

Εισαγωγή στην Κρυπτογραφία

Κ. Α. Δραζιώτης

Enigma Machine. CWI, Amsterdam, ©K. A. Draziotis

0101010001000001000100001
0100101011111101001010010
0101001110101001101010100



Κ. Α. Δραζιώτης

*Επ. Καθηγητής Τμήματος Πληροφορικής,
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης*

Εισαγωγή στην Κρυπτογραφία



Εισαγωγή στην Κρυπτογραφία

Συγγραφή

Κ. Α. Δραζιώτης

Συντελεστές έκδοσης

Γλωσσική Επιμέλεια: Γεωργία Τριανταφυλλίδου

Γραφιστική Επιμέλεια: Κ. Α. Δραζιώτης

Τεχνική Επεξεργασία: Κ. Α. Δραζιώτης

Copyright © 2021, ΚΑΛΛΙΠΟΣ, ΑΝΟΙΚΤΕΣ ΑΚΑΔΗΜΑΪΚΕΣ ΕΚΔΟΣΕΙΣ
(ΣΕΑΒ + ΕΛΚΕ-ΕΜΠ)



Το παρόν έργο αδειοδοτείται υπό τους όρους της άδειας Creative Commons Αναφορά Δημιουργού - Μη Εμπορική Χρήση - Παρόμοια Διανομή 4.0. Για να δείτε ένα αντίγραφο της άδειας αυτής επισκεφτείτε τον ιστότοπο <https://creativecommons.org/licenses/by-nc-sa/4.0/deed.el>

ΚΑΛΛΙΠΟΣ

Εθνικό Μετσόβιο Πολυτεχνείο

Ηρώων Πολυτεχνείου 9, 15780 Ζωγράφου

www.kallipos.gr

ISBN : 978-618-85820-5-7

Στην Ελένη

Περιεχόμενα

Πίνακας συντομεύσεων-ακρωνυμίων	viii
Κατάλογος Σχημάτων	ix
Πρόλογος	x
1 Γιατί είναι χρήσιμη η κρυπτογραφία;	1
1.1 Εισαγωγή	3
1.2 Οι στόχοι της Κρυπτογραφίας	4
1.3 Μερικές πρακτικές εφαρμογές της κρυπτογραφίας	4
1.3.1 SSL/TLS	4
1.3.2 Tor	4
1.3.3 PGP	6
1.3.4 Ομομορφικά κρυπτοσυστήματα-Κρυπτογραφία με Πλέγματα	6
1.3.5 Μετα-κβαντική κρυπτογραφία	6
2 Κρυπτογραφία ιδιωτικού κλειδιού	9
2.1 Εισαγωγή	11
2.2 Η Αρχή του Kerckhoffs	11
2.3 Συστήματα αντικατάστασης	11
2.4 Οι συχνότητες γραμμάτων του ελληνικού αλφαβήτου	13
2.5 Συστήματα μετατόπισης	13
2.6 Κρυπτοσυστήματα μετάθεσης	14
2.7 Το σύστημα του Playfair	15
2.8 Το σύστημα του Vigenère	16
2.9 Σημειωματάριο Μίας Φοράς	19
2.9.1 Διαχωριστές - Distinguishers	21
3 Κρυπτοσυστήματα Ροής (stream ciphers)	25
3.1 Εισαγωγή	27
3.2 LFSR	28
3.2.1 Επιπρόσθετη θεωρία για τα LFSR	31
3.3 RC4 - Ron's Code	35
4 Κρυπτοσυστήματα τμήματος (block ciphers)	38
4.1 Εισαγωγή	40
4.1.1 Ασφάλεια ενός κρυπτοσυστήματος τμήματος	40
4.2 DES	41
4.2.1 Η συνάρτηση Feistel F	41
4.2.2 Παραγωγή υποκλειδιών K_i	44
4.2.3 Σχόλια για το σύστημα DES	45
4.2.4 Καταστάσεις λειτουργίας συμμετρικού συστήματος τμήματος	46
4.3 AES	47
4.3.1 Πρόσθεση και πολλαπλασιασμός nibbles στον mini AES	48
4.3.2 Κρυπτογράφηση στο mini AES	49

5	Συναρτήσεις Κατακερματισμού & Κώδικες αυθεντικοποίησης	52
5.1	Εισαγωγή	54
5.2	Εύρεση διενέξεων-Επίθεση των γενεθλίων (birthday attack)	56
5.3	Σχήμα των Merkle-Damgård	59
5.4	Εφαρμογές	60
5.4.1	Το σύστημα αποθήκευσης κωδικών σε μηχανές linux	60
5.4.2	Blockchain και Bitcoin	60
5.5	Κώδικες αυθεντικοποίησης: MAC	63
6	Το πρόβλημα ανταλλαγής κλειδιού	67
6.1	Το Πρωτόκολλο Diffie-Hellman	69
6.2	Diffie-Hellman χωρίς αλληλεπίδραση	70
6.3	Άλλα ζητήματα ασφάλειας	71
7	Πολυπλοκότητα & Μηχανές Turing	73
7.1	Εισαγωγή	75
7.2	Πολυπλοκότητα	75
7.2.1	Μηχανές Turing	76
7.2.2	Πολυπλοκότητα Βασικών Πράξεων	81
7.2.3	Γρήγορη ύψωση σε δύναμη	83
8	Εισαγωγή στη Θεωρία αριθμών	86
8.1	Εισαγωγή	88
8.2	Διαιρετότητα	88
8.2.1	Πρώτοι αριθμοί	91
8.3	Μέγιστος Κοινός Διαιρέτης	94
8.4	Ισοτιμίες (ή Ισοδυναμίες)	102
8.4.1	Παραγοντοποίηση και υπολογισμός της $\phi(n)$	107
8.5	Το Θεώρημα των Πρώτων Αριθμών	108
8.6	e -οστές ρίζες mod p	111
8.7	Κινεζικό Θεώρημα Υπολοίπων (CRT)	113
9	Τεστ Πιστοποίησης πρώτων αριθμών	117
9.1	Τεστ Πιστοποίησης του Fermat και τάξη ακεραίου	119
9.2	Ψευδοπρώτοι και αριθμοί Carmichael	121
9.2.1	Πιθανότητα αποτυχίας του Test του Fermat	123
9.2.2	Αριθμοί Carmichael	124
9.3	Τεστ πιστοποίησης των Miller-Rabin	128
9.4	Κατασκευή μεγάλων πρώτων	131
10	Παραγοντοποίηση & Διακριτός λογάριθμος	133
10.1	Παραγοντοποίηση	135
10.1.1	Δοκιμαστική διαίρεση (trial division)	135
10.1.2	Η μέθοδος του Fermat για παραγοντοποίηση	136
10.1.3	Η εκδοχή του Lehman	138
10.1.4	Οι ιδέες του Maurice Kraitchik	138
10.1.5	Αλγόριθμος του Dixon/Quadratic Sieve	141
10.1.6	B-smooth numbers	142
10.1.7	Ο αλγόριθμος Quadratic Sieve	144
10.2	Διακριτός Λογάριθμος (discrete logarithm)	147

