R}$
Έξοδος. $[a_0; a_1, a_2, ...] = x$

```
1 i = 0, a_0 = ⌊x⌋, x = x − a_0
2 while x > 0 do
      i = i + 1
      a_i = ⌊1/x⌋
      x = 1/x − a_i
  end
```

Αποδεικνύεται ότι το συνεχές κλάσμα είναι πεπερασμένο αν και μόνο αν ο $x \in \mathbb{Q}$ [14, Θεώρημα 6.2]. Επομένως, για ρητούς αριθμούς μπορούμε ξεκινώντας από τον τελευταίο όρο του συνεχούς κλάσματος $a_{k-1} + 1/a_k$ και πηγαίνοντας προς τα πίσω να βρούμε όλα τα (ανάγωγα) κλάσματα N_i/D_i. Τα κλάσματα αυτά ονομάζονται *συγκλίνων κλάσματα* του x.

Παράδειγμα 6.1.1. Έστω $x = 2677/3599$. Στο πρώτο βήμα θέτουμε $a_0 = \lfloor x \rfloor = 0$. Κατόπιν, $a_1 = \lfloor 1/x \rfloor = \lfloor 3599/2677 \rfloor = 1$ και η νέα τιμή του x είναι $1/x - a_1 = 3599/2677 - 1 = 922/2677$. Άρα $a_2 = \lfloor 2677/922 \rfloor = 2$ κ.ο.κ. βρίσκουμε $x = [0; 1, 2, 1, 9, 2, 1, 3, 1, 1, 3]$. Δηλαδή, κάνουμε Ευκλείδεια διαίρεση, το πηλίκο που προκύπτει είναι το a_i. Ξανακάνουμε Ευκλείδεια διαίρεση στο αντίστροφο του

υπολοίπου και το νέο πηλίκο είναι το a_{i+1} κ.ο.κ. έως ότου προκύψει υπόλοιπο 1. Τα συγκλίνοντα κλάσματα του x είναι

$$[0;1] = 1, [0;1,2] = \frac{1}{1+\frac{1}{2}} = \frac{2}{3}, [0;1,2,1] = \frac{3}{4}, ..., x.$$

+++

Πολυπλοκότητα του αλγορίθμου 6.1.1. Έστω $x = m/n$ ανάγωγο κλάσμα με $m, n > 0$.

Πριν αποδείξουμε το βασικό θεώρημα της ενότητας χρειαζόμαστε το παρακάτω.

Θεώρημα 6.1.1 (Legendre) Αν a ρητός αριθμός και p/q θετικός ρητός με $\gcd(p,q) = 1$ ώστε $|a - p/q| < 1/2q^2$, τότε ο p/q είναι ένας συγκλίνων του a. Δηλαδή, υπάρχει ακέραιος $k > 0$ τέτοιος ώστε $p = N_k,\ q = D_k$.

Θεώρημα 6.1.2 (Wiener) Έστω $N = pq$ με p, q πρώτους με ίδιο πλήθος bits, τέτοιοι ώστε $p < q$. Αν $d < \frac{1}{3}N^{1/4}$ τότε σε χρόνο $O(\ln e \ln N)$ μπορούμε να βρούμε τον μυστικό εκθέτη d.

Απόδειξη. Από την σχέση $ed \equiv 1 (\mod \phi(N))$ υπάρχει ακέραιος k τέτοιος ώστε $ed = 1 + k\phi(N) = 1 + k(p-1)(q-1) = 1 + k(N - p - q + 1)$. Αν διαιρέσουμε με το Nd έχουμε

$$\left|\frac{e}{N} - \frac{k}{d}\right| = \left|\frac{1 + k(1-p-q)}{dN}\right| < \frac{k(p+q)}{Nd} < \frac{p+q}{N}.$$

Αλλά, το $2p$ έχει ένα περισσότερο bit από το q άρα $p < q < 2p$. Οπότε $p < \sqrt{N}$ και $p + q < 3p < 3\sqrt{N}$, δηλαδή

$$\frac{p+q}{N} < \frac{3\sqrt{N}}{N} = \frac{3}{\sqrt{N}}.$$

Τέλος

$$\left|\frac{e}{N} - \frac{k}{d}\right| < \frac{p+q}{N} < \frac{3}{\sqrt{N}}.$$

Από την υπόθεση $d < \frac{1}{3}N^{1/4}$, συνεπώς $d^2 < \frac{1}{9}\sqrt{N}$ επομένως, $3d^2 < \frac{1}{3}\sqrt{N}$. Άρα,

$$\frac{3}{\sqrt{N}} < \frac{1}{3d^2} < \frac{1}{2d^2}.$$

Από το θεώρημα του Legendre προκύπτει το ζητούμενο.

□

Ο Wiener πρότεινε να χρησιμοποιούμε $e > N^{1.5}$ διότι τότε ο αλγόριθμος του δεν δίνει καμία πληροφορία για το d.

Παράδειγμα 6.1.2. Αν ο N έχει 1024-bits τότε ο $N^{1/4}$ έχει το πολύ $1024/4 + 1 = 257$ bits. Επομένως πρέπει να διαλέξουμε τον μυστικό εκθέτη d με τουλάχιστον 258 bits.

Μία υλοποίηση της επίθεσης του Wiener είναι η εξής.

Αλγόριθμος 6.1.2. : Επίθεση του Wiener
Είσοδος. N, e $(e < N)$
Έξοδος. Ο μυστικός εκθέτης d ή FAIL

```
1  e/N = [0; a_1, ..., a_n]
2  for i = 1 to n do
3  |   N_i/D_i = [0; a_1, ..., a_i]
   end
4  for i = 1 to n do
5  |   if (2^e)^{D_i} ≡ 2 mod N then
6  |   |   return D_i
7  |   |   EXIT
   |   end
   end
8  FAIL
```

Μια άλλη υλοποίηση που δίνει τους πρώτους παράγοντες είναι η εξής.

Αλγόριθμος 6.1.3. : Επίθεση του Wiener
Είσοδος. N, e $(e < N)$
Έξοδος. Οι πρώτοι αριθμοί p, q ή FAIL

```
1  e/N = [0; a_1, ..., a_n]
2  for i = 1 to n do
3  |   N_i/D_i = [0; a_1, ..., a_i]
4  |   φ_i = (eD_i − 1)/N_i
   end
5  for i = 1 to n do
6  |   if φ_i ∈ ℤ then
7  |   |   Βρείτε τις ρίζες x_1, x_2 της εξίσωσης: x^2 − (N − φ_i + 1)x + N = 0
8  |   |   if x_1, x_2 ∈ ℤ then
9  |   |   |   return x_1, x_2
10 |   |   |   EXIT
   |   |   end
   |   end
   end
11 FAIL
```

Στην γραμμή 7 γίνεται έλεγχος αν το D_i που διαλέξαμε είναι το σωστό, δηλ. το d. Ο έλεγχος γίνεται έμμεσα. Η εξίσωση της γραμμής 7 έχει ως ρίζες του πρώτους p, q. Πράγματι (π.χ. για $x = p$ και $\phi_i = \phi(N)$),

$$p^2 - (N - \phi(N) + 1)p + N = p^2 - Np + (p-1)(q-1)p - p + N =$$

$$p^2 - Np + p^2q - p^2 - pq + p - p + N =$$

$$-p^2q + p^2q - pq + pq = 0.$$

Παρόμοια και για $x = q$.

Παράδειγμα 6.1.3. Έστω ότι $(N, e) = (5697733, 3105251)$ τότε

$$\frac{e}{N} = [0; 1, 1, 5, 17, 1, 9, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 6, 2]$$

και

$$\frac{N_i}{D_i} \in \{1, 1/2, 6/11, 103/189, 109/200, ...\},$$

$$\phi_i \in \{310250, 6210501, 5692960, ...\}.$$

Για $i = 3$ προκύπτει $(x_1, x_2) = (2393, 2381)$, άρα $d = 11$.

Μία βελτίωση της επίθεσης του Wiener δόθηκε από τους Boneh και Durfee.

Επίθεση 6.1.1 (Boneh-Durfee). *Αν $d < N^{0.292}$, τότε σε πολυωνυμικό χρόνο μπορούμε να υπολογίσουμε το d (η επίθεση αυτή δεν είναι ντετερμινιστική όπως του Wiener).*

Παρατήρηση 6.1.1. Γενικά, οι επιθέσεις στο RSA, ανήκουν σε μια απο τις παρακάτω κατηγορίες. Ευρετική, πιθανοτική ή ντετερμινιστική. Στις ευρετικές επιθέσεις, κάνουμε μια υπόθεση την οποία δεν μπορούμε να αποδείξουμε αλλά ούτε και να βρούμε μια πιθανότητα του κατά πόσο ισχύει. Στις πιθανοτικές κάνουμε μια υπόθεση, όπου μπορούμε να αποδείξουμε ότι ισχύει με μια μεγάλη πιθανότητα. Στις ντετερμινιστικές, δεν χρειαζόμαστε κάποια επιπλέον υπόθεση.

Επομένως, αν δούμε το προηγούμενο παράδειγμα με την νέα επίθεση των Boneh-Durfee η απαίτηση είναι ο d να έχει τουλάχιστον 301 bits. Η προηγούμενη επίθεση υλοποιείται όσο το $e < N^{1.875}$. Η επίθεση των Boneh-Durfee δεν είναι πρακτική για $d > N^{0.275}$ διότι οδηγεί σε πλέγματα μεγάλης διάστασης, που ακόμη και ο LLL (δείτε υποενότητα 8.5) αν και πολυωνυμικού χρόνου είναι (πρακτικά) αργός. Το προηγούμενο εφαρμόστηκε για την περίπτωση όπου $e \approx N$. Αν $e = N^{\alpha}$ $(0 < \alpha < 1.875)$ ισχύει το παρακάτω.

Επίθεση 6.1.2 (Boneh-Durfee). *Αν $e = N^{\alpha}$ και $d < N^{\delta(\alpha)}$, $\delta(\alpha) = 7/6 - 1/3(1 + 6\alpha)^{1/2}$, τότε σε πολυωνυμικό χρόνο μπορούμε να υπολογίσουμε το d.*

Επομένως, για $e = 2^{16} + 1$ δηλαδή $\alpha = 0.015$ (για N 1024 bits) και $\alpha = 0.078$ (για N 2048 bits), τότε το d πρέπει να έχει τουλάχιστον $\delta(0.015) \cdot 1024 \approx 838$ bits. Ενώ για N 2048 bits και $e = 2^{16} + 1$, τότε το d πρέπει να έχει τουλάχιστον $\delta(0.078) \cdot 2048 \approx 1562$ bits.

Η Επίθεση του Håstad.

Ας υποθέσουμε ότι ο Bob στέλνει το ίδιο μήνυμα σε τουλάχιστον τρεις αποδέκτες με χρήση του textbook RSA με δημόσιο εκθέτη $e = 3$. Η Εύα που παρακολουθεί το κανάλι έχει στην κατοχή της τρία διανύσματα (N_i, e_i, C_i), όπου $C_i \equiv M_i^e (\text{mod } N_i)$, με $e_i = 3$. Από CRT βρίσκει $C \equiv C_i (\text{mod } N_i)$. Επομένως, $C \equiv M^3 (\text{mod } N_1 N_2 N_3)$. Αλλά $M < N_i$ οπότε $M^3 < N_1 N_2 N_3$. Άρα $C = M^3$. Η Εύα υπολογίζει την κυβική πραγματική ρίζα του C. Η επίθεση αυτή γενικεύεται με το παρακάτω θεώρημα.

Θεώρημα 6.1.3 Έστω $N_1, ..., N_k$ θετικοί ακέραιο ανά δύο πρώτοι και $N_{min} = \min_i N_i$ $(i = 1, 2, ..., k)$. Αν $g_i(x) \in \mathbb{Z}_{N_i}[x]$ βαθμού $d < k$ και $M < N_{min}$ με $g_i(x) \equiv 0 (\text{mod } N_i)$, τότε υπάρχει αποδοτικός αλγόριθμος που υπολογίζει το M.

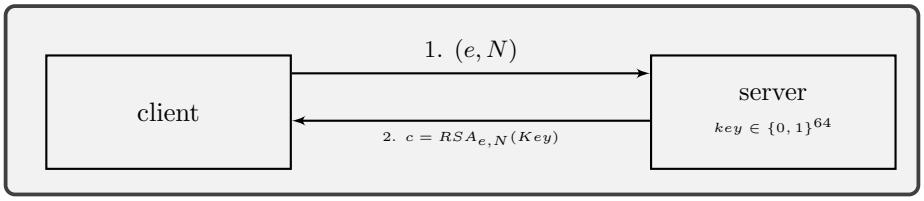
Για να αποφύγουμε την προηγούμενη επίθεση προτείνεται η εφαρμογή τυχαίου padding πριν την κρυπτογράφηση αντί σταθερού padding. Για παράδειγμα ο Bob στέλνει στον $i-$παραλήπτη αντί του μηνύματος M το μήνυμα $i \cdot 2^m + M$ (linear padding), όπου m το πλήθος των bits του M. Αλλά και σε αυτή την περίπτωση το προηγούμενο θεώρημα σε συνδυασμό με την επίθεση του Coppersmith μπορεί επίσης να αποκαλύψει το μήνυμα M.

6.2 Επιθέσεις στο RSA βασισμένες σε πλέγματα

Σε αυτή την ενότητα θα αναφέρουμε επιθέσεις στο RSA που βασίζονται σε πλέγματα (μια εισαγωγή δίνεται στην ενότητα 8). Στόχος είναι να καταλάβουμε πότε υλοποιούνται αυτές οι επιθέσεις και όχι να δώσουμε μία μαθηματική ανάλυση αυτών των επιθέσεων.

6.2.1 Επίθεση σε μικρά μηνύματα

Έχουμε ήδη αναφέρει ότι δεν πρέπει να χρησιμοποιούμε απευθείας την TDF-RSA (και γενικά μια οποιαδήποτε TDF). Για παράδειγμα ας θεωρήσουμε την παρακάτω κατάσταση.



Ο πελάτης και ο διακομιστής ανταλλάσσουν κλειδί με χρήση της TDF-RSA. Η Εύα, που ως συνήθως παρακολουθεί το κανάλι επικοινωνίας, κλέβει το c. Με μεγάλη πιθανότητα ($\approx 25\%$ δείτε και την άσκηση 12.7)το key γράφεται $key = m_1 m_2$ με $m_1, m_2 < 2^{32}$. Αλλά, $c = key^e \pmod N$ οπότε $c/m_1^e = m_2^e \pmod N$. Η Εύα σχηματίζει δύο λίστες (στο $\mathbb{Z}_N$):

$$L_1 = \{c/1^e, c/2^e, c/3^e, ..., c/2^{32e}\}, \quad L_2 = \{1^e, 2^e, 3^e, ..., 2^{32e}\}.$$

Επομένως, σε χρόνο $O(2^{32})$ (για την ακρίβεια χρειαζόμαστε 2^{33} υψώσεις mod N) μπορεί να βρει την τομή στο $\mathbb{Z}_N$ (εφόσον πρώτα ταξινομήσει τις λίστες. Η ταξινόμηση της κάθε λίστας γίνεται σε χρονική πολυπλοκότητα $\Theta(2^{32}\log_2(2^{32}))$). Τότε, βρίσκει όλα τα (k_1, k_2) έτσι ώστε $c/k_1^e = k_2^e$. Κάποια επιλογή από τα k_1, k_2 δίνει $m_1 = k_1, m_2 = k_2$ και υπολογίζουμε $key = m_1 m_2$ (για την σωστή επιλογή θα ισχύει $RSA_{e,N}(key) = c$.) Δηλαδή, μπορεί να βρει πολύ γρήγορα (σε χρόνο $\tilde{O}(2^{32})$), το κλειδί. Η επίθεση αυτή είναι αποτελεσματική μόνο για μικρά μηνύματα και εφικτή μόνο αν χρησιμοποιήσουμε το textbook-RSA. Ένας τρόπος να αποφύγουμε αυτήν την επίθεση είναι να προσθέσουμε ένα σταθερό pad πριν στείλουμε το κλειδί. Δηλαδή, αντί του $RSA_{e,N}(key)$ ο διακομιστής να στείλει για παράδειγμα

$$(RSA_{e,N}(key + 2^{1024} + 2^{128} + 1), 2^{1024} + 2^{128} + 1).$$

6.2.2 Merge Sort

(*MergeSort*). Χρησιμοποιήσαμε στην προηγούμενη ανάλυση το γεγονός ότι, αν έχουμε μια λίστα μήκους n, τότε μπορούμε να την ταξινομήσουμε σε χρόνο $\Theta(n \log n)$. Πράγματι ο αλγόριθμος MergeSort έχει αυτή την πολυπλοκότητα. Ανακαλύφθηκε από τον John von Newman το 1945. Ξεκινάμε, διαιρώντας στην μέση την λίστα, σε δύο ίδιου μήκους υπό-λίστες (ας υποθέσουμε για απλότητα ότι ο n είναι άρτιος), έστω L_{right}, L_{left}. Ταξινομούμε, τις δύο υπο-λίστες με τον ίδιο τρόπο, δηλαδή τις χωρίζουμε στην μέση, έτσι θα προκύψουν συνολικά τέσσερις υπό-λίστες, κ.ο.κ., θα καταλήξουμε σε λίστες με δύο στοιχεία. Έτσι, έχουμε ένα δυαδικό δέντρο με δύο βασικά κλαδιά, και το κάθε κλαδί διακλαδίζεται σε άλλα δύο, αριστερό και δεξί, κ.ο.κ. Αφού φτάσουμε στις λίστες με τα δύο στοιχεία, τις ταξινομούμε ανά δύο (κάνοντας το πολύ μία σύγκριση). Τώρα, οι λίστες με τα δύο στοιχεία είναι ταξινομημένες. Στο επόμενο βήμα, χρησιμοποιούμε την συνάρτηση merge η οποία ενώνει και ταξινομεί ανά δύο τις λίστες με τα δύο στοιχεία (θα χρειαστούμε το πολύ 2 συγκρίσεις για κάθε ζευγάρι). Θα καταλήξω να έχω ταξινομημένες λίστες με τέσσερα στοιχεία. Ξαναεφαρμόζουμε την συνάρτηση merge (θα χρειαστούμε το πολύ 4 συγκρίσεις για κάθε ζευγάρι), αργά ή γρήγορα θα καταλήξω στην ζητούμενη ταξινομημένη λίστα. Ο αλγόριθμος αυτός (όπως και του Karatsuba) ανήκει στην κατηγορία Divide and Conquer.

Αλγόριθμος 6.2.1. : MergeSort
Είσοδος. Μία λίστα L με n στοιχεία
Έξοδος. Η Ταξινομημένη λίστα

```
1 def Mergesort(L)
2 if |L| ≤ 1 then
  |   return L
  end
3 Take L_left
4 Take L_right
5 L_right =Mergesort(L_right)
6 L_left =Mergesort(L_left)
7 L← merge(left,right)
8 return L
```

Ο προηγούμενος αλγόριθμος καλεί ως υπορουτίνα τον παρακάτω αλγόριθμο.

Αλγόριθμος 6.2.2. merge
Είσοδος. Δύο ταξινομημένες λίστες L, R.
Έξοδος. Η Ταξινομημένη λίστα που περιέχει τα στοιχεία $L \cup R$.

```
1  def merge(L, R)
2  while |L| > 0 and |R| > 0 do
3  |  if L[0] > R[0] then
4  |  |  C.appendright(R[0]) and R.remove(R[0])
   |  else
5  |  |  C.appendright(L[0]) and L.remove(L[0])
   |  end
   end
6  while |L| > 0 do
7  |  C.appendright(L[0]) and L.remove(L[0])
   end
8  while |R| > 0 do
9  |  C.appendright(R[0]) and R.remove(R[0])
   end
10 return C
```

Οι συγκρίσεις που γίνονται μπορούν να υπολογιστούν ως εξής. Ας είναι $T(n)$ το πλήθος των συγκρίσεων μιας λίστας μήκους n (για απλότητα στις πράξεις υποθέτουμε ότι το n είναι δύναμη του 2). Τότε $T(n/2)$ είναι το πλήθος των συγκρίσεων στις γραμμές 5,6 και $n-1$ συγκρίσεις στην προτελευταία γραμμή. Άρα $T(n) = 2T(n/2) + n$ (για απλότητα, αντί $n-1$ θέσαμε n). Βρίσκουμε, λύνοντας την αναγωγική εξίσωση με $T(1) = 0$, ότι $T(n) = n\log_2(n)$. Άρα, το πλήθος των συγκρίσεων είναι $\Theta(n \log_2 n)$. Η μνήμη που απαιτεί αυτός ο αλγόριθμος είναι $O(n)$. Ο MergeSort είναι ο καλύτερος αλγόριθμος, όσον αφορά το πλήθος των συγκρίσεων.

6.2.3 Η μέθοδος του Coppersmith

Οι επιθέσεις του Coppersmith βασίζονται σε μία κατασκευή που μπορεί να περιγραφεί σε μια διαδικασία τριών βημάτων:
1. Κατασκευή του πλέγματος και εκτέλεση αλγορίθμου LLL[28].
2. Εφαρμογή της ανισότητας Howgrave-Graham.
3. Εφόσον η επίθεση υλοποιείται (θα μας το πει το βήμα 2), εύρεση ακέραιων λύσεων μιας πολυωνυμικής εξίσωσης με συντελεστές στους ακεραίους.
Στο RSA οι επιθέσεις που προκύπτουν είναι οι εξής:
1. Κρυπτανάλυση του RSA με μικρό εκθέτη και σταθερό padding.
2. Κρυπτανάλυση του RSA με μικρό εκθέτη και τυχαίο padding.
3. Κρυπτανάλυση του RSA όταν γνωρίζουμε *αρκετά* bits του πρώτου p ή q.
Ειδικότερα για την επίθεση 3, η μέθοδος του Coppersmith μας δίνει έναν αλγόριθμο παραγοντοποίησης του N χρόνου $O(2^{0.25N})-$bits. Επίσης στη επίθεση 1, έχουμε την ισοδύναμη επίθεση, όπου η Εύα γνωρίζει αρκετά bits από το μήνυμα. Δηλαδή, αν το μήνυμα γράφεται για παράδειγμα $M = 2^{1024} - 2^{200} + x$ και στόχος της Εύας είναι να βρει το x (σε αυτή την περίπτωσή το x έχει $200-$bits και γνωρίζει τα 824 πιο σημαντικά bits του M).

[28] δείτε υποενότητα 8.5

Θεώρημα 6.2.1 (Coppersmith) Έστω $0 < \varepsilon < 1/d$. Ας είναι $F(x)$ ένα κανονικό πολυώνυμο (δηλ. ο συντελεστής του μεγιστοβάθμιου όρου είναι μονάδα) βαθμού d με μία τουλάχιστο ρίζα x_0 στο $\mathbb{Z}_N$ με $|x_0| < X = \lceil 0.5N^{1/d-\varepsilon} \rceil$. Τότε, μπορούμε να βρούμε το x_0 σε χρόνο $poly(d, 1/\varepsilon, \ln N)$.

Αλγόριθμος 6.2.3. : Univariate Coppersmith Method
Είσοδος. Ένα πολυώνυμο $F(x)$ βαθμού d, $\varepsilon \in (0, 1/d)$ και ένας θετικός ακέραιος N (με άγνωστη παραγοντοποίηση)
Έξοδος. Οι ρίζες x_0 της ισοδυναμίας $F(x) \equiv 0 (\text{mod } N)$ με $|x_0| < X$, όπου $X = \lceil 0.5N^{1/d-\varepsilon} \rceil$

1 $h \leftarrow \lceil 1/(\varepsilon d) \rceil$
2 $X \leftarrow \lceil 0.5N^{1/d-\varepsilon} \rceil$
3 $n \leftarrow d \cdot h$
4 Κατασκεύασε απεικόνιση $\Phi_X : \mathbb{Z}[x] \to \mathbb{R}^n$ με $\Phi_X(f_0 + f_1 x + \cdots + f_n x_n) = (f_0, f_1 X, \ldots, f_n X^n)$
5 Κατασκεύασε τα πολυώνυμα $G_{i,j}(x) = N^{h-1-j} F(x)^j x^i,\ 0 \le j < h,\ 0 \le i < d$
6 Κατασκεύασε τα διανύσματα $\Phi_X(G_{i,j})$ και σχημάτισε ένα πίνακα M που τα περιέχει ως γραμμές του
7 LLL-αναγωγή στις γραμμές του πίνακα M. Ας είναι $\mathbf{b}_1$ το πρώτο ανηγμένο διάνυσμα
8 Σχημάτισε το πολυώνυμο $\Phi_X^{-1}(\mathbf{b}_1)$ και εξάγαγε τις ρίζες του στους ακεραίους

Στον προηγούμενο αλγόριθμο τα i, j διατρέχουν το κάθε ένα το σύνολο στο οποίο ορίζονται, ώστε ο πίνακας M να προκύψει τριγωνικός άνω [29]. Ας δούμε ένα παράδειγμα.

Παράδειγμα 6.2.1. Ας υποθέσουμε ότι η Εύα παρακολουθεί το κανάλι επικοινωνίας μεταξύ της Alice και του Bob. Η Alice και ο Bob χρησιμοποιούν textbook-RSA. Ο δημόσιος εκθέτης του Bob είναι $e = 3$ και το RSA-modulus είναι $N = 1797693134 \ldots 9211$ $(1024 - bits)$. Υποθέτουμε ότι το μήνυμα x είναι 290 bits. Στον Bob η Alice δεν στέλνει το x, αλλά προσθέτει ένα pad που αποτελείται από 768 bits, όλα άσοι. Επομένως, η Alice στέλνει το παρακάτω μήνυμα

$$M = 111 \cdots 1111 || bin(x).$$

Η Εύα γνωρίζει το pad, η οποία ψάχνει τα υπόλοιπα 290 bits του M. Επίσης, η Εύα γνωρίζει την κρυπτογράφηση του M που είναι $C = M^e (\text{mod } N)$. Η Εύα σχηματίζει το πολυώνυμο $F(x) = (x + 2^{1024} - 2^{290})^3 - C$ και εκτελεί τον αλγόριθμο 6.2.3. Οπότε υπολογίζει διαδοχικά
1. $h = 7$
2. $X = 109095309735365398756518898013497620784775746772155687711811$
$69473087725596041917979688 96$
3. $n = 21$
4. Ένας πίνακας 21×21 στον οποίο εκτελούμε τον LLL. Το πρώτο διάνυσμα την ανηγμένης βάσης δίνει το πολυώνυμο

$$g(x) = 2977907 \ldots 116524 x^{19} - 109432 \ldots 6372752 x^{18} + \cdots + 3272 \ldots 616608$$

το οποίο έχει ρίζα στο $\mathbb{Z}$ τον ακέραιο

$$x_0 = 8594215741625 \ldots 76551218155386392$$

[29] Μία υλοποίηση του προηγούμενου αλγόριθμου στο σύστημα Sagemath μπορείτε να βρείτε στην github.com/AristotleUniversity/sage/blob/master/our-projects/univariate_coppersmith

που είναι και το ζητούμενο μήνυμα x.

Στο προηγούμενο παράδειγμα το σενάριο θα μπορούσε να ήταν αυτό του 1, που αναφέραμε στην αρχή της ενότητας. Δηλαδή, η Εύα δεν γνωρίζει κάποια bits του μηνύματος αλλά το pad που χρησιμοποιεί η Alice.

6.2.4 Συμπεράσματα

Σήμερα χρησιμοποιούμε το σχήμα pad OAEP [30] και για υπογραφές RSA το σύστημα PSS. Το σχήμα pad OAEP[31] ανακαλύφθηκε από τους Rogaway και Bellare. Το πρότυπο που υλοποιεί το σχήμα OAEP είναι το PKCS#1 ver. 2. Βέβαια στην πράξη βλέπουμε να χρησιμοποιείται και το πρότυπο PKCS#1 ver. 1.5 (που δεν υλοποιεί το σχήμα OAEP). Ιδιαίτερα για το τελευταίο υπάρχει η επίθεση του Bleichenbacher που έδωσε μια επίθεση CCA που χρησιμοποιεί 2^{20} κρυπτογραφημένα μηνύματα. Η επίθεση αυτή δεν εφαρμόζεται στο πρότυπο PKCS#1 ver. 2. Τέλος, έχει προταθεί και το πρότυπο PKCS#1 ver. 2.2.

Ο Pointcheval απέδειξε ότι το σύστημα RSA-OAEP είναι IND-CCA. Με άλλα λόγια υλοποιώντας μια επίθεση κρυπτογραφημένου κειμένου (CCA) το σύστημα είναι σημασιολογικά ασφαλές (Semantically Secure). Δηλαδή ο επιτιθέμενος δεν μπορεί να βρεί ούτε ένα bit από το κρυπτογραφημένο μήνυμα. Η απόδειξη ότι το RSA-OAEP είναι IND-CCA απαιτεί την ύπαρξη ενός τυχαίου μαντείου (random oracle) και δεν βασίζεται μόνο σε υποθέσεις της θεωρίας αριθμών.

Το textbook-RSA έχει ντερμενιστική κρυπτογράφηση επομένως δεν είναι SS. Δηλ. Το textbook-RSA δεν είναι ούτε IND-CPA secure. Αυτό μπορούμε να το δούμε με το εξής παράδειγμα. Ας υποθέσουμε ότι η Alice στέλνει τρεις αριθμούς στον Bob. Η Εύα γνωρίζει ότι οι αριθμοί που στέλνει η Alice είναι οι a, b, c. Η Εύα επίσης γνωρίζει τις κρυπτογραφήσεις των a, b, c (εφόσον το δημόσιο κλειδί είναι γνωστό σε όλους) επομένως, μπορεί να ελέγξει ποιο μήνυμα στέλνει η Alice στον Bob.

Λήμμα 6.2.1. *Το textbook-RSA δεν είναι IND-CCA.*

Απόδειξη. Έστω ότι θέλουμε να αποκρυπτογραφήσουμε το c το οποίο έχει κρυπτογραφηθεί με textbook-RSA. Από την υπόθεση (εφόσον υλοποιούμε CCA-attack) έχουμε ένα μαντείο-RSA που δέχεται ως είσοδο κρυπτογραφημένα μηνύματα και απαντάει με την αποκρυπτογράφηση αυτών (Decryption Oracle). Υποθέτουμε οτι οι παράμετροι του RSA ειναι (e, N) και η αποκρυπτογράφηση του c είναι ο αριθμός m. Ο μόνος περιορισμός είναι ότι δεν μπορούμε να του στείλουμε το c (challenge ciphertext).

Διαλέγουμε ένα τυχαίο ακέραιο s από το σύνολο

$$\{1, 2, .., N-1\} - \{c\}.$$

Υποθέτουμε $\gcd(s, N) = 1$. Αν $\gcd(s, N) > 1$ τότε το $s = p$ ή q. Οπότε μπορούμε να αποκρυπτογραφήσουμε οποιοδήποτε μήνυμα. Στέλνουμε στο μαντείο τον αριθμό $c' = s^e c \bmod N$. Έστω m' ο αριθμός που επιστρέφει το μαντείο. Τότε, υπολογίζουμε τον αριθμό $m's^{-1} \bmod N$. Παρατηρούμε ότι $(m's^{-1})^e \equiv s^e m(s^e)^{-1} \equiv$

[30] ftp://ftp.di.ens.fr/pub/users/pointche/Papers/2002_cryptobytes.pdf
[31] **O**ptimal **A**symmetric **E**ncryption **P**adding

$m \bmod N$. Επομένως το μήνυμα m υπολογίζεται ως $(m's^{-1})^e \bmod N$ και τα m', s, e, N είναι γνωστά.

□

Στις υλοποιήσεις μας δεν πρέπει να χρησιμοποιούμε το textbook-RSA.

Επίσης, έχει απόδειχτει το εξής αποτέλεσμα όσον αφορά το textbook-RSA.

Θεώρημα 6.2.2 Έστω ότι έχουμε ένα μαντείο που δέχεται ως είσοδο κρυπτογραφημένα μηνύματα και επιστρέφει το LSB[32] του μηνύματος. Δηλαδή, με είσοδο c επιστρέφει $\text{LSB}(m)$, όπου $m = RSA_d(c)$. Τότε, μπορούμε να αποκρυπτογραφήσουμε όποιο μήνυμα θέλουμε.