10.2.1	Ο αλγόριθμος του Shanks	148
10.2.2	Μέθοδος Pollard- ρ	150
11	Trapdoor Functions (TDF)	155
11.1	Εισαγωγή	157
11.1.1	Υβριδικό κρυπτοσύστημα και Key Encapsulation Mechanism	159
11.2	RSA TDF	160
11.2.1	RSA-KEM	161
11.2.2	Η γνώση του d αποκαλύπτει τα p, q	162
11.2.3	CRT και RSA	164
11.3	Rabin TDF	165
11.3.1	Ασφάλεια	166
11.3.2	Coin flipping over the phone	166
11.4	ElGamal	167
11.4.1	Κρυπτογραφία με Ελλειπτικές καμπύλες	167
11.4.2	Post Quantum ECC	168
12	Επιθέσεις στο Κρυπτοσύστημα RSA	170
12.1	Η επίθεση του Wiener	172
12.2	Επιθέσεις στο RSA βασισμένες σε πλέγματα	176
12.2.1	Επίθεση σε μικρά μηνύματα	176
12.2.2	Merge Sort	177
12.2.3	Η μέθοδος του Coppersmith	178
12.3	Συμπεράσματα	179
13	Ψηφιακές Υπογραφές	183
13.1	Ορισμός ψηφιακής υπογραφής	185
13.2	Ασφάλεια μιας ψηφιακής υπογραφής	185
13.2.1	Επιθέσεις σε ψηφιακές υπογραφές	185
13.2.2	Ευπάθειες ψηφιακής υπογραφής	186
13.2.3	Ορισμός Ασφάλειας ψηφιακής υπογραφής	186
13.3	Ψηφιακές Υπογραφές από TDF	186
13.3.1	Υπογραφή RSA	187
13.3.2	DSA	188
13.3.3	Επίλυση διαφωνιών	188
13.4	Συστήματα Ταυτοποίησης - id-schemes	189
13.4.1	Σύστημα ταυτοποίησης του Schnorr	190
14	Πλέγματα	195
14.1	Εισαγωγή στα Πλέγματα	197
14.2	Διαδικασία Gram-Schmidt	201
14.3	Θεωρήματα του Minkowski	207
14.3.1	GSA : Geometric Series Assumption	211
14.4	Αλγόριθμος των Gauss-Lagrange	212
14.5	Ο αλγόριθμος LLL	216
14.5.1	Βοηθητικά λήμματα	218
14.6	SVP	218
14.6.1	Ο αλγόριθμος απαρίθμησης των Kannan-Pohst-Fincke	219
14.6.2	Ο ψευδοκώδικας του αλγορίθμου απαρίθμησης	223
14.6.3	Schnorr-Euchner	224
14.7	CVP	225
14.8	Συστήματα που βασίζονται σε πλέγματα	226

14.8.1 Learning With Errors (LWE)	227
14.8.2 Ένα LWE-ομομορφικό κρυπτοσύστημα (BV11)	229
14.8.3 Ομομορφικές ιδιότητες του συστήματος	231
15 SSL/TLS & PGP	236
15.1 SSL/TLS	238
15.2 GNU PG	241

Πίνακας συντομεύσεων-ακρωνυμίων

AES	: Advanced Encryption Standard
AKS	: Agrawal - Kayal - Saxena
CBC	: Cipher Block Chaining
CCA	: Chosen Ciphertext Attack
CPA	: Chosen Plaintext Attack
CRC	: Cyclic Redundancy Check
CRT	: Chinese Remainder Theorem
DH	: Diffie-Hellman
ECDSA	: Elliptic Curve Digital Signature Algorithm
ECB	: Electronic Codebook Mode
DARPA	: Defense Advanced Research Projects Agency
DES	: Data Encryption Standard
DSA	: Digital Signature Standard
GPG	: GNU Privacy Guard
GNFS	: General Number Field Sieve
GNU	: GNU's Not Unix
GPL	: General Public License
LFSR	: Linear Feedback Shift Register
LWE	: Learning with Errors
MAC	: Message Authentication Codes
MD	: Message Digest
NBS	: National Bureau of Standards
NIST	: National Institute of Standards and Technology
NP	: Non deterministic Polynomial
OAEP	: Optimal Asymmetric Encryption Padding
OTP	: One Time Pad
PGP	: Pretty Good Privacy
PNT	: Prime Number Theorem
PRG	: Pseudo Random Generator
QS	: Quadratic Sieve
RSA	: Rivest - Shamir - Adleman
RHF	: Root Hermite Factor
SHA	: Secure HAsH
SPN	: Substitution and Permutation Network
Tor	: The Onion Router
TDF	: TrapDoor Function
αν-ν	: αν και μόνο αν
αντ.	: αντίστοιχα
γ.α.	: γραμμικώς ανεξάρτητα
κά.	: και άλλα
ν.δ.ό.	: να δείξετε ότι
ν.α.ό.	: να αποδείξετε ότι

Κατάλογος Σχημάτων

2.1	Μηχανή Enigma στο Ινστιτούτο CWI, Amsterdam ©K. A. Draziotis.	11
4.1	Σχήμα Feistel.	42
4.2	ο i -οστός γύρος της F κατά την κρυπτογράφηση και αποκρυπτογράφηση (κάτω σχήμα).	43
4.3	Στιγμιότυπο της συνάρτησης $F : K \times \{0, 1\}^{32} \rightarrow \{0, 1\}^{32}$	43
4.4	Whit Diffie and Martin Hellman. Turing Medal 2016. CC BY-SA 2.0/CC BY-SA 3.0, δείτε https://commons.wikimedia.org/wiki/File:Whit_Diffie_at_CFP_2007.jpg , https://commons.wikimedia.org/wiki/File:Martin-Hellman.jpg	45
4.5	ECB mode (κρυπτογράφηση).	46
4.6	CBC mode (κρυπτογράφηση) [4].	47
4.7	Μια τυπική επανάληψη του mini-AES.	48
5.1	Σχήμα των Merkle-Damgård	59
5.2	CMAC Εικόνα από wikipedia commons, άδεια: CC BY-SA 3.0. https://en.wikipedia.org/wiki/File:CMAC_-_Cipher-based_Message_Authentication_Code.pdf	65
5.3	Κονσόλες που ελέγχουν το λογισμικό με MAC	65
7.1	Η μηχανή ENIAC. Άδεια CC BY-SA 3.0 https://commons.wikimedia.org/wiki/File:ENIAC_Penn1.jpg	77
7.2	Μια μηχανή Turing. Το σχήμα είναι παραλλαγή του σχήματος του Sebastian Sardina και έχει άδεια CC-BY 2.5. http://www.texample.net/tikz/examples/turing-machine-2	78
7.3	Υποθέσαμε ότι $NP \neq co-NP$	81
10.1	Η συνάρτηση του Dickman $\rho(u)$. Παρατηρήστε ότι $\rho(u) \leq 1$ και τείνει στο μηδέν πολύ γρήγορα.	143
10.2	Στον κάθετο άξονα είναι οι τιμές της συνάρτησης $\psi(y, \sqrt{y})$ και στον οριζόντιο άξονα το y	143
10.3	ρ -Pollard.	150
11.1	Michael O. Rabin (1931-) Turing Medal 1979. Ήταν μαθητής του Alonzo Church. CC BY-SA 2.0 de licence, https://en.wikipedia.org/wiki/Michael_O._Rabin#/media/File:M_O_Rabin.jpg	165
11.2	Το γράφημα της ελλειπτικής καμπύλης $y^2 = x^3 + 7$ στο $\mathbb{R}$	168
14.1	Ένα πλέγμα που παράγεται από τα $\mathbf{b}_1, \mathbf{b}_2$	198
14.2	(Εμπειρική) επαλήθευση της υπόθεσης LWE.	229
15.1	Δομή πρωτοκόλλου TLS.	238
15.2	TLS- Key Derivation Protocol.	240
15.3	TLS-record layer protocol.	241

Πρόλογος

Η κρυπτογραφία είναι το απαραίτητο εργαλείο, για να προστατεύσουμε ευαίσθητες πληροφορίες σε υπολογιστικά συστήματα. Χρησιμοποιείται καθημερινά από πάρα πολλούς χρήστες. Υλοποιείται σε διάφορα πρωτόκολλα, όπως το SSL/TLS. Το τελευταίο υλοποιεί κρυπτογραφικά συστήματα και έχει ένα εύρος εφαρμογών.

Σε αυτό το βιβλίο παρουσιάζουμε τη βασική κρυπτογραφία, συμμετρική και δημόσιου κλειδιού. Στη συμμετρική κρυπτογραφία ασχολούμαστε, εκτός των συμμετρικών κρυπτοσυστημάτων, με τις συναρτήσεις αυθεντικοποίησης (MAC) και τις συναρτήσεις κατακερματισμού (hash functions). Στην κρυπτογραφία δημόσιου κλειδιού παρουσιάζουμε την απαραίτητη θεωρία αριθμών που χρειάζεται, για να αναπτύξουμε τα συστήματα RSA και Diffie-Hellman. Στην παρουσίαση της θεωρίας αριθμών παρατίθενται αρκετές αποδείξεις, καθώς και πολλές άλυτες ασκήσεις. Συνολικά υπάρχουν 257 άλυτες ασκήσεις από απλές που βασίζονται στους ορισμούς έως αρκετά δύσκολες. Επίσης, παρουσιάζουμε τη βασική θεωρία πλεγμάτων (lattices) και ένα ομομορφικό κρυπτοσύστημα που βασίζεται στο πρόβλημα LWE. Τα πλέγματα είναι απαραίτητα για την κρυπτανάλυση του RSA και τη σχεδίαση των σύγχρονων κρυπτοσυστημάτων που αντιστέκονται σε επιθέσεις κβαντικών υπολογιστών. Τέλος, έχει δοθεί έμφαση στην πρακτική της κρυπτογραφίας μέσα από χρήση κώδικα python.

Τυχόν λάθη/αβλεψίες/παραλείψεις είναι ευθύνη του συγγραφέα και μπορούν να αποσταλούν σχόλια και επισημάνσεις στη διεύθυνση drazioti@csd.auth.gr. Στην επόμενη έκδοση θα υπάρχουν credits.

Ευχαριστίες.

Εύα Περόντση, Μαρία Πεγιά, Γρηγόριος Γιαννακόπουλος, Κωνσταντίνος Σταυράτης, Ελένη Σταματάκη, Λώρα Παπακωνσταντίνου, Αναστάσιος Γεωργόπουλος, Γεωργία Τριανταφυλλίδου, Εμμανουήλ Παππάς, Χριστόφορος Μαρινόπουλος, Φίλιππος Ρωσσίδης, Κεντρική Ομάδα Υποστήριξης Κάλλιπος.

Κεφάλαιο 1

Γιατί είναι χρήσιμη η κρυπτογραφία;

Σύνοψη - Περίληψη. Σε αυτό το κεφάλαιο παρουσιάζουμε μερικές εφαρμογές της κρυπτογραφίας με βασικότερη το πρωτόκολλο SSL/TLS που έχει ως στόχο την κρυπτογράφηση και αυθεντικοποίηση όλης της δικτυακής κυκλοφορίας μεταξύ δύο υπολογιστών, πελάτη (client) και διακομιστή (server). Επίσης, η κρυπτογραφία έχει εφαρμογές σε πρωτόκολλα ανωνυμοποίησης, σε ψηφιακές υπογραφές, ψηφιακές εκλογές και άλλες, τις οποίες συνοπτικά παραθέτουμε.

Προαπαιτούμενη γνώση. Για αυτό το κεφάλαιο δεν χρειάζονται ειδικές γνώσεις, αλλά μόνο βασικές έννοιες της πληροφορικής. Μπορείτε να συμβουλευτείτε και την παρακάτω βιβλιογραφία [6, Κεφ. 16], [7, Κεφ. 1]. Επίσης, προτείνεται η αξιοποίηση του πολύ χρήσιμου λεξικού όρων της κρυπτογραφίας [1], και του βιβλίου του William Stallings [4].

1.1 Εισαγωγή

Τα πρώτα συστήματα κρυπτογράφησης χρησιμοποιούσαν το ίδιο κλειδί κρυπτογράφησης και αποκρυπτογράφησης (ονομάζονται και συμμετρικά συστήματα). Ένα τέτοιο σύστημα είναι το σύστημα κρυπτογράφησης AES : **A**dvanced **E**ncryption **S**tandard, το οποίο χρησιμοποιείται στο πρωτόκολλο SSL/TLS, που με τη σειρά του χρησιμοποιείται όπου απαιτείται ασφαλής ηλεκτρονική συναλλαγή (π.χ. σε τράπεζες, σε συστήματα που απαιτούν τη χρήση κωδικού, στο ηλεκτρονικό εμπόριο κ.ά.). Στη σύγχρονη εποχή, το SSL/TLS χρησιμοποιείται από μεγάλες εταιρείες, όπως στη μηχανή αναζήτησης της Google που κρυπτογραφεί όλη την επικοινωνία μεταξύ του υπολογιστή μας και του διακομιστή της Google, ο οποίος θα επεξεργαστεί το αίτημά μας. Δηλαδή, αν κάποιος παρακολουθεί την επικοινωνία μας, δεν μπορεί να καταλάβει ποιο είναι το αίτημά μας. Κάποιος, αρκετά φιλόδοξος, μπορεί να αναρωτηθεί πώς θα είναι εφικτό να κρύβουμε τα αιτήματά μας, και από την ίδια τη Google. Όσο παράδοξο και αν ακούγεται το ερώτημα αυτό, απαντήθηκε (εν μέρει) από τον Craig Gentry. Αυτό που απομένει είναι να γίνει ένα τέτοιο πρακτικό σύστημα.