Αυτή η απόδειξη δόθηκε από τους S. Goldwasser, S. Micali and P. Tong στην εργασία: *Why and how to establish a private code on a public network* το 1982. Γενικεύτηκε το 1998 για οποιοδήποτε bit. Δηλαδή, το λιγότερο σημαντικό bit είναι το ίδιο ασφαλές με ολοκλήρο το μήνυμα. Ο Daniel Bleichenbacher χρησιμοποίησε παρόμοιο RSA-μαντείο για να *σπάσει* το RSA-PKCS#1 ver.1.5 που χρησιμοποιούνταν στο SSL v.3.[33] Συμπεραίνουμε ότι, ενώ το textbook-RSA βασίζει την ασφάλεια του, στη δυσκολία της παραγοντοποίησης (ή καλύτερα στην εύρεση $e-$οστων ριζών $\bmod N$ ενώ δεν γνωρίζουμε την παραγοντοποίηση του N) δεν δημιουργεί ένα ασφαλές κρυπτοσύστημα. Συμπεραίνουμε μέχρι στιγμής τα παρακάτω,

[32] LSB : Least Significant Bit

[33] Daniel Bleichenbacher, Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS #1, 1998

• Η υπόθεση ότι η παραγοντοποίηση είναι δύσκολη, συνεπάγεται ότι η RSA-TDF (textbook-RSA) είναι *μίας φοράς* (one way) υλοποιώντας επίθεση επιλεγμένου κειμένου (CPA-attack). Η ασφάλεια που απαιτούμε από ένα κρυπτοσύστημα είναι η IND-CCA.
• Το textbook-RSA δεν είναι σημασιολογικά ασφαλές υπό παθητικές επιθέσεις [α΄](αποκαλύπτει πληροφορία-leaks partial information).
• Το textbook-RSA δεν είναι ασφαλές σε παθητικές επιθέσεις όταν χρησιμοποιούμε *μικρά* μηνύματα (μπορεί κάποιος να αποκρυπτογραφήσει το μήνυμα). Η επίθεση αυτή δεν είναι πολυωνυμική, αλλά γίνεται πρακτική για μηνύματα που έχουν το πολύ 100-bits. Δείτε την ενότητα 6.2.1.
• Το textbook-RSA δεν είναι ασφαλές αν το μυστικό κλειδί είναι μικρό, έχει δυαδικό μήκος $< 0.292\times$(το δυαδικό μήκος του N) (Boneh-Durfee attack).
• Το textbook-RSA δεν είναι ασφαλές σε CCA-attacks.
• Αν έχουμε μια υπορουτίνα που με είσοδο c επιστρέφει ένα bit του m ($m = RSA_e(C)$), τότε μπορούμε να αποκρυπτογραφήσουμε όποιο μήνυμα θέλουμε.
• Το RSA-PKCS#1 ver.1.5, δεν είναι ασφαλές υπό CCA-attacks (Bleichenbacher attack).
• Το textbook-RSA με constant/random padding δεν είναι ασφαλές αν το $e = 3$ υπό παθητικές επιθέσεις (Coppersmith attack).

[α΄]Η Εύα δεν χρησιμοποιεί κάποιο μαντείο όπως στις επιθέσεις CPA, CCA. Απλά παρακολουθεί την επικοινωνία.

Όσον αφορά την επιλογή $e = 3$ έχουμε να παρατηρήσουμε τα παρακάτω. Αν το $e = 3$ (μία επιλογή αρκετά συνηθισμένη στο παρελθόν), το σύστημα είναι ασφαλές αν χρησιμοποιήσουμε ένα ασφαλές σχήμα pad (π.χ. OAEP). Ειδικά όταν η ταχύτητα κρυπτογραφήσης ή υπογραφής είναι αρκετά σημαντική η επιλογή του $e = 3$ κάνει τον αλγόριθμο RSA ταχύτερο κατά περίπου $8.5-$φορές σε σχέση με την επιλογή $e = 2^{16} + 1$ (την οποία χρησιμοποιούμε σήμερα).

Κεφάλαιο 7

Ψηφιακές Υπογραφές

7.1 Ορισμός ψηφιακής υπογραφής

Οι ψηφιακές υπογραφές χρησιμοποιούνται για την απόδειξη της γνησιότητας ενός ψηφιακού εγγράφου. Είναι συστήματα δημόσιου κλειδιού. Η διαφορά με τα κρυπτοσυστήματα είναι ότι η Alice υπογράφει με το ιδιωτικό κλειδί και ο Bob επαληθεύει με το δημόσιο κλειδί του αποστολέα (Alice). Δηλ. ένα μήνυμα ψηφιακά υπογεγραμμένο από την Alice μπορεί να επαληθευτεί από οποιονδήποτε. Επίσης η Alice δεν μπορεί να αρνηθεί ότι έχει στείλει αυτό το μήνυμα. Αυτό ονομάζεται στην κρυπτογραφία μη αποποίηση του εγγράφου (non-repudiation). Επίσης, κατά αναλογία με την ιδιόχειρη υπογραφή όπου τοποθετούμε το υπογεγραμμένο μήνυμα σε ένα φάκελο και το σφραγίζουμε, μπορούμε το ψηφιακά υπογεγραμμένο μήνυμα να το κρυπτογραφήσουμε με το δημόσιο κλειδί του παραλήπτη. Έκτος από την υπογραφή μηνυμάτων οι ψηφιακές υπογραφές χρησιμοποιούνται στις παιχνιδομηχανές (Xbox, PS3). Παράδοση που ξεκίνησε από την Nintendo το 1985 και συνεχίστηκε με την Atari. Η Nintendo χρησιμοποίησε MAC συναρτήσεις για την αυθεντικοποίηση των παιχνιδιών.

Για να ορίσουμε τι είναι ψηφιακή υπογραφή χρειάζεται να γνωρίζουμε τρεις αλγόριθμους (G, S, V).
G : παραγωγή κλειδιών (pk, sk) (Generation).
Ως συνήθως είναι ένας πιθανοτικός πολυωνυμικός αλγόριθμος που παράγει ένα ζεύγος (pk, sk).
S : αλγόριθμος υπογραφής (Sign).
Είναι ένας πιθανοτικός πολυωνυμικός αλγόριθμος που δέχεται δύο παραμέτρους ως είσοδο sk, m όπου m :μήνυμα, $S(sk, m) = s$. Η έξοδος ονομάζεται υπογραφή του μηνύματος.
V : αλγόριθμος επαλήθευσης (Verify).
Είναι πιθανοτικός αλγόριθμος πολυωνυμικού χρόνου ο οποίος δέχεται ως είσοδο την υπογραφή s και το μήνυμα m και επιστρέφει True αν η υπογραφή είναι σωστή και False διαφορετικά.

7.2 Αδυναμίες ψηφιακής υπογραφής

Υπαρκτή Πλαστογράφηση. Λέμε ότι έχουμε υπαρκτή πλαστογράφηση αν κάποιος τρίτος καταφέρνει να υπογράψει ένα μήνυμα. Το μήνυμα δεν έχει απαραίτητα κάποιο νόημα. Αυτή η αδυναμία αν υπάρχει σε μια ψηφιακή υπογραφή δεν συνεπάγεται ότι το σχήμα ψηφιακής υπογραφής δεν είναι ασφαλές.

Ολική Πλαστογράφηση. Σε αυτή την περίπτωση κάποιος τρίτος, μπορεί και υπογράφει όποιο μήνυμα θέλει. Το σχήμα της ψηφιακής υπογραφής με αυτή την αδυναμία είναι μη ασφαλές. Δεν είναι απαραίτητο κάποιος να έχει το μυστικό κλειδί για να πετυχεί ολική πλαστογράφηση.
Επιλεκτική Πλαστογράφηση. Σε αυτή την περίπτωση κάποια μηνύματα τα οποία διάλεξε κάποιος τρίτος μπορεί να τα υπογράψει.
Ολικό σπάσιμο (total crack). Σε αυτή την περίπτωση το μυστικό κλειδί μπορεί να υπολογιστεί από κάποια τρίτη οντότητα.

7.3 Ψηφιακές Υπογραφές από TDF

Οι Diffie και Hellman παρατήρησαν ότι ένα κρυπτοσύστημα δημόσιου κλειδιού που προκύπτει από TrapDoor Functions, έστω f, μπορεί να χρησιμοποιηθεί και για να υπογράφουμε μηνύματα, απλά προεκτείνοντας το μήνυμα με την υπογραφή $s = f^{-1}(sk, m)$, όπου sk το ιδιωτικό κλειδί της TrapDoor function και m το μήνυμα που επιθυμούμε να υπογράψουμε. Στέλνει στον παραλήπτη (m, s). Ο παραλήπτης ελέγχει την εγκυρότητα της ψηφιακής υπογραφής ελέγχοντας αν ισχύει η ισότητα, $f(pk, s) = m$.
Οι ιδιότητες της TDF συνεπάγονται τις εξής ιδιότητες της ψηφιακής υπογραφής.

(i). Μόνο ο χρήστης που κατέχει το ιδιωτικό κλειδί μπορεί να υπογράφει.
(ii). Οι χρήστες που γνωρίζουν το δημόσιο κλειδί, μπορούν να ελέγξουν την εγκυρότητα της υπογραφής ενός μηνύματος.
(iii). Υπογεγραμμένα μηνύματα μπορούν να στέλνονται σε μη ασφαλείς διαύλους με ασφάλεια.
(iv). Δεν μπορεί ο κάτοχος του ιδιωτικού κλειδιού να αρνηθεί ότι έστειλε το υπογεγραμμένο μήνυμα (m, s).

7.3.1 Υπογραφή RSA

7.3.2 DSA

DSA : Digital Signature Algorithm είναι μια ψηφιακή υπογραφή που χρησιμοποιείται επίσημα απο το 1991, από την αμερικάνικη κύβερνηση αλλά και στο SSL/TLS. Χρησιμοποίειται αρκετά η εκδόχη της με τις ελλειπτικές καμπύλες[34]. Αυτή η ψηφιακη υπογραφή βασίζει την ασφάλεια της στον διακριτό λογάριθμο και δεν προέρχεται από TDF.

Έστω ότι η Alice θέλει να υπογράψει ένα μήνυμα και να το στείλει στον Bob. Αρχικά η Alice παράγει δύο πρώτους p, q με $q|p-1$. Τα μήκη των (p, q) είναι $(160, 1024)$ ή $(224, 2048)$ ή $(256, 2048)$ bits. Επίσης διαλέγει τυχαία έναν αριθμό a (ιδιωτικό κλειδί) από το σύνολο $\{1, 2, ..., q-1\}$ και υπολογίζει $R = g^a \pmod p$. Το δημοσιο κλειδί είναι η τετράδα (p, q, g, R) καθώς και μια κρυπτογραφικά ασφαλής συνάρτηση κατακερματισμού έστω

$$h : \{0,1\}^* \to \{1, 2, ..., q-1\}.$$

[34] το 1998 προτάθηκε η εκδοχή του DSA με ελλειπτικές καμπύλες. Για παράδειγμα το bitcoin χρησιμοποιεί αυτή την εκδόχη του DSA.

Έστω $m < q$ το μήνυμα που θέλει να υπογράψει. Διαλέγει ένα (εφήμερο) κλειδί k τυχαία από το σύνολο $\{1, 2, ..., q-1\}$ και υπολογίζει τους αριθμούς

$$r = (g^k \bmod p) \bmod q, \;\; s = k^{-1}(h(m) + ar) \bmod q.$$

Το (r, s) είναι η υπογραφή του μηνύματος m. Ο Bob επαληθέυει την υπογραφή υπολογίζοντας τον αριθμό

$$\left(g^{s^{-1}h(m) \bmod q} R^{s^{-1}r \bmod q} \bmod p\right) \bmod q$$

και ελέγχει αν ισούται με r.

Άσκηση 7.1 Να αποδείξετε ότι αν στην ψηφιακή υπογραφή DSA κάποιος χρησιμοποιήσει δύο φόρες το ίδιο εφήμερο κλειδί (για δύο διαφόρετικά μηνύματα), τότε μπορεί να υπολογίσει το ιδιωτικό κλειδί a.

Κεφάλαιο 8

Πλέγματα

8.1 Εισαγωγή στα Πλέγματα

Ορισμός 8.1.1. *Ένα υποσύνολο $L \subset \mathbb{R}^n$ καλείται πλέγμα (lattice), αν υπάρχουν γραμμικός ανεξάρτητα διανύσματα $\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_k$ του $\mathbb{R}^n$ $(n \geq k)$ τέτοια ώστε*

$$L = L(\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_k) = \Big\{ \sum_{j=1}^{k} \alpha_j \mathbf{b}_j : \alpha_j \in \mathbb{Z}, 1 \leq j \leq k \Big\} = \{\mathbf{x}B : \mathbf{x} \in \mathbb{Z}^k\},$$

όπου B έχει ως γραμμές τα $\mathbf{b}_1, ..., \mathbf{b}_k$. Τα διανύσματα $\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_k$ καλούνται βάση του πλέγματος L.

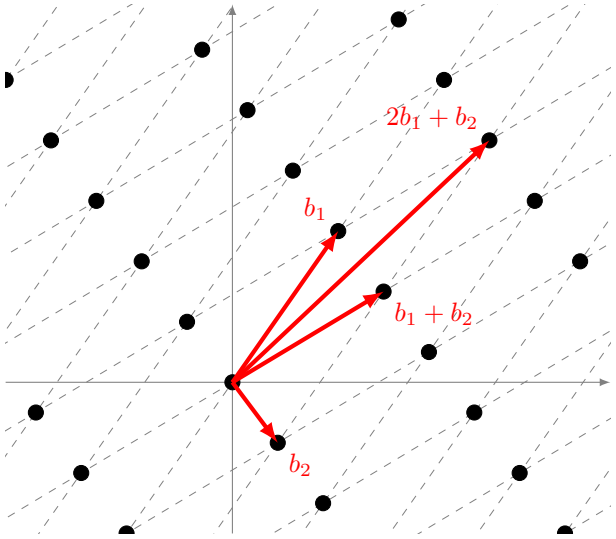
Τα βασικά στοιχεία των πλεγμάτων (όπως και η προχωρημένη θεωρία) μπορεί να βρεθεί στις αναφορές [4],[7, Κεφ.17]. Παρατηρήστε ότι αν $\mathbf{x}, \mathbf{y} \in L$ τότε $(\mathbf{x} - \mathbf{y}) \in L$, δηλαδή L υποομάδα της προσθετικής ομάδας $(\mathbb{R}^n, +)$. Άρα, αν $\mathbf{0} \notin L$ τότε το L δεν είναι πλέγμα. Επιπλέον, όλες οι βάσεις έχουν τον ίδιο αριθμό στοιχείων, αυτός ο κοινός αριθμός k, ονομάζεται *τάξη (rank)* του πλέγματος L. Ο πίνακας M_L διαστάσεων $k \times n$ που σχηματίζεται από τα διανύσματα $\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_k$ ονομάζεται *πίνακας βάσης του πλέγματος (basis matrix)*. Παρατηρήστε ότι η τάξη του πίνακα βάσης ισούται με την τάξη του πλέγματος L δηλ. k. Κάθε στοιχείο του L π.χ. το $a_1\mathbf{b}_1 + \cdots + a_k\mathbf{b}_k$ ισούται με $\mathbf{a}M_L$, όπου $\mathbf{a} = (a_1, ..., a_k)$. Δηλ., $L = \mathbb{Z}^k M_L$. *Διάσταση* του πλέγματος ονομάζουμε τον αριθμό n. Αν $n = k$, λέμε τότε ότι το πλέγμα είναι *μέγιστης τάξης (full rank lattice)*. Στην περίπτωση αυτή λέμε ότι είναι διάστασης n, αντί τάξης n.

Παρατήρηση 8.1.1. Θεωρούμε τώρα την γραμμική απεικόνιση $T : \mathbb{R}^k \to \mathbb{R}^n$ με

$$T(x_1, ..., x_k) = x_1\mathbf{b}_1 + \cdots + x_k\mathbf{b}_k.$$

Παρατηρήστε ότι $T(\mathbb{Z}^k) = L$. Ο πίνακας M_T διαστάσεων $(n \times k)$, της γραμμικής απεικόνισης T δεν ισούται με τον πίνακα του πλέγματος L, αλλά ο ανάστροφος $M_T^T = M_L$. Ο πίνακας M_T της T ορίζεται αν υπολογίσω τις τιμές της T επί της κανονικής βάσης του $\mathbb{R}^k$. Δηλαδή $T(\mathbf{e}_1) = T((1, 0, ..., 0)) = \mathbf{b}_1$ κ.ο.κ. $T(\mathbf{e}_k) = \mathbf{b}_k$. Κατόπιν σχηματίζουμε τον πίνακα που έχει ως στήλες τα στοιχεία $T(\mathbf{e}_1), ..., T(\mathbf{e}_k)$.

Παράδειγμα 8.1.1. (**i**). Έστω $A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 1 \end{bmatrix}$ τότε, το σύνολο $L_1 = \{\mathbf{x} \in \mathbb{Z}^2 : \mathbf{x}A = \mathbf{0}\}$, είναι ένα πλέγμα. Πράγματι, λύνοντας το σύστημα που προκύπτει υπολογίζουμε την λύση $(0, 0)$, έτσι το πλέγμα είναι το $L_1 = \mathbb{Z}\cdot\mathbf{0}$ που είναι μηδενικής τάξης.



Σχήμα 8: Ένα πλέγμα που παράγεται από τα $\mathbf{b}_1, \mathbf{b}_2$.

(**ii**). Αν $B = \begin{bmatrix} 2 & 2 & 2 \\ 1 & 1 & 1 \\ 2 & 3 & 4 \end{bmatrix}$, τότε το σύνολο $L_2 = \{\mathbf{x} \in \mathbb{Z}^3 : \mathbf{x}B = \mathbf{0}\}$, είναι επίσης ένα πλέγμα. Πράγματι λύνοντας το σύστημα προκύπτουν οι λύσεις $\{(t, -2t, 0) : t \in \mathbb{Z}\}$. Δηλαδή το πλέγμα παράγεται από το διάνυσμα $\mathbf{b} = (1, -2, 0)$, άρα $L_2 = \mathbb{Z}\ \mathbf{b}$ (είναι τάξης 1).
(**iii**). Το σύνολο L των ακεραίων λύσεων της γραμμικής εξίσωσης

$$a_1 x_1 + \cdots + a_n x_n = a_0,$$

με $a_0 \neq 0$ και $\gcd(a_1, .., a_n) = 1$ δεν είναι πλέγμα, διότι $\mathbf{0} \notin L$.
(**iv**). Τα σύνολα

$$A_n = \{(x_0, ..., x_n) \in \mathbb{Z}^{n+1} : \sum_{i=0}^{n} x_i = 0\}\ (n \geq 1) \tag{8.1.1}$$

και

$$D_n = \{(x_1, ..., x_n) \in \mathbb{Z}^n : \sum_{i=1}^{n} x_i \equiv 0 \pmod 2\}\ (n \geq 3) \tag{8.1.2}$$

είναι πλέγματα[35].

Ορισμός 8.1.2. *Ονομάζουμε στοιχειώδη παραλληλόγραμμο ενός πλέγματος $L(B)$, όπου B $k \times n$, το σύνολο $\mathcal{P}(B) = \{\mathbf{x}B : \mathbf{x} \in [0, 1)^k\}$.*

Αν $n = k = 2$, και $\{\mathbf{b}_1, \mathbf{b}_2\}$ μια βάση του πλέγματος, τότε το $\mathcal{P}(B)$ είναι το ημί-ανοιχτό παραλληλεπίπεδο με κορυφές $\mathbf{O}, \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_1 + \mathbf{b}_2$.

[35]Τα πλέγματα αυτά ονομάζονται και root lattices, διότι προέρχονται από σύνολα διανυσμάτων που είναι root systems.

Ορισμός 8.1.3. *Ονομάζουμε όγκο (volume) ενός πλέγματος $L(B)$ τον όγκο του στοιχειώδες παραλληλόγραμμου του πλέγματος. Τον συμβολίζουμε* $\det(L)$ *ή* $vol(L)$.

Άσκηση 8.1 Να απόδειξετε ότι ο όγκος του πλέγματος (8.1.1) για $n = 2$, ισούται με $\sqrt{3}$. Δηλ. $vol(A_2) = \sqrt{3}$. Επίσης, να αποδείξετε ότι $vol(D_3) = 2$.

Άσκηση 8.2 Έστω $\mathcal{L}$ ένα πλέγμα. Το σύνολο,

$$V(\mathcal{L}) = \{\mathbf{x} \in \mathbb{R}^n : \text{για κάθε } \mathbf{y} \in \mathcal{L}, ||\mathbf{x}|| \leq ||\mathbf{x} - \mathbf{y}||\}$$

καλείται κλειστό Voronoi κελί του πλέγματος $\mathcal{L}$. Περιγράψτε γεωμετρικά το $V(\mathbb{Z}^2)$.

Παρατήρηση 8.1.2. Μερικές φορές ονομάζουμε τον όγκο ενός πλέγματος και co-volume και ισούται με τον όγκο του τόρου (torus) $\text{span}(L)/L$.

Λήμμα 8.1.1. *Έστω B, B' δύο πίνακες $k \times n$ τάξης k, και οι γραμμές του B' είναι διανύσματα του $L(B)$. Τότε, η B' είναι μία βάση του πλέγματος $L(B)$ αν και μόνο αν $\mathcal{P}(B') \cap L(B) = \{\mathbf{0}\}$.*

Απόδειξη. Έστω B' μία βάση του πλέγματος $L(B)$. Ας είναι επίσης, $\mathbf{x} \in \mathcal{P}(B') \cap L(B)$. Θα αποδείξουμε ότι $\mathbf{x} = \mathbf{0}$. Εφόσον, το $\mathbf{x} \in L(B)$ και B' βάση του $L(B)$, το $\mathbf{x} = \mathbf{y}B'$, για κάποιο $\mathbf{y} \in \mathbb{Z}^k$ και επειδή $\mathbf{x} \in \mathcal{P}(B)$, έχουμε ότι, $\mathbf{x} = \mathbf{z}B'$, για κάποιο $\mathbf{z} \in [0,1)^k$. Άρα, $\mathbf{y}B' = \mathbf{z}B'$ έχουμε $(\mathbf{y} - \mathbf{z})B' = \mathbf{0}$, επομένως $\mathbf{y} - \mathbf{z} \in Ker(B'^T)$, όπου $Ker(B'^T)$ ο αριστερός μηδενοχώρος (cokernel) του B'. Ισχύει, $\dim[Ker(B'^T)] = k - rank(B') = 0$. Καταλήγουμε ότι $\mathbf{y} = \mathbf{z}$. Αναγκαστικά, το διάνυσμα $\mathbf{y}$ του πλέγματος $L(B')$, θα είναι το μηδενικό, δηλαδή $\mathbf{x} = \mathbf{0}$.
(Αντίστροφα). Θα δείξουμε ότι $L(B) \subseteq L(B')$. Έστω $\mathbf{x} \in L(B) \subset \mathbb{R}^n$. Επειδή

$$\text{span}(\mathbf{b}_1, ..., \mathbf{b}_k) = \text{span}(\mathbf{b}'_1, ..., \mathbf{b}'_k),$$

υπάρχει $\mathbf{y} \in \mathbb{R}^k$, τέτοιο ώστε $\mathbf{x} = \mathbf{y}B'$. Αλλά $(\mathbf{y} - \lfloor \mathbf{y} \rfloor) \in [0,1)^k$. Επομένως, $\mathbf{x}' = (\mathbf{y} - \lfloor \mathbf{y} \rfloor)B' \in \mathcal{P}(B')$. Επίσης, $\lfloor \mathbf{y} \rfloor B' \in L(B)$ διότι οι γραμμές του B' είναι διανύσματα του $L(B)$. Επομένως, $\mathbf{x}' = \mathbf{x} - \lfloor \mathbf{y} \rfloor B' \in L(B)$. Άρα, $\mathbf{x}' = 0$ και επομένως $\mathbf{x} = \lfloor \mathbf{y} \rfloor B' \in L(B')$. Δηλαδή, $L(B) \subseteq L(B')$. Τέλος, επειδή οι πίνακες B, B' έχουν ίδια τάξη, προκύπτει ότι $L(B) = L(B')$. Δηλαδή, B' βάση του $L(B)$. □

Παράδειγμα 8.1.2. Έστω, $B = \{\mathbf{b}_1 = (0,2),\ \mathbf{b}_2 = (1,4)\}$. Ας είναι $B' = \{\mathbf{b}'_1 = (0,1),\ \mathbf{b}'_2 = (-1,1)\}$. Τότε, το B' δεν είναι βάση του $L(B)$, διότι $\mathbf{b}'_1 \in \mathcal{P}(L(B))$.

Πρόταση 8.1.1. *Ας είναι B, B' δύο $n \times n$ πίνακες βάσης ενός μέγιστης τάξης πλέγματος L. Τότε υπάρχει πίνακας $U \in SL_n(\mathbb{Z})$ με $B = UB'$, και αντίστροφα.*

Απόδειξη. ($\Rightarrow$) Ας είναι $B = [\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_n]^T$, $B' = [\mathbf{b}'_1, \mathbf{b}'_2, ..., \mathbf{b}'_n]^T$. Τότε,

$$\mathbf{b}'_i = \sum_{j=1}^{n} a_{ij}\mathbf{b}_j.$$

Άρα υπάρχει πίνακας U με ακέραια στοιχεία , $U = [a_{ij}]_{1\leq i,j\leq n}$, τ.ω. $B' = UB$. Παρόμοια, υπάρχει πίνακας U' τ.ω. $B = U'B'$. Άρα $B' = UU'B'$. Επειδή τα διανύσματα $\mathbf{b}'_1, \mathbf{b}'_2, ..., \mathbf{b}'_n$ είναι γ.α. (ως διανύσματα βάσης) προκύπτει ότι ο πίνακας B' αντιστρέφεται. Άρα $I_n = B'B'^{-1} = UU'$. Επομένως, $\det(UU') = 1$, άρα $\det U = \pm 1$. Δηλαδή $U \in SL_n(\mathbb{Z})$.
($\Leftarrow$) Έστω $B' = UB$, τότε $\mathbf{x}B \in L$. Αλλά, $\mathbf{x}B = \mathbf{x}U'B' = \mathbf{y}B' \in L'$ έτσι $L \subseteq L'$. Παρόμοια το $\mathbf{y}B' \in L'$ οπότε $\mathbf{y}UB = \mathbf{x}B \in L$. Άρα $L' \subseteq L$. Επομένως $L = L'$. □

Η προηγούμενη Πρόταση ισχύει και γενικότερα.

Πρόταση 8.1.2. *Ας είναι B, B' δύο $k \times n$ πίνακες βάσης ενός πλέγματος L. Τότε υπάρχει πίνακας $U \in SL_k(\mathbb{Z})$ με $B = UB'$, και αντίστροφα.*

Απόδειξη. [7, Κεφ. 17, λήμμα 16.1.6] □

Άσκηση 8.3 Υπολογίστε τον όγκο του πλέγματος που έχει βάση $\mathbf{b}_1 = (1, 1, 1)$, $\mathbf{b}_2 = (0, -1, 2)$.

Άσκηση 8.4 Αποδείξτε ότι $\dim(rowspace(B)) = rank(L(B))$, όταν το $L(B)$ είναι μέγιστης τάξης.

Άσκηση 8.5 Έστω το πλέγμα $L = \mathbb{Z}\mathbf{b}_1 + \mathbb{Z}\mathbf{b}_2$ με $\mathbf{b}_1 = (1, 2)$, $\mathbf{b}_2 = (-1, 2)$. Υπολογίστε το δείκτη ορθογωνιότητας $\delta(L) = \dfrac{||\mathbf{b}_1|| \cdot ||\mathbf{b}_2||}{vol(L)}$. Κατόπιν αποδείξτε ότι $\delta(L) = 1$ αν και μόνο αν τα $\mathbf{b}_1, \mathbf{b}_2$ είναι ορθογώνια.

Άσκηση 8.6 (*) Να αποδείξετε ότι $vol(V(\mathcal{L})) = vol(\mathcal{L})$ (όπου $V(\mathcal{L})$ το Voronoi cell του $\mathcal{L}$, δείτε Άσκηση 8.2).

8.2 Διαδικασία Gram-Schimdt

Την προβολή του διανύσματος $\mathbf{u} \in \mathbb{R}^n$ επί του $\mathbf{v} \neq \mathbf{0}$ την συμβολίζουμε $\text{proj}_{\mathbf{v}}\mathbf{u}$. Υπάρχει $\lambda \in \mathbb{R}$ τέτοιο ώστε $\text{proj}_{\mathbf{v}}\mathbf{u} = \lambda\mathbf{v}$. Αν $\mathbf{u}, \mathbf{v}$ δεν είναι κάθετα, τότε $\lambda \neq 0$. Άλλά, το διάνυσμα $\text{proj}_{\mathbf{v}}\mathbf{u} - \mathbf{u}$ είναι κάθετο στην $\text{proj}_{\mathbf{v}}\mathbf{u}$, άρα

$$(\text{proj}_{\mathbf{v}}\mathbf{u} - \mathbf{u}) \cdot \text{proj}_{\mathbf{v}}\mathbf{u} = (\lambda\mathbf{v} - \mathbf{u}) \cdot \lambda\mathbf{v} = 0.$$

Επομένως ($\lambda \neq 0$),

$$\lambda = \frac{\mathbf{u} \cdot \mathbf{v}}{||\mathbf{v}||^2}.$$

Άρα,

$$\text{proj}_{\mathbf{v}}\mathbf{u} = \frac{\mathbf{u} \cdot \mathbf{v}}{\mathbf{v} \cdot \mathbf{v}}\mathbf{v}.$$

Έστω τώρα $\{\mathbf{b}_1, ..., \mathbf{b}_k\} \subset \mathbb{R}^n$ μια διατεταγμένη βάση ενός διανυσματικού υποχώρου του $\mathbb{R}^n$ ($n \geq k$). Με την παρακάτω διαδικασία προκύπτει μια ορθογώνια βάση.

$$\mathbf{b}_1^* = \mathbf{b}_1,\ \mathbf{b}_2^* = \mathbf{b}_2 - \text{proj}_{\mathbf{b}_1^*}\mathbf{b}_2 = \mathbf{b}_2 - \frac{\mathbf{b}_1 \cdot \mathbf{b}_2}{\mathbf{b}_1 \cdot \mathbf{b}_1}\mathbf{b}_1,$$

$$\mathbf{b}_i^* = \mathbf{b}_i - \sum_{j=1}^{i-1} \text{proj}_{\mathbf{b}_j^*}\mathbf{b}_i, ...$$

$$\mathbf{b}_k^* = \mathbf{b}_k - \sum_{j=1}^{k-1} \text{proj}_{\mathbf{b}_j^*}\mathbf{b}_k.$$

Θέτουμε,

$$\mu_{ij} = \frac{\mathbf{b}_i \cdot \mathbf{b}_j^*}{\mathbf{b}_j^* \cdot \mathbf{b}_j^*}\ (i > j)$$

Η προηγούμενη διαδικασία γράφεται,

$$\mathbf{b}_1^* = \mathbf{b}_1, \mathbf{b}_2^* = \mathbf{b}_2 - \mu_{21}\mathbf{b}_1^* \quad \dots$$

$$\mathbf{b}_k^* = \mathbf{b}_k - \sum_{j=1}^{k-1} \mu_{kj}\mathbf{b}_j^*.$$

Επαγωγικά μπορούμε να δούμε ότι $\mathbf{b}_i^* \cdot \mathbf{b}_j^* = 0$ για $i \neq j$. Η προηγούμενη διαδικασία ονομάζεται *ορθογωνοποίηση κατά Gram-Schimdt*. Αν $\mu = (\mu_{ij})$ ο τριγωνικός κάτω πίνακας $k \times k$ με διαγώνια στοιχεία άσους, τότε η προηγούμενη διαδικασία γράφεται $B = \mu B^*$ όπου B, B^* $k \times n$ πίνακες με γραμμές τα διανύσματα $\mathbf{b}_i$ και $\mathbf{b}_i^*$, αντίστοιχα. Παρακάτω παρουσιάζουμε τον ψευδοκώδικα για την προηγούμενη διαδικασία.

Αλγόριθμος 8.2.1. : GSO Ψευδοκώδικας
Είσοδος. Μια βάση $\mathcal{B} = \{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n\}$ του $V = \text{span}(\mathcal{L})$
Έξοδος. Μια ορθογώνια βάση $\mathcal{B}'$ του V και τον Gram-Schmidt πίνακα (μ_{ij}).

$\mathbf{R}_1 \leftarrow \mathbf{b}_1$
for $i = 1$ *to* n **do**
  $\mathbf{t}_1 \leftarrow \mathbf{b}_i$
  $\mathbf{t}_3 \leftarrow \mathbf{t}_1$
  for $j = i - 1$ *to* 1 **do**
    $\mu_{i,j} \leftarrow \frac{\mathbf{t}_1 \cdot \mathbf{R}_j}{||\mathbf{R}_j||^2}$
    $\mathbf{t}_2 \leftarrow \mathbf{t}_3 - \text{proj}_{\mathbf{R}_j}(\mathbf{t}_1)$
    $\mathbf{t}_3 \leftarrow \mathbf{t}_2$
  end
  $\mathbf{R}_j \leftarrow \mathbf{t}_2$
end
return $\mathcal{B}' = (\mathbf{R}_1, ..., \mathbf{R}_n), \mu = (\mu_{i,j})_{i,j}$

Παράδειγμα 8.2.1. Έστω η διατεταγμένη βάση του $\mathbb{R}^2$,

$$\{\mathbf{b}_1 = (1,1),\ \mathbf{b}_2 = (1,0)\}.$$

Τότε, $\mathbf{b}_1^* = \mathbf{b}_1$ και $\mathbf{b}_2^* = \mathbf{b}_2 - \mu_{21}\mathbf{b}_1^*$. Το

$$\mu_{21} = \frac{\mathbf{b}_2 \cdot \mathbf{b}_1^*}{\mathbf{b}_1^* \cdot \mathbf{b}_1^*} = \frac{1}{2}.$$

Άρα, $\mathbf{b}_2^* = (1/2, -1/2)$.

Μερικές ιδιότητες της βάσης $\{\mathbf{b}_1^*, ..., \mathbf{b}_k^*\}$ είναι οι εξής,

(**i**). $\text{span}(\mathbf{b}_1^*, ..., \mathbf{b}_i^*) = \text{span}(\mathbf{b}_1, ..., \mathbf{b}_i),\ 1 \leq i \leq k$,
(**ii**). $\mathbf{b}_i^* = \mathbf{b}_i - (\alpha_1\mathbf{b}_1 + \cdots + \alpha_{i-1}\mathbf{b}_{i-1})\ (\alpha_1, ..., \alpha_{i-1} \in \mathbb{R})$,
(**iii**). $\mathbf{b}_i^* \cdot \mathbf{b}_j = 0,\ j < i$,
(**iv**). $||\mathbf{b}_i^*|| = \text{distance}(\mathbf{b}_i, \text{span}(\mathbf{b}_1, ..., \mathbf{b}_{i-1}))$, $i \geq 2$.

Η πρώτη προκύπτει διότι

$$\mathbf{b}_i^* = \mathbf{b}_i - \sum_{j=1}^{i-1} \text{proj}_{\mathbf{b}_j^*}\mathbf{b}_i \tag{8.2.1}$$

επομένως,

$$\mathbf{b}_i = \mathbf{b}_i^* + \sum_{j=1}^{i-1} \text{proj}_{\mathbf{b}_j^*}\mathbf{b}_i$$

άρα $\mathbf{b}_i \in \text{span}(\mathbf{b}_1^*, ..., \mathbf{b}_i^*)$, συνεπώς

$$\text{span}(\mathbf{b}_1^*, ..., \mathbf{b}_i^*) \subset \text{span}(\mathbf{b}_1^*, ..., \mathbf{b}_i^*).$$

Αντίστροφα, εξ ορισμού της κατασκεύης Gram-Schimdt έχουμε

$$\text{span}(\mathbf{b}_1^*, ..., \mathbf{b}_i^*) \subset \text{span}(\mathbf{b}_1, ..., \mathbf{b}_i).$$

Η δεύτερη επίσης προκύπτει άμεσα, διότι

$$\mathbf{b}_i^* - \mathbf{b}_i = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{ij}\mathbf{b}_j^* - \mathbf{b}_i =$$

$$-\sum_{j=1}^{i-1} \mu_{ij}\mathbf{b}_j^* \in \text{span}(\mathbf{b}_1^*, ..., \mathbf{b}_{i-1}^*) = \text{span}(\mathbf{b}_1, ..., \mathbf{b}_{i-1}).$$

Η τρίτη ιδιότητα μπορεί να αποδειχθεί επαγωγικά. Για $i = 2$ ισχύει. Πράγματι,

$$\mathbf{b}_2^* \cdot \mathbf{b}_1 = \mathbf{b}_2 \cdot \mathbf{b}_1 - \mu_{21}\mathbf{b}_1^* \cdot \mathbf{b}_1 =$$

$$\mathbf{b}_1 \cdot \mathbf{b}_2 - \frac{\mathbf{b}_2 \cdot \mathbf{b}_1}{||\mathbf{b}_1||^2} \, ||\mathbf{b}_1||^2 = 0.$$

Υποθέτουμε ότι ισχύει για $\leq i-1$ δηλαδή,

$$\mathbf{b}_\rho^* \cdot \mathbf{b}_j = 0, \ 1 \leq j < \rho \leq i-1.$$

Θα αποδείξουμε ότι ισχύει για i. Για $j < i$ έχουμε,

$$\mathbf{b}_i^* \cdot \mathbf{b}_j = \cdots = 0.$$

Αυτή η ιδιότητα ισοδύναμα γράφεται

$$\mathbf{b}_i^* \in \text{span}(\mathbf{b}_1, ..., \mathbf{b}_{i-1})^\perp.$$

Η τέταρτη ιδιότητα εξ ορισμού ισχύει για $i = 2$. Γενικά, αρκεί να παρατηρήσουμε ότι αν $\mathcal{A}_i = \text{span}(\mathbf{b}_1, ..., \mathbf{b}_{i-1})^\perp$, τότε

$$\text{proj}_{\mathcal{A}_i}(\mathbf{b}_i) = \mathbf{b}_i - \sum_{j=1}^{i-1} \text{proj}_{\mathbf{b}_j^*} \mathbf{b}_i = \mathbf{b}_i^*.$$

Παράδειγμα 8.2.2. Έστω $\mathbf{b}_1 = (1,1,1)$, $\mathbf{b}_2 = (0,1,-1)$. Έστω το σημείο $P = (2,-1,3)$. Βρείτε την απόσταση του P από το επίπεδο $\langle \mathbf{b}_1, \mathbf{b}_2 \rangle$.

Η

$$\pi(\mathbf{x}) = \sum_{j=1}^{k} \text{proj}_{\mathbf{b}_j^*}(\mathbf{x}),$$

είναι η ανάλυση του $\mathbf{x}$ στις ορθογώνιες συνιστώσες $\{\mathbf{b}_1^*, ..., \mathbf{b}_k^*\}$. Ενώ,

$$\pi_i(\mathbf{x}) = \sum_{j=i}^{k} \text{proj}_{\mathbf{b}_j^*}(\mathbf{x}),$$

είναι η ανάλυση του $\mathbf{x}$ στις ορθογώνιες συνιστώσες $\{\mathbf{b}_i^*, \ldots, \mathbf{b}_k^*\}$. Ισχύει $\pi_i(\mathbf{b}_i) = \mathbf{b}_i^*$. Εξ ορισμού,

$$\pi_i(\mathbf{x}) \in \text{span}(\mathbf{b}_i^*, ..., \mathbf{b}_k^*).$$

Ειδικότερα από την ιδιότητα (iii)

$$\mathbf{b}_i^* \cdot \mathbf{b}_j = 0 \ (j < i),$$

άρα

$$\mathbf{b}_\ell^* \in \mathcal{A}_i = \text{span}(\mathbf{b}_1, ..., \mathbf{b}_{i-1})^\perp$$

για $\ell = i, i+1, ..., k$. Επίσης, $\dim \mathcal{A}_i = k - i + 1$ επομένως

$$\mathcal{A}_i = \text{span}(\mathbf{b}_i^*, ..., \mathbf{b}_k^*).$$

Δηλ. η προβολή $\pi_i(\mathbf{x}) \in \mathcal{A}_i$.

Ορισμός 8.2.1. *Ονομάζουμε ορθογώνια προβολή στον υποχώρο* $span(\mathbf{b}_i^*, \dots, \mathbf{b}_k^*) \subset \mathbb{R}^n$, *την απεικόνιση*

$$\pi_i : \mathbb{R}^n \to \text{span}(\mathbf{b}_i^*, \dots, \mathbf{b}_k^*) \ (1 \leq i \leq k),$$

$$\pi_i(\mathbf{x}) = \sum_{j=i}^{k} \text{proj}_{\mathbf{b}_j^*}(\mathbf{x}) = \sum_{j=i}^{k} \frac{\mathbf{x} \cdot \mathbf{b}_j^*}{\mathbf{b}_j^* \cdot \mathbf{b}_j^*} \mathbf{b}_j^*$$

και

$$\pi(\mathbf{x}) = \pi_1(\mathbf{x}).$$

Παρατήρηση 8.2.1. Παρατηρήστε ότι ισχύει

$$\pi_i(\mathbf{x}) + \pi_i(\mathbf{y}) = \sum_{j=i}^{n} \frac{\mathbf{x} \cdot \mathbf{b}_j^*}{\mathbf{b}_j^* \cdot \mathbf{b}_j^*} \mathbf{b}_j^* + \sum_{j=i}^{n} \frac{\mathbf{y} \cdot \mathbf{b}_j^*}{\mathbf{b}_j^* \cdot \mathbf{b}_j^*} \mathbf{b}_j^* =$$

$$\sum_{j=i}^{n} \frac{(\mathbf{x} + \mathbf{y}) \cdot \mathbf{b}_j^*}{\mathbf{b}_j^* \cdot \mathbf{b}_j^*} \mathbf{b}_j^* = \pi_i(\mathbf{x} + \mathbf{y})$$

και

$$\pi_i(\lambda \mathbf{x}) = \lambda \pi_i(\mathbf{x}) \quad (\lambda \in \mathbb{R}).$$

Παράδειγμα 8.2.3. Έστω $\mathbf{b}_1 = (1, 0, 0)$, $\mathbf{b}_2 = (1, 1, 1), \mathbf{b}_3 = (1, 0, 2)$ και $\mathcal{L}$ το πλέγμα που παράγεται από αυτά. Θέλουμε να υπολογίζουμε μια βάση του $\pi(\mathcal{L})$. Μέτα από πράξεις βρίσκουμε $\mathbf{b}_1^* = (1, 0, 0) = \mathbf{b}_1$, $\mathbf{b}_2^* = (0, 1, 1) = \mathbf{b}_2 - \mathbf{b}_1$,

$$\mathbf{b}_3^* = (0, -1, 1) = \mathbf{b}_3 - \mathbf{b}_1 - \mathbf{b}_2^* = \mathbf{b}_3 - \mathbf{b}_2.$$

Έστω η προβολή

$$\pi : \mathcal{L} \to \text{span}(\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3)^{\perp}.$$

Ας είναι $\mathbf{x} = (3, 2, 0) \in \mathcal{L}$, τότε $\pi(\mathbf{x}) = 3\mathbf{b}_1^* + \mathbf{b}_2^* - \mathbf{b}_3^*$. Το σύνολο $\pi(L)$ είναι πλέγμα δίοτι αν $\mathbf{x}, \mathbf{y} \in \pi(\mathcal{L})$ τότε $\mathbf{x} - \mathbf{y} = \pi(\mathbf{a}) - \pi(\mathbf{b})$ για κάποια $\mathbf{a}, \mathbf{b}$ του $\mathcal{L}$. Οπότε, $\mathbf{x} - \mathbf{y} = \pi(\mathbf{a} - \mathbf{b}) \in \pi(\mathcal{L})$ και $\mathbf{0} \in \pi(\mathcal{L})$. Το πλέγμα που παράγεται από τα διανύσματα $\pi(\mathbf{b}_1), \pi(\mathbf{b}_2), \pi(\mathbf{b}_3)$ είναι το $\pi(\mathcal{L})$. Δηλ. $\pi(\mathcal{L}) = \text{span}_{\mathbb{Z}}\{\pi(\mathbf{b}_1), \pi(\mathbf{b}_2), \pi(\mathbf{b}_3)\}$. Επίσης $\pi(\mathbf{b}_1) = \mathbf{b}_1^*$,

$$\pi(\mathbf{b}_2) = \mathbf{b}_1^* + \mathbf{b}_2^* = (1, 1, 1)$$

και

$$\pi(\mathbf{b}_3) = \mathbf{b}_1^* + \frac{3}{2}\mathbf{b}_2^* + \mathbf{b}_3^* = (1, 1/2, 5/2).$$

Παρατηρούμε ότι το $\pi(\mathcal{L})$ είναι ρητό πλέγμα δηλ. είναι $\subset \mathbb{Q}^3$. Καταλήγουμε ότι

$$\pi(\mathcal{L}) = \text{span}_{\mathbb{Z}}\{(1, 0, 0), (1, 1, 1), (1, 1/2, 5/2)\}.$$

Τέλος ο όγκος του $\pi(L)$ είναι $vol(\pi(\mathcal{L})) = 2$.

Λήμμα 8.2.1. *Έστω $\mathcal{L}$ πλέγμα με* $\text{rank}(\mathcal{L}) = n$. *Τα σύνολα* $\pi_i(\mathcal{L})$ $(1 \leq i \leq n)$ *είναι πλέγματα με* $\text{rank}(\pi_i(\mathcal{L})) = n + 1 - i$ *και* $vol(\pi_i(\mathcal{L})) = \prod_{j=i}^{n} ||\mathbf{b}_j^*||$.

Απόδειξη. Αρχικά εφόσον $\mathcal{L}$ διακριτό σύνολο και οι προβολές $\pi_i(\mathcal{L})$ είναι διακριτά σύνολα. Θα αποδείξουμε ότι $\pi_i(\mathcal{L})$ είναι ομάδα. Έστω $\mathbf{x}, \mathbf{y} \in \pi_i(\mathcal{L})$. Τότε υπάρχουν $\mathbf{a}, \mathbf{b} \in \mathcal{L}$ τέτοια ώστε $\mathbf{x} = \pi_i(\mathbf{a})$ και $\mathbf{y} = \pi_i(\mathbf{b})$. Οπότε (δείτε την παρατήρηση 8.2.1)

$$\mathbf{x} - \mathbf{y} = \pi_i(\mathbf{a}) - \pi_i(\mathbf{b}) = \pi_i(\mathbf{a} - \mathbf{b}) \in \pi_i(\mathcal{L}).$$

Παρατηρούμε ότι τα διανύσματα $\pi_i(\mathbf{b}_j)$ για $j = 1, 2, ..., n$ παράγουν το $\pi_i(\mathcal{L})$. Αλλά, εξ ορισμού της π_i έχουμε $\pi_i(\mathbf{b}_j) = \mathbf{0}$ για $j < i$. Άρα το πλέγμα $\pi_i(\mathcal{L})$ παράγεται από τον σύνολο $\{\pi_i(\mathbf{b}_i), \pi_i(\mathbf{b}_{i+1}), ..., \pi_i(\mathbf{b}_n)\}$. Αυτό το σύνολο είναι και γραμμικά ανεξάρτητο. Τέλος επειδή

$$\text{span}\{\pi_i(\mathbf{b}_i), \pi_i(\mathbf{b}_{i+1}), ..., \pi_i(\mathbf{b}_n)\} = \text{span}\{\mathbf{b}_i^*, \mathbf{b}_{i+1}^*, ..., \mathbf{b}_n^*\}$$

και το σύνολο $\{\mathbf{b}_i^*, \mathbf{b}_{i+1}^*, ..., \mathbf{b}_n^*\}$ είναι γραμμικά ανεξάρτητο, προκύπτει ότι ο όγκος $vol(\pi_i(\mathcal{L})) = \prod_{j=i}^{n} ||\mathbf{b}_j^*||$. □

Πόρισμα 8.2.1. $vol(\mathcal{L}) = vol(\pi(\mathcal{L}))$ $\big(\pi(\mathcal{L}) = \pi_1(\mathcal{L})\big)$.

Άσκηση 8.7 Έστω $\mathbf{x} = (1, 1)$ και $\{\mathbf{b}_1 = (1, -1),\ \mathbf{b}_2 = (0, 2)\}$ μια βάση του πλέγματος $\mathcal{L}$. Να υπολογίσετε την προβολή $\pi(\mathbf{x})$. Τέλος, να βρείτε μια βάση του $\pi(\mathcal{L})$.

Άσκηση 8.8 Ν.α.ο.

$$||\pi_{i-1}(\mathbf{b}_i)||^2 = \frac{(\mathbf{b}_i \cdot \mathbf{b}_{i-1}^*)^2}{||\mathbf{b}_{i-1}^*||} + ||\mathbf{b}_i^*|| \ (i \geq 2).$$

Άσκηση 8.9 Έστω $\mathbf{x}$ τυχαίο διάνυσμα του $\mathbb{R}^n$ και $\mathbf{b}_1, ..., \mathbf{b}_n$, μια βάση του $\mathbb{R}^n$.
(i). Ν.α.ο.

$$||\pi_{i-1}(\mathbf{x})||^2 = ||\pi_i(\mathbf{x})||^2 + \frac{(\mathbf{x} \cdot \mathbf{b}_{i-1}^*)^2}{||\mathbf{b}_{i-1}^*||} \geq ||\pi_i(\mathbf{x})||^2 \ (i = 2, 3, ..., n).$$

(ii). Τι είδους μονοτονία έχει η ακολουθία $\{||\pi_i(\mathbf{x})||\}$; $(i = 1, 2, ..., n)$.

Παρατήρηση 8.2.2. (**i**). Έστω

$$\hat{B} = \begin{bmatrix} | & | & \cdots & | \\ \hat{\mathbf{b}}_1^T & \hat{\mathbf{b}}_2^T & \cdots & \hat{\mathbf{b}}_n^T \\ | & | & \cdots & | \end{bmatrix},$$

όπου $\hat{\mathbf{b}}_j = \frac{\mathbf{b}_j^*}{||\mathbf{b}_j^*||}$. Η διαδικασία κατασκευής των $\hat{\mathbf{b}}_j$ ονομάζεται *ορθοκανονικοποίηση κατά Gram-Schimdt.*
(**ii**). Ο πίνακας $\hat{B}$ έχει στήλες ανά δύο κάθετες και μοναδιαίες, επομένως είναι ορθογώνιος, $\hat{B} \in \mathbb{O}_n(\mathbb{R})$. Οπότε η γραμμική απεικόνιση $P : L \to \mathbb{R}^k,\ P(\mathbf{v}) = \mathbf{v}\hat{B}$ είναι ισομετρία. Επομένως, $\mathbf{a} \cdot \mathbf{b} = \mathbf{a}\hat{B} \cdot \mathbf{b}\hat{B}$.

(**iii**). Ο πίνακας $B^*\hat{B}$ είναι διαγώνιος $k \times k$ πίνακας που έχει στην κύρια διαγώνιο τα στοιχεία $||\mathbf{b}_i^*||$. Επομένως,

$$\det[B^*\hat{B}] = \prod_{j=1}^{k} ||\mathbf{b}_j^*||.$$

Πράγματι,

$$\begin{bmatrix} -\mathbf{b}_1^*- \\ -\mathbf{b}_2^*- \\ \vdots \\ -\mathbf{b}_k^*- \end{bmatrix} \cdot \begin{bmatrix} | & | & \cdots & | \\ \frac{\mathbf{b}_1^*}{||\mathbf{b}_1^*||} & \frac{\mathbf{b}_2^*}{||\mathbf{b}_2^*||} & \cdots & \frac{\mathbf{b}_k^*}{||\mathbf{b}_k^*||} \\ | & | & \cdots & | \end{bmatrix} = \left[\frac{\mathbf{b}_i^* \cdot \mathbf{b}_j^*}{||\mathbf{b}_j^*||} \right]_{1 \le i \le j \le k}$$

Από την τελευταία παρατήρηση και το γεγονός ότι $vol(\mathcal{P}(B)) = vol(\mathcal{P}(B^*))$, προκύπτει το επόμενο.

Λήμμα 8.2.2. *Ο όγκος του πλέγματος L ισούται με τον θετικό πραγματικό αριθμό* $\det(B^*\hat{B}) = \prod_{i=1}^{k} ||\mathbf{b}_i^*||$.

Παρατήρηση 8.2.3. Ο όγκος του πλέγματος L είναι ανεξάρτητος του πίνακα βάσης B.

Παράδειγμα 8.2.4. Έστω $L = L(\mathbf{b}_1, \mathbf{b}_2)$, όπου $\mathbf{b}_1 = (1, 1, -2)$, $\mathbf{b}_2 = (2, 1, 0)$. Έχουμε,

$$\hat{B} = \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{3/2}{\sqrt{7/2}} \\ \frac{1}{\sqrt{6}} & \frac{1/2}{\sqrt{7/2}} \\ \frac{-2}{\sqrt{6}} & \frac{1}{\sqrt{7/2}} \end{bmatrix}.$$

Τέλος, $vol(L) = \det[B^*\hat{B}] = \sqrt{21}$.

check

Παρατήρηση 8.2.4. Αν $\{\mathbf{b}_1, ..., \mathbf{b}_n\}$ γραμμικά ανεξάρτητα και $\{\mathbf{b}_1^*, ..., \mathbf{b}_n^*\}$ η Gram-Schmidt βάση, τότε $||\mathbf{b}_i^*|| \le ||\mathbf{b}_i||$ και $\mathbf{b}_i \cdot \mathbf{b}_i^* = \mathbf{b}_i^* \cdot \mathbf{b}_i^* \quad (i \ge 1)$.

Άσκηση 8.10 Να αποδείξετε την παρατήρηση 8.2.3.

Άσκηση 8.11 Να αποδείξετε ότι $vol(A_n) = \sqrt{n+1}$ όπου $A_n \subset \mathbb{Z}^{n+1}$ το οποίο ορίζεται ως

$$A_n = \{(x_0, x_1, ..., x_n) \in \mathbb{Z}^{n+1} : \sum_{i=0}^{n} x_i = 0\}.$$

Άσκηση 8.12 Να αποδείξετε ότι $vol(D_n) = 2$ όπου $D_n \subset \mathbb{Z}^n$ που ορίζεται

$$D_n = \{(x_1, ..., x_n) \in \mathbb{Z}^n : \sum_{i=0}^{n} x_i = 0 \bmod 2\}.$$

8.3 Θεωρήματα του Minkowski

Η εύρεση ενός διανύσματος με το μικρότερο μήκος σε ένα πλέγμα L ονομάζεται *πρόβλημα του μικρότερου διανύσματος* ή SVP : **S**hortest **V**ector **P**roblem και είναι *δύσκολο* πρόβλημα. Ειδικότερα ο Ajtai απέδειξε ότι είναι NP-hard υπό τυχαίες αναγωγές. Επίσης, ο Miccianchio απέδειξε ότι το SVP_γ είναι NP-hard για $\gamma = \sqrt{2}$ (υπό τυχαίες αναγωγές). Αν

$$\lambda_1(L) = \inf_{L \ni \mathbf{x} \neq \mathbf{0}} \{||\mathbf{x}||\},$$

τότε το SVP αναζητά διανύσματα του πλέγματος L με μέτρο $\lambda_1(L)$. Αρχικά θα δείξουμε ότι το $\lambda_1(L)$ υπάρχει.

Θεώρημα 8.3.1 Υπάρχει διάνυσμα $\mathbf{z} \in L$ τέτοιο ώστε $||\mathbf{z}|| = \lambda_1(L)$.

Απόδειξη. Από την χαρακτηριστική ιδιότητα του inf υπάρχει ακολουθία $(\mathbf{z}_n)_n \in L$ τέτοια ώστε $||\mathbf{z}_n|| \to \lambda_1$. Εφόσον η ακολουθία συγκλίνει υπάρχει θετικός αριθμός δ τέτοιος ώστε $||\mathbf{z}_i - \mathbf{z}_j|| \leq \delta\lambda_1$, για όλα τα i, j. Άρα,

$$\mathbf{z}_i \in \overline{B}(\delta\lambda_1) = \{\mathbf{x} \in \mathbb{R}^n : ||\mathbf{x}|| \leq \delta\lambda_1\}.$$

Επομένως, υπάρχει υπακολουθία $\mathbf{z}_{ij}$ της ακολουθίας $\mathbf{z}_n$ που συγκλίνει, έστω στο διάνυσμα $\mathbf{z}$. Τότε, για κάθε $\varepsilon > 0$ υπάρχει $j_0 > 0$ τέτοιο ώστε $||\mathbf{z}_{ij} - \mathbf{z}|| < \varepsilon/\delta$ για όλα τα $j > j_0$. Επόμενως για $j, v > j_0$ έχουμε

$$||\mathbf{z}_{ij} - \mathbf{z}_{iv}|| = ||(\mathbf{z}_{ij} - \mathbf{z}) + (\mathbf{z} - \mathbf{z}_{iv})|| \leq 2\varepsilon/\delta.$$

Διαλέγοντας, αντί ε το $\varepsilon\delta/2$ προκύπτει $||\mathbf{z}_{ij} - \mathbf{z}_{iv}|| < \varepsilon$. Εφόσον τα πλέγματα είναι διακριτές ομάδες, προκύπτει $\mathbf{z}_{ij} = \mathbf{z}_{iv} = \mathbf{z} \in L$. Άρα το ζητούμενο διάνυσμα είναι το $\mathbf{z}$. □

Παρατήρηση 8.3.1. Ο αριθμός $\lambda_1(L)$ δεν εξαρτάται από την επιλογή βάσης του πλέγματος. Επίσης, ονομάζεται first successive minima.

Θεώρημα 8.3.2 (1ο θεώρημα του Minkowski). Έστω L ένα πλέγμα διάστασης n και $R > 0$. Αν V_n ο όγκος της μοναδιαίας $n-$διάστατης σφαίρας του $\mathbb{R}^n$ και $V_n R^n > \det L$, τότε υπάρχει $\mathbf{x} \in L - \{\mathbf{0}\}$ με $||\mathbf{x}|| \leq 2R$.

Απόδειξη. Έστω $f : \mathbb{R}^n \to \mathbb{R}^n/L$,

$$f(\mathbf{x}) = \mathbf{x} \pmod{L}.$$

Η συνάρτηση αυτή είναι επί. Θεωρούμε τον περιορισμό,

$$rest_{\overline{B}_n(R)} f : \overline{B}_n(R) \to \mathbb{R}^n/L,$$

όπου $\overline{B}_n(R)$ η σφαίρα ακτίνας R του $\mathbb{R}^n$. Ο περιορισμός δεν είναι 1-1, λόγο της υπόθεσης με τον όγκο. Πράγματι αν ήταν 1-1, τότε

$$vol(\overline{B}_n(R)) = V_n R^n \leq vol(\mathbb{R}^n/L) = \det L,$$

το οποίο αντίκειται στην υπόθεση του θεωρήματος. Επομένως, υπάρχουν διανύσματα $\mathbf{x}, \mathbf{y}$ του $\overline{B}_n$ διαφορετικά μεταξύ τους, τέτοια ώστε $f(\mathbf{x}) = f(\mathbf{y})$. Επομένως, $\mathbf{x} - \mathbf{y} \in L$ άρα, το διάνυσμα $\mathbf{v} = \mathbf{x} - \mathbf{y}$ έχει μέτρο $||\mathbf{v}|| \leq ||\mathbf{x}|| + ||\mathbf{y}|| \leq 2R$. □

Παρατήρηση 8.3.2. Ο όγκος της n-διάστατης μοναδιαίας σφαίρας,

$$\overline{B}_n = \{\mathbf{x} \in \mathbb{R}^n : ||\mathbf{x}|| \leq 1\}$$

είναι

$$V_n = \frac{\pi^{n/2}}{\Gamma(\frac{n}{2}+1)} \approx \left(\frac{2\pi e}{n}\right)^{n/2} \frac{1}{\sqrt{\pi n}},$$

όπου

$$\Gamma(z) = \int_0^\infty x^{z-1} e^{-x}\, dx,$$

για $Re(z) > 0$. Ισχύει

$$\Gamma(n) = (n-1)! \ (n \in \mathbb{Z}_{\geq 1}). \tag{8.3.1}$$

Αυτή η ισότητα μας επιτρέπει να επεκτείνουμε το σύμβολο ! σε πραγματικούς αριθμούς. Π.χ. $(\frac{1}{2})! = \Gamma(1/2)$.

Ενώ, της n-διάστατης σφαίρας ακτίνας R,

$$V_n(R) = vol(\overline{B}_n(R)) = V_n R^n.$$

Επίσης η ακτίνα $R_n(V)$ μιας σφαίρας όγκου V, ισούται με

$$R_n(V) = \frac{\left(\Gamma(\frac{n}{2}+1)V\right)^{1/n}}{\sqrt{\pi}}.$$

Το θεώρημα 8.3.2 του Minkowski μας λέει ότι μια σφαίρα με ακτίνα τουλάχιστον

$$R_0 = 2\Big(\frac{\det L}{V_n}\Big)^{1/n} = \frac{2(\det L)^{1/n}\Gamma(\frac{n}{2}+1)^{1/n}}{\sqrt{\pi}} \tag{8.3.2}$$

περιέχει ένα μη μηδενικό σημείο του L.

Δίνουμε τον παρακάτω ορισμό.

Ορισμός 8.3.1. *Συμβολίζουμε με GH : **G**aussian **H**euristic, τον θετικό πραγματικό αριθμό*

$$GH(L) = \Big(\frac{\det L}{V_n}\Big)^{1/n} = \frac{(\det L)^{1/n}\Gamma(\frac{n}{2}+1)^{1/n}}{\sqrt{\pi}} \approx \sqrt{\frac{n}{\pi e}}(\det(L))^{1/n},$$

όπου $n = rank(L)$.

Πόρισμα 8.3.1. $\lambda_1(L) < 2GH(L)$.

Απόδειξη. Για κάθε ακτίνα R με $GH(L) < R < GH(L) + \varepsilon$ (για κάθε $\varepsilon > 0$) η προυπόθεση του θεωρήματος 8.3.2 του Minkowski ισχύει. Άρα υπάρχει μη-μηδενικό διάνυσμα $\mathbf{x}$ του πλέγματος L, τέτοιο ώστε, $||\mathbf{x}|| \leq 2R < 2(GH(L) + \varepsilon)$ (για κάθε $\varepsilon > 0$). Το ζητούμενο έπεται. □

Δίνουμε τον ορισμό μιας πολύ βασικής ευρετικής υπόθεσης που ισχύει σε τυχαία (ακέραια) πλέγματα.

Ορισμός 8.3.2. *(Ευρετική του Gauss). Σε τυχαίο ακέραιο πλέγμα L τάξης n και διάστασης m (δηλ. $L \subset \mathbb{Z}^m$) υποθέτουμε ότι ισχύει*

$$\lambda_1(L) \approx GH(L).$$

Αυτή η ευρετική δεν ισχύει για όλα τα ακέραια πλέγματα[36]. Με άλλα λόγια η ευρετική του Gauss προτείνει ότι, για τυχαία ακέραια πλέγματα, η ελάχιστη ακτίνα που χρειάζεται να έχει μια σφαίρα με κέντρο την αρχή, ώστε να περιέχει ένα μη μηδενικό σημείο του πλέγματος, είναι $R_0/2$ αντί R_0 στον τύπο (8.3.2).

Στην βιβλιογραφία συνήθως η ευρετική του Gauss δίνεται ως εξής.

Gauss Heuristic. *Αν έχουμε ένα πλέγμα L και ένα σύνολο S τότε το πλήθος των σημείων $S \cap L$ είναι προσεγγιστικά ίσο με $vol(S)/vol(L)$.*

Παρατήρηση 8.3.3. Αν ισχύει η ευρετική του Gauss σε ένα πλέγμα L, τότε η ελάχιστη ακτίνα R ώστε

$$|L \cap \overline{B}_n(R)| = 1 \approx \frac{vol(\overline{B}_n(R))}{\det L} = \frac{R^n V_n}{\det L}.$$

είναι η $R = GH(L)$.

Υπάρχουν περιπτώσεις που η ευρετική του Gauss δεν ισχύει. Για παράδειγμα αν θεωρήσουμε το πλέγμα $\mathbb{Z}^n$ και την *μπάλα* διάστασης n και ακτίνας $R_n = \alpha\sqrt{n}$ και $0 < \alpha < n^{-\varepsilon}$ γιά κάποιο $\varepsilon > 0$. Τότε, καθώς το n αυξάνει αποδυκνείεται ότι η ευρετική του Gauss δεν ισχύει[37].

Άσκηση 8.13 Να αποδείξετε την ισότητα 8.3.1.

Άσκηση 8.14 Να αποδείξετε ότι

$$\left(\frac{1}{2}\right)! = \frac{\sqrt{\pi}}{2}.$$

[36] J. E. Mazo and A. M. Odlyzko, Lattice points in high dimensional spheres (1990), *Monats. Math.* vol. 17

[37] Mazo, Odlyzko, Lattice points in high dimensional spheres. Monatsheft Mathematik, p. 17–47, 1990.

Άσκηση 8.15 (Complement formula). Να αποδειξετε ότι

$$\Gamma(s)\Gamma(1-s) = \frac{\pi}{\sin(\pi s)}.$$

Υπόδειξη.

$$J = \Gamma(s)\Gamma(1-s) = \int_{y=0}^{\infty} e^{-y}\Big[\int_{x=0}^{\infty} e^{-x}\big(\frac{x}{y}\big)^{s-1}\frac{dx}{y}\Big]\, dy.$$

Συνεχίστε με αντικατάσταση $u = x/y$. Μετά από πράξεις

$$J = \int_{u=0}^{\infty}\frac{u^{s-1}}{u+1}\, du = \int_{u=0}^{1}\frac{u^{s-1}}{u+1}\, du + \int_{v=0}^{1}\frac{v^{-s}}{v+1}\, dv.$$

Κατόπιν αναλύστε την γεωμετρική σειρά και χρησιμοποιήστε

$$\frac{\pi}{\sin(\pi s)} = \frac{1}{s} - \sum_{n=1}^{\infty}\frac{2s}{n^2-s^2}.$$

Μια άλλη βασική παράμετρος που αντιστοιχίζεται σε μια βάση B ενός πλέγματος L τάξης n, είναι ο παράγοντας Hermite (HF : **H**ermite **F**actor).

$$HF(L,B) = \frac{||\mathbf{b}_1||}{(\det L)^{1/n}}.$$

Ο Hermite factor εξαρτάται μόνο από την βάση Gram-Schimdt, διότι $\mathbf{b}_1 = \mathbf{b}_1^*$ και $\det L = \prod ||\mathbf{b}_i^*||$. Επομένως, μπορούμε να γράψουμε $HF(L,B) = HF(B^*)$, όπου B^* η GS-βάση του B. Ο Hermite το 1850 απόδειξε ότι

$$\frac{||\mathbf{b}_1||}{(\det L)^{1/n}} = f(n),$$

για κάποια συνάρτηση f. Οπότε και ο λόγος

$$\frac{\lambda_1(L)}{(\det L)^{1/n}},$$

εξαρτάται μόνο από την τάξη του πλέγματος n.