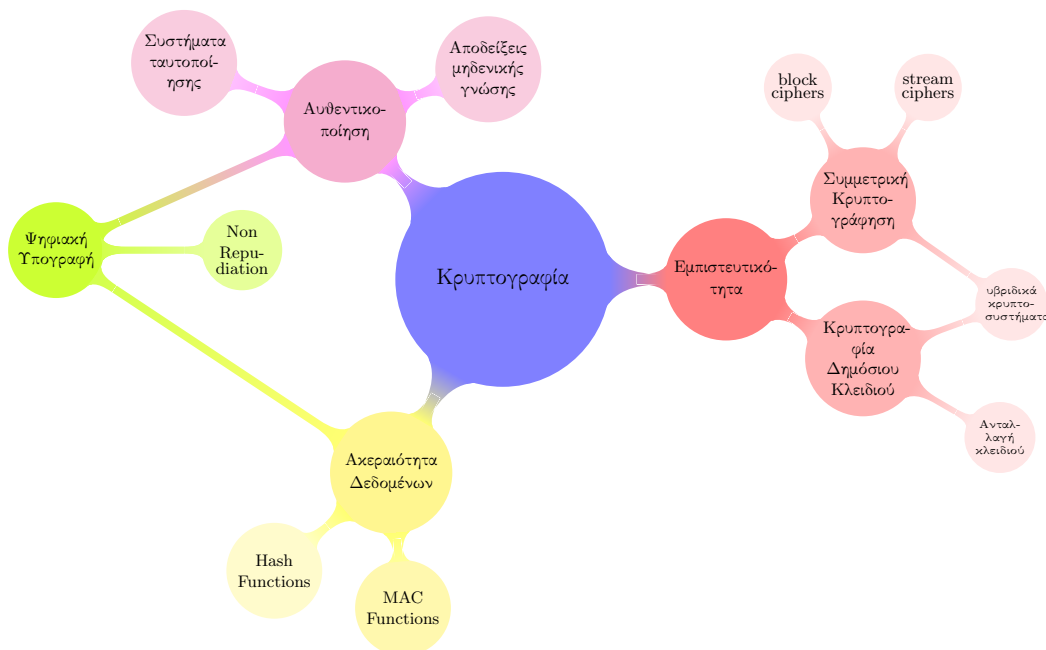
Σε αυτήν την ενότητα, αφού αρχικά δούμε μερικές πρακτικές εφαρμογές της κρυπτογραφίας, θα ξεκινήσουμε από πολύ απλά κρυπτοσυστήματα και θα καταλήξουμε σε σύγχρονα (αρκετά πολύπλοκα) κρυπτοσυστήματα, όπως το AES¹. Ένα από τα πιο απλοϊκά κρυπτοσυστήματα ήταν αυτό του Καίσαρα. Σε αυτό το σύστημα, κάθε γράμμα αντικαθίσταται από το τρίτο διαδοχικό γράμμα του αλφαβήτου που χρησιμοποιούμε. Για παράδειγμα, η λέξη *σήμερα* γράφεται *φκυθιδ*, διότι το *φ* είναι το τρίτο διαδοχικό γράμμα μετά το *σ*. Το κλειδί σ' αυτό το σύστημα είναι η τιμή της μετάθεσης, δηλαδή 3. Για να αποκρυπτογραφήσουμε, ακολουθούμε την αντίστροφη πορεία. Συστήματα σαν το προηγούμενο ονομάζονται *συστήματα μετατόπισης*. Από τα προηγούμενα γίνεται φανερό ότι το βασικό στοιχείο του συστήματος είναι το μυστικό κλειδί. Θεωρούμε ότι ο εχθρός δεν γνωρίζει το κλειδί, αλλά γνωρίζει πώς δουλεύει το σύστημα. Επομένως, δύο άτομα (ή οντότητες²) που θέλουν να επικοινωνήσουν πρέπει να ανταλλάξουν με ασφαλή τρόπο το μυστικό κλειδί. Στο διαδίκτυο το πρόβλημα ανταλλαγής κλειδιών είναι πολύ σημαντικό, διότι οι γραμμές επικοινωνίας δεν είναι ασφαλείς δίαυλοι επικοινωνίας. Όλα τα συστήματα κρυπτογραφίας πριν τους Diffie-Hellman (π.D-H) βασίζονταν στην παραδοχή ότι η ανταλλαγή κλειδιών γίνεται με ασφαλή τρόπο. Αυτό δεν ήταν πρακτικό, γιατί απαιτούσε κόστος και πολλές φορές υπήρχε και καθυστέρηση στον διαμερισμό του κλειδιού. Συστήματα όπου δημοσιοποιούμε ένα κλειδί σε μη ασφαλή δίαυλο επικοινωνίας και με τη βοήθειά του οι ενδιαφερόμενες οντότητες υπολογίζουν ένα κοινό μυστικό κλειδί ονομάζονται κρυπτοσυστήματα δημόσιου κλειδιού (Public Key Cryptography) και ανακαλύφθηκαν από τους Whitfield Diffie και Martin Hellman το 1976 [3].

Αρχικά, όπως προαναφέρθηκε, σε αυτό το κεφάλαιο θα δούμε μερικές πρακτικές εφαρμογές της κρυπτογραφίας, χωρίς να εμβαθύνουμε ιδιαίτερα στις αρχές λειτουργίας των συστημάτων που θα παρουσιάσουμε. Για παράδειγμα, θα δούμε ότι η κρυπτογραφία, εκτός της βασικής της λειτουργίας, της απόκρυψης ενός μηνύματος, μπορεί να χρησιμοποιηθεί, για να κατασκευάσουμε ψηφιακά νομίσματα, να υπογράψουμε ψηφιακά κάποια αρχεία μας, να εξασφαλίζουμε ανωνυμία στο διαδίκτυο (π.χ. το Tor) ή να ψηφίζουμε ηλεκτρονικά. Επίσης, πολλά από τα συστήματα, που θα περιγράψουμε σε αυτό το βιβλίο, υλοποιούνται στο πρόγραμμα κρυπτογράφησης emails, το διάσημο PGP : **P**retty **G**ood **P**rivacy. Προχωρώντας, θα ασχοληθούμε με τα συστήματα κρυπτογραφίας πριν τους Diffie-Hellman (συστήματα ιδιωτικού κλειδιού). Το πιο διάσημο σύστημα ιδιωτικού κλειδιού είναι το σύστημα DES : **D**ata **E**ncryption **S**tandard. Το DES και μεταγενέστερα συστήματα αυτού θα μελετηθούν ξεχωριστά. Ειδικότερα, θα δούμε κλασικά συστήματα, όπως το σύστημα μετατόπισης, αντικατάστασης και συνδυασμούς αυτών. Μπορείτε να ανατρέξετε και στο αγγλικό βιβλίο [2], καθώς και στο πολύ καλό βιβλίο [5] στην ελληνόγλωσση βιβλιογραφία. Σε αρκετά σημεία του βιβλίου παρουσιάζονται αλγόριθμοι (σε μορφή ψευδοκώδικα) και σε μερικές ασκήσεις ζητάμε την υλοποίησή τους. Οι υλοποιήσεις, που προτείνουμε, δίνονται στη γλώσσα python (ο κώδικας τρέχει και στις δύο εκδόσεις python 2, python 3 με μικροαλλαγές. Συνήθως δίνεται σε python 3). Επίσης, σε προχωρημένες εφαρμογές, όπως στα πλέγματα, χρησιμοποιούμε το πρόγραμμα ανοιχτού κώδικα SageMath (το οποίο επίσης είναι γραμμένο σε python).

¹<http://www.moserware.com/2009/09/stick-figure-guide-to-advanced.html>

²οντότητα εννοούμε ένα άτομο ή οργανισμό ή μία συσκευή ή μία διαδικασία

1.2 Οι στόχοι της Κρυπτογραφίας



1.3 Μερικές πρακτικές εφαρμογές της κρυπτογραφίας

Θα αναφερθούμε πολύ συνοπτικά σε μερικές πρακτικές εφαρμογές, πριν ασχοληθούμε αναλυτικά με τα κρυπτοσυστήματα ιδιωτικού κλειδιού (συμμετρική κρυπτογραφία).