Ορισμός 8.3.3. *(Σταθερά του Hermite). Ορίζουμε την σταθερά του Hermite διάστασης n,*

$$\gamma_n = \sup_{L\subset\mathbb{Z}^n, rank(L)=n}\Big(\frac{\lambda_1(L)}{(\det L)^{1/n}}\Big)^2.$$

Λήμμα 8.3.1. *Ισχύει, $\sqrt{\gamma_n} \leq \frac{2}{V_n^{1/n}}$.*

Απόδειξη. Από το πόρισμα 8.3.1, έχουμε

$$\lambda_1(L) \leq 2GH(L) = 2\frac{(\det L)^{1/n}}{V_n^{1/n}}.$$

Επομένως,

$$\sqrt{\gamma_n} \leq \frac{2\frac{(\det L)^{1/n}}{V_n^{1/n}}}{(\det L)^{1/n}} = \frac{2}{V_n^{1/n}}$$

□

Παρατήρηση 8.3.4. Μέχρι στιγμής έχουμε τρεις βασικές ποσότητες ανεξάρτητες της βάσης του πλέγματος. Την ποσότητα $\lambda_1(L)$, τον όγκο $\det(L)$ και την σταθερά Hermite $\gamma_n(L)$. Η τελευταία ποσότητα είναι και ανεξάρτητη του πλέγματος, δηλ. $\gamma_n(L) = \gamma_n$.

Παρατήρηση 8.3.5. Ο Lagrange απόδειξε ότι $\gamma_2 = \sqrt{\frac{4}{3}}$. Επίσης οι Korkine-Zolotarev απέδειξαν ότι $\gamma_4 = \sqrt[4]{4}, \gamma_5 = \sqrt[5]{8}$. Ενώ ο Hofreiter (καθώς και ο Blichfeld) απόδειξαν ότι $\gamma_6 = \sqrt[6]{\frac{64}{3}}$. Ο Blichfeld απόδειξε $\gamma_7 = \sqrt[7]{64}, \gamma_8 = 2$. Η σταθερά του Hermite ορίστηκε αρχικά από τον Hermite για θετικά ορισμένες τετραγωνικές μορφές, δηλ. πολυώνυμα της μορφής

$$q(x_1, ..., x_n) = \sum_{1 \leq i,j \leq n} a_{ij} x_i x_j \ (a_{ij} = a_{ji} \in \mathbb{Z}),$$

με $q(\mathbf{x}) \geq 0$ για κάθε $\mathbf{x} \in \mathbb{Z}^n$. Θέτουμε $\Delta_q = \det([a_{ij}])$. Ο Hermite απόδειξε ότι

$$||q|| = \min_q \{q(\mathbf{x}) : \mathbf{x} \in \mathbb{Z}^n - \{\mathbf{0}\}\} \leq \Delta_q^{1/n} \sqrt{\frac{4}{3}}^{n-1}.$$

Επομένως, ο λόγος $||q||/\Delta_q^{1/n}$ εξαρτάται μόνο από την διάσταση n. Έτσι ο Hermite όρισε

$$\gamma_n = \sup_q \frac{||q||}{\Delta_q^{1/n}}.$$

Πρόταση 8.3.1. *(Ανισότητα του Hermite).*

$$\gamma_n < \gamma_2^{(n-1)/2} = \left(\frac{4}{3}\right)^{(n-1)/2} (n > 2).$$

Απόδειξη. [8, Θεώρημα 7.5] □

Μια αλγοριθμική μορφή της προηγούμενης πρότασης είναι ο αλγόριθμος LLL.

Παρατήρηση 8.3.6. Υπάρχουν και γραμμικά (άνω) φράγματα για την σταθερά του Hermite. Π.χ. για $n \geq 2$ ισχύει $\gamma_n \leq \frac{2n}{3}$.

Η ανισότητα του Hermite γενικεύτηκε από τον Mordell.

Πρόταση 8.3.2. *(Ανισότητα του Mordell).*

$$\sqrt{\gamma_n} < \sqrt{\gamma_k}^{(n-1)/(k-1)} \ (k < n).$$

Μια αλγοριθμική μορφή της προηγούμενης πρότασης είναι ο αλγόριθμος BKZ. Εφόσον

$$\frac{||\mathbf{b}_1||}{(\det L)^{1/n}} = O(\delta^n),$$

ορίζουμε Root Hermite Factor : RHF(L,B) τον πραγματικό αριθμό δ έτσι ώστε

$$||\mathbf{b}_1|| = \delta^n (\det L)^{1/n}.$$

Δηλ. $\delta^n = HF(L, B)$. Όπως είδαμε $\delta \geq 1$. Γενικά, θα δούμε παρακάτω ότι ένας αλγόριθμος που βρίσκει μια νέα βάση, είναι καλύτερος όταν κατά μέσο όρο τα δ που παράγει είναι πιο κοντά στη μονάδα.

8.3.1 GSA : Geometric Series Assumption

Μια άλλη βασική υπόθεση που κάνουμε πολύ συχνά στα πλέγματα είναι η GSA. Πρώτη φορά δόθηκε από τους Schnor-Euchner.

Ορισμός 8.3.4. *Λέμε ότι ισχύει η GSA σε ένα πλέγμα L για την βάση B αν*

$$\frac{||\mathbf{b}_i^*||}{||\mathbf{b}_1||} = r^{i-1}$$

για κάποιο $r \in (0, 1)$. Το r ονομάζεται και συντελεστής GSA.

Παρατήρηση 8.3.7. Δηλαδή, η ακολουθία

$$(||\mathbf{b}_1^*||, ||\mathbf{b}_2^*||, \ldots, ||\mathbf{b}_n^*||)$$

είναι μια γεωμετρική πρόοδος με λόγο $r < 1$.

Ο συντελεστής GSA και ο RHF συνδέονται μεταξύ τους.

Λήμμα 8.3.2. *Έστω ένα πλέγμα L και για μια βάση του B υπάρχει ο Root Hermite Factor δ. Τότε ο συντελεστής GSA της βάσης B είναι $r \approx \delta^{-2}$.*

Απόδειξη. Από την *GSA* έχουμε $||\mathbf{b}_i^*|| = r^{i-1}||\mathbf{b}_1||$. Οπότε

$$\det L = ||\mathbf{b}_n^*|| \cdots ||\mathbf{b}_1^*|| =$$

$$r^{n-1} \cdots 1 \cdot ||\mathbf{b}_1||^n =$$

$$r^{(n-1)n/2} ||\mathbf{b}_1||^n.$$

Αλλά, $||\mathbf{b}_1|| = \delta^n (\det L)^{1/n}$, επομένως

$$\det L = r^{(n-1)n/2} \delta^{n^2} \det L.$$

Τέλος,

$$r = \delta^{-2n^2/(n(n-1))} \approx \delta^{-2}.$$

□

Πόρισμα 8.3.2. *Αν έχουμε ένα πλέγμα με βάση B που έχει Root Hermite Factor δ και ισχύει η GSA, τότε*

$$||\mathbf{b}_i^*|| \approx \delta^{n-2(i-1)}(\det L)^{1/n} \quad (i \geq 1).$$

Μια εφαρμόγη του πορίσματος είναι στους αλγορίθμους απαρίθμησης για να κάνουμε πρόβλεψη των $||\mathbf{b}_i^*||$ ξεκινώντας από μια βάση με Root Hermite Factor δ.

8.4 Αλγόριθμος των Gauss-Lagrange

Έστω $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^2$ γραμμικά ανεξάρτητα και L το πλέγμα που παράγεται από αυτά.

Ορισμός 8.4.1. *Μια βάση $\mathbf{b}_1, \mathbf{b}_2$ του L ονομάζεται ανηγμένη κατά Gauss-Lagrange αν-ν*

$$||\mathbf{b}_1|| \leq ||\mathbf{b}_2|| \leq ||\mathbf{b}_2 + q\mathbf{b}_1|| \tag{8.4.1}$$

για κάθε $q \in \mathbb{Z}$.

Παρατήρηση 8.4.1. Ισοδύναμα ο ορισμός ισχύει αν στο δεύτερο μέλος της ανισότητας έχουμε $||\mathbf{b}_2 - q\mathbf{b}_1||$.

Θεώρημα 8.4.1 Αν η διατεταγμένη βάση $\{\mathbf{b}_1, \mathbf{b}_2\}$ του $L = L(\mathbf{b}_1, \mathbf{b}_2)$ είναι ανηγμένη κατά Gauss-Lagrange τότε $||\mathbf{b}_i|| = \lambda_i(L)$.

Απόδειξη. Έστω $\mathbf{v} = m_1\mathbf{b}_1 + m_2\mathbf{b}_2$. Αν $m_2 = 0$ τότε $||\mathbf{v}|| = |m_1|||\mathbf{b}_1|| \geq ||\mathbf{b}_1||$. Επομένως, $\mathbf{b}_1 = \lambda_1(L)$. Επίσης $||\mathbf{b}_2|| = \lambda_2(L)$. Πράγματι, αν υπάρχει $\mathbf{b}'$ γραμμικά ανεξάρτητο από το $\mathbf{b}_1$ τ.ω. $||\mathbf{b}_1|| \leq ||\mathbf{b}'|| < ||\mathbf{b}_2||$, τότε $\mathbf{b}' = k\mathbf{b}_2$, για κάποιο $k \in \mathbb{Z}$. Τότε όμως $||\mathbf{b}'|| \geq ||\mathbf{b_2}||$. Άτοπο. Άρα $||\mathbf{b}_2|| = \lambda_2(L)$. Έστω $m_2 \neq 0$. Τότε από την Ευκλείδεια διαίρεση υπάρχουν ακέραιοι a, b τ.ω. $m_2 = m_1 a + b$, $0 \leq b < |m_2|$. Θα δείξουμε ότι $||\mathbf{v}|| > ||\mathbf{b}_2||$. Τότε, τελειώσαμε. Έχουμε,

$$\mathbf{v} = b\mathbf{b}_1 + m_2(\mathbf{b}_2 + a\mathbf{b}_1).$$

Επομένως από τριγωνική,

$$||\mathbf{v}|| \geq |m_2|||\mathbf{b}_2 + a\mathbf{b}_1|| - b||\mathbf{b}_1|| =$$

$$-b||\mathbf{b}_1|| + |m_2|||\mathbf{b}_2 + a\mathbf{b}_1|| + b||\mathbf{b}_2 + a\mathbf{b}_1|| - b||\mathbf{b}_2 + a\mathbf{b}_1|| =$$

$$(|m_2| - b)||\mathbf{b}_2 + a\mathbf{b}_1|| + b(||\mathbf{b}_2 + a\mathbf{b}_1|| - ||\mathbf{b}_1||).$$

Εφόσον η βάση είναι ανηγμένη $||\mathbf{b}_2 + a\mathbf{b}_1|| - ||\mathbf{b}_1|| \geq 0$, και $|m_2| - b > 0$, έχουμε:

$$(|m_2| - b)||\mathbf{b}_2 + a\mathbf{b}_1|| + b(||\mathbf{b}_2 + a\mathbf{b}_1|| - ||\mathbf{b}_1||) \geq (|m_2| - b)||\mathbf{b}_2 + a\mathbf{b}_1|| >$$

$$||\mathbf{b}_2 + a\mathbf{b}_1||.$$

Ξανά, αφού η βάση είναι ανηγμένη $||\mathbf{b}_2 + a\mathbf{b}_1|| \geq ||\mathbf{b}_2||$, και προκύπτει τελικά ότι

$$||\mathbf{v}|| > ||\mathbf{b}_2||.$$

□

Ισχύει το ακόλουθο λήμμα.

Λήμμα 8.4.1. *Αν για την βάση* $\{\mathbf{b}_1, \mathbf{b}_2\}$ *του* L *ισχύει*

$$||\mathbf{b}_1|| \leq ||\mathbf{b}_2|| \leq ||\mathbf{b}_1 + \mathbf{b}_2||, ||\mathbf{b}_1 - \mathbf{b}_2||$$

τότε,
(**i**). $|\cos\theta| \leq \frac{||\mathbf{b}_2||}{2||\mathbf{b}_1||}$, *όπου* θ *η γωνία των* $\mathbf{b}_1, \mathbf{b}_2$.
(**ii**). $\frac{|\mathbf{b}_1 \cdot \mathbf{b}_2|}{||\mathbf{b}_1||^2} \leq \frac{1}{2}$.

Απόδειξη. Θέτω $B_1 = ||\mathbf{b}_1||^2, B_2 = ||\mathbf{b}_2||^2$.
(**i**). Από την υπόθεση

$$B_1 \leq B_1 \pm 2\mathbf{b}_1 \cdot \mathbf{b}_2 + B_2.$$

Ισοδύναμα $0 \leq B_2 \pm 2\mathbf{b}_1 \cdot \mathbf{b}_2$ επομένως $\pm 2\mathbf{b}_1 \cdot \mathbf{b}_2 \leq B_2$ ή $\pm 2||\mathbf{b}_1||\cos\theta \leq ||\mathbf{b}_2||$. Το ζητούμενο έπεται.
(**ii**). Δουλεύοντας παρόμοια με την σχέση

$$B_2 \leq B_1 \pm 2\mathbf{b}_1 \cdot \mathbf{b}_2 + B_2$$

καταλήγουμε στην ανισότητα

$$|\cos\theta| \leq \frac{||\mathbf{b}_1||}{2||\mathbf{b}_2||}.$$

Ισχύει :

$$\frac{|\mathbf{b}_1 \cdot \mathbf{b}_2|}{||\mathbf{b}_1||^2} = \frac{||\mathbf{b}_2||}{||\mathbf{b}_1||} \cdot |\cos\theta| \leq \frac{||\mathbf{b}_2||}{||\mathbf{b}_1||} \frac{||\mathbf{b}_1||}{2||\mathbf{b}_2||} = \frac{1}{2}$$

□

Παρατήρηση 8.4.2. Η γωνία μεταξύ των δύο διανυσμάτων $\mathbf{b}_1$ και $\mathbf{b}_2$ του προηγούμενου λήμματος ανήκει στο διάστημα $(\pi/3, 2\pi/3)$. Δηλ. διαφέρουν το πολύ κατά 30° για να γίνουν κάθετα.

Λήμμα 8.4.2. *Έστω* $\mathbf{b}_1, \mathbf{b}_2$ *μια διατεταγμένη βάση ενός πλέγματος* L. *Αν* $g(x) = ||\mathbf{b}_2 - x\mathbf{b}_1||$ *και* $g(0) \leq g(1), g(-1)$ *τότε,*

$$g(q) = ||\mathbf{b}_2 - q\mathbf{b}_1|| < ||\mathbf{b}_2 - (q+1)\mathbf{b}_1|| = g(q+1)$$

για κάθε ακέραιο $q > 1$ *και*

$$g(q) = ||\mathbf{b}_2 - q\mathbf{b}_1|| < ||\mathbf{b}_2 - (q-1)\mathbf{b}_1|| = g(q-1)$$

για κάθε ακέραιο $q < -1$.

Απόδειξη. Θέτω $f(x) = g(x)^2$. Υποθέτω αρχικά ότι $q \geq 0$. Για $q = 0$ από την υπόθεση έχω $f(0) \leq f(1) = f(q+1)$. Έστω ότι ισχύει $f(q-1) \leq f(q)$ (για κάποιο $q > 1$) και θ.α.ο. $f(q) < f(q+1)$. Θεωρούμε το τρίγωνο με κορυφές $\{\mathbf{b}_2, (q-1)\mathbf{b}_1, (q+1)\mathbf{b}_1\}$. Τότε $g(q)$ είναι το μήκος της διαμέσου που αντιστοιχεί στην πλευρά με κορυφές $(q-1)\mathbf{b}_1$ και $(q+1)\mathbf{b}_1$. Οπότε από το θεώρημα των διαμέσων έχω

$$4f(q) = 2f(q-1) + 2f(q+1) - 4||\mathbf{b}_1||^2.$$

Επομένως, $2f(q) \leq f(q) + f(q+1) - 2||\mathbf{b}_1||^2$. Άρα

$$f(q) \leq f(q+1) - 2||\mathbf{b}_1||^2 < f(q+1).$$

Άρα ισχύει για κάθε $q > 1$.

Έστω ότι $q < -1$. Τότε από την υπόθεση του θεωρήματος έχω $f(0) < f(-1)$. Θ.α.ο. $f(q) < f(q-1)$ για κάθε $q < -1$. Για $q = 0$ από την υπόθεση έχω $f(0) \leq f(-1)$. Υποθέτω ότι $f(q+1) \leq f(q)$ και θ.α.ο. $f(q) \leq f(q-1)$. Στην πραγματικότητα θα δείξω ότι για $q < -1$ έχω αυστηρά ανισότητα. Το ίσον ισχύει μόνο για $q = -1$. Θεωρούμε το τρίγωνο με κορυφές $\{\mathbf{b}_2, (q+1)\mathbf{b}_1, (q-1)\mathbf{b}_1\}$. Τότε $g(q)$ είναι το μήκος της διαμέσου που αντιστοιχεί στην πλευρά με κορυφές $(q+1)\mathbf{b}_1$ και $(q-1)\mathbf{b}_1$. Οπότε από το θεώρημα των διαμέσων έχω

$$4f(q) = 2f(q+1) + 2f(q-1) - 4||\mathbf{b}_1||^2.$$

Επομένως, $2f(q) = f(q) + f(q-1) - 2||\mathbf{b}_1||^2$. Άρα

$$f(q) = f(q-1) - 2||\mathbf{b}_1||^2 < f(q-1)$$

(αυστηρή ανισότητα δίοτι $\mathbf{b}_1 \neq \mathbf{0}$ ως διάνυσμα βάσης). Άρα ισχύει για κάθε $q < -1$. Το ζητούμενο έπεται.

□

Πόρισμα 8.4.1. *Η διατεταγμένη βάση $\{\mathbf{b}_1, \mathbf{b}_2\}$ του πλέγματος $L = L(\mathbf{b}_1, \mathbf{b}_2)$ είναι ανηγμένη κατά Gauss-Lagrange αν-ν*

$$||\mathbf{b}_1|| \leq ||\mathbf{b}_2|| \leq ||\mathbf{b}_1 + \mathbf{b}_2||, ||\mathbf{b}_1 - \mathbf{b}_2||. \tag{8.4.2}$$

Απόδειξη. ($\Rightarrow$) Για $q = \pm 1$ στον ορισμό 8.4.1, προκύπτει άμεσα.
($\Leftarrow$) Αν θέσουμε $g(x) = ||\mathbf{b}_2 - x\mathbf{b}_1||$ τότε $g(0) \leq g(1), g(-1)$ επομένως οι προϋποθέσεις του προηγούμενου λήμματος ικανοποιούνται. Συνεπώς, $g(0) < g(q)$ για κάθε ακέραιο $|q| > 1$ (για $q = -1, 0, 1$ ισχύει από την υπόθεση). Επομένως, $||\mathbf{b}_2|| \leq ||\mathbf{b}_2 - q\mathbf{b}_1||$ για κάθε ακέραιο q. Άρα η βάση είναι ανηγμένη κατά Gauss-Lagrange. □

Λήμμα 8.4.3. *Ας είναι $\{\mathbf{b}_1, \mathbf{b}_2\}$ μια βάση του πλέγματος $L = (\mathbf{b}_1, \mathbf{b}_2)$. Τότε*

$$\min\{||\mathbf{b}_2 - x\mathbf{b}_1|| : x \in \mathbf{R}\} = ||\mathbf{b}_2 - \mu\mathbf{b}_1||,$$

όπου

$$\mu = \frac{\mathbf{b}_1 \cdot \mathbf{b}_2}{||\mathbf{b}_1||^2}.$$

Απόδειξη. Θέτουμε, $B_i = ||\mathbf{b}_i||^2$, $(i = 1, 2)$. Έστω $f(x) = ||\mathbf{b}_2 - x\mathbf{b}_1||^2 = B_1x^2 - 2(\mathbf{b}_1 \cdot \mathbf{b}_2)x + B_2$. Το σημείο ελαχίστου είναι $\mu = \frac{\mathbf{b}_1 \cdot \mathbf{b}_2}{B_1}$. □

Παρατήρηση 8.4.3. (**i**). $f(\mu) = -\frac{(\mathbf{b}_1 \cdot \mathbf{b}_2)^2}{B_1} + B_2 = B_2(\sin\theta)^2$.
(**ii**). Ο συμβολισμός για την συνάρτηση που είναι πλησιέστερα σε ένα ακέραιο είναι

$$\lfloor x \rceil = \lfloor x + \frac{1}{2} \rfloor, \lceil x \rfloor = \lceil x - \frac{1}{2} \rceil$$

Έτσι, $\lfloor 0.5 \rceil = 1$, ενώ $\lceil 0.5 \rfloor = 0$.

Λήμμα 8.4.4. *Ας είναι* $\{\mathbf{b}_1, \mathbf{b}_2\}$ *μια βάση του πλέγματος* $L = (\mathbf{b}_1, \mathbf{b}_2)$. *Τότε*

$$\min\{||\mathbf{b}_2 - x\mathbf{b}_1|| : x \in \mathbb{Z}\} = ||\mathbf{b}_2 - \mu_0\mathbf{b}_1||,$$

όπου

$$\mu_0 = \left\lceil \frac{\mathbf{b}_1 \cdot \mathbf{b}_2}{||\mathbf{b}_1||^2} \right\rfloor.$$

Αν η βάση είναι ανηγμένη τότε $\mu_0 = 0$.

Απόδειξη. Θέτω $\mu = \frac{\mathbf{b}_1 \cdot \mathbf{b}_2}{||\mathbf{b}_1||^2}$ και $g(x) = ||\mathbf{b}_2 - x\mathbf{b}_1||$. Θ.α.ο. $g(\mu_0) \leq g(q)$ για κάθε ακέραιο q. Θεωρούμε το τρίγωνο με κορυφές

$$\mathbf{b}_2, (\mu_0 - 1)\mathbf{b}_1, (\mu_0 + 1)\mathbf{b}_1.$$

Τότε, το σημείο $\mu_0\mathbf{b}_1$ είναι από κατασκευής πιο κόντα στο σημείο $\mu\mathbf{b}_1$ από τα σημεία $(\mu_0 \pm 1)\mathbf{b}_1$. Επομένως $g(\mu_0) \leq g(\mu_0 \pm 1)$. Για τα πιο απομακρυσμένα σημεία από το σημείο $\mu_0\mathbf{b}_1$ ισχύει το ίδιο.

Αν η βάση είναι ανηγμένη από το λήμμα 8.4.1(ii) προκύπτει $\mu_0 = 0$. □

Αλγόριθμος 8.4.1. : Αλγόριθμος των Gauss-Lagrange
Είσοδος. $\mathbf{b}_1, \mathbf{b}_2$ γραμμικα ανεξάρτητα
Έξοδος. Μια ανηγμένη κατά Gauss-Lagrange βάση

```
1  B_1 ← ||b_1||^2
2  μ ← b_1 · b_2 / B_1
3  b_2 ← b_2 − ⌈μ⌋ b_1
4  B_2 ← ||b_2||^2
5  while ||b_2|| < ||b_1|| do
6  |   swap(b_1, b_2)
7  |   B_1 ← B_2
8  |   μ ← b_1 · b_2 / B_1
9  |   b_2 ← b_2 − ⌈μ⌋ b_1
10 |   B_2 ← ||b_2||^2
   end
11 return b_1, b_2
```

Ορθότητα του αλγορίθμου

Αν το loop δεν εκτελεστεί, τότε καταλήγω σε δύο διανύσματα $\mathbf{b}_1$ και $\mathbf{b}_2 = \mathbf{b}_2 - \lceil \mu \rfloor \mathbf{b}_1$. Επομένως, από το προηγούμενο λήμμα 8.4.4, έχω $||\mathbf{b}_2|| < ||\mathbf{b}_2 + q\mathbf{b}_1||$ για κάθε ακέραιο q, επομένως η βάση είναι ανηγμένη. Αν εκτελεστεί το loop τότε καταλήγουμε σε δύο διανύσματα τέτοια ώστε $||\mathbf{b}_1|| < ||\mathbf{b}_2||$ (γραμμή 6, διαφορετικά τερματίζει). Το νέο $\mathbf{b}_1$ που τώρα είναι $\mathbf{b}_2$ το ονομάζω $\mathbf{b}'_1$. Επίσης, το νέο $\mathbf{b}_2$ που τώρα είναι $\mathbf{b}_1$ το ονομάζω $\mathbf{b}'_2$. Επομένως, το επόμενο βήμα, γραμμή 9: $\mathbf{b}'_2 \leftarrow \mathbf{b}'_2 - \mu_0 \mathbf{b}'_1 = \mathbf{b}_1 - \mu_0 \mathbf{b}_2$. Το νέο αυτό διάνυσμα από το προηγούμενο λήμμα είναι μικρότερο από το $\mathbf{b}_1$. Οπότε πάλι θα καταλήξουμε σε δύο διανύσματα $\mathbf{b}'_1, \mathbf{b}'_2$ τέτοια ώστε $||\mathbf{b}'_1|| < ||\mathbf{b}'_2|| < ||\mathbf{b}_1||$. Αν το loop συνεχιστεί τα νέα διανύσματα θα είναι μικρότερα από το $||\mathbf{b}'_1||$. Κ.ο.κ όλα τα νεα διανύσματα θα είναι αυστηρά μικρότερα. Αλλά μέσα στο κύκλο με ακτίνα $||\mathbf{b}_1||$ βρίσκονται πεπερασμένα σημεία του πλέγματος (αφού το πλέγμα είναι διακριτό σύνολο). Άρα ο αλγόριθμος θα τερματίσει μετά από πεπερασμένα βήματα και θα καταλήξει στα δύο μικρότερα διανύσματα που είναι γραμμικά ανεξάρτητα και ανήκουν στο πλέγμα.

Τέλος, η πολυπλοκότητα αυτού του αλγορίθμου αποδεικνύεται ότι είναι πολυωνυμική.

Θεώρημα 8.4.2 Έστω $||\mathbf{b}_i|| \leq B$ $(i = 1, 2)$. Μετά από $O(\log_2(B))^3)$ bit-πράξεις ο αλγόριθμος θα τερματίσει.

Απόδειξη. [7, Θεώρημα 17.1.10] □

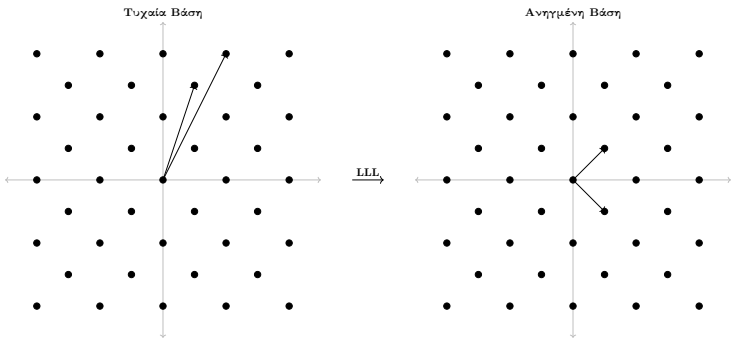
8.5 Ο αλγόριθμος LLL

Συνήθως ένα πλέγμα ορίζεται από διανύσματα που δεν έχουν *καλές* ιδιότητες. Θα επιθυμούσαμε να δουλεύουμε με διανύσματα μικρού μήκους και ανά δύο κάθετα. Ο λόγος είναι ότι μπορούμε πολύ αποδοτικά να κάνουμε πράξεις σε αυτό το πλέγμα. Γενικά, σε πολλά προβλήματα κρυπτανάλυσης πρέπει να βρούμε ένα αρκετά μικρό διάνυσμα του πλέγματος. Ο αλγόριθμος LLL που ανακαλύφθηκε από τους Arjen **L**enstra, Hendrik **L**enstra και László **L**ovász είναι πολυωνυμικού χρόνου (ως προς τα μήκη των διανυσμάτων που ορίζουν το πλέγμα) και επιστρέφει ένα διάνυσμα $2^{(n-1)/4}$ το μήκος του $\lambda_1(L)$, όπου $n = rank(L)$. Στην πράξη, μετά από πειράματα που εκτελέστηκαν από τους Nguyen-Gama, διαπιστώθηκε ότι ο LLL επιστρέφει με μεγάλη πιθανότητα το μικρότερο διάνυσμα για πλέγματα με τάξη το πολύ 35. Να αναφέρουμε ότι για διδιάστατα πλέγματα (μέγιστης τάξης) το SVP ο LLL το λύνει. Ο αλγόριθμος για την διάσταση δύο ονομάζεται αλγόριθμος των Gauss-Lagrange και ανακαλύφθηκε πολύ νωρίτερα από τον LLL. Είναι ανοιχτό πρόβλημα στα πλέγματα η εύρεση ενός αλγορίθμου πολυωνυμικού χρόνου που να επιστρέφει ένα διάνυσμα του πλέγματος με μέτρο $< Poly(n)\lambda_1(L)$, όπου $Poly(n)$

ένα πολυώνυμο του n.

Αλγόριθμος 8.5.1. : LLL Pseudocode
Είσοδος. Μια διατεταγμένη βάση $\mathcal{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \subset \mathbb{Z}^m$ του πλέγματος $\mathcal{L}(\mathcal{B})$ και έναν πραγματικό αριθμό $\delta \in (1/4, 1)$. Χρησιμοποιούμε την συνάρτηση $gso = GSO[\mathcal{B}]$ (αλγόριθμος 8.2.1) η οποία επιστρέφει μια ορθογώνια βάση $gso[0]$ και τον Gram-Schmidt matrix gso[1]
Έξοδος. Μια $LLL-$ανηγμένη βάση του $\mathcal{L}$

```
--Initialization:
i = 2
gso ← GSO(B)
(b_i)_i ← gso[1]
(μ_{i,j})_{i,j} ← gso[2]
-- Size Reduction Step:
while i ≤ n do
    for j = i − 1 to 1 do
        c_{i,j} ← ⌊μ_{i,j}⌉          #⌊x⌉ = ⌊x + 0.5⌋
        b_i ← b_i − c_{i,j} b_j
        update B and gso ← GSO(B)
        (b_i)_i ← gso[1]
        (μ_{i,j})_{i,j} ← gso[2]
    end
    --    Swap step:
    if δ||b*_i||^2 > ||μ_{i+1,i} b*_i + b*_{i+1}||^2 then
        b_i ↔ b_{i+1}
        i = max(2, i − 1)
        update B and gso ← GSO(B)
        (b_i)_i ← gso[1]
        (μ_{i,j})_{i,j} ← gso[2]
    else
        i ← i + 1
    end
end
return B
```



Το βασικό θεώρημα που αποδείχτηκε από τους Lenstra, Lenstra και Lovász είναι το παρακάτω.

Θεώρημα 8.5.1 Έστω $L(\mathcal{B})$ με $B = \{\mathbf{b}_1, ..., \mathbf{b}_n\}$, $\mathbf{b}_i \in \mathbb{Z}^m$, $\Lambda = \max_i(||\mathbf{b}_i||^2)$ και $\delta = 3/4$. Τότε ο προηγούμενος αλγόριθμος τερματίζει μετά από $O(n^2 \log_2 \Lambda)$

επαναλήψεις και $O(n^2)$ αριθμητικές πράξεις ανά επανάληψη. Το συνολικό κόστος σε bit είναι $O(n^5 m(\log_2 \Lambda)^2)$.

8.5.1 Βοηθητικά λήμματα

Λήμμα 8.5.1. *Αν $\{\mathbf{b}_1, ..., \mathbf{b}_n\}$ είναι μια ανηγμένη LLL βάση, τότε για την GS βάση ισχύει*

$$||\mathbf{b}^*_{i-1}|| < \frac{2}{\sqrt{4\delta - 1}}||\mathbf{b}^*_i||,\ 2 \leq i \leq n.$$

Απόδειξη. Από την συνθήκη του Lovász έχουμε

$$\delta||\mathbf{b}^*_{i-1}||^2 \leq ||\mu_{i,i-1}\mathbf{b}^*_{i-1}||^2 + ||\mathbf{b}^*_i||^2,\ 2 \leq i \leq n.$$

Παρατηρούμε ότι

$$\mu_{i,i-1}||\mathbf{b}^*_{i-1}|| = \mathbf{b}_i \cdot \frac{\mathbf{b}^*_{i-1}}{||\mathbf{b}^*_{i-1}||} = ||\mathbf{b}_i|| \cos\phi,$$

όπου ϕ η γωνία των διανυσμάτων $\mathbf{b}_i, \mathbf{b}^*_{i-1}$. Εφόσον είναι ανηγμένη κατά μέγεθος (size reduced) έχουμε,

$$|\cos\phi| < \frac{||\mathbf{b}^*_{i-1}||}{2||\mathbf{b}_i||}.$$

Επομένως, η συνθήκη του Lovász γράφεται

$$\sqrt{\delta - \frac{1}{4}}||\mathbf{b}^*_{i-1}|| < ||\mathbf{b}^*_i||.$$

Το ζητούμενο έπεται. □

8.6 SVP

Γενικά υπάρχουν δύο κατηγορίες αλγορίθμων για την επίλυση αυτού του προβλήματος. Αλγόριθμοι απαρίθμησης (enumeration algorithms) και αλγόριθμοι κοσκινίσματος (sieving algorithms). Υπάρχει έντονη έρευνα τα τελευταία χρόνια γι′αυτό το πρόβλημα, διότι η επίλυση του *σπάει* κρυπτοσυστήματα που πιστεύουμε ότι έχουν ασφάλεια από κβαντικούς υπολογιστές. Ασυμπτωτικά οι αλγόριθμοι κοσκινίσματος έχουν καλύτερη πολυπλοκότητα από τους αλγόριθμους απαρίθμησης. Ο καλύτερος αλγόριθμος[38] απαιτεί χρόνο $O(2^{2.465n})$ και μνήμη $O(2^{1.23n})$. Το μειονέκτημα τους είναι η εκθετικά μεγάλη απαίτηση σε μνήμη. Στην πράξη χρησιμοποιούμε και τις δύο κατηγορίες αλγορίθμων. Το πανεπιστήμιο του Darmstadt[39] τρέχει έναν διαγωνισμό για το approximation SVP όπου φαίνεται ότι στην

[38] Xavier Pujol and Damien Stehle, Solving the shortest lattice vector problem in time $2^{2.465n}$

[39] `https://www.latticechallenge.org/svp-challenge/halloffame.php`

πράξη και οι δύο κατηγορίες αλγορίθμων δουλεύουν αρκετά καλά (ίσως οι Sieving algorithms να δουλέυουν λίγο καλύτερα).

Ο Ajtai το 1996 απόδειξε ότι το SVP_γ είναι NP-hard under randomized reductions για $\gamma = O(1)$. Για $\gamma = O(\sqrt{n})$ είναι co-NP∩NP και για $\gamma = 2^{\sqrt{n}}$ έχουμε υπο-εκθετικούς αλγόριθμους. Για $\gamma = O(2^n)$ έχουμε πολυωνυμικούς αλγόριθμους (LLL).

8.6.1 Ο αλγόριθμος απαρίθμησης των Kannan-Pohst-Fincke

Ο αλγόριθμος του KPF : **K**annan-**P**ohst-**F**incke, επιστρέφει όλα τα διανύσματα ενός πλέγματος L που είναι μικρότερα από ένα θετικό αριθμό R. Πάντα υπάρχουν διανύσματα του πλέγματος $\leq R$ π.χ. το $\mathbf{0}$. Ο αλγόριθμος αυτός κατασκευάζει ένα δένδρο που αποτελείται από όλα τα διανύσματα των πλεγμάτων

$$\pi_n(L), \pi_{n-1}(L), ..., \pi_1(L)$$

με μήκος $\leq R$. Έστω $\{\mathbf{b}_1, ..., \mathbf{b}_n\}$ μια βάση του πλέγματος L. Ισχύει,

$$\mathbf{b}_i = \mathbf{b}_i^* + \sum_{j=1}^{i-1} \mu_{i,j}\mathbf{b}_j^*. \tag{8.6.1}$$

Οπότε, ένα διάνυσμα

$$\mathbf{v} = \sum_{j=1}^{n} v_i \mathbf{b}_i \in L$$

γράφεται,

$$\mathbf{v} = v_1\mathbf{b}_1^* + v_2(\mathbf{b}_2^* + \mu_{2,1}\mathbf{b}_1^*) + \cdots + v_n(\mathbf{b}_n^* + \sum_{j=1}^{n-1} \mu_{i,j}\mathbf{b}_j^*) =$$

$$\mathbf{b}_1^*(v_1 + \mu_{2,1}v_2 + \cdots + \mu_{n,1}v_n) + \cdots + \mathbf{b}_n^* v_n =$$

$$\sum_{j=1}^{n}(v_j + \sum_{i=j+1}^{n} \mu_{i,j}v_i)\mathbf{b}_j^*.$$

Επομένως,

$$||\mathbf{v}||^2 = \sum_{j=1}^{n}\Big(v_j + \sum_{i=j+1}^{n} \mu_{i,j}v_i\Big)^2 ||\mathbf{b}_j^*||^2,$$

και

$$||\pi_r(\mathbf{v})||^2 = \sum_{j=r}^{n}(v_j + \sum_{i=j+1}^{n} \mu_{i,j}v_i)^2 ||\mathbf{b}_j^*||^2 \ (1 \leq r \leq n). \tag{8.6.2}$$

Παρατηρούμε ότι,

$$||\pi_n(\mathbf{v})||^2 \leq ||\pi_{n-1}(\mathbf{v})||^2 \leq \cdots \leq ||\pi_1(\mathbf{v})||^2 = ||\mathbf{v}||^2 \leq R^2.$$

Δηλαδή, το άθροισμα (8.6.2) αυξάνεται όταν πάμε από τον δείκτη r στον $r-1$. Η ιδέα του αλγορίθμου είναι να φράξουμε αρχικά το v_n (την $n-$συντεταγμένη του $\mathbf{v}$) από την συνθήκη $||\pi_n(\mathbf{v})||^2 \leq R^2$, δηλαδή, να υπολογίσουμε ένα διάστημα I_1 όπου παίρνει τιμές το v_n. Κατόπιν, υπολογίζουμε ένα διάστημα I_2 για το v_{n-1} κ.ο.κ. έως ότου φτάσουμε στο διάστημα I_n.

Έστω $1 \leq k \leq n$, τότε

$$||\pi_{n+1-k}(\mathbf{v})||^2 \leq R^2.$$

ισοδύναμα

$$\sum_{j=n+1-k}^{n} \Big(v_j + \sum_{i=j+1}^{n} \mu_{i,j} v_i\Big)^2 ||\mathbf{b}_j^*||^2 \leq R^2,$$

ισοδύναμα

$$\Big(v_{n+1-k} + \sum_{i=n+2-k}^{n} \mu_{i,n+1-k} v_i\Big)^2 ||\mathbf{b}_{n+1-k}^*||^2 + \sum_{j=n+2-k}^{n} \Big(v_j + \sum_{i=j+1}^{n} \mu_{i,j} v_i\Big)^2 ||\mathbf{b}_j^*||^2 \leq R^2,$$

ισοδύναμα

$$\Big|v_{n+1-k} + \sum_{i=n+2-k}^{n} \mu_{i,n+1-k} v_i\Big| \leq \frac{\sqrt{R^2 - \sum_{j=n+2-k}^{n} \Big(v_j + \sum_{i=j+1}^{n} \mu_{i,j} v_i\Big)^2 ||\mathbf{b}_j^*||^2}}{||\mathbf{b}_{n+1-k}^*||}.$$

Αν θέσουμε

$$l_j = \Big(v_j + \sum_{i=j+1}^{n} \mu_{i,j} v_i\Big)^2 ||\mathbf{b}_j^*||^2, \tag{8.6.3}$$

τότε

$$\Big|v_{n+1-k} + \sum_{i=n+2-k}^{n} \mu_{i,n+1-k} v_i\Big| \leq \frac{\sqrt{R^2 - \sum_{j=n+2-k}^{n} l_j}}{||\mathbf{b}_{n+1-k}^*||}.$$

Για $k = 1$ έχουμε

$$|v_n| \leq \frac{R}{||\mathbf{b}_n^*||}.$$

Άρα,

$$v_n \in \Big[-\frac{R}{||\mathbf{b}_n^*||}, \frac{R}{||\mathbf{b}_n^*||}\Big].$$

Επειδή, τα $\pm\mathbf{v}$ έχουν το ίδιο μέτρο, θεωρούμε

$$v_n \in I_1 = \Big[0, \frac{R}{||\mathbf{b}_n^*||}\Big]. \tag{8.6.4}$$

Για $k = 2$ προκύπτει,

$$\big|v_{n-1} + \mu_{n,n-1} v_n\big| \leq \frac{\sqrt{R^2 - v_n^2 ||\mathbf{b}_n^*||^2}}{||\mathbf{b}_{n-1}^*||},$$

επομένως,

$$v_{n-1} \in I_2 = \left[-\mu_{n,n-1}\, v_n - \frac{\sqrt{R^2 - v_n^2||\mathbf{b}_n^*||^2}}{||\mathbf{b}_{n-1}^*||}, -\mu_{n,n-1} v_n + \frac{\sqrt{R^2 - v_n^2||\mathbf{b}_n^*||^2}}{||\mathbf{b}_{n-1}^*||} \right].$$

Ισοδύναμα,

$$v_{n-1} \in I_2 = \left[-\mu_{n,n-1}\, v_n - \frac{\sqrt{R^2 - l_n}}{||\mathbf{b}_{n-1}^*||}, -\mu_{n,n-1} v_n + \frac{\sqrt{R^2 - l_n}}{||\mathbf{b}_{n-1}^*||} \right] \tag{8.6.5}$$

Γενικά,

$$v_{n+1-k} \in I_k =$$

$$\left[-\sum_{i=n+2-k}^{n} \mu_{i,n+1-k}\, v_i - \frac{\sqrt{R^2 - \sum_{j=n+2-k}^{n} l_j}}{||\mathbf{b}_{n+1-k}^*||}, -\sum_{i=n+2-k}^{n} \mu_{i,n+1-k}\, v_i + \frac{\sqrt{R^2 - \sum_{j=n+2-k}^{n} l_j}}{||\mathbf{b}_{n+1-k}^*||} \right].$$

Enumeration Tree.
Ο αλγόριθμος αυτός δουλεύει με ένα δένδρο ύψους (height of tree) n. Ο αλγόριθμος που χρησιμοποιούμε για να διασχίσουμε αυτό το δένδρο είναι ο DFS : **D**epth **F**irst **S**earch. Ο λόγος είναι για να χρειαστούμε λίγη μνήμη. Ως ρίζα του δένδρου (root) θεωρούμε το μηδενικό διάνυσμα $\mathbf{0} = \pi_{n+1}(L)$. Οι κόμβοι (nodes) του δένδρου σε βάθος (depth) $k = n + 1 - d$ $(d = n, n-1, ..., 1)$ [40] είναι όλα τα διανύσματα του πλέγματος $\pi_d(L)$ με μέτρο το πολύ R. Αν ο μοναδικός κόμβος σε όλα τα πιθανά βάθη k είναι το μηδενικό διάνυσμα, ο αλγόριθμος επιστρέφει *αποτυχία*. Αν ο αλγόριθμος καταλήξει σε ένα φύλλο (leaf), τότε επιστρέφει το διάνυσμα αυτό (και έχει μέτρο $\leq R$).

Αν $\overline{B}_r(R)$ η $r-$διάστατη σφαίρα ακτίνας R, τότε σε ύψος d (ή σε βάθος $k = n + 1 - d$) οι κόμβοι είναι όλα τα διανύσματα του συνόλου

$$\overline{B}_{n+1-d}(R) \cap \pi_d(L) = \overline{B}_k(R) \cap \pi_d(L).$$

Το πλήθος των κόμβων σε ένα συγκεκριμένο βάθος εξαρτάται και από το πόσο ανηγμένη είναι η βάση. Οπότε, όταν χρησιμοποιούμε τον αλγόριθμο KPF πρέπει να προσέχουμε η βάση μας να είναι ανηγμένη, διαφορετικά ο αλγόριθμος θα είναι αρκετά πιο αργός. Πολλές φορές η διαδικασία αναγωγής (συνήθως είναι απλά η εκτέλεση του LLL) ονομάζεται preprocessing phase.

Υπάρχουν, δύο παραλλαγές που μπορούμε να εφαρμόσουμε για να κάνουμε απαρίθμηση στα διαστήματα I_k. Η πρώτη οδηγεί στον κλασικό αλγόριθμο των KFP και η δεύτερη στον αλγόριθμο Schnorr-Euchner που χρησιμοποιείται στον αλγόριθμο BKZ (είναι μια βελτιωμένη παραλλαγή του LLL που δίνει καλύτερα διανύσματα, αλλά δεν είναι πολυωνυμικού χρόνου).

Άσκηση 8.16 Αν ισχύει η GSA σε ένα πλέγμα και ο συντελεστής GSA είναι $r > 1/2$, τότε $|I_1 \cap \mathbb{Z}| < |I_2 \cap \mathbb{Z}|$.

[40] Βάθος ενός κόμβου σ΄ ένα δένδρο είναι το πλήθος των ακμών που ενώνουν την ρίζα με τον κόμβο. Ενώ ύψος (height of node) είναι το πλήθος των ακμών που ενώνουν τον κόμβο με ένα φύλλο. Επομένως στον τύπο $k = n + 1 - d$ το k είναι το βάθος του κόμβου, ενώ το d το ύψος του.

Παράδειγμα 8.6.1. Ας θεωρήσουμε το πλέγμα που παράγεται από τις γραμμές του πίνακα,

$$B = [\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3] = \begin{bmatrix} 3 & 6 & 13 \\ 11 & 3 & 15 \\ 12 & 12 & 0 \end{bmatrix}.$$

Εύκολα υπολογίζουμε

$$(||\mathbf{b}_1^*||^2, ||\mathbf{b}_2^*||^2, ||\mathbf{b}_3^*||^2) \approx (214, 72.21, 206.86)$$

και

$$M = (\mu_{i,j})_{i,j} = \begin{bmatrix} 1 & 0 & 0 \\ 1.149 & 1 & 0 \\ 0.504 & 0.607 & 1 \end{bmatrix}.$$

Από την σχέση (8.6.4) για $R = ||\mathbf{b}_1|| = \sqrt{214} \approx 14.62$, έχουμε

$$v_n = v_3 \in \left[0, \frac{R}{||\mathbf{b}_3^*||}\right] = [0, 1.01].$$

Επομένως, καθώς το v_3 είναι ακέραιος, αναγκαστικά παίρνει τις τιμές $v_3 = 0$ ή $v_3 = 1$.
Περίπτωση $v_3 = 1$.
Από την σχέση (8.6.5) έχουμε,

$$I_2(1) = \left[-\mu_{3,2}v_3 - \frac{\sqrt{R^2 - l_3}}{||\mathbf{b}_2^*||}, -\mu_{3,2}v_3 + \frac{\sqrt{R^2 - l_3}}{||\mathbf{b}_2^*||} \right],$$

όπου $l_3 = v_3^2||\mathbf{b}_3^*||^2 = 206.86$. Επομένως,

$$I_3(1) = \left[-0.607 - \frac{\sqrt{214 - 206.86}}{\sqrt{72.21}}, -0.607 + \frac{\sqrt{214 - 206.86}}{\sqrt{72.21}} \right] = [-0.92, -0.29].$$

Άρα, δεν υπάρχει κάποιος ακέραιος στο $I_3(1)$.
Περίπτωση $v_3 = 0$.
Παρόμοια, για το v_2 έχουμε από την σχέση (8.6.5) (έχουμε μηδενίσει το v_3),

$$v_2 \in I_2 = \left[\frac{-\sqrt{R^2 - l_3}}{||\mathbf{b}_2^*||}, \frac{\sqrt{R^2 - l_3}}{||\mathbf{b}_2^*||} \right],$$

αλλά $||\pi_3(\mathbf{v})||^2 = l_3 = v_3^2||\mathbf{b}_3^*||^2 = 0$, επομένως

$$v_2 \in \left[-\frac{\sqrt{214}}{\sqrt{72.21}}, \frac{\sqrt{214}}{\sqrt{72.21}} \right] = [-1.72, 1.72],$$

άρα $v_2 \in \{-1, 0, 1\}$. Τέλος, πρέπει να υπολογίσουμε το $v_1 \in I_3$, όπου

$$I_3 = \left[-\frac{\sqrt{R^2 - l_2 - l_3}}{||\mathbf{b}_1||} - \mu_{2,1}v_2 - \mu_{3,1}v_3, \frac{\sqrt{R^2 - l_2 - l_3}}{||\mathbf{b}_1||} - \mu_{2,1}v_2 - \mu_{3,1}v_3 \right].$$

Το I_3 εξαρτάται από τα v_2, v_3, αλλά το $v_3 = 0$. Άρα εξαρτάται μόνο από το $v_2 \in \{-1, 0, 1\}$. Επίσης και το $l_3 = 0$. Δηλ.

$$I_3(v_2) = \Big[-\frac{\sqrt{R^2 - l_2}}{||\mathbf{b}_1||} - \mu_{2,1} v_2, \frac{\sqrt{R^2 - l_2}}{||\mathbf{b}_1||} - \mu_{2,1} v_2 \Big]$$

και

$$l_2(v_2) = (v_2 + \mu_{3,2} v_3)^2 ||\mathbf{b}_2^*||^2 = v_2^2 ||\mathbf{b}_2^*||^2.$$

Άρα

$$l_2(-1) = l_2(1) = 72.21, l_2(0) = 0.$$

Για $v_2 = -1$ έχουμε

$$I_3(-1) = [0.33, 1.96].$$

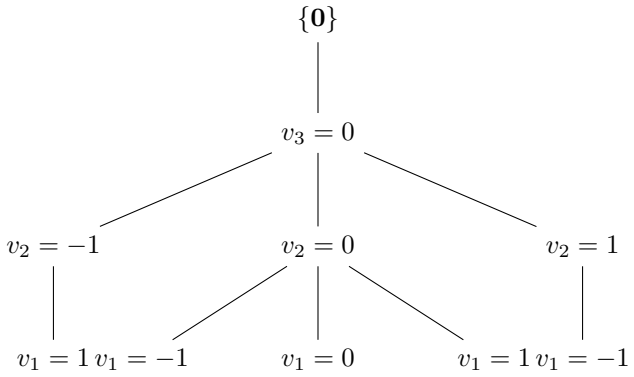
Επομένως, $v_1 = 1$. Για $v_2 = 1$ έχουμε

$$I_3(1) = [-1.96, -0.33],$$

άρα $v_1 = -1$. Ενώ, $I_3(0) = [-1, 1]$. Άρα, $v_1 \in \{-1, 0, 1\}$. Τα μη μηδενικά διανύσματα του πλέγματος που έχουν μέτρο $\leq R = \sqrt{214}$, είναι αυτά που αντιστοιχούν στις τιμές

$$(v_1, v_2, v_3) = (-1, 1, 0), (1, -1, 0), (\pm 1, 0, 0).$$

Το υποδέντρο απαρίθμησης για την περίπτωση $v_3 = 0$, $R = ||\mathbf{b}_1||$ είναι το παρακάτω.



Στην τριάδα $(v_1, v_2, v_3) = (-1, 1, 0)$ αντιστοιχεί το διάνυσμα $\mathbf{v}_1 = -\mathbf{b}_1 + \mathbf{b}_2 = (8, -3, 2)$ που έχει μέτρο 8.77, ενώ στην τριάδα $(1, -1, 0)$ αντιστοιχεί το διάνυσμα $-\mathbf{v}_1$. Στην τριάδα $(-1, 0, 0)$ αντιστοιχεί το διάνυσμα $\mathbf{v}_2 = -\mathbf{b}_1$ που έχει μέτρο ίσο με $R = \sqrt{214} \approx 14.62$. Το ίδιο για την τριάδα $(-1, 0, 0)$ στην οποία αντιστοιχεί το διάνυσμα $-\mathbf{v}_1$. Επομένως έχουμε δύο μη-μηδενικά διανύσματα (αν αγνοήσουμε τα αντίθετα αυτών), τα $\mathbf{v}_1 = (8, -3, 2)$ και $\mathbf{v}_2 = (3, 6, 13)$.

Παρατήρηση 8.6.1. Αν στο δέντρο απαρίθμησης ανήκει το κλαδί $[v_n, v_{n-1}, ..., v_1]$, τότε και το κλαδί $[-v_n, -v_{n-1}, ..., -v_1]$ ανήκει στο δέντρο. Επομένως, στον αλγόριθμό μας μπορούμε να θεωρήσουμε τα μισά σημεία από το δέντρο απαρίθμησης. Επίσης, το δένδρο απαρίθμησης εξαρτάται από την ακτίνα R. Όσο μικρότερη είναι η ακτίνα τόσο λιγότερους κόμβους έχει το δένδρο. Το ίδιο συμβαίνει αν είναι ανηγμένη η βάση. Γι΄ αυτό το λόγο πριν εφαρμόσουμε τον αλγόριθμο απαρίθμησης χρησιμοποιούμε κάποιον αλγόριθμο αναγωγής (LLL/BKZ).

Όσον αφορά την πολυπλοκότητα του αλγορίθμου έχουμε το παρακάτω αποτέλεσμα.

Θεώρημα 8.6.1 (Harnot-Stehlé). Υπάρχει πολύωνυμο $p(x, y) \in \mathbb{R}[x, y]$ τέτοιο ώστε για κάθε πλέγμα τάξης n και διάστασης m και βάση που έχει συντελεστές που φράσσονται από το B, η έξοδος του αλγορίθμου των KFP απαιτεί

$$p(\log_2(B), m)n^{n/2e+o(n)}$$

bit πράξεις (όπου $e = exp(1)$).

Οι αλγόριθμοι της κατηγορίας Sieving έχουν καλύτερη ασυμπτωτική πολυπλοκότητα. Για παράδειγμα ο αλγόριθμος AKS είναι πιθανοτικός αλγόριθμος με ασυμπτωτική πολυπλοκότητα $2^{2.247n+o(n)}$ και μνήμη $2^{1.325n+o(n)}$. Επίσης υπάρχει ντετερμενιστικός αλγόριθμος που βασίζεται στα κελιά Voronoi, με ασυμπτωτική πολυπλόκότητα $2^{2n+o(n)}$ και μνήμη $2^{n+o(n)}$.

8.6.2 Ο ψευδοκώδικας του αλγορίθμου απαρίθμησης

Αλγόριθμος 8.6.1. : Αλγόριθμος Απαρίθμησης (KFP enumeration algorithm)
Είσοδος. Μια διατεταγμένη βάση $\mathcal{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \subset \mathbb{Z}^m$ του πλέγματος $\mathcal{L}(\mathcal{B})$ και ένα θετικό αριθμό R.
Έξοδος. Όλα τα διανύσματα $\mathbf{x} \in \mathcal{L}$ με $||\mathbf{x}|| \leq R$.

01. `Compute` $\{\mu_{ij}\}$ `and` $B_i = ||\mathbf{b}_i^*||^2$
02. $\mathbf{x} = (x_i) \leftarrow \mathbf{0}_n, \mathbf{c} = (c_i) \leftarrow \mathbf{0}_n,\ \ell = (\ell_i) \leftarrow \mathbf{0}_n,\ sumli \leftarrow 0, S = \emptyset, i \leftarrow 1$
03. **While** $i \leq n$
04. $\quad c_i \leftarrow -\sum_{j=i+1}^{n} x_j \mu_{ji}$
05. $\quad \ell_i \leftarrow B_i(x_i - c_i)^2$
06. $\quad sumli \leftarrow \sum_{j=i}^{n} \ell_j$
07. $\quad$ **If** $sumli \leq R^2$
08. $\quad\quad$ **If** $i = 1$
09. $\quad\quad\quad S \leftarrow S \cup \{\sum_{j=1}^{n} x_j \mathbf{b}_j\}$
10. $\quad\quad\quad x_1 \leftarrow x_1 + 1$
11. $\quad\quad$ **else**
12. $\quad\quad\quad i \leftarrow i - 1$
13. $\quad\quad\quad x_i \leftarrow$ `left part of the interval` I_i
14. $\quad\quad$ **end if**
15. $\quad$ **else**
16. $\quad\quad i \leftarrow i + 1$

17. $x_i \leftarrow x_i + 1$
18. **end if**
19. `return S`

Στην γραμμή 13, το x_i παίρνει την τιμή

$$\left\lceil -\sum_{j=i+1}^{n} \mu_{j,i}\, x_j - \sqrt{\frac{R^2 - \sum_{j=i+1}^{n} l_j}{B_i}} \right\rceil$$

που είναι το αριστερό άκρο του διαστήματος I_{n+1-i}. Στον ψευδοκώδικα το i είναι το height του δένδρου απαρίθμησης. Ο αλγόριθμος αυτός δεν παίρνει έτοιμα τα διάστηματα I_i αλλά φτιάχνει αυτά τα διαστήματα με τον έλεγχο που γίνεται στην γραμμή 7, που είναι $l_i \leq R^2$, όπου το l_i δίνεται από την ισότητα (8.6.3). Όταν το $i = 1$ (γραμμή 8) σημαίνει ότι έχουμε βρει ένα διάνυσμα με μέτρο $\leq R$, οπότε το αποθηκεύουμε στην λίστα S. Αν το $i > 1$ (γραμμή 11) τότε μειώνουμε το ύψος (δηλ. αυξάνουμε το βάθος και πλησιάζουμε πιο κοντά σ' ένα πιθανό φύλλο) και ξεκινάμε το x_i από το αριστερό άκρο του διαστήματος I_{n+1-i}. Στη συνέχεια θα κατέβουμε πιο χαμηλά στο δένδρο, όσο το τρέχον μήκος της προβολής του διανύσματος είναι $\leq R$, αντίθετα (γραμμή 15) θα αύξησουμε το ύψος (θα ανέβουμε ψηλότερα στο δένδρο) και θα αυξήσουμε κατά 1 το x_i. Ο αλγόριθμος μπορεί να αποτύχει, με την έννοια να μην βρει κάποιο μη μηδενικό διάνυσμα.

Αν εκτελέσουμε τον αλγόριθμο στο προηγούμενο παράδειγμα θα πάρουμε τα διανύσματα (με την εξης σειρά):

$$(0,0,0), (1,0,0), (-1,1,0)$$

διατρέχοντας το δένδρο από κάτω προς τα πάνω. Παρατηρήστε ότι ο αλγόριθμος θα κατασκευάσει τα μισά διανύσματα του δένδρου απαρίθμησης. Τα υπόλοιπα προκύπτουν θεωρώντας τα αντίθετα αυτών. Η μνήμη που χρησιμοποιούμε είναι $O(n)$.

Οι Schnorr-Euchner πρότειναν το x_i να ξεκινά από την τιμή

$$\left\lceil -\sum_{j=i+1}^{n} \mu_{j,i}\, x_j \right\rfloor$$

που είναι πολύ κοντά στο κέντρο του διαστήματος I_i και να αυξομειώνεται το x_i κατά 1. Η παραλαγή αυτή ονομάζεται αλγοριθμος των Schnorr-Euchner και χρησιμοποιείται ως υπορουτίνα στον αλγόριθμο BKZ. Στον αλγόριθμο των Schnorr-Euchner υπάρχει μια ακόμη παραλλάγη όπου επιτρέπουμε αντί του R, όταν βρισκόμαστε σε βάθος i στη γραμμή 7, να έχουμε $sumli \leq R_i$ όπου $R_i = \alpha_i R_i$ για κάποιο $\alpha_i \in (0,1)$. Τότε λέμε ότι έχουμε απαρίθμιση με κλάδεμα (enumeration with pruning) που επίσης πρότειναν οι Schnorr-Euchner-Horner.

Αλγόριθμος 8.6.2. : Αλγόριθμος Απαρίθμησης των Schnorr-Euchner (Schnorr-Euchner enumeration algorithm)[41]

Είσοδος. Μια διατεταγμένη βάση $\mathcal{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \subset \mathbb{Z}^m$ του πλέγματος $\mathcal{L}(\mathcal{B})$ και ένα θετικό αριθμό R.

Έξοδος. Όλα τα διανύσματα $\mathbf{x} \in \mathcal{L}$ $(x_n \geq 0)$ $||\mathbf{x}|| \leq R$.

[41] Δείτε Fig. 1 του Harnot, Stehle, Rigorous and Efficient Short Lattice Vectors Enumeration

01. Compute $\{\mu_{ij}\}$ and $B_i = ||\mathbf{b}_i^*||^2$
02. $\mathbf{x} = (x_i) \leftarrow \mathbf{0}_n, \mathbf{c} = (c_i) \leftarrow \mathbf{0}_n,\ \ell = (\ell_i) \leftarrow \mathbf{0}_n$
03. $\Delta\mathbf{x} \leftarrow (1, \mathbf{0}_{n-1}),\ \Delta^2\mathbf{x} \leftarrow (1, (-\mathbf{1})_{n-1}),\ sumli \leftarrow 0, S = \emptyset, i \leftarrow 1$
04. **While** $i \leq n$
05. $\quad c_i \leftarrow -\sum_{j=i+1}^{n} x_j \mu_{ji}$
06. $\quad \ell_i \leftarrow B_i(x_i - c_i)^2$
07. $\quad sumli \leftarrow \sum_{j=i}^{n} \ell_j$
08. $\quad$ **If** $sumli \leq R^2$ and $i = 1$
09. $\quad\quad S \leftarrow S \cup \{\sum_{j=1}^{n} x_j \mathbf{b}_j\}$
10. $\quad$ **end if**
11. $\quad$ **If** $sumli \leq R^2$ and $i > 1$
12. $\quad\quad i \leftarrow i - 1$
13. $\quad\quad c_i \leftarrow -\sum_{j=i+1}^{n} x_j \mu_{ji}$ # center of the interval I_{n+1-i}
14. $\quad\quad x_i \leftarrow \lfloor c_i \rceil$
15. $\quad\quad \Delta x_i \leftarrow 0$
16. $\quad\quad$ **If** $c_i < x_i$
17. $\quad\quad\quad \Delta^2 x_i \leftarrow 1$
18. $\quad\quad$ **else**
19. $\quad\quad\quad \Delta^2 x_i \leftarrow -1$
20. $\quad\quad$ **end if**
21. $\quad$ **elif** $i = n$ # i.e. if $sumli > R^2$ and $i = n$
22. $\quad\quad$ **break**
23. $\quad$ **else** # i.e. if $sumli > R^2$ or $i = 1$
24. $\quad\quad i \leftarrow i + 1$
25. $\quad\quad \Delta^2 x_i \leftarrow -\Delta^2 x_i$
26. $\quad\quad \Delta x_i \leftarrow -\Delta x_i + \Delta^2 x_i$
27. $\quad\quad x_i \leftarrow x_i + \Delta x_i$
28. $\quad$ **end if**
29. return S

Στην υλοποίηση αυτή, ο δείκτης i, είναι το ύψος του κόμβου στο δέντρο απαρίθμησης. Οι μεταβλητές $\Delta x_i, \Delta^2 x_i$ μας βοηθούν να ορίσουμε τον zig-zag αλγόριθμο στα διαστήματα I_{n+1-i}. Στα υπόλοιπα δεν διαφέρει από τον αλγόριθμο των KFP.

Αν στην γραμμή 08 (και 11) του αλγορίθμου, αντί R^2 θέσουμε $R^2_{n+1-i} = a^2_{n+1-i}R^2$, για κάποια $a_i \in (0,1)$, τότε έχουμε τον αλγόριθμο enumeration with prunning και prunning vector $\mathbf{a} = (a_1, a_2, ..., a_n)$ με $0 < a_1 \leq a_2 \leq \cdots \leq a_n = 1$. Δηλ. η γραμμή 8 θα αντικατασταθεί από την
08. $\quad$ **If** $sumli \leq (a_{n+1-i}R)^2$ and $i = 1$
και η γραμμή 11 από την
11. $\quad$ **If** $sumli \geq (a_{n+1-i}R)^2$ and $i > 1$
και στην είσοδο του αλγορίθμου χρειαζόμαστε και το διάνυσμα $\mathbf{a}$. Επομένως, αν στην είσοδο δοθεί το διάνυσμα $\mathbf{a} = (a_1, ..., a_n)$ τότε για κάθε i στον αλγόριθμο θεωρούμε την συντεταγμένη a_{n+1-i}. Παρατηρήστε ότι οι αριθμοί R_j έχουν ως δείκτη το βάθος και όχι το ύψος του δένδρου (ισχύει $j + \text{height} = n + 1$). Αν θέσουμε $\mathbf{a} = (1, 1, \ldots, 1)$ τότε εκτελείται ο αλγόριθμος χωρίς prunning. Όταν

χρησιμοποιούμε prunning ο αλγόριθμος μπορεί να αποτύχει ακόμη και αν υπάρχει διάνυσμα μήκους $\leq R$.

Ο αλγόριθμος αυτός χρησιμοποιείται ως υπορουτίνα στον βασικό αλγόριθμο αναγωγής BKZ. Έχει αναλυθεί επίσης η floating point εκδοχή του και έχει υλοποιηθεί στα υπολογιστικά πακέτα fplll (και το αντίστοιχο σε python που ονομάζεται fpylll) και NTL[42].