1.3.1 SSL/TLS

Το πρωτόκολλο SSL/TLS είναι το βασικό πρωτόκολλο κρυπτογράφησης και αυθεντικοποίησης στο διαδίκτυο. Όταν ενεργοποιείται, οι σελίδες από http γίνονται https, όπου το s σημαίνει secure. Σε αυτό το αρκετά πολύπλοκο πρωτόκολλο βασίζεται η ασφάλεια όλων των συναλλαγών που αφορούν το ηλεκτρονικό εμπόριο, τράπεζες, δημόσιο κ.ά. Γενικά, όπου χρειάζεται η εισαγωγή μυστικού κωδικού, πρέπει να υπάρχει το SSL/TLS. Θα δούμε παρακάτω, όταν θα μιλήσουμε και για την κρυπτογραφία δημόσιου κλειδιού, πώς δουλεύει αυτό το πρωτόκολλο και για μερικά προβλήματα που έχουν παρουσιαστεί στο επίπεδο υλοποίησης. Γενικά, εφόσον χρησιμοποιούνται οι κατάλληλες πολιτικές ασφάλειας (επιλογή σωστών αλγορίθμων, τακτικές αναβαθμίσεις, σωστά ψηφιακά πιστοποιητικά, καλή γνώση του sysadmin σε θέματα ασφάλειας κ.ά.) εμπιστευόμαστε αυτό το πρωτόκολλο.

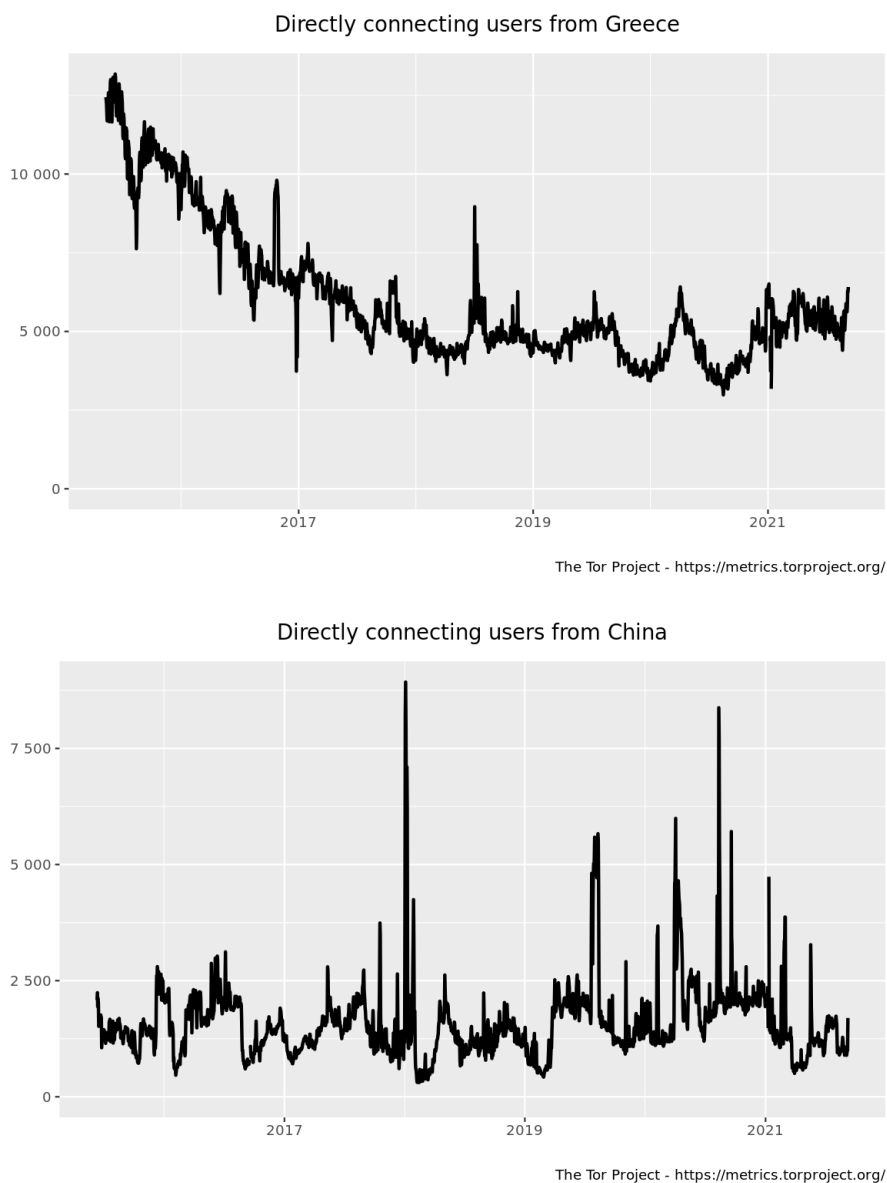
1.3.2 Tor

Το **Tor**: The onion router, είναι ένα αποκεντρωμένο σύστημα ανώνυμης πλοήγησης στο διαδίκτυο^{3 4 5}. Είχε αρχικά χρηματοδοτηθεί από την Αμερικανική κυβέρνηση (DARPA), αλλά αργότερα υποστηρίχθηκε από την κοινότητα ανοιχτού λογισμικού, και πλέον σήμερα δίνεται με τους όρους της άδειας GNU GPL. Επιτρέπει στους χρήστες του, μέσα από τη χρήση ενός μεγάλου δικτύου εθελοντών, τη δρομολόγηση όλης της διαδικτυακής μας κυκλοφορίας από τυχαίους relays, με αποτέλεσμα η σύνδεσή μας να μην πραγματοποιείται απευθείας, αλλά να αποκρύπτεται η ταυτότητά μας (IP) στην τελευταία μας σύνδεση. Παρακάτω παρουσιάζουμε με διαγράμματα τη χρήση του Tor στην Ελλάδα και στην Κίνα κατά την περίοδο 2014-2021. Στο σύστημα αυτό υλοποιείται συμμετρική και ασύμμετρη κρυπτογραφία. Τέλος, το tor χρησιμοποιείται σε συνδυασμό με την πλατφόρμα ανοιχτού λογισμικού secure drop. Η πλατφόρμα αυτή

³D. Goldschlag, M. Reed, P. Syverson, Onion Routing, Communications of the ACM (1999) <https://dl.acm.org/doi/pdf/10.1145/293411.293443>

⁴<https://www.fsf.org/news/2010-free-software-awards-announced>

⁵[https://en.wikipedia.org/wiki/Tor_\(network\)](https://en.wikipedia.org/wiki/Tor_(network))



υλοποιήθηκε από τους Aaron Swartz και Kevin Poulsen και είχε ως στόχο οι πηγές (whistleblowers) να επικοινωνούν με ασφάλεια και ανωνυμία με δημοσιογράφους. Μερικές εφημερίδες που χρησιμοποιούν αυτήν την πλατφόρμα είναι οι: The Guardian⁶, Washington Post⁷, Al Jazeera Media Network⁸ και Wall Street Journal⁹, καθώς και ο βραβευμένος δημοσιογράφος Barton Gellman χρησιμοποιεί το secure drop¹⁰.