Άσκηση 8.17 Έστω $\{\mathbf{b}_i\}_{1\leq i\leq n}$ μια βάση ενός πλέγματος $\mathcal{L}$. Να υλοποιηθεί ο αλγόριθμος (8.6.2) με prunning vector τέτοιο ώστε $a_i = \min\{1, 1.05\sqrt{i/n}\}$ $(1 \leq i \leq n)$.

8.7 CVP

Το πρόβλημα CVP_γ, $\gamma \geq 1$ (approximate **C**losest **V**ector **P**roblem) είναι το εξής.

Ορισμός 8.7.1. $(CVP_\gamma(\mathcal{L}, \mathbf{t}))$ *Δοθέντος πλέγματος* $\mathcal{L} \subset \mathbb{Z}^m$ *και διανύσματος* $\mathbf{t} \in \mathbb{R}^m$ *βρείτε διάνυσμα* $\mathbf{x} \in \mathcal{L}$, *τέτοιο ώστε για κάθε διάνυσμα* $\mathbf{y} \in L$ *να ισχύει*

$$||\mathbf{x} - \mathbf{t}|| \leq \gamma ||\mathbf{y} - \mathbf{t}||,$$

για κάποιο $\gamma \geq 1$.

Το πρόβλημα απόφασης είναι το παρακάτω·.

Ορισμός 8.7.2. (*Decision* $CVP_\gamma(L, \mathbf{t})$) *Δοθέντος πλέγματος* $\mathcal{L} \subset \mathbb{Z}^m$, *διανύσματος* $\mathbf{t} \in \mathbb{R}^m$ *και αριθμού* $r \in \mathbb{Q}$, *αποφάσισε αν η απόσταση*

$$dist(\mathbf{t}, \mathcal{L}) = \min\{||\mathbf{x} - \mathbf{t}|| : \mathbf{x} \in \mathcal{L}\},$$

είναι $\leq r$ *ή* $> \gamma r$.

Με χρήση του αλγορίθμου LLL μπορούμε να λύσουμε το προσεγγιστικό $\text{CVP}_\gamma(L)$ για κάποιο πλέγμα $L \subset \mathbb{Z}^m$, για $\gamma = 2^{n/2}$ και $n = rank(L)$ σε πολυωνυμικό χρόνο. Ο αλγόριθμος που προκύπτει ονομάζεται αλγόριθμος του Babai.

Αλγόριθμος 8.7.1. : Babai's nearest plane algorithm
Είσοδος. Μια διατεταγμένη βάση $\mathcal{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \subset \mathbb{Z}^m$ του πλέγματος $\mathcal{L}(\mathcal{B})$ και έναν διάνυσμα $\mathbf{t} \in \mathbb{R}^m$.
Έξοδος. $\mathbf{x} \in \mathcal{L}$ τέτοιο ώστε $||\mathbf{x} - \mathbf{t}|| \leq 2^{n/2} dist(\mathcal{L}, \mathbf{t})$

```
B ← LLL(B) (δ = 3/4)
Compute B* ← GSO(B)
b ← t
for j = n to 1 do
    c_j ← ⌊ (b·b*_j) / (||b*_j||²) ⌉        # ⌊x⌉ = ⌊x + 0.5⌋
    b ← b − c_j b_j
end
return t − b
```

[42] `https://github.com/fplll/fplll` και `https://shoup.net/ntl/`

Αρχικά να παρατηρήσουμε ότι το διάνυσμα $\mathbf{t} - \mathbf{b} \in \mathcal{L}$. Το διάνυσμα $\mathbf{b}$ που θα προκύψει μετά το πέρας των επαναλήψεων είναι της μορφής $\mathbf{t} - \sum_{i=1}^{n} \lambda_j \mathbf{b}_j$ για κάποια $\lambda_j \in \mathbb{Z}$. Άρα, πράγματι το $\mathbf{t} - \mathbf{b} \in \mathcal{L}$.

Η ιδέα του αλγορίθμου.

Περιγράφουμε στην διάσταση 2. Έστω ε_k η ευθεία που είναι παράλληλη στο διάνυσμα $\mathbf{b}_1$ και περνάει από το σημείο $k\mathbf{b}_2^*$. Δηλαδή, $\varepsilon_k = k\mathbf{b}_2^* + \text{span}(\mathbf{b}_1)$. Αρχικά, περιορίζουμε το διάνυσμα $\mathbf{t}$ ανάμεσα σε δύο ευθείες ε_c και ε_{c+1} (ή ε_{c-1}). Διαλέγουμε το c έτσι ώστε $\text{dist}(\varepsilon_c, \mathbf{t}) \leq \frac{||\mathbf{b}_2^*||}{2}$. Το σχέδιο είναι να βρω ένα σημείο του πλέγματος επί της ε_c ώστε να είναι πιο κοντά στο $\mathbf{t}$. Αρχικά παρατηρούμε ότι η ευθεία ε_c (υπερεπίπεδο στις μεγαλύτερες διαστάσεις) περιέχει σημεία του πλέγματος. Π.χ. το σημείο $c\mathbf{b}_2$ είναι σημείο της ε_c.

Λήμμα 8.7.1. $c\mathbf{b}_2 \in \varepsilon_c$.

Απόδειξη. $\mathbf{b}_2^* = \mathbf{b}_2 - \mu_{2,1}\mathbf{b}_1$. Επομένως,

$$c\mathbf{b}_2 = c\mathbf{b}_2^* + c\mu_{2,1}\mathbf{b}_1 \in c\mathbf{b}_2^* + \text{span}(\mathbf{b}_1) = \varepsilon_c.$$

□

Κεφάλαιο 9

Συστήματα που βασίζονται σε πλέγματα

Το πλεονέκτημα αυτών των συστημάτων είναι ότι συνήθως έχουμε αποδείξεις ασφάλειας που βασίζονται σε δύσκολα προβλήματα των πλεγμάτων στην χειρότερη περίπτωση. Ειδικότερα, οι αναγωγές αυτές είναι από την μέση-περίπτωση στην χειρότερη περίπτωση (average case to worst case).

Στα αρχικά στάδια της ανάπτυξης κρυπτοσυστημάτων δημόσιου κλειδιού, πολλοί ερευνητές προσπάθησαν να φτιάξουν κρυπτοσυστήματα που βασίζονται σε δύσκολα προβλήματα, ειδικότερα σε προβλήματα της κλάσης NP-complete. Ένα τέτοιο παράδειγμα είναι το κρυπτοσύστημα που βασίζεται στο πρόβλημα υποσυνόλου (Subset Sum). Σε κάθε περίπτωση, αυτά τα συστήματα δεν ήταν πετυχημένα. Αυτό διότι ήταν δύσκολα στην χειρότερη περίπτωση και όχι κατά μέσο όρο. Δηλ. αν διαλέξω τυχαία ένα πρόβλημα subset sum πολύ πιθανό να μπορώ να το λύσω εύκολα.

Μια σπουδαία εργασία του Miklos Ajtai (1996), ήρθε και κατά κάποιο τρόπο έλυσε το προηγούμενο πρόβλημα. Βρήκε μια κατηγορία προβλημάτων, που είναι κατά μέσο όρο δύσκολα (average case hard problems) και απέδειξε ότι η επίλυση τους οδηγεί στην επίλυση ενός προβλήματος που είναι δύσκολο στην χειρότερη περίπτωση. Αυτή η εργασία οδήγησε σε κρυπτοσυστήματα, συναρτήσεις κατακερματισμού, ψηφιακές υπογραφές και συστήματα ταυτοποίησης (id-schemes) που βασίζονται σε προβλήματα των πλεγμάτων που είναι δύσκολα στην χειρότερη περίπτωση.

Οι περισσότερες εργασίες που αφορούν σε συστήματα που βασίζονται σε πλέγματα, είναι αποδείξεις του τύπου average-worst case reductions. Δηλ. οι αποδείξεις αυτές μετασχηματίζουν μια επιτυχημένη επίθεση στο κρυπτοσύστημα, σε έναν αποδοτικό αλγόριθμο που λύνει οποιοδήποτε παράδειγμα ενός προβλήματος στα πλέγματα στην χειρότερη περίπτωση. Η έκφραση απόδειξη ασφαλείας είναι παραπλανητική και αυτό που εννοούμε είναι αναγωγή της ασφάλειας του κρυπτοσυστήματος (security reduction) σε ένα πρόβλημα που θεωρείται δύσκολο στην χειρότερη περίπτωση.

Επίσης, υπάρχουν κβαντικές επιθέσεις σε μερικά προβλήματα που βασίζονται σε πλέγματα. Η κβαντική επίθεση των Campbell-Groves-Shepherd μπορεί να εφαρμοστεί στο σύστημα δημόσιου κλειδιού Soliloquy. Αυτή η επίθεση μπορεί να εφαρμοστεί και στο ομομορφικό σύστημα των Smart-Vercauteren καθώς και

στην πλειότιμη συνάρτηση (multiavriate map) των Garg-Gentry-Halevi[43]. Τα τρία προηγούμενα συστήματα ανήκουν στην κατηγορία των ideal/lattice based cryptosystems.

Εξαιρετικό ενδιαφέρων παρουσιάζει η κβαντική επίθεση των Eldar-Shor [44]. Όταν παρουσιάστηκε αυτή η επίθεση, πολλοί ερευνητές ασχολήθηκαν προσεκτικα με την προηγούμενη εργασία και πολύ σύντομα βρέθηκε ένα κενό στη μέθοδο τους από τον Oded Regev. Φυσικά η εργασία αποσύρθηκε. Μερικοί, πιστεύουν [45] ότι αυτό ήταν ικανό να δημιουργήσει αμφιβολίες για τα συστήματα που βασίζονται σε πλέγματα, ώστε η Google να διακόψει το πρόγραμμα CECPQ[46] το οποίο αφορούσε σε ένα σύστημα ανταλλαγής κλειδιών που βασίζεται σε πλέγματα (τα συστήματα X25519 και NEW HOPE). Ο NEWHOPE είναι Ring LWE τύπου σύστημα ανταλλαγής κλειδιού και σε περίπτωση που ένας κβαντικός υπολογιστής έσπαζε αυτό το σύστημα ο αλγόριθμος X25519 εγγυάται τουλάχιστον την κλασική ασφάλεια του συστήματος.

9.1 Learning With Errors (LWE)

Το πρόβλημα Learning With Errors (LWE) μας δίνει μια σειρά από κρυπτοσυστήματα (ομομορφικά και μη) ανθεκτικά σε κβαντικές επιθέσεις. Πρώτη φορά παρουσιάστηκε από τον Regev το 2005.[47] Ας ξεκινήσουμε με ένα απλό πρόβλημα, το πρόβλημα επίλυσης ενός γραμμικού συστήματος σε ένα πεπερασμένο σώμα $\mathbb{Z}_q$. Έστω $q = 11$ και $n = 3$, και αναζητούμε λύση $\mathbf{s} \in \mathbb{Z}_{11}^3$, του συστήματος

$$\begin{cases} 2s_1 + 3s_2 + 4s_3 &= 5 \\ 4s_1 + s_2 + s_3 &= 8 \\ -s_1 + 2s_2 - s_3 &= 2 \end{cases}$$

Με αναγωγή Gauss, εύκολα βρίσκουμε $\mathbf{s} = (1, 2, 2)$. Ας θεωρήσουμε κάποια τυχαία στοιχεία του σύνολου $\mathbb{Z}_{11}$. Έστω $e_1, e_2, e_3 \in \{-1, 0, 1\}$. Επίσης ας υποθέσουμε ότι δεν γνωρίζουμε τα e_1, e_2, e_3. Τότε το νέο σύστημα

$$\begin{cases} 2s_1 + 3s_2 + 4s_3 + e_1 &= 5 \\ 4s_1 + s_2 + s_3 + e_2 &= 8 \\ -s_1 + 2s_2 - s_3 + e_3 &= 2 \end{cases}$$

είναι πιο δύσκολο να λυθεί. Ενώ το κλασικό σύστημα είναι εύκολο να λυθεί, το *θορυβώδες* σύστημα (noisy system) είναι λίγο πιο δύσκολο. Θέτουμε $\mathcal{A}_{\mathbf{s},\chi}$ την κατανομή πιθανότητας που διαλέγει τυχαία $\mathbf{a}_i \in \mathbb{Z}_q^n$ και e_i σύμφωνα με την κατανομή χ από το σύνολο $\mathbb{Z}_q$ $(i = 1, 2, ..., k)$ και εξάγει τα $k-$ζεύγη

$$(\mathbf{a}_i, \mathbf{a}_i \cdot \mathbf{s} + e_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q.$$

[43] http://web.eecs.umich.edu/~cpeikert/soliloquy.html
[44] https://arxiv.org/abs/1611.06999
[45] https://groups.google.com/forum/#!topic/cryptanalytic-algorithms/WNMuTfJuSRc
[46] https://en.wikipedia.org/wiki/CECPQ1
[47] Oded Regev, On Lattices, Learning with Errors, Random Linear Codes, and Cryptography, STOC 2005

Ορισμός 9.1.1. *(LWE ή Search-LWE) Λέμε ότι ένας αλγόριθμος λύνει το πρόβλημα LWE με modulus q και κατανομή λάθους χ, αν για κάθε $\mathbf{s} \in \mathbb{Z}_q^n$, δοθέντων αυθαίρετου πλήθους k δειγμάτων από την κατανομή $\mathcal{A}_{\mathbf{s},\chi}$, εξάγει το $\mathbf{s}$.*

Οι παράμετροι του LWE είναι η τετράδα (n, k, q, χ) όπου n, k θετικοί ακέραιοι, $q = q(n)$ και χ μια κατανομή επί του $\mathbb{Z}_q$, δηλ. $LWE_{n,k,q,\chi}$. Αν $q = 2$ και χ η κατανομή Bernouli, δηλ. $\chi(0) = \tau,\ \chi(1) = 1 - \tau$ $(\tau \in (0, 1))$, τότε το πρόβλημα LWE ονομάζεται LPN : **L**earning **P**arity with **N**oise problem.[48]

Ορίζουμε επίσης και το πρόβλημα απόφασης.

Ορισμός 9.1.2. *(DLWE ή Desicion-LWE) Λέμε ότι ένας αλγόριθμος λύνει το πρόβλημα DLWE με modulus q και κατανομή λάθους χ, αν δοθέντων αυθαίρετου πλήθους k δειγμάτων, εξάγει True αν το δειγμα προέρχεται από την $\mathcal{A}_{\mathbf{s},\chi}$ για κάποιο $\mathbf{s}$, διαφορετικά False.*

Τα δύο προβλήματα SLWE και DLWE είναι πολυωνυμικά ισοδύναμα για την περίπτωση που ο q είναι γινόμενο πρώτων αριθμών $q_1 q_2 \cdots q_n$ όπου $q_i = poly(n)$.

Πριν δούμε την δυσκολία αυτού του προβλήματος χρειαζόμαστε έναν ορισμό για ένα κλασικό πρόβλημα στα πλέγματα.

Ορισμός 9.1.3. *(SIVP) Έστω πλέγμα $\mathcal{L}$ διάστασης n. Το πρόβλημα SIVP: Shortest Independent Vectors Problem, είναι η εύρεση n γραμμικά ανεξάρτητων διανυσμάτων $\mathbf{x}_i$ του $\mathcal{L}$ τέτοια ώστε, η ποσότητα $B = \max ||\mathbf{x}_i||$ να γίνεται ελάχιστη. Αν αντί του B θεωρήσουμε το $B_\gamma = \gamma B$ $(\gamma \in \mathbb{R}_{>1})$ τότε, έχουμε το προσεγγιστικό SIVP που συμβολίζεται $SIVP_\gamma$.*

Γενικότερα, ο Regev απόδειξε ότι αν έχουμε ένα μαντείο LWE, τότε υπάρχει κβαντικός αλγόριθμος που λύνει το $SIVP_\gamma$. Δηλαδή, έχουμε μια κβαντική αναγωγή του $SIVP_\gamma$ στο πρόβλημα LWE. Επομένως, το πρόβλημα LWE είναι τόσο δύσκολο, όσο η χειρότερη περίπτωση του προβλήματος $SIVP_\gamma$, για $\gamma = n/\alpha$ και $\alpha = \sigma\sqrt{2\pi}/q$. Το τελευταίο για $\gamma = O(1)$ είναι NP-hard.

Ο Peikert το 2009, απέδειξε ότι υπάρχει και κλασική αναγωγή (όχι κβαντική) για την περίπτωση που το $q = O(2^n)$. Το 2013, αποδείχτηκε και κλασική αναγωγή[49] όταν $q = O(poly(n))$. Οπότε η μελέτη της δυσκολίας αυτού του προβλήματος έχει απαντηθεί ικανοποιητικά.

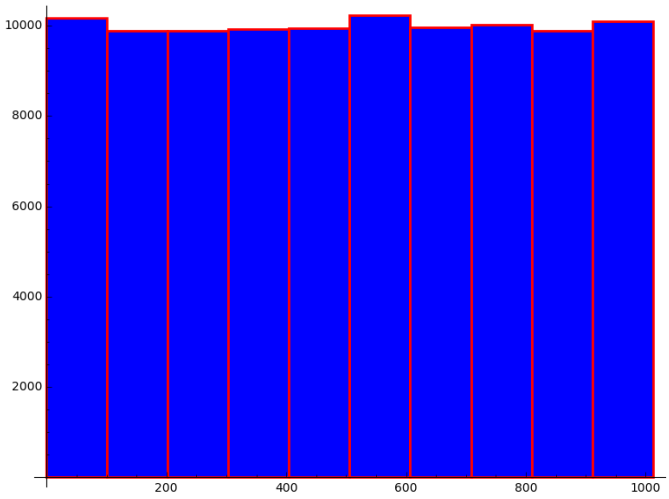
Θεώρημα 9.1.1 Έστω $q = poly(n)$ και $aq > \sqrt{n}$ $(\alpha \in (0, 1))$. Η λύση του LWE_q συνεπάγεται μια λύση σε ένα δύσκολο πρόβλημα των πλεγμάτων στην χειρότερη περίπτωση.

Η κατανομή χ είναι συνήθως η (διακριτή) κανονική κατανομή του Gauss επί του $\mathbb{Z}_q$, με τυπική απόκλιση αq και $\alpha > 1/poly(n)$.

Μπορούμε να ελέγξουμε εμπειρικά την υπόθεση του DLWE κάνοντας το εξής πείραμα. Διαλέγουμε από την ομοιόμορφη κατανομή $r = 100000$ διανύσματα από το $\mathbb{Z}_q^n$. Ας πάρουμε $q = 1013$ και $n = 10$. Επίσης, διαλέγουμε r ακέραιους από την διακριτή Gaussian με $\sigma = q/2$. Αν κάνουμε το ιστόγραμμα του συνόλου των ακεραίων $\mathbf{a}_i \cdot \mathbf{s} + e_i \pmod q$, παίρνουμε το ιστόγραμμα στο σχήμα (9).

[48] A. Blum, M. L. Furst, M. J. Kearns and R. J. Lipton, Cryptographic Primitives Based on Hard Learning Problems, Crypto 1993

[49] Brakerski et al. Classical Hardness of Learning with Errors



Σχήμα 9: (εμπειρική) επαλήθευση της υπόθεσης LWE.

Επιθέσεις. Μια απλή επίθεση στο LWE είναι η εξής. Αν έχω ένα LWE μαντείο (δηλ. μια κατανομή $\mathcal{A}_{\mathbf{s},\chi}$), τότε ζητάω $poly(n)$ δείγματα. Αν τύχει ένα ζευγάρι $(\mathbf{a}, b)$ με $\mathbf{a} = (1, 0, ..., 0)$ τότε, έχω την εξίσωση $b = \mathbf{a} \cdot \mathbf{s} + e = s_1 + e$. Επομένως, μπορώ να βρω το $s_1 = b - e \approx b$. Η πιθανότητα να πετύχει αυτή η επίθεση είναι q^{-n}. Άρα έχω πολυπλοκότητα $O(2^{n \log n})$.

Μια άλλη επίθεση είναι να ζητήσω $n-$δείγματα και να εφαρμόσω αναγωγή Gauss στον επαυξημένο $[A|\mathbf{b}]$. Τότε η προσεγγιστική λύση που θα βρω (έχω $n \times n$ γραμμικό σύστημα) είναι τελικά η ζητούμενη.

Επίσης υπάρχουν επιθέσεις που βασίζονται σε πλέγματα, όπως η επίθεση που βασίζεται στο BDD : **B**ounded **D**istance **D**ecoding problem. Χρειαζόμαστε τον παρακάτω ορισμό.

Ορισμός 9.1.4. *Το BDD_α με είσοδο ένα πλέγμα L και ένα διάνυσμα $\mathbf{t}$ με $d(L, \mathbf{t}) \leq \alpha\lambda_1(L)$ ζητάει την εύρεση του διανύσματος $\mathbf{t}$.*

Αν το $\alpha \geq 1/2$ είναι NP-complete αλλά άγνωστο για την περίπτωση $\alpha < 1/2$. Επίσης υπάρχουν επιθέσεις που βασίζονται στη απαρίθμηση των KFP. Τέλος υπάρχει η συνδυαστική επίθεση (BKW) που ανάγει το DLWE στο SIS : **S**hort **I**nteger **S**olution problem[50]. Εμείς θα δούμε αναλυτικά την επίθεση που ανάγει το πρόβλημα μας στο BDD_α, διότι βάσει αυτής της επίθεσης θα διαλέξουμε τις παραμέτρους μας.

Γενικά, από την κλάση των προβλημάτων average case χρησιμοποιούμε το SIS και LWE τα οποία έχουν αναγωγές σε δύσκολα προβλήματα των πλεγμάτων (στην χειρότερη περίπτωση). Οπότε και είναι υποψήφια για κρυπτοσυστήματα, με το δεύτερο να έχει το πλεονέκτημα ότι παράγει και ομομορφικά κρυπτοσυστήματα.

[50] Fitzpatrick, Some Algorithms for Learning with Errors, Thesis (2014)

Επίσης, υπάρχουν συναρτήσεις κατακερματισμού που βασίζονται στο SIS και συστήματα ταυτοποίησης (id-schemes).

Μπορούμε να κατασκευάσουμε ένα ομομορφικό σύστημα κρυπτογράφησης που βασίζεται στο πρόβλημα LWE. Το σύστημα που θα παρουσιάσουμε είναι συμμετρικό αλλά μπορεί εύκολα να μετατραπεί σε ένα σύστημα δημόσιου κλειδιού.

9.2 Ένα LWE-ομομορφικό κρυπτοσύστημα (BV11)

Το σύστημα που θα παρουσιάσουμε δημιουργήθηκε από τους Brakerski - Vaikuntanathan. Διαλέγουμε q, n όπως προηγουμένως και χ η διακριτή κατανομή του Gauss, $D_{\mathbb{Z},\sigma}$, με κέντρο 0 και τυπική απόκλιση σ. Έστω $\mathbf{s} \in \mathbb{Z}_q^n$ το μυστικό κλειδί. Διαλέγουμε θετικό ακέραιο $t \ll q$ και υποθέτουμε ότι ο χώρος μηνυμάτων $\mathcal{M} \subset \mathbb{Z}_q^n \times \mathbb{Z}_t$. Η Alice επιθυμεί να στείλει ένα μήνυμα m στον Bob. Για κάθε μήνυμα m, επιλέγουμε τυχαία ένα διάνυσμα $\mathbf{a} \in \mathbb{Z}_q^n$ και e από την κατανομή $D_{\mathbb{Z},\sigma}$. Επεκτείνουμε το μήνυμα m ως $(\mathbf{a}, m) \in \mathcal{M}$. Τότε,

$$\mathbf{c} = Enc_{\mathbf{s}}(m) = (\mathbf{a}, \mathbf{a} \cdot \mathbf{s} + te + m) = (\mathbf{a}, b) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$$

και

$$Dec_{\mathbf{s}}(c) = (b - \mathbf{a} \cdot \mathbf{s} \pmod t) = (te + m) \pmod t = m \pmod t = m.$$

Καθώς το σύστημα είναι συμμετρικό η Alice γνωρίζει το κλειδί αποκρυπτογράφησης $\mathbf{s}$. Η κρυπτογράφηση είναι πιθανοτική (και όχι ντετερμινιστική) δηλ. για το ίδιο μήνυμα γενικά παίρνουμε διαφορετικό κρυπτογραφημένο μήνυμα. Η ασφάλεια του LWE δεν επηρεάζεται αν αντί του $\mathbf{a} \cdot \mathbf{s} + e$ χρησιμοποιήσουμε $\mathbf{a} \cdot \mathbf{s} + te + m$.

Για κάθε κρυπτογραφημένο μήνυμα ορίζεται η γραμμική συνάρτηση $\mathbb{Z}_q^n \to \mathbb{Z}_q$

$$f_{\mathbf{a},b}(\mathbf{x}) = b - \mathbf{a} \cdot \mathbf{x} = b - \sum_{i=1}^{n} a_i x_i.$$

Η αποκρυπτογράφηση τότε γράφεται

$$Dec_{\mathbf{s}}(\mathbf{c}) = f_{\mathbf{a},b}(\mathbf{s}) \pmod t.$$

Αλγόριθμος 9.2.1. : Encryption-LWE
Παράμετροι. $q, n, t, \chi = D_{\mathbb{Z},\sigma}$ (discrete Gaussian). Επίσης θεωρούμε ως πλήρες σύστημα αντιπροσώπων του $\mathbb{Z}_q$ το σύνολο $\{0, 1, ..., q-1\}$.
Είσοδος. Ένα μήνυμα $m \in \mathbb{Z}_q$ και το μυστικό κλειδί $\mathbf{s} \in \mathbb{Z}_q^n$
Έξοδος. Το κρυπτογραφημένο μήνυμα $c = enc(m) \in \mathbb{Z}_q^{n+1}$.

1 $e \leftarrow D_{\mathbb{Z},\sigma}^n$
2 $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_q^n$
3 $enc(m) = \big(\mathbf{a}, \mathbf{a} \cdot \mathbf{s} + te + m \pmod q\big)$
4 **return** $enc(m)$

Η αποκρυπτογράφηση γίνεται ως εξής:

Αλγόριθμος 9.2.2. : Decryption-LWE
Παράμετροι. $q, n, t, \chi = D_{\mathbb{Z},\sigma}$ (discrete Gaussian)
Είσοδος. Το κρυπτογραφημένο μήνυμα $\mathbf{c} = (\mathbf{a}, b) \in \mathbb{Z}_q^{n+1}$ και το μυστικό κλειδί $\mathbf{s} \in \mathbb{Z}_q^n$
Έξοδος. Το μήνυμα $m = dec(\mathbf{c}) \in \mathbb{Z}_q$.

1 $dec(\mathbf{c}) = b - \mathbf{a} \cdot \mathbf{s} \pmod t$
2 **return** $dec(\mathbf{c})$

Παράδειγμα 9.2.1. Αρχικά, επιλέγουμε τις παραμέτρους (n, q, t, σ). Στην πράξη το e είναι μικρό. Χρησιμοποιούμε $\sigma = 3$ και κέντρο 0 για την κατανομή του Gauss. Έστω $(n, q, t) = (4, 2^{32} + 15, 2^{20})$. Το μυστικό κλειδί

$$\mathbf{s} = (3305290141, 418184802, 2955413319, 1604802104) \in \mathbb{Z}_q^4.$$

Αυτό είναι σταθερό για όλα τα μηνύματα.

Κάθε φορά που κρυπτογραφούμε, διαλέγουμε το $(\mathbf{a}, e)$. Ας είναι το $e = 3$ και επιλέγουμε το $\mathbf{a}$ τυχαία από το σύνολο $\mathbb{Z}_q^4 = \{-(q-1)/2, ..., 0, ..., (q-1)/2\}^4$ (o q είναι πρώτος). Διαλέξαμε αυτό το σύστημα αντιπροσώπων του $\mathbb{Z}_q$ για να μπορούμε να επιλέγουμε και αρνητικούς αριθμούς ως μηνύματα. Ας είναι

$$\mathbf{a} = (1211599272, -718252110, -1325070467, -1050322899).$$

Έστω ότι θέλουμε να κρυπτογραφήσουμε τον αριθμό $-5 \in \mathbb{Z}_q$. Αν η Alice επιθυμεί να στείλει αυτό το μήνυμα στον Bob , υπολογίζει

$$c = Enc(m) = (\mathbf{a}, \mathbf{a} \cdot \mathbf{s} + te + m \pmod q) = (\mathbf{a}, 1474618740 = b).$$

Ο ακέραιος αυτός b είναι $< q/2$ οπότε τον αφήνουμε ως έχει. Αν ήταν $> q/2$ τότε θα παίρναμε $b - q$ (θα ήταν ένας αρνητικός ακέραιος $> (q-1)/2$).

Τώρα για την αποκρυπτογράφηση (αν υποθέσουμε ότι ο Bob γνωρίζει το μυστικό κλειδί $\mathbf{s}$) ο Bob υπολογίζει τον αριθμό

$$b - \mathbf{a} \cdot \mathbf{s} \pmod t.$$

Επειδή $b - \mathbf{a} \cdot \mathbf{s} \pmod t = 1048571 > t/2$ έχουμε

$$Dec(c) = 1048571 - t = 1038571 - 2^{20} = -5.$$

Ομομορφικές ιδιότητες του συστήματος.
Έστω ότι ο Bob έχει στα χέρια του δύο κρυπτογραφημένα μηνύματα $\mathbf{c}_1, \mathbf{c}_2$ και έστω ότι θέλει να τα προσθέσει. Τότε υπολογίζει το άθροισμα των διανυσμάτων $\mathbf{c}_1 + \mathbf{c}_2 = (\mathbf{a}_1 + \mathbf{a}_2, b_1 + b_2) = (\mathbf{a}, b)$ και στέλνει στην Alice τις συντεταγμένες του αθροίσματος. Κατόπιν, η Alice υπολογίζει το πολυώνυμο

$$F = (b - (\mathbf{a} \cdot \mathbf{x})) \in \mathbb{Z}_q[x_1, ..., x_n].$$

Το οποίο γράφεται:

$$F = b_1 + b_2 - (\mathbf{a}_1 + \mathbf{a}_2) \cdot \mathbf{x}.$$

Τέλος, υπολογίζει την τιμή του στο $\mathbf{s}$, modulo t,

$$F(\mathbf{s}) \pmod t = (b_1 + b_2 - (\mathbf{a}_1 + \mathbf{a}_2) \cdot \mathbf{s}) \pmod t =$$

$$(b_1 - \mathbf{a}_1 \cdot \mathbf{s} + b_2 - \mathbf{a}_2 \cdot \mathbf{s}) \pmod t =$$

$$te_1 + m_1 + te_2 + m_2 \pmod t = m_1 + m_2.$$

Το $F(\mathbf{x}) = f_{\mathbf{a}_1+\mathbf{a}_2, b_1+b_2}(\mathbf{x}) = f_{\mathbf{a}_1,b_1}(\mathbf{x}) + f_{\mathbf{a}_2,b_2}(\mathbf{x})$. Δηλ. το σύστημα είναι ομομορφικό ως προς την πρόσθεση. Παρατηρούμε ότι το μήκος του κρυπτογραφημένου μηνύματος είναι $n+1$ όσο και το μήκος του μηνύματος.

Αν είχαμε πολλά μηνύματα $m_1, ..., m_k$, τότε αρκεί να γίνει ο προηγούμενος υπολογισμός για το διάνυσμα,

$$\Big(\sum_{i=1}^{k} \mathbf{c}_i, \sum_{i=1}^{k} b_i\Big).$$

Δηλαδή, το σύστημα μας επιτρέπει όσες προσθέσεις θέλουμε και οι υπολογισμοί γίνονται αποδοτικά (αφού δεν αυξάνει το μήκος του κρυπτογραφημένου μηνύματος μετά την ομομορφική πράξη).

Δυστυχώς, για τον πολλαπλασιασμό δεν ισχύει κάτι αντίστοιχο, αλλά μόνο $\log n$ πολλαπλασιασμοί μπορούν να γίνουν αποδοτικά. Τώρα ο Bob στέλνει τους συντελεστές του πολυωνύμου

$$f_{\mathbf{a}_1,b_1}(\mathbf{x}) \cdot f_{\mathbf{a}_2,b_2}(\mathbf{x}) =$$

$$h_0 + h_1 x_1 + \cdots + h_n x_n + \sum_{i,j} h_{i,j} x_i x_j \in \mathbb{Z}_q[x_1, ..., x_n].$$

Δηλ. στέλνει το διάνυσμα

$$(h_0, h_1, ..., h_n, h_{11}, ..., h_{nn}) \in \mathbb{Z}_q^{(n+1)(n+2)/2}.$$

Σε αυτήν την περίπτωση η Alice υπολογίζει το πολυώνυμο G,

$$G = h_0 + h_1 x_1 + \cdots + h_n x_n + \sum_{i,j} h_{i,j} x_i x_j$$

και υπολογίζει την τιμή του στο $\mathbf{s}$, modulo t,

$$G(\mathbf{s}) \pmod t = f_{\mathbf{a}_1,b_1}(\mathbf{s}) \cdot f_{\mathbf{a}_2,b_2}(\mathbf{s}) \pmod t =$$

$$(b_1 - a_1 \cdot \mathbf{s}) \cdot (b_1 - a_1 \cdot \mathbf{s}) \pmod t = m_1 \cdot m_2 \pmod t = m_1 \cdot m_2.$$

Για να ισχύει η τελευταία ισότητα πρέπει $m_1 m_2 < t$. Αρκεί να διαλέξουμε $m_1, m_2 \in \mathbb{Z}_{\lceil \sqrt{t} \rceil}$. Δηλ. το σύστημα είναι ομομορφικό και ως προς τον πολλαπλασιασμό. Το μήκος του διανύσματος που στέλνει ο Bob στην Alice είναι ίσο με το πλήθος των συντελεστών του πολλαπλασιασμού των δύο πολυώνυμων, δηλαδή $(n+1)(n+2)/2$. Δηλαδή, ο πολλαπλασιασμός, αν και ομομορφικός, δεν είναι μια αποδοτική διαδικασία για την Alice , αφού θα χρειαστεί να υπολογίσει $(n+1)(n+2)/2$ γινόμενα. Αν πολλαπλασιάζαμε τρεις αριθμούς, τότε το πλήθος των συντελεστών θα γινόταν $O(n^3)$. Για μια μέτρια επιλογή του n μπορούμε να πούμε ότι το σύστημα μπορεί να κάνει αποδοτικά όσες προσθέσεις θέλουμε και έναν πολλαπλασιασμό.

Για να αποφύγουμε αυτή την *εκτόξευση* του πλήθους των συντελεστών χρησιμοποιούμε μια τεχνική που ονομάζεται Relinearization. Η Alice διαλέγει ένα τυχαίο $\mathbf{r} \in \mathbb{Z}_q^n$ και κρυπτογραφεί τις συντεταγμένες του μυστικού κλειδιού $\mathbf{s} = (s_1, ..., s_n)$ καθώς και τους αριθμούς $s_i s_j$ με χρήση του κλειδιού $\mathbf{r}$. Δηλαδή, $Enc_{\mathbf{r}}(s_i) = (\mathbf{a}_i, b_i)$, $Enc_{\mathbf{r}}(s_{ij}) = (\mathbf{a}_{ij}, b_{ij})$, όπου

$$\mathbf{a}_i = (a_{ik})_k, b_i = \mathbf{a}_i \cdot \mathbf{r} + te_i + s_i, \quad \mathbf{a}_{ij} = (a_{ijk})_k, b_{ij} = \mathbf{a}_{ij} \cdot \mathbf{r} + te_{ij} + s_i s_j. \quad (9.2.1)$$

Υποθέτουμε ότι η Alice δημοσιεύει τις προηγούμενες κρυπτογραφήσεις (n^2+n το πλήθος).

Λήμμα 9.2.1. *Έστω ότι η* Alice κρυπτογραφεί δύο αριθμούς $m_1, m_2 \in \mathbb{Z}_t$ ($m_1 m_2 < t$), με το κλειδί $\mathbf{s}$ και τους στέλνει στον Bob . Άρα ο Bob κατέχει τα

$$Enc_{\mathbf{s}}(m_1) = (\mathbf{A}_1, B_1), Enc_{\mathbf{s}}(m_2) = (\mathbf{A}_2, B_2)$$

καθώς και τις κρυπτογραφήσεις (9.2.1). Κατόπιν, ο Bob υπολογίζει το γινόμενο των πολυωνύμων

$$f_{\mathbf{A}_1, B_1}(\mathbf{x}) \cdot f_{\mathbf{A}_2, B_2}(\mathbf{x}) = h_0 + h_1 x_1 + \cdots + h_n x_n + \sum_{i,j} h_{i,j} x_i x_j$$

και στέλνει στην Alice το διάνυσμα του $\mathbb{Z}_q^{n+1}$,

$$(H_0, H_1, ..., H_n) =$$

$$\Big(h_0 + \sum_{i=1}^{n} h_i b_i + \sum_{i=1}^{n} h_{ij} b_{ij}, \sum_{i=1}^{n} h_i a_{i,1} + \sum_{i,j} h_{ij} a_{i,j,1}, \ldots, \sum_{i=1}^{n} h_i a_{i,n} + \sum_{i,j} h_{ij} a_{i,j,n}\Big).$$

Τότε, η αποκρυπτογράφηση από την Alice με το κλειδί $\mathbf{r}$ θα εμφανίσει το γινόμενο $m_1 m_2$.

Απόδειξη. Έστω, $G(x_1, ..., x_n) = H_0 + H_1 x_1 + \cdots + H_n x_n$. Τότε, $G(r_1, ..., r_n) = H_0 + H_1 r_1 + \cdots + H_n r_n$. Εξ ορισμού,

$$m_1 m_2 = f_{\mathbf{A}_1, B_1}(\mathbf{s}) f_{\mathbf{A}_2, B_2}(\mathbf{s}) \pmod t.$$

Αρκεί να αποδείξουμε ότι $G(\mathbf{r}) = m_1 m_2$. Αναπτύσσουμε το δεύτερο μέλος,

$$f_{\mathbf{A}_1, B_1}(\mathbf{s}) f_{\mathbf{A}_2, B_2}(\mathbf{s}) \pmod t =$$

$$h_0 + \sum_{i=1}^{n} h_i s_i + \sum_{i,j} h_{ij} s_i s_j \pmod t =$$

$$h_0 + \sum_{i=1}^{n} h_i (b_i - \mathbf{a}_i \cdot \mathbf{r} - te_i) + \sum_{i,j} h_{ij} (b_{ij} - \mathbf{a}_{ij} \cdot \mathbf{r} - te_{ij}) \pmod t =$$

$$h_0 + \sum_{i=1}^{n} h_i (b_i - \mathbf{a}_i \cdot \mathbf{r}) + \sum_{i,j} h_{ij} (b_{ij} - \mathbf{a}_{ij} \cdot \mathbf{r}) =$$

$$H_0 + H_1 r_1 + \cdots + H_n r_n = G(r_1, ..., r_n) = G(\mathbf{r}).$$

Άρα η Alice απλά υπολογίζει το G επί του $\mathbf{r}$.

□

Επομένως ο Bob στέλνει διάνυσμα μήκους $n+1$ αντί $(n+1)(n+2)/2$. Παρατηρούμε ότι χωρίς Relinearization η συνάρτηση αποκρυπτογράφησης $Dec : \mathbb{Z}_q^{(n+1)(n+2)/2} \to \mathbb{Z}_q$ ενώ με Relinearization είναι $Dec : \mathbb{Z}_q^{n+1} \to \mathbb{Z}_q$.

Το σύστημα όπως το παρουσιάσαμε χρησιμοποιεί μόνο μια εξίσωση. Μπορούμε εύκολα να γενικεύσουμε σε πίνακες διάστασης $d \times n$.

Όπως γράψαμε και προηγουμένως, το σύστημα μπορεί να κάνει ομομορφικά όσες προσθέσεις επιθυμούμε και αποδοτικά έναν πολλαπλασιασμό. Αν εφαρμόσουμε την προηγούμενη διαδικασία χρησιμοποιώντας μια αλυσίδα νέων κλειδιών $\mathbf{r}_1, ..., \mathbf{r}_L$, αντί ενός μόνο νέου κλειδιού $\mathbf{r}$ τότε, μπορούμε να πολλαπλασιάζουμε ομομορφικά $L+1$ αριθμούς αντί μόνο 2. Δηλαδή να έχει το κύκλωμα (circuit) υπολογισμού του Bob , βάθος $L-$ πολλαπλασιασμούς. Αν το $L \approx \varepsilon \log n$ ($\varepsilon > 0$), τότε μπορούμε να πετύχουμε ένα αποδοτικό ομομορφικό σύστημα (όχι πλήρως ομομορφικό fully homomorphic) που να επιτρέπει πολλαπλασιασμούς *βάθους* L. Στη συνέχεια πρέπει να χρησιμοποιηθεί η διαδικασία του Bootstraping για να κατασκευάσουμε ένα πλήρως ομομορφικό σύστημα fully homomorphic scheme. Τέτοια συστήματα, σαν αυτό που μόλις παρουσιάσαμε, ονομάζονται σχεδόν ομομορφικά (somewhat homomorphic). Στην πράξη όμως, το προηγούμενο σύστημα, στην μορφή που είναι μας ικανοποιεί για απλούς υπολογισμούς (π.χ. το τετράγωνο της απόστασης δύο σημείων του $\mathbb{R}^n$).

Επιλογή παραμέτρων Η επιλογή παραμέτρων σε ένα κρυπτοσύστημα είναι πολύ σημαντική διότι, η λάθος επιλογή παραμέτρων μπορεί να οδηγήσει στη κατάρρευση του κρυπτοσυστήματος. Συνήθως, η επιλογή γίνεται με το σκεπτικό να είναι το σύστημα ανθεκτικό στις επιθέσεις που είναι γνωστές.

Η επιλογή των παραμέτρων στο LWE γίνεται έτσι ώστε να οδηγούμαστε σε συστήματα ανθεκτικά στις επιθέσεις των Miccianchio-Regev[51] και Lindner-Peikert (LP)[52]. Η πρώτη επίθεση είναι η κατασκευή ενός διαχωριστή (distinguisher) που χρησιμοποιεί το SIS-problem για να λύσει το πρόβλημα απόφασης LWE (DLWE), ενώ στην δεύτερη επίθεση γίνεται μια επίθεση επί του BDD-problem σε κατάλληλο

[51] Miccianchio-Regev, 2009, https://eprint.iacr.org/2011/405.pdf
[52] Lindner-Peikert, 2011, http://web.eecs.umich.edu/ cpeikert/pubs/lwe-analysis.pdf

$q-$αδικό πλέγμα $\Lambda_q(A)$, και σε περίπτωση επιτυχίας λύνουμε το πρόβλημα εύρεσης LWE (SLWE). Θα ακολουθήσουμε την προσέγγιση LP για να εκτιμήσουμε τις παραμέτρους μας. Να πούμε εδώ ότι η εκδοχή RLWE : Ring LWE, οδηγεί σε πολύ μικρότερου μήκους κλειδιά και σε αποδοτικότερες πράξεις στον χώρο μηνυμάτων (που στην περίπτωση του κλασικού LWE είναι ο $\mathbb{Z}_q$ ενώ στην περίπτωση του RLWE είναι ό δακτύλιος $\mathbb{Z}_p[x]/\langle x^n+1\rangle$, για κάποιο πρώτο p.)

LP-προσέγγιση. Έστω ότι η Εύα έχει έναν LWE-oracle και με είσοδο $(M_1,...,M_m)$ το μαντείο απαντάει με $m-$δείγματα $(\mathbf{a}_i,b_i)\leftarrow \mathcal{A}_{\mathbf{s},\chi},\ i=1,2,...,m$. Ισχύουν οι εξισώσεις, $\mathbf{a}_i\cdot\mathbf{s}\equiv e_it+M_i \pmod q$. Επομένως, η Εύα έχει έναν πίνακα $A\in\mathbb{Z}_q^{n\times m}$ που έχει ως στήλες τα διανύσματα $\mathbf{a}_i$. Επίσης, $\mathbf{e}$ είναι το διάνυσμα στήλη που έχει ως στοιχεία τα e_i. Στόχος της Εύας είναι να βρει έναν αλγόριθμο που με είσοδο

$$(A, A^T\mathbf{s}+\mathbf{e} \pmod q),$$

να εξάγει το μυστικό κλειδί $\mathbf{s}$.

Ορίζουμε το πλέγμα

$$\Lambda_q(A^T)=\{\mathbf{z}\in\mathbb{Z}_q^m : \exists \mathbf{s}\in\mathbb{Z}_q^n, \mathbf{z}=A^T\mathbf{s} \pmod q\}\subset\mathbb{Z}_q^m.$$

Το πλέγμα αυτό είναι $m-$διάστατο και θα απαιτήσουμε να είναι αρκετά μεγάλο $m\approx 200$. Τότε, το $\mathbf{b}=(b_i)_{i=1,..,m}\in\mathbb{Z}_q^m$ είναι ένα σημείο *κοντά* στο $\Lambda_q(A^T)$. Για να αποκωδικοποιήσουμε το σημείο $\mathbf{b}$ εφαρμόζουμε την παρακάτω επίθεση.

1. Εύρεση μιας ανηγμένης βάσης $B=(\mathbf{b}_1,...,\mathbf{b}_n)$ του $\Lambda_q(A^T)$
2. Εφαρμογή του Babai για την εύρεση ενός διανύσματος $\mathbf{z}$ κοντά στο $\mathbf{b}$.

Αν διαλέξουμε όλα τα μηνύματα $M_1=M_2=\cdots=M_m=0$, τότε θα έχουμε την εξίσωση $\mathbf{b}=A^T\mathbf{s}+\mathbf{e}t$. Ορίζουμε B^* να είναι η Gram-Schmidt βάση της B και

$$\mathcal{P}_{t/2}(B^*)=\left\{\sum_{i=1}^n x_i\mathbf{b}_i^* : \frac{-t}{2}\le x_i<\frac{t}{2}\right\}.$$

Η επίθεση θα πετύχει αν και μόνο αν $\mathbf{b}-\mathbf{z}\in\mathcal{P}_{t/2}(B)$. Οι Lindner -Peikert προσέγγισαν την πιθανότητα επιτυχίας, δηλ. έδωσαν μια εκτίμηση για την πιθανότητα $Pr(\mathbf{z}:\mathbf{b}-\mathbf{z}\in\mathcal{P}_{t/2}(B))$. Προσέγγισαν την διακριτή κατανομή του Gauss με την συνεχή κατανομή του Gauss.

Λήμμα 9.2.2.

$$Pr(\mathbf{z}:\mathbf{b}-\mathbf{z}\in\mathcal{P}_{t/2}(B))=\prod_{i=1}^m erf\left(\frac{td_i||\mathbf{b}_i^*||\sqrt{\pi}}{2s}\right)$$

Η συνάρτηση

$$erf(x)=\frac{1}{\sqrt{\pi}}\int_{-x}^{x}e^{-x^2}\,dx$$

Αρχικά πρέπει να αποδεχθούμε κάποιες υποθέσεις σε σχέση με τα πλέγματα της μορφής $\Lambda_q(A)$. Η βασική υπόθεση είναι η GSA:Geometric Series Assumption

η οποία διατυπώθηκε από τον Schnorr. Αυτή η υπόθεση ελέγχθηκε σε τυχαία πλέγματα και τουλάχιστον πειραματικά επαληθεύει την υπόθεση GSA. Αυτή την υπόθεση την χρησιμοποιούμε για να εκτιμήσουμε το μήκος των διανυσμάτων $\mathbf{b}_i^*$ δεδομένου του παράγοντα Hermite δ.

Η επιλογή του m που πρότειναν οι Lindner-Peikert είναι η

$$m = \sqrt{\frac{n \log_2(q)}{\log_2(\delta)}}.$$

Κεφάλαιο 10

Συστήματα Ταυτοποίησης - id-schemes

Σε αυτά τα συστήματα η Alice θέλει να αποδείξει στον Bob ότι κατέχει ένα μυστικό, χωρίς φυσικά να το αποκαλύψει στον Bob. Με άλλα λόγια στο τέλος της συνομιλίας τους ο Bob θα πειστεί ότι στο άλλο άκρο της επικοινωνίας είναι η Alice εφόσον μόνο αυτή γνωρίζει το μυστικό. Το πιο απλό σύστημα ταυτοποίησης είναι το σύστημα των κωδικών (password hash scheme). Σε αυτά τα συστήματα η Alice θέλει να αποδείξει στην μηχανή την οποία συνδέεται ότι έχει πράγματι τα δικαιώματα που ισχυρίζεται ότι έχει, εφόσον μόνο η Alice γνωρίζει τον κωδικό. Όπως είδαμε ο κωδικός της Alice, έστω pw, είναι αποθηκευμένος στην μηχανή ως $hash(pw)$. Οπότε με την εισαγωγή του κωδικού, υπολογίζεται τοπικά το hash και κατόπιν στέλνεται και συγκρίνεται αυτή η τιμή με την τιμή της hash που είναι αποθηκευμένη στην (απομακρυσμένη) μηχανή. Ο στόχος της Εύας είναι να πλαστογραφήσει την ταυτότητα της Alice. Αν Εύα παρακολουθεί την επικοινωνία μπορεί να κλέψει την τιμή $hash(pw)$ και να την υποβάλει στον διακομιστή ως δική της (replay attack), χωρίς να χρειαστεί να βρεί τον αρχικό κωδικό της Alice. Συνήθως τα συστήματα που βασίζονται σε κωδικούς, για να αποφύγουν την προηγούμενη επίθεση χρησιμοποιούν και κάποια τυχαία bits στην επικοινωνία τους. Η πιο συνηθισμένη πρακτική είναι αυτά τα συστήματα να τρέχουν πάνω από το SSL/TLS.

Όπως είδαμε προηγουμένως όταν η Εύα έχει τον ρόλο του ωτακουστή, τότε μπορεί να παραβιάσει το σύστημα. Υπάρχουν συστήματα ταυτοποίησης που είναι ασφαλή σε τέτοιου τύπου επιθέσεις. Θα εξετάσουμε το σύστημα ταυτοποίησης του Schnorr. Στο σύστημα αυτό η Alice πρέπει να αποδείξει ότι γνωρίζει κάποιο μυστικό. Είναι ένα σύστημα τριών βημάτων (three move id-schemes). Η ελάχιστη ασφάλεια που απαιτούμε από τέτοια συστήματα είναι, καθώς η Εύα παρακολουθεί την επικοινωνία να μην μπορεί να αποκτήσει κάποιες πληροφορίες για το μυστικό κλειδί που κατέχει μόνο η Alice. Τα συστήματα τριών βημάτων μπορούν με μια απλή τροποποίηση να μετατραπούν σε ψηφιακές υπογραφές (μετασχηματισμός των Fiat-Shamir). Το σύστημα του Schnorr βασίζεται στον διακριτό λογάριθμο. Υπάρχουν άλλα συστήματα όπως το σύστημα ταυτοποίησης των Fiat-Shamir που βασίζεται στην δυσκολία υπολογισμού των τετραγωνικών ριζών mod n, όταν δεν γνωρίζουμε την παραγοντοποίηση του n. Επίσης, πιο νέα συστήματα ταυτοποίησης βασίζουν την ασφάλεια τους σε δύσκολα προβλήματα των πλεγμάτων. Το μειονέκτημα των τελευταίων είναι η μεγάλη πολυπλοκότητα όσον αφορά την ανταλλαγή

των δύο οντοτήτων. Τέλος συστήματα ταυτοποίησης όπως του Schnorr είναι κατάλληλα και για χρήση σε έξυπνες κάρτες (smart cards) αλλά όχι τσιπάκια όπως τα rfid-tags. Στα τελευταία, ενώ έχουν βρεθεί λύσεις όσον αφορά την συμμετρική κρυπτογραφία π.χ. ο αλγόριθμος humingbird και SEA. Στην κρυπτογραφία δημόσιου κλειδιού δεν υπάρχουν ανάλογοι (lightweight) αλγόριθμοι.

10.1 Σύστημα ταυτοποίησης του Schnorr

Έστω p, q πρώτοι αριθμοί τέτοιοι ώστε, $q|p-1$. Επίσης ο g είναι ένας γεννήτορας της ομάδας $\mathbb{Z}_q^*$.

Ιδιωτικό κλειδί : s τέτοιο ώστε $v = g^{-s} \pmod p$.
Δημόσιο κλειδί : $(p, q, g; v)$,
Πριν παρουσιάσουμε το σύστημα του Schnorr δίνουμε την παρακάτω απλοποίηση.
1ο βήμα. Η Alice διαλέγει έναν τυχαίο αριθμό $r \in \{1, 2, ..., q-1\}$ και στέλνει στον Bob $x = g^r \pmod p$.
2ο βήμα. Ο Bob διαλέγει ένα τυχαίο αριθμό $e \in \{1, 2\}$ και τον στέλνει στην Alice
3ο βήμα. Η Alice υπολογίζει τον αριθμό $y = r + es (\text{mod } q)$ και το στέλνει στον Bob.
Ο Bob κάνει την επαλήθευση (verification) ελέγχοντας $g^y v^e \pmod p = x$.
Πράγματι (οι πράξεις στο $\mathbb{Z}_p$), $g^y v^e = g^{r+es} g^{-se} = g^{r+es-es} = g^r = x$.

Ορισμός 10.1.1. *Λέμε ότι ένα σύστημα ταυτοποίησης τριών κινήσεων, είναι σημασιολογικά ασφαλές (sound) αν και μονο αν, η Εύα γνωρίζοντας μόνο το δημόσιο κλειδί, μπορεί να περάσει τον έλεγχο αυθεντικοποίησης με αμελητέα πιθανότητα.*

Θα δείξουμε ότι η ασφάλεια του προηγούμενου συστήματος εξαρτάται μόνο από την επιλογή του e. Ας υποθέσουμε ότι η Εύα προβλέπει το σωστό $e' \in \{1, 2\}$. Τότε μπορεί να πλαστογραφήσει την ταυτότητα της Alice.

Η Εύα διαλέγει ένα τυχαίο r από το σύνολο $\{1, 2, ..., q-1\}$ και μαντεύει το σωστό e' του Bob. Θέτει $y = r$ και υπολογίζει το $g^y v^{e'} \pmod p$ και το θέτει ίσο με x. Τότε το ζευγάρι (x, y) περνάει το τεστ του Bob. Πράγματι ο Bob θα υπολογίσει το $g^y v^{e'}$ το οποίο είναι ίσο με x. Άρα η Εύα έχει επιτυχία στην προηγούμενη επίθεση με πιθανότητα $1/2$. Το πρωτόκολλο δεν είναι ασφαλές διότι η Εύα με μη αμελητέα πιθανότητα $(1/2)$, πετυχαίνει να περάσει το τεστ αυθεντικοποίησης. Προτείνουμε την παρακάτω εύλογη τροποποίηση (Schnorr).

1ο βήμα. Η Alice διαλέγει έναν τυχαίο αριθμό $r \in \{1, 2, ..., q-1\}$ και στέλνει στον Bob $x = g^r \pmod p$.
2ο βήμα. Ο Bob διαλέγει ένα τυχαίο αριθμό $e \in \{1, 2, ..., 2^t\}$ και τον στέλνει στην Alice
3ο βήμα. Η Alice υπολογίζει τον αριθμό $y = r + es (\text{mod } q)$ και το στέλνει στον Bob.
Ο Bob κάνει την επαλήθευση (verification) ελέγχοντας $g^y v^e \pmod p = x$.

Πράγματι, $g^y v^e = g^{r+es} g^{-se} = g^{r+es-es} = g^r = x$.

Σύμφωνα με την προηγούμενη ανάλυση η Εύα μπορεί με πιθανότητα 2^{-t} να πλαστογραφήσει την ταυτότητα της Alice. Αν υποθέσουμε ότι το t είναι 80, τότε η προηγούμενη επίθεση της Εύας έχει αμελητέα πιθανότητα επιτυχίας. Αυτό δεν συνεπάγεται ότι το σύστημα είναι ασφαλές. Για να αποδείξουμε ότι το σύστημα είναι sound, πρέπει να αποδείξουμε ότι η πιθανότητα επιτυχίας της Εύας δεν μπορεί να αυξηθεί περισσότερο από 2^{-t}. Πρέπει να αποδείξουμε το επόμενο θεώρημα.

Θεώρημα 10.1.1 (soundness). Έστω ότι υπάρχει πιθανότικός αλγόριθμος $\mathcal{A}$ τέτοιος ώστε, με είσοδο (pk, e), όπου το e είναι τυχαίο από το σύνολο $\{1, ..., 2^t\}$, επιστρέφει με πιθανότητα $\varepsilon > 2^{-t+1}$ ένα ζευγάρι (x, y) το οποίο περνάει το τεστ αυθεντικοποίησης και αυτό γίνεται σε χρόνο $|\mathcal{A}|$. Τότε, με θετική σταθερή πιθανότητα και σε χρόνο $O(|\mathcal{A}|/\varepsilon)$, μπορεί να υπολογιστεί ο διακριτός λογάριθμος $\log_g(v)$ στο $\mathbf{Z}_p^*$.

Αυτό το θεώρημα, μας λέει ότι το σύστημα είναι ασφαλές (για $t \geq 80$), διότι η Εύα δεν μπορεί να πετύχει κάτι καλύτερο από το να διαλέξει τυχαία το e από το σύνολο $\{1, ..., 2^t\}$, εκτός και αν μπορεί να υπολογίσει τον διακριτό λογάριθμο του v. Με άλλα λόγια αν ο Bob συμπεριφέρεται τίμια (δηλαδή διαλέγει το e τυχαία), τότε το σύστημα είναι sound (με την προϋπόθεση ότι ο υπολογισμός του διακριτού λογαρίθμου είναι δύσκολος).

Σταθεροποιούμε το δημόσιο κλειδί pk. Ο αλγόριθμος $\mathcal{A}$ εκτός του pk, εξαρτάται από μια εσωτερική κατάσταση $IS_{\mathcal{A}}$, που είναι μια τυχαία δυαδική λέξη. Επίσης, κατά την παραγωγή του y ο αλγόριθμος $\mathcal{A}$ εξαρτάται και από το e. Ορίζουμε, $S_{\mathcal{A}}(IS_{\mathcal{A}}, pk, e)$ μια συνάρτηση Boole, η οποία παίρνει την τιμή 1 όταν ο αλγόριθμος $\mathcal{A}$ πετύχει για την είσοδο $(pk, \mathcal{A}, e)$, διαφορετικά ισούται με 0. Δηλαδή, η $S_{\mathcal{A}}(IS_{\mathcal{A}}, pk, e) = 1$ όταν για το ζευγάρι $(IS_{\mathcal{A}}, e)$ ο αλγόριθμος $\mathcal{A}$ εξάγει ένα ζευγάρι (x, y) τέτοιο ώστε, να περνάει το τεστ αυθεντικοποίησης του σχήματος ταυτοποίησης.

Ορισμός 10.1.2. *Ονομάζουμε πίνακα επιτυχίας M_{pk}, έναν πίνακα που αποτελείται από τα bits, $S_{\mathcal{A}}(IS_{\mathcal{A}}, pk, e)$ ως εξής: Για μια τιμή του $IS_{\mathcal{A}}$ θεωρούμε όλες τις τιμές του $e = 1, 2, ..., 2^t$ και σχηματίζουμε την γραμμή που αποτελείται από τα $S_{\mathcal{A}}(IS_{\mathcal{A}}, pk, e)_{e=1,2,...}$. Αυτό το κάνουμε για όλες τις τυχαίες εσωτερικές καταστάσεις $IS_{\mathcal{A}}$, που υποθέτουμε ότι είναι N. Ο πίνακας αυτός έχει 2^t στήλες και N γραμμές.*

Ο πίνακας

$$M_{pk} = \begin{array}{c} \\ IS_1 \\ IS_2 \\ \vdots \\ IS_N \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & ... & 2^t \end{array} \\ \begin{pmatrix} 0 & 1 & ... & 1 \\ 1 & 0 & ... & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & ... & 1 \end{pmatrix} \end{array}$$

Η επιλογή της $i-$γραμμής μας δίνει ένα x, κατόπιν η επιλογή μιας στήλης π.χ. $j-$στήλη, μας δίνει ένα ζευγάρι (e, y). Αν το στοιχείο στην θέση (i, j) είναι 1, τότε

η τριάδα (x, e, y) περνάει το τεστ ελέγχου. Η ιδέα της απόδειξης του θεωρήματος είναι να βρούμε μια γραμμή, που να έχει *αρκετούς* άσσους, ώστε με σταθερή πιθανότητα οι δύο στήλες που διαλέγουμε τυχαία να μας προμηθεύουν με δύο επιτυχημένες τριάδες $(x, e_1, y_1), (x, e_2, y_2)$. Δηλ. $M_{pk}(IS_i, e_1) = 1, M_{pk}(IS_i, e_2) = 1$ με $e_1 \neq e_2$. Τότε, εφαρμόζουμε το παρακάτω λήμμα. Η παρακάτω ιδιότητα που θα αποδείξουμε στο λήμμα ονομάζεται και special soundness.

Λήμμα 10.1.1. *(special soundness). Αν έχουμε δύο τριάδες $(x, e, y), (x, e', y')$ με $e \neq e'$, τότε μπορούμε να υπολογίσουμε αποδοτικά τον διακριτό λογάριθμο $s = \log_g v$.*

Απόδειξη. Άσκηση για τον αναγνώστη. □

Παρατήρηση 10.1.1. Συστήματα που είναι special sound ονομάζονται και συστήματα απόδειξης γνώσης (proof of knowledge). Σε αυτά τα συστήματα η πλαστογραφία (impersonation) και η γνώση του μυστικού κλειδιού (knowledge of secret key) είναι ισοδύναμες. Επόμενως το σύστημα του Schnorr είναι proof of knowledge.

Ορίζουμε $\epsilon = Pr[M_{pk}(IS, e) = 1]$. Δηλαδή, ϵ είναι η πιθανότητα επιτυχίας, αν διαλέξουμε τυχαία ένα στοιχείο του πίνακα M_{pk}.

Ορισμός 10.1.3. *Μία γραμμή του πίνακα M_{pk} την ονομάζουμε βαριά (heavy), αν η σχετική συχνότητα των 1 στην γραμμή είναι $\geq \varepsilon/2$.*

Λήμμα 10.1.2. *Έστω g το πλήθος των άσσων στις βαριές γραμμές του πίνακα M_{pk}. Επίσης, $h = 2^t \times N$ είναι το πλήθος των στοιχείων του M_{pk}. Τότε, $g/h \geq \varepsilon/2$.*

Απόδειξη. Έστω M' ο υποπίνακας που αποτελείται από τις μη-βαριές γραμμές του M_{pk}. Ας είναι h' το πλήθος των στοιχείων του M' και από την υπόθεση h είναι το πλήθος των στοιχείων του M_{pk}. Εξ ορισμού το πλήθος των 1 στον πίνακα M_{pk} είναι $h\varepsilon$, ενώ στον M' είναι $< h'\varepsilon/2$. Επομένως,

$$g \geq h\varepsilon - h'\varepsilon/2 \geq h\varepsilon - h\varepsilon/2 = h\varepsilon/2.$$

□

Από το λήμμα, προκύπτει εύκολα ότι υπάρχουν βαριές γραμμές και αυτές είναι περισσότερες από τις μισές γραμμές του πίνακα M_{pk}. Αν υποθέσουμε ότι $\epsilon \geq 2^{-t+2}$, τότε το πλήθος των 1 στις βαριές γραμμές είναι τουλάχιστον 2. Πράγματι,

$$g \geq 2^{-t+2} \cdot N \cdot 2^t/2 = 2N \geq 2.$$

Δηλαδή, μια βαριά γραμμή του πίνακα M_{pk}, έχει τουλάχιστον 2 θέσεις με 1.

Λήμμα 10.1.3. *Αν διαλέξουμε τυχαία τουλάχιστον $\frac{1}{\varepsilon}$ γραμμές IS_i, τότε με πιθανότητα $1/2$ θα έχω μία βαριά γραμμή.*

Απόδειξη. Επιλέγουμε εμείς τυχαία την εσωτερική κατάσταση. Με αυτόν τον τρόπο κάνουμε τον αλγόριθμο $\mathcal{A}$ ντετερμινιστικό. Το (x, y), παράγεται από τον αλγόριθμο $\mathcal{A}$ ως εξής. Το $x = x(\mathcal{A}, pk, IS_{\mathcal{A}})$ (δεν εξαρτάται από την επιλογή

του e) και το $y = y(\mathcal{A}, pk, IS_{\mathcal{A}}, e)$. Επομένως, αν εκτελέσουμε τον αλγόριθμο $\mathcal{A}$ τουλάχιστον $\frac{1}{\varepsilon}$ φορές, διαλέγοντας κάθε φορά ένα νέο τυχαίο $IS_{\mathcal{A}}$, τότε υπολογίζουμε x. Η επιλογή της εσωτερικής κατάστασης $IS_{\mathcal{A}}$ μας δίνει μια γραμμή του M_{pk}. Κατόπιν, διαλέγουμε ένα e (δηλαδή την στήλη), οπότε υπολογίζουμε το y. Αν η γραμμή είναι βαριά, τότε με πιθανότητα $1/2$, η τριάδα (x, e, y) είναι πετυχημένη (δηλ. περνάει το τεστ αυθεντικοποίησης). Αλλά μια γραμμή είναι βαριά με πιθανότητα $\varepsilon/2$. Επομένως, μετά από $\frac{1}{\varepsilon}$ τυχαίες επιλογές του $IS_{\mathcal{A}}$ θα πετύχουμε μια βαριά γραμμή με πιθανότητα $\frac{1}{2}$. □

Απόδειξη του Θεωρήματος. Καλούμε τον αλγόριθμο $\mathcal{A}$ έως ότου βρούμε τον πρώτο άσσο. Αυτό μπορεί να επιτευχθεί ύστερα από το πολύ $1/\varepsilon$ κλήσεις στον αλγόριθμο $\mathcal{A}$ με είσοδο $(pk, IS_{\mathcal{A}}, e)$, όπου το pk είναι σταθερό και τα $IS_{\mathcal{A}}$, e είναι τυχαία. Πράγματι, όλα τα στοιχεία του πίνακα υποθέτουμε ότι είναι h. Τότε, η πιθανότητα να πετύχουμε 1, στον πίνακα M_{pk} είναι

$$\frac{h\varepsilon/2}{h} \cdot \#\text{κλήσεις του } \mathcal{A} = \frac{\varepsilon}{2} \cdot \frac{1}{\varepsilon} = \frac{1}{2}.$$

Εκτελούμε τον παρακάτω αλγόριθμο.
1.Καλούμε $1/\varepsilon$ φορές τον αλγόριθμο $\mathcal{A}$.
2.