⁶<https://www.theguardian.com/securedrop> ανάκτηση 14-11-2021

⁷<https://www.washingtonpost.com/securedrop/> ανάκτηση 14-11-2021

⁸<https://securedrop.org/directory/al-jazeera/> ανάκτηση 14-11-2021

⁹<https://www.wsj.com/tips> ανάκτηση 14-11-2021

¹⁰<https://www.bartongellman.com/pgp/>

1.3.3 PGP

I counted Werner Koch, who wrote the software and maintains it still, among the great contributors to civil society
– B. Gellman (Dark Mirror)

Το σύστημα PGP : Pretty Good Privacy, χρησιμοποιείται για την κρυπτογράφηση και ψηφιακή υπογραφή των email μας. Η περιγραφή του θα γίνει αρκετά αργότερα, όταν θα έχουμε μιλήσει για την κρυπτογραφία δημόσιου κλειδιού. Θεωρείται αρκετά ασφαλές, εφόσον χρησιμοποιηθεί σωστά. Υπάρχουν υλοποιήσεις που δίνονται με άδεια GNU.

Άσκηση 1.1 Εγκαταστήστε το GPG¹¹ και δημιουργήστε ένα ζευγάρι pgp κλειδιού.

1.3.4 Ομομορφικά κρυπτοσυστήματα-Κρυπτογραφία με Πλέγματα

Τα ομομορφικά συστήματα (Homomorphic schemes) επιτρέπουν υπολογισμούς πάνω στα κρυπτογραφημένα δεδομένα. Αυτή η τεχνολογία έχει εφαρμογές κυρίως σε υποδομές νέφους (cloud infrastructures). Το πρώτο ομομορφικό σύστημα που παρουσιάστηκε από τον Craig Gentry κάνει χρήση των πλεγμάτων (lattices). Σε αυτό το βιβλίο θα παρουσιάσουμε ένα σύγχρονο ομομορφικό κρυπτοσύστημα που βασίζεται στο πρόβλημα LWE : Learning With Errors. Βέβαια, τα κρυπτοσυστήματα που βασίζονται στα πλέγματα, εκτός της χρήσης τους σε ομομορφικά κρυπτοσυστήματα, είναι υποψήφια και για την κρυπτογραφία μετά τους χβαντικούς υπολογιστές (Post Quantum Cryptography). Τα χαρακτηρίζουμε υποψήφια, διότι μελλοντικά μπορεί να βρεθεί χβαντικός αλγόριθμος που να λύνει αποδοτικά δύσκολα προβλήματα σε πλέγματα. Υπάρχουν αρκετές υλοποιήσεις όπως η HeLib¹², SEAL της Microsoft, Transpiler¹³ της Google κ.ά.

1.3.5 Μετα-χβαντική κρυπτογραφία

Με τον όρο *μετα-χβαντική κρυπτογραφία* εννοούμε την κρυπτογραφία μετά τους χβαντικούς υπολογιστές, όπου συστήματα όπως το RSA και το Diffie-Hellman δεν θα είναι πλέον ασφαλή¹⁴. Μετά από διαγωνισμό του NIST¹⁵, προέκυψε ότι το σύστημα Crystals-Kyber¹⁶ θα είναι το νέο πρότυπο κρυπτογράφησης για την κρυπτογραφία δημόσιου κλειδιού. Το συγκεκριμένο κρυπτοσύστημα βασίζεται στη θεωρία πλεγμάτων και μέχρι σήμερα (2024), δεν έχει βρεθεί κάποιος χβαντικός αλγόριθμος που να το σπάει. Συνιστάται η μετάβαση σε αυτά τα νέα συστήματα, ώστε να είμαστε προετοιμασμένοι όσο το δυνατόν καλύτερα, όταν βρεθεί ένας χβαντικός υπολογιστής που να μπορεί να σπάει το σύστημα RSA με τις σημερινές παραμέτρους (δηλ. το RSA modulus να είναι τουλάχιστον 2048-bits).

Από την άλλη μεριά, ο όρος *Κβαντική κρυπτογραφία* χρησιμοποιείται για να περιγράψει χβαντικά πρωτόκολλα ανταλλαγής κλειδιού (π.χ. το πρωτόκολλο BB84), δηλαδή πρωτόκολλα που βασίζονται σε ιδιότητες της χβαντικής φυσικής και δεν πρέπει να συγχέεται με τον όρο «μετα-χβαντική κρυπτογραφία». Σύστημα ανταλλαγής κλειδιού που βασίζει τη λειτουργία του στην χβαντική φυσική υπάρχει και ήδη χρησιμοποιείται από μερικούς οργανισμούς¹⁷.

¹¹<https://gnupg.org/download/>

¹²<https://github.com/homenc/HElib>

¹³<https://github.com/google/fully-homomorphic-encryption/tree/main/transpiler>

¹⁴Ο ερευνητής (φυσικός) της IBM Peter Shor με τον ομώνυμο χβαντικό αλγόριθμο μπορεί να παραγοντοποιεί (μεγάλους) αριθμούς, καθώς και να λύνει το πρόβλημα του διακριτού λογαρίθμου σε πολυωνυμικό χρόνο.

¹⁵National Institute of Standards and Technology

¹⁶Δείτε τη μεταπτυχιακή εργασία εδώ

¹⁷https://en.wikipedia.org/wiki/Quantum_key_distribution

Άσκηση 1.2 Να κάνετε μια βιβλιογραφική έρευνα όπου να αναλύεται σε ποιο στάδιο βρίσκεται ο διαγωνισμός του NIST. Ποια είναι τα μαθηματικά προβλήματα που κυριαρχούν στα νέα αυτά πρωτόκολλα που έχουν προταθεί;

Βιβλιογραφία

- [1] Aumasson, Jean-Philippe. *Crypto Dictionary*. No starch press, 2021.
- [2] Cozzens, M.; Miller, S. J. *The Mathematics of Encryption-An Elementary Introduction*. Mathematical World, 1st edition, vol. 29, 2013. AMS.
- [3] Diffie, W.; Hellman, M. *New Directions in Cryptography*. IEEE Transactions on Information Theory, vol. **22**, no.6. 1976.
- [4] Stallings, W. *Cryptography and Network Security: Principles and Practice*. Fifth Edition, Pearson, 2011.
- [5] Κάτος, Β.; Στεφανίδης, Γ. *Τεχνικές Κρυπτογραφίας και Κρυπτανάλυσης*, Εκδ. Ζυγός, 2003.
- [6] Ζάχος, Ε.; Παγουρτζής, Α.; Σούλιου, Θ. *Θεμελίωση επιστήμης υπολογιστών*, Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. <http://dx.doi.org/10.57713/kallipos-491>, 2015.
- [7] Ζάχος, Ε.; Παγουρτζής, Α.; Γροντάς, Π. *Υπολογιστική Κρυπτογραφία*, Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. <http://dx.doi.org/10.57713/kallipos-492>, 2015.

Κεφάλαιο 2

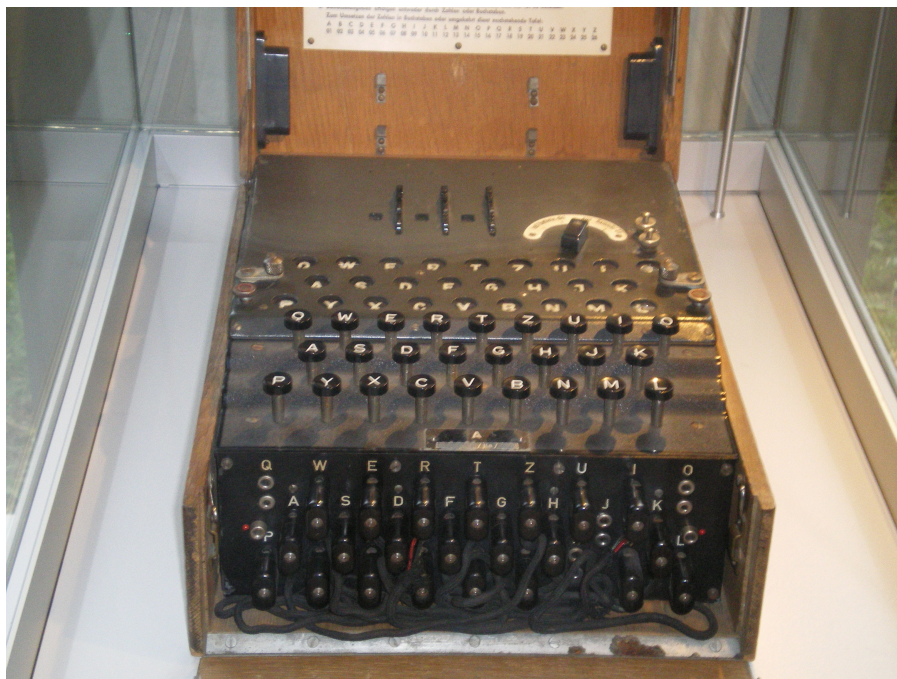
Κρυπτογραφία ιδιωτικού κλειδιού

Σύνοψη - Περίληψη. Γίνεται μια εισαγωγή στην κρυπτογραφία ιδιωτικού κλειδιού ή συμμετρική κρυπτογραφία. Αρχικά, παρουσιάζονται κάποια απλά ιστορικά κρυπτοσυστήματα, όπως το σύστημα της αντικατάστασης, της μετάθεσης (ή σύστημα του Καίσαρα), καθώς και κάποιοι ορισμοί ασφάλειας. Στην ενότητα 2.9 παρουσιάζεται το One Time Pad μαζί με την απόδειξη τέλει ασφάλειας του Shannon.

Προαπαιτούμενη γνώση. Σε αυτό το κεφάλαιο χρειάζονται βασικές γνώσεις θεωρίας πιθανοτήτων και ειδικότερα διακριτή πιθανότητα (discrete probability). Μπορείτε να συμβουλευτείτε για μια εισαγωγή στις διακριτές πιθανότητες το βιβλίο [7, Κεφ. 11] και [5]. Τέλος, μπορείτε να δείτε για τα κρυπτοσυστήματα αντικατάστασης και μετατόπισης τις αναφορές στην ελληνόγλωσση βιβλιογραφία [1, Κεφάλαιο 5], [6].

2.1 Εισαγωγή

Σε αυτά τα κρυπτοσυστήματα το κλειδί κρυπτογράφησης και αποκρυπτογράφησης είναι το ίδιο. Ένα τέτοιο κρυπτοσύστημα είναι η μηχανή αίνιγμα (σχήμα 2.1) που χρησιμοποιήθηκε από τους Γερμανούς στον Β΄ Παγκόσμιο Πόλεμο.



Σχήμα 2.1: Μηχανή Enigma στο Ινστιτούτο CWI, Amsterdam ©K. A. Draziotis.

2.2 Η Αρχή του Kerckhoffs

Στην προηγούμενη ενότητα αναφέρθηκε ότι η ασφάλεια ενός κρυπτοσυστήματος δεν πρέπει να βασίζεται στη γνώση του κρυπτοσυστήματος, αλλά στη γνώση του μυστικού κλειδιού. Η αρχή αυτή διατυπώθηκε πρώτη φορά τον 19ο αιώνα από τον Auguste Kerckhoffs (Ολλανδός γλωσσολόγος και κρυπτογράφος). Οι λόγοι που διατυπώθηκε είναι αρκετοί. Πρώτον, για να είναι το σύστημα μαθηματικά ελκυστικό, εννοώντας με αυτό ότι ένα σύστημα που δεν γνωρίζουμε πώς δουλεύει δεν μπορεί να αναλυθεί και επομένως η ασφάλειά του δεν μπορεί να είναι σίγουρη. Ένας άλλος λόγος είναι η απαίτηση το κρυπτοσύστημά μας να είναι μακρόβιο. Εύκολα ο εχθρός, με εκβιασμό ή δωροδοκία θα μπορούσε να μάθει πώς δουλεύει το κρυπτοσύστημα και συναχόλουθα δεν θα διασφαλιζόταν η μακροβιότητά του.

2.3 Συστήματα αντικατάστασης

Σε αυτά τα συστήματα αντικαθιστούμε γράμματα με κάποια σύμβολα τα οποία επίσης μπορεί να είναι γράμματα ή αριθμοί ή οποιαδήποτε σύμβολα. Πρακτικά φτιάχνουμε έναν πίνακα όπως τον παρακάτω:

A	B	Γ	Δ	E	Z	H	Θ	I	K	Λ	M
B	A	Γ	Δ	H	Θ	E	K	Z	I	M	Λ

N	Ξ	O	Π	P	Σ	T	Υ	Φ	X	Ψ	Ω
Σ	Π	O	T	Υ	X	Ψ	Ω	Φ	P	N	Ξ

Έστω ότι θέλουμε να κρυπτογραφήσουμε το παρακάτω κείμενο.

Στις δέκα το βράδυ σήμερα θα γίνει επίθεση.

Κάθε γράμμα αντικαθίσταται από έναν χαρακτήρα της δεύτερης γραμμής του πίνακα, οπότε έχουμε το κείμενο

XΨZXΔHIBΨOATBΔΩXELHTBKBΓZΣHZHTZKHXE.

Για την αποκρυπτογράφηση, αρκεί κάποιος να γνωρίζει τον πίνακα αντικατάστασης. Ένα σοβαρό πρόβλημα αυτών των συστημάτων είναι ότι, όταν ένα γράμμα εμφανίζεται συχνά στο αρχικό κείμενο, τότε με την ίδια συχνότητα εμφανίζεται και στο κρυπτογραφημένο κείμενο. Έτσι, αν κάποιος γνωρίζει σε ποια γλώσσα είναι γραμμένο το κείμενο και τη συχνότητα εμφάνισης των γραμμάτων στη συγκεκριμένη γλώσσα και, εφόσον το κείμενο είναι αρκετά μεγάλο, τότε εύκολα αποκρυπτογραφείται χωρίς τη γνώση του κλειδιού. Ας πάρουμε το κρυπτογραφημένο κείμενο

TPNALXCTPΦΛΘΞΞΘΘOYMNΘYΘTΓTKTΓEOEΩEMNΘΨKΘTBΘΨEMΘ

Το κείμενο αυτό είναι μικρό και πιθανόν να χρειαστεί περισσότερες δοκιμές απ' ότι αν ήταν μεγαλύτερο. Στην υποενότητα 2.4 δίνονται οι συχνότητες του ελληνικού αλφαβήτου. Πιο συχνά εμφανίζονται τα Α,Ο,Ι,Ε,Τ,Σ. Στο κείμενό μας το Θ έχει τη μεγαλύτερη συχνότητα και επομένως το πιθανότερο είναι να είναι το α. 5 φορές εμφανίζεται το Υ, οπότε το πιθανότερο είναι να είναι το ο. Όμως, θα μπορούσε να είναι και κάποιο από τα ι, ε, τ, σ. Από 4 φορές εμφανίζονται το Τ και το Ε. Δηλαδή, για τα Υ,Τ,Ε ∈ {ο, ι, ε, σ}. Το κανονικό κείμενο το γράφουμε με μικρά γράμματα.

$\Theta \rightarrow \alpha$

TPNALXCTPΦΛαΞΞααOYMNαΥαΤΥαTKTΓEOEΩEMNaΨKaTBaΨEMa

Επικεντρώνουμε την προσοχή μας στη διαδοχή των γραμμάτων αΥαΤΥαΤ. Με δοκιμές βρίσκουμε ότι, αν $Υ \rightarrow \tau$, $T \rightarrow \sigma$, σχηματίζεται αταστας. Οπότε, καταλαβαίνουμε ότι

K → η, N → κ, M → ι, O → ν.

τPκΛXCTPΦΛαΞΞα αντικαταστασης ΓΕνΕΩΕικαΨηασΒαΨΕια

Παρατηρούμε ότι η διαδοχή γραμμάτων **ασΒαΨΕια** θα μπορούσε να είναι η λέξη **ασφάλεια**. Οπότε, δοκιμάζουμε $B \rightarrow \phi$, $\Psi \rightarrow \lambda$ και $E \rightarrow \epsilon$. Τελικά παίρνουμε

τPκΛXCTPΦΛαΞΞααντικαταστασηςΓενΕΩΕικαΨηασφαλεια

Δοκιμάζουμε την αντικατάσταση $\Gamma \rightarrow \delta$, $\Omega \rightarrow x$.

τPκΛXCTPΦΛαΞΞααντικαταστασηςδενεχεικαλησφαλεια

Τέλος, προκύπτει το μήνυμα

το κρυπτόγραμμα αντικατάστασης δεν έχει καλή ασφάλεια

Τέτοια συστήματα χρησιμοποίησαν οι συγγραφείς Conan Doyle & E. A. Poe στα έργα τους *The Dancing Men* και ο *Ο Χρυσός σκαρβαίος*, αντίστοιχα. Επίσης, μπορείτε να δείτε το [4].

2.4 Οι συχνότητες γραμμάτων του ελληνικού αλφαβήτου

Σε κάθε φυσική γλώσσα η συχνότητα εμφάνισης των γραμμάτων παραμένει στατιστικά ίδια. Στην ελληνική γλώσσα συλλέγοντας κείμενα από διάφορες πηγές (εφημερίδες, βιβλία, blogs, Wikipedia κ.ά.) και χρησιμοποιώντας προγράμματα που μετράνε τη συχνότητα των χαρακτήρων καταλήξαμε στον παρακάτω πίνακα (η δεύτερη γραμμή δίνει τις συχνότητες %). Το δείγμα μας αποτελούνταν από 52693 χαρακτήρες.

A	B	Γ	Δ	E	Z	H	Θ	I	K	Λ	M
11.68	0.65	1.82	1.74	8.33	0.37	5.11	1.21	9.63	4.28	2.88	3.61

N	Ξ	O	Π	P	Σ	T	Υ	Φ	X	Ψ	Ω
6.28	0.45	10.25	4.11	4.13	7.73	7.81	4.32	0.88	0.46	0.16	2.11

2.5 Συστήματα μετατόπισης

Μια ειδική κατηγορία των κρυπτοσυστημάτων αντικατάστασης είναι τα συστήματα μετατόπισης, στα οποία το κρυπτογραφημένο μήνυμα προκύπτει με μετακίνηση όλων των γραμμάτων κατά k θέσεις. Για παράδειγμα, ο Ιούλιος Καίσαρας χρησιμοποίησε $k = 3$. Έτσι, το μήνυμα *συνάντησέ με στις πέντε* γίνεται

ΦΨΠΔΧΚΦΘΟΘΦΧΜΦΤΘΠΧΘ.

Αν διαλέξουμε $k = 5$, τότε το ίδιο μήνυμα κρυπτογραφείται ως

ΨΑΣΖΩΜΨΚΡΚΨΩΞΨΦΚΣΩΚ.

Παρατηρήστε ότι το $Y \rightarrow A$. Εύκολα διαπιστώνουμε ότι το ίδιο αποτέλεσμα θα πάρουμε, αν αντί του $k = 5$ θέσουμε $k = 29$ ή $k = 53$. Επομένως, υπάρχουν μόνο 24 δυνατότητες, για να βρούμε το κλειδί (πρακτικά έχουμε 23 δυνατότητες). Το απλό αυτό κρυπτοσύστημα ονομάζεται και κρυπτοσύστημα του Καίσαρα. Ένας άλλος τρόπος να παρουσιάσουμε αυτό το σύστημα είναι με πρόσθεση mod 24. Κάνουμε την ακόλουθη σύμβαση:

A	B	Γ	Δ	E	Z	H	Θ	I	K	Λ	M	N	Ξ
0	1	2	3	4	5	6	7	8	9	10	11	12	13

O	Π	P	Σ	T	Υ	Φ	X	Ψ	Ω
14	15	16	17	18	19	20	21	22	23

Τότε γράφουμε το αρχικό μας μήνυμα σύμφωνα με την προηγούμενη κωδικοποίηση και σε κάθε αριθμό προσθέτουμε το $k \bmod 24$. Δηλαδή, η συνάρτηση κρυπτογράφησης είναι

$$E_k(x) = x + k \bmod 24$$

και η συνάρτηση αποκρυπτογράφησης είναι

$$D_k(x) = x - k \bmod 24.$$

Στο σύστημα αυτό το σύνολο κλειδιών $\mathcal{K}$, το σύνολο μηνυμάτων $\mathcal{M}$ και το σύνολο κρυπτομηνυμάτων $\mathcal{C}$ είναι όλα ίσα με το σύνολο $\{0, 1, \dots, 23\}$.

Μια γενίκευση του συστήματος του Καίσαρα είναι τα αφφινικά κρυπτοσυστήματα. Σε αυτήν την περίπτωση η συνάρτηση κρυπτογράφησης είναι $E(x) = a \cdot x + b \bmod 24$. Ο πολλαπλασιασμός $a \cdot x$ είναι

$\text{mod } 24$. Δηλαδή, κάνω τον πολλαπλασιασμό, κατόπιν διαιρώ με το 24 και παίρνω το υπόλοιπο. Π.χ. αν $a = 3, x = 12$, τότε $3 \cdot 12 = 36$ και $36 = 24 \cdot 1 + 12$, άρα $a \cdot x = 12 \text{ mod } 24$. Η συνάρτηση αποκρυπτογράφησης είναι $D(y) = a^{-1}y - a^{-1}b \text{ mod } 24$. Εδώ χρειάζεται προσοχή, διότι, για να υπάρχει το αντίστροφο του $a \text{ mod } 24$, πρέπει το a και το 24 να μην έχουν κοινούς διαιρέτες. Για να βρω το αντίστροφο του $a \text{ mod } 24$, αρκεί να βρω έναν ακέραιο z με $0 < z \leq 23$, τέτοιον ώστε $z \cdot a \equiv 1 \text{ mod } 24$. Στην περίπτωση μας με λίγες δοκιμές μπορώ να το υπολογίσω. Στα αφφινικά συστήματα το κλειδί είναι το (a, b) . Το πλήθος των δυνατών κλειδιών είναι 192, αφού έχω 24 δυνατότητες