Μετά από $1/\varepsilon$ προσπάθειες θα βρούμε το πρώτο 1, αν ψάξουμε στον H τυχαία (δηλ. αν διαλέγουμε μια τυχαία εσωτερική κατάσταση και ένα τυχαίο διάνυσμα $\mathbf{e}$). Η πιθανότητα είναι μεγαλύτερη από $1/2$. Αν ο 1 βρίσκεται σε μια βαριά γραμμή, τότε μπορούμε να βρούμε ένα δεύτερο 1 στην ίδια γραμμή με πιθανότητα $\frac{\frac{\varepsilon}{2}2^t - 1}{2^t}$. Και γι΄ αυτό χρειαζόμαστε $\frac{2^t}{\frac{\varepsilon}{2}2^t - 1}$ προσπάθειες για επιτυχία. Αφού $\frac{2^t}{\frac{\varepsilon}{2}2^t - 1} < \frac{2^t}{\frac{\varepsilon}{2}2^{t-1}} = 4/\varepsilon$, αυτό σημαίνει ότι με λιγότερες από $\frac{4}{\varepsilon}$ προσπάθειες θα πάρουμε και το δεύτερο 1, με πιθανότητα $1/2$. Από την άλλη μεριά, αν το πρώτο 1 που βρήκαμε βρισκόταν σε μια μη-βαριά γραμμή, μπορεί να σπαταλήσουμε πάρα πολύ χρόνο ψάχνοντας για δεύτερο 1. Για να αποφύγουμε κάτι τέτοιο, εφαρμόζουμε έναν αλγόριθμο, έστω $\mathcal{A}_1$, ο οποίος σταματάει τη διαδικασία μετά από ένα συγκεκριμένο πλήθος προσπαθειών. Ο αλγόριθμος αποτελείται από δυο βήματα, τα οποία εκτελούνται συγχρόνως:
Βήμα 1: Ψάχνουμε τυχαία στην ίδια γραμμή μέχρι να βρεθεί ένα δεύτερο 1.
Βήμα 2: Επαναλαμβανόμενα, ψάχνουμε ένα τυχαίο στοιχείο του H και διαλέγουμε ένα τυχαίο αριθμό μεταξύ των $1, 2, \ldots, d$ (το d θα το επιλέξουμε αργότερα). Αυτό το βήμα σταματάει, αν το στοιχείο που βρήκαμε είναι 1 και ο αριθμός d είναι 1.

Ο αλγόριθμος $\mathcal{A}_1$ εκτελείται με χρόνο $\mathrm{O}(|\mathcal{A}|/\varepsilon)$. Αλλά θέλουμε το βήμα 1 να σταματήσει πρώτο (με μεγάλη πιθανότητα), έτσι ώστε να έχουμε τελικά δυο 1 σε μια γραμμή. Η πιθανότητα το βήμα 2 να τερματίσει πρώτο μετά από k προσπάθειες είναι $\varepsilon/d(1 - \varepsilon/d)^{k-1}$. Χρησιμοποιώντας την υπόθεση του ε όπως πριν, έχουμε ότι $(1 - \varepsilon/d)^{k-1} \leq 1$ και αυτό σημαίνει ότι η πιθανότητα να τελειώσει μετά από k ή λιγότερες προσπάθειες είναι το πολύ $k\varepsilon/d$. Θεωρούμε ότι $k = \lfloor d/(2\varepsilon) \rfloor$, ώστε η η πιθανότητα επιτυχίας για το βήμα βήμα 2 να γίνει $1/2$. Επιπλέον, αν επιλέξουμε $d = 16$, έχουμε ότι βήμα 2 τερματίζει μετα από περισσότερες από

$8/\varepsilon = k$ προσπάθειες με πιθανότητα τουλάχιστον $1/2$. Κι έτσι, το βήμα 1 θα τερματίσει πριν το βήμα 2 με πιθανότητα μεγαλύτερη του $\frac{1}{2}\frac{1}{2} = \frac{1}{4}$. Οπότε, αν συμβεί αυτό, τότε με πιθανότητα τουλάχιστον $1/8$ παίρνουμε ένα δεύτερο 1.

$$\mathcal{A} \xrightarrow[\Pr>1/2]{O(|\mathcal{A}|/\varepsilon)} \text{1η δοκιμή} \longrightarrow \mathcal{A}_1 \xrightarrow[\Pr>1/8]{O(|\mathcal{A}|/\varepsilon)} \text{2η δοκιμή}$$

Έτσι λοιπόν, πρέπει να έχουμε δυο 1 σε μια βαριά γραμμή μετά από $12/\varepsilon$ προσπάθειες με σταθερή πιθανότητα τουλάχιστον $\frac{1}{2} \cdot \frac{1}{8} = \frac{1}{16}$. Αυτό σημαίνει, ότι ο αλγόριθμος εκτελείται σε χρόνο $O(|\mathcal{A}|/\varepsilon)$ και πετυχαίνει να βρει δυο 1 στην ίδια γραμμή με σταθερή πιθανότητα $> 1/16$. Κι αφού έχουμε δυο 1 (στην ίδια γραμμή), παίρνουμε δυο τριάδες $(\mathbf{r}, (\mathbf{s}_i)_i, \mathbf{e}), (\mathbf{r}, (\mathbf{s}'_i)_i, \mathbf{e}')$, με $\mathbf{e} \neq \mathbf{e}'$.

Εφαρμόζοντας το λήμμα 10.1.1, το θεώρημα έπεται. □

Αφού υποθέσαμε ότι το να βρούμε στοιχεία από το Σ_b είναι δύσκολο, δεν μπορούμε να βελτιώσουμε την πιθανότητα επιτυχίας, ώστε $> 2^{-t+2}$.

Κεφάλαιο 11

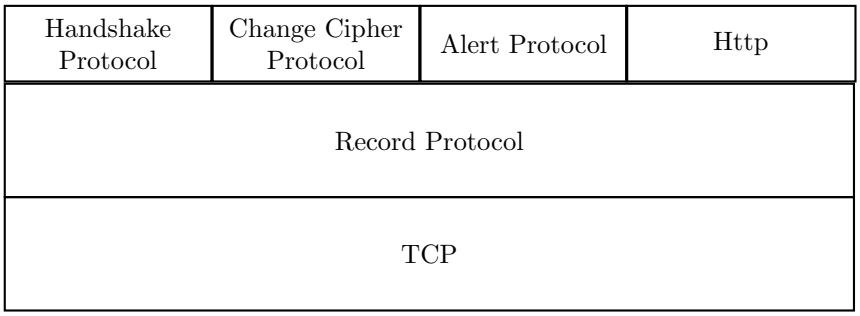
SSL/TLS & PGP

11.1 SSL/TLS

Το πρωτόκολλο SSL αναπτύχθηκε από την NETSCAPE το 1994 για να διασφαλίσει την ασφάλεια στην επικοινωνία μεταξύ πελάτη (client και εξυπηρετητή server. Η έκδοση 2 του SSL δεν χρησιμοποιείται ενώ σε χρήση είναι η έκδοση 3. Η έκδοση του TLS που χρησιμοποιείται σήμερα είναι η 1.2 (υποστηρίζεται από όλους τους γνωστούς browsers). Χρησιμοποιεί περίπου 200 ρουτίνες (ciphersuites), όπως
TLS_RSA_WITH_AES_128_CBC_SHA256 (ασφαλής)
TLS_KRB5_WITH_3DES_EDE_CBC_MD5 (αδύναμη),
TLS_NULL_WITH_NULL_NULL (χωρίς ασφάλεια!!). Μερικές από τις επιθέσεις στο σύστημα αυτό :
• BEAST(2011),CRIME(2012),Lucky13-RC4 attacks (2013)
• Renegotiation attack (2009),triple Handshake attack (2014).
Επίσης, έχουμε πολύ χαμηλού επιπέδου κώδικα ειδικά στη διαχειριση πιστοποιητικών :
• Why Eve and Mallory Love Android (2012),
• The most dangerous code in the world (2012)
• Apple goto fail (2013),
• GnuTLS certificate processing bug (2013),
• Truncation and cookie cutter attacks (2013,2014),
• OpenSSL CCS bug (2014),
• Frankencerts (2014)
και ίσως το πιο διάσημο bug : Heartbleed bug.
Τέλος, να αναφέρουμε την FREAK attack που επειτρέπει με MiMT να χρησιμοποιεί ο server αλγόριθμο RSA με 512-bit modulus. Όπως, έχουμε πει τέτοιου μήκους modulus παραγοντοποιούνται εύκολα. Συνεχίζουμε με την περιγραφή του πρωτοκόλλου. Το SSL/TLS αποτελείται από δύο επιμέρους πρωτόκολλα, το πρωτόκολλο χειραψίας (handshake protocol) και το πρωτόκολλο καταγραφής (record protocol).

- *Πρωτόκολλο χειραψίας (Handshake Protocol).*

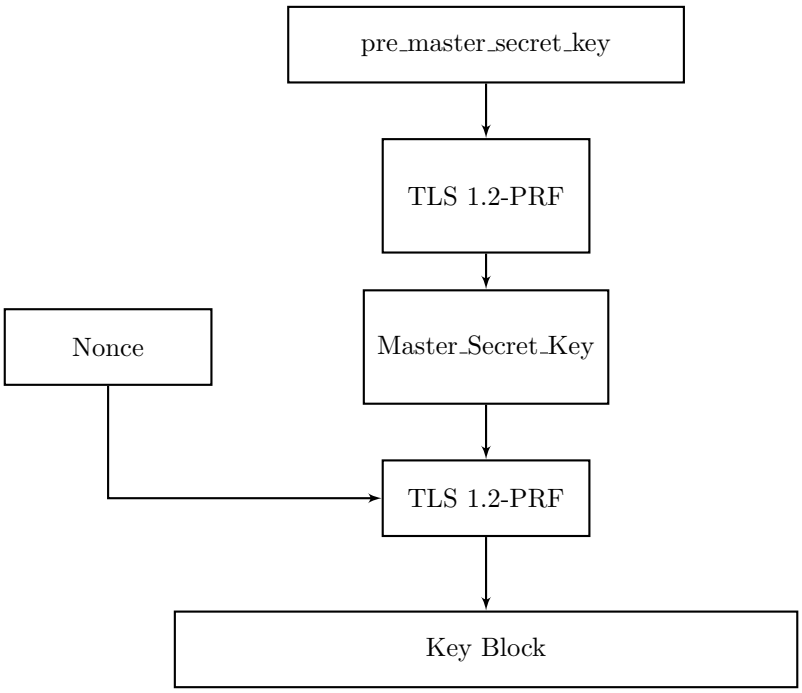
Χρησιμοποιείται για την αυθεντικοποίηση του ενός ή και των δύο σημείων επικοινωνίας και για την εγκαθίδρυση ενός κοινού κλειδιού που θα χρησιμοποιηθεί από το record layer. Επίσης, κάποιοι συμμετρικοί αλγόριθμοι κρυπτογράφησης χρησιμοποιούν *IV*, και αυτά θα παραχθούν κατά την διάρκεια αυτού του πρωτοκόλλου καθώς και τα κλειδιά που θα χρησιμοποιηθούν στις MAC. Συνήθως το



Σχήμα 10: Δομή πρωτοκόλλου TLS.

SSL/TLS αυθεντικοποιεί τον εξυπηρετητή(server) (σπάνια γίνεται αυθεντικοποίηση του client). Κατά την διάρκεια αυτού του πρωτοκόλλου γίνεται με ασφάλεια η επιλογή της έκδοσης του SSL/TLS, η επιλογή των αλγορίθμων και των hash, η επιλογή των αλγορίθμων ανταλλαγής κλειδιού. Ιδιαίτερη προσοχή δίνεται ώστε να μην υπάρχει η δυνατότητα επιλογής κάποια παλαιότερης έκδοσης του SSL/TLS ή επιλογή ασθενούς αλγορίθμου. Πιο αναλυτικά η αυθεντικοποίηση με RSA γίνεται ως εξής:
1. Ο Client θα στείλει ένα nonce στον Server.
2. Ο Server θα στείλει ένα nonce και το ψηφιακό πιστοποιητικό του (ServerCert) στον Client.
3. Ο Client θα εξάγει το δημόσιο κλειδί από το ServerCert.
4. Ο Client θα επιλέξει ένα Pre_master_key:pms.
5. Ο Client θα στείλει $RSA_{OAEP}(pms, pk_{server})$.
6. Ο Server θα αποκρυπτογραφήσει και θα έχει τώρα το pms (Key Derivation Protocol).
7. Ο Server θα σχηματίσει το master_secret_key : ms από το pms και τα nonce (Key Derivation Protocol).
8. Ο Server θα στείλει ServerFin=PRF(ms,transcript) (PRF:HMAC).
9. Ο Client θα επαληθεύσει το ServerFin.

Ποια μέθοδος θα χρησιμοποιηθεί αποφασίζεται από τον client όταν στέλνει clienthello ενώ ο server επιβεβαιώνει την επιλογή του client και απαντά με ServerHello. 'λλες επιλογές, αντί του RSA-PKCS ver.2, είναι :
◦ Static Diffie-Hellman. Ο server στέλνει τις παραμέτρους του Diffie-Hellman καθώς και την (στατική) τιμή g^x ενώ ο client επιλέγει ένα y και στέλνει το g^y στον server. Το κοινό κλειδί είναι g^{xy}.
◦ Ephemeral Diffie-Hellman. Ο server και ο client ανταλλάσσουν τις παραμέτρους (η ομάδα και ο γεννήτορας αποφασίζεται από τον server). Η υπογραφή των παραμέτρων γίνεται αποκλειστικά από τον server.
◦ Anonymous Diffie-Hellman. Το ίδιο όπως προηγουμένως αλλά δεν γίνεται υπογραφή (man in the middle attack).
Το pre_master_secret δεν είναι το τελικό κλειδί που θα χρησιμοποιηθεί.

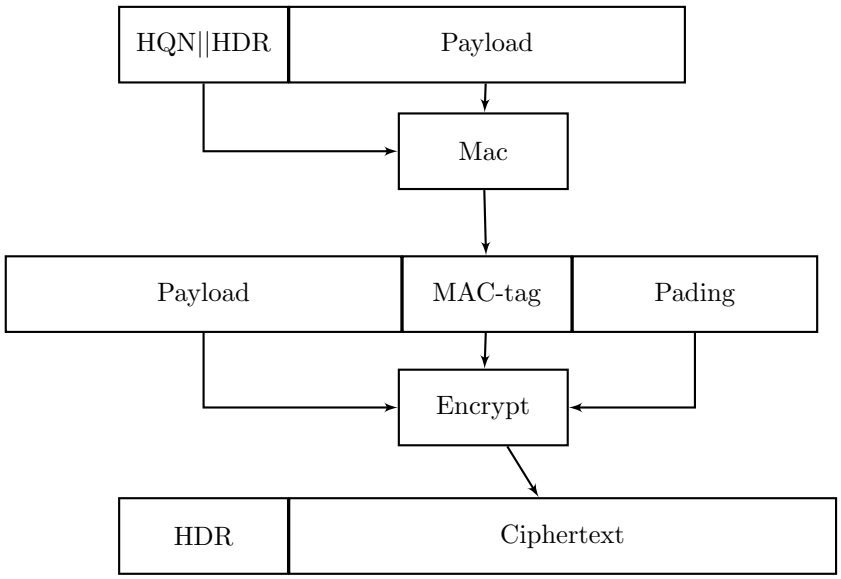


Σχήμα 11: TLS- Key Derivation Protocol.

Key derivation Protocol. Το τελικό κλειδί θα προκύψει από το pre_master_secret_key με την βοήθεια μιας PRF που συνήθως είναι η HMAC-SHA256. Τα nonces και η PRF έχουν ανταλλαχθεί κατά την διάρκεια του πρωτοκόλλου χειραψίας. Από το master_secret θα προκύψει το key block όπου το επόμενο πρωτόκολλο record layer θα δανειστεί τα Mac-keys, encryption keys καθώς και τα IV.

• *Πρωτόκολλο καταγραφής (record layer).*

Παρέχει τις υπηρεσίες, της *Ε*μπιστευτικότητας με χρήση ενός συμμετρικού αλγόριθμου, της *Α*κεραιότητας δεδομένων με χρήση MAC, της *Α*κεραιότητας της πηγής των δεδομένων, της προστασία από replay attacks με χρήση ακολουθίας που προστατεύεται από MAC, και προαιρετικά μας δίνει την δυνατότητα συμπίεσης των δεδομένων. Στο διάγραμμα 12 φαίνεται η διαδικασία που ακολουθεί αυτό το πρωτόκολλο. Από το πρωτόκολλο χειραψίας ο client και ο server έχουν το ίδιο keyblock από το οποίο θα πάρουν όποια κλειδιά χρειάζονται. Αρχικά έχουμε κάποια δεδομένα (payload) που πρέπει να κρυπτογραφηθούν. Η MAC υπολογίζεται στο HQN (sequence number) στην επικεφαλίδα (HDR) και στα δεδομένα (payload). Προστίθεται το Mac-tag και το pad αν χρειάζεται, προστίθεται. Κατόπιν γίνεται η κρυπτογράφηση με ένα συμμετρικό αλγόριθμο και κλειδί από το keyblock. Η επικεφαλίδα (header-HDR) επεκτείνεται κατά 5 bytes, όπου προστίθεται η έκδοση του SSL/TLS (2 bytes), handshake message (1 byte) και 2 bytes καταλαμβάνει



Σχήμα 12: TLS-record layer protocol.

το μήκος του. Αφού κρυπτογραφηθεί στέλνεται στο πρωτόκολλο TCP. Αυτός που θα λάβει το μήνυμα server ή client θα κάνει τα εξής.
1. Θα ελέγξει το μήκος του μηνύματος που υπάρχει στην επικεφαλίδα με το μήκος του μηνύματος που έλαβε.
2. Θα αποκρυπτογραφήσει.
3. Θα αφαιρέσει το padding.
4. Θα ελέγξει την Mac.
5. Θα αποσυμπιέσει (προαιρετικά).
6. Θα στείλει τα δεδομένα στο πιο πάνω layer, π.χ. στο http.
Αποτυχία σε κάποιο από τα προηγούμενα βήματα δίνει fatal error. Αν ενεργοποιηθεί το πρωτόκολλο ChangeCipherProtocol νέο keyblock θα παραχθεί.

11.2 GNU PG

the way to make people
trustworthy is to trust them.
Ernest Hemingway
Selected Letters 1917–1961

Το GNU PG ή GPG μας επιτρέπει να κρυπτογραφούμε ένα μήνυμα ηλεκτρονικού ταχυδρομείου καθώς και να το υπογράφουμε με το ιδιωτικό κλειδί μας. Βασίζεται στο PGP : Pretty Good Privacy το οποίο αναπτύχθηκε από τον Phil Zimmermann. Το PGP αποτελείται από τέσσερις υπηρεσίες. Αυθεντικοποίηση,

Εμπιστευτικότητα, Συμπίεση, Κατακερματισμός. Στην αυθεντικοποίηση πρέπει ο παραλήπτης να είναι σίγουρος ότι το μήνυμα ήρθε από τον σωστό αποστολέα. Επομένως, πρέπει να υπογράφεται ψηφιακά από τον αποστολέα. Απαιτείται επιπλέον η αυθεντικοποίηση του δημόσιου κλειδιού του αποστολέα, ώστε να μην είναι το σχήμα ευάλωτο σε μια επίθεση ενδιαμέσου (man in the middle). Στο PGP αυτή η αυθεντικοποίηση επιτυγχάνεται με τον κύκλο εμπιστοσύνης (web of trust). Υποθέτουμε ότι η εμπιστοσύνη έχει την μεταβατική ιδιότητα. Αν ο Α ισχυρίζεται ότι το δημόσιο κλειδί είναι του αποστολέα και ο Β εμπιστεύεται τον Α, τότε και ο Β εμπιστεύεται το δημόσιο κλειδί του αποστολέα. Όταν ο αποστολέας στέλνει ένα μήνυμα στον παραλήπτη, πριν κρυπτογραφηθεί το μήνυμα (εμπιστευτικότητα), υπογράφεται ψηφιακά. Δηλαδή, το PGP ακολουθεί το παράδειγμα first sign then encrypt. Ειδικότερα αυτή η διαδικασία επιτελείται ως ακολούθως (ο Α είναι ο αποστολέας και ο Β ο παραλήπτης). Το ζευγάρια (PK_A, SK_A) και (PK_B, SK_B) είναι ένα ζεύγος δημόσιου και ιδιωτικού κλειδιού του Α και Β, αντίστοιχα. Επίσης χρησιμοποιείται η ψηφιακή υπογραφή RSA καθώς και μια ασφαλής συνάρτηση κατακερματισμού hash. Τέλος, χρησιμοποιούμε και έναν συμμετρικό αλγόριθμο (E_s, D_s), με (τυχαίο) κλειδί K_s το οποίο χρησιμοποιείται μόνο μία φορά).

Αλγόριθμος 11.2.1. : PGP (για τον αποστολέα)
Είσοδος. Ένα μήνυμα m
Έξοδος. Αυθεντικοποιημένη κρυπτογράφηση του m από τον αποστολέα

Αυθεντικοποίηση
1 $H \leftarrow hash(m)$
2 $S \leftarrow RSA_{SK_A}(H)$
3 $m' \leftarrow [m, S]$
Εμπιστευτικότητα
4 $c' \leftarrow E_s(K_s, m')$
5 $K' \leftarrow RSA_{PK_B}(K_s)$
6 $c \leftarrow [c', K']$
7 Ο Α στέλνει στον Β το c

Στα βήματα 1-3 γίνεται η αυθεντικοποίηση του μηνύματος, δηλαδή το μήνυμα υπογράφεται με τον αλγόριθμο RSA αφού προηγούμενος έχουμε υπολογίσει τον κατακερματισμό του μηνύματος. Μερικές φορές μετά το βήμα 3, γίνεται συμπίεση του μηνύματος m'. Στα υπόλοιπα βήματα, γίνεται η κρυπτογράφηση του αυθεντικοποιημένου μηνύματος.

Αλγόριθμος 11.2.2. : PGP (για τον παραλήπτη)
Είσοδος. Ένα μήνυμα c
Έξοδος. Αποκρυπτογράφηση του c από τον παραλήπτη

1 $K_s \leftarrow RSA_{SK_B}(c[2])$ (με αυτόν τον τρόπο ο Β υπολογίζει το συμμετρικό κλειδί K_s)
2 $m' \leftarrow D_s(K_s, c[1])$ (με αυτόν τον τρόπο ο Β υπολογίζει το m')
3 $H' \leftarrow Hash(m'[1])$
4 **if** $H' = RSA_{PK_A}(m'[2])$ **then**
  return True,$m'[1]$ (εδώ γίνεται επαλήθευση της ψηφιακής υπογραφής και σε περίπτωση επιτυχίας τυπώνουμε το μήνυμα m)
  else
    return FAIL
  end
end

Παρατήρηση 11.2.1. Όπως το SSL/TLS έτσι και το PGP χρησιμοποιεί το σχήμα *first Sign then Encrypt*.

Παρατήρηση 11.2.2. Μια επικίνδυνη επίθεση βρέθηκε στο PGP (καθώς και σε ένα άλλο πρωτόκολλο για emails, το S/MIME[53]) όταν χρησιμοποιούμε email clients όπως ο Thunderbird, Apple email, Outlook 2017, Win 10 mail κ.α. Η επίθεση αυτή έχει το όνομα eFail [54]. Δεν *χτυπάει* το πρωτόκολλο PGP αλλά μια ευπάθεια των email clients. Ο πιο απλός τρόπος να προστατευτούμε από αυτή την επίθεση είναι να μην χρησιμοποιούμε email clients για την αποκρυπτογράφηση (π.χ. μπορούμε να χρησιμοποιούμε την κονσόλα του λειτουργικού μας συστήματος).

Παρατήρηση 11.2.3. Το πρωτόκολλο S/MIME χρησιμοποιεί το πρότυπο X.509 και τα δύο μέλη που επικοινωνούν εμπιστεύονται μια αρχή πιστοποίησης (CA) υπεύθυνη για την υπογραφή των δημόσιων κλειδιών, σε αντίθεση με το PGP που το μοντέλο διαχείρησης των πιστοποιητικών βασίζεται στο web of trust. Το πρόγραμμα ανοιχτού κώδικα GNU/GPG υλοποιεί και τα δύο πρωτόκολλα, PGP και S/MIME.

Άσκηση 11.1 Στείλτε ένα κρυπτογραφημένο μήνυμα στον συγγραφέα. (Το δημόσιο κλειδί του έχει αναγνωριστικό 0xEB1185F82713D6D).

[53] Secure/Multipurpose Internet Mail Extensions
[54] `https://efail.dea`

Κεφάλαιο 12

Ασκήσεις

Άσκηση 12.1 Έστω ότι έχετε ένα CPA-μαντείο στο textbook-RSA. Ας είναι (N, e) το δημόσιο κλειδί και υποθέτουμε ότι δεν το γνωρίζετε. Ας είναι, (m_i, c_i) τα ζευγάρια που έχετε. Υλοποιείστε μια (πιθανοτική) επίθεση, που να σας επιστρέφει το N. Μπορείτε να μετρήσετε εμπειρικά την πιθανότητα επιτυχίας;
(Υποδ. θεωρήστε και τα $(m_i^2, c_i') \in \mathbb{Z}_N^2$, τότε τα $c_i' - c_i$ διαιρούνται από το N.)

Άσκηση 12.2 Να λυθούν τα γραμμικά συστήματα ισοτιμιών

$$\text{(i)}. \begin{cases} x \equiv 2 \pmod 3 \\ x \equiv 3 \pmod 5 \\ x \equiv 2 \pmod 7 \end{cases}, \text{(ii)}. \begin{cases} 2x \equiv 1 \pmod 5 \\ 3x \equiv 9 \pmod 6 \\ 4x \equiv 1 \pmod 7 \\ 5x \equiv 9 \pmod{11} \end{cases}.$$

Άσκηση 12.3 Να υπολογίσετε (*με το χέρι*) έναν μη τετριμμένο διαιρέτη του αριθμού 2581.

Άσκηση 12.4 Να αποδείξετε ότι $x^{\frac{p-1}{2}} \in \{\pm 1\}$ όταν $x \in \mathbb{Z}_p^*$.

Άσκηση 12.5 Να βρείτε το πλήθος των Τ.Υ. mod N αν $N = p_1 p_2 \cdots p_n$ όπου p_i περιττοί διακεκριμένοι πρώτοι αριθμοί.

Άσκηση 12.6 Με χρήση του θεωρήματος του Euler βρείτε ποια στοιχεία του $\mathbb{Z}_{13}$ έχουν τετραγωνικές ρίζες.

Άσκηση 12.7 Γράψτε ένα μικρό πρόγραμμα που να υπολογίζει την πιθανότητα ένας τυχαίος θετικός ακέραιος m με n-bits, να έχει έναν παράγοντα d με δυαδικό μήκος στο σύνολο

$$\{\lfloor n/2 \rfloor, \lfloor n/2 \rfloor \pm 1\}.$$

Χριαζόμαστε την παρακάτω ρουτίνα η οποία θα εκτελεστει K φορές, ώστε να υπολογίζουμε την ζητούμενη πιθανότητα (αυτή θα είναι πλήθος άσων δια του K).

---

input : Ένας ακέραιος m δυαδικού μήκους $n-$bits
output: 1, αν υπάρχει ακέραιος d διαιρέτης του n δυαδικού μήκους περίπου $n/2$ bits
1: $x \leftarrow \{2^{n-1}, ..., 2^n - 1\}$;
2: L = divisors(x)
3: **For** i from 1 **to** $len(L)$
4: **If** $len(d) \in \{\lfloor n/2 \rfloor, \lfloor n/2 \rfloor \pm 1\}$ then return 1

Στην επίθεση που παρουσιάσαμε στην ενότητα 6.2.1, υποθέσαμε ότι το κλειδί γράφεται ως γινόμενο δύο παραγόντων με περίπου το ίδιο μήκος.

Επομένως, για την περίπτωση που το κλειδί έχει μήκος $n = 64-$bits η επίθεση που παρουσιάσαμε στη ενότητα 6.2.1 έχει πιθανότητα επιτυχίας $1/4$ και χρονική πολυπλοκότητα $O(2^{32})$. Να το επαληθεύσετε πειραματικά.

Άσκηση 12.8 Έστω $p = 5, q = 11$ και το κρυπτογραφημένο μήνυμα $c = 14$ έχει προκύψει από την εφαρμογή της TDF του Rabin σε ένα μήνυμα m. Βρείτε το αρχικό μήνυμα m αν γνωρίζετε ότι το $m < 20$.

Άσκηση 12.9 Σχηματίστε ένα πίνακα που να έχει τρεις στήλες και 5 γραμμές. Στην πρώτη στήλη θα υπολογιστεί η πολυπλοκότητα για παραγοντοποίηση σύμφωνα με τον αλγόριθμο του Coppersmith για αριθμούς με $\{100, 200, 1024, 2048, 4096\}$ bit. Στην δεύτερη στήλη σύμφωνα με τον αλγόριθμο Quadratic Sieve και στην τρίτη στήλη σύμφωνα με τον αλγόριθμο GNFS (τα αποτέλεσματα που θα προκύψουν από τον πίνακα έχουν προκύψει κάνοντας μια παρατυπία! Στην ασυμπτωτική πολυπλοκότητα, εμεις θέσαμε συγκεκριμένες τιμές).

Άσκηση 12.10 Βρείτε τα συνεχή κλάσματα των ρητών $\frac{12345}{6789}$, $\frac{54321}{9876}$.

Άσκηση 12.11 Αποδείξτε την παρατήρηση 8.2.3 για την περίπτωση $n = k = 2$.

Άσκηση 12.12 Στο textbook RSA έχουμε εκθέτη $e = 3$ και modulus N 2048$-$bits. Το μήνυμα x έχει το πολύ 682$-$bits. Μπορείτε να βρείτε το x;

Ελληνική Βιβλιογραφία

[1] Ε. Ζάχος; Α. Παγουρτζής; Π. Γροντάς. *Υπολογιστική Κρυπτογραφία*. Σύνδεσμος Ελληνικών ακαδημαικών Βιβλιοθηκών, (2015).

[2] Δημήτριος Πουλάκης. *Θεωρία Αριθμών*, Εκδόσεις Ζήτη, (2009).

[3] Δημήτριος Πουλάκης. *Υπολογιστική Θεωρία Αριθμών*. Σύνδεσμος Ελληνικών ακαδημαικών Βιβλιοθηκών, (2015).

[4] Ελένη Τζανάκη. *Γεωμετρία των Αριθμών*. Διπλωματική Εργασία, Πανεπίστημιο Κρήτης, (2000).

[5] Νίκος Τζανάκης. *Θεμελιώδης Θεωρία Αριθμών*. Σημειώσεις, Πανεπιστήμιο Κρήτης, (2012).

Bibliography

[6] W. Diffie and E. M. Hellman. New directions in cryptography. IEEE Transactions on Information Theory, vol.22, no.6, 1976.

[7] Steven Galbraith. Mathematics of public key cryptography (version 1.1). http://www.math.auckland.ac.nz/ sgal018/crypto-book/crypto-book.html, 2011.

[8] L. J. Gerstein. *Basic Quadratic Forms*, volume 90 of *Graduade studies in mathematics*. AMS, 2008.

[9] R. Grandall and C. Pommerance. *Prime Numbers. A Computational Perspective*. Springer, second edition, 2005.

[10] Donald E. Knuth. *The art of computer programming*, volume 2: Seminumerical algorithms. Addison-Wesley, Reading, Mass., 3 edition, 1998.

[11] A. Menezes, P. van Oorschot, and S. Vanstone. *Handbook of applied cryptography*. CRC Press, 1996.

[12] Victor Shoup. *A computational Introduction to Number Theory and Algebra (Second edition)*. Cambridge University Press, 2008.

[13] W. Stein. Elementary Number Theory : Primes, Congruences, and Secrets, (2011).

[14] Samuel Wagstaff. *The joy of factoring*, volume 68 of *Student Mathematical Library*. AMS, 2013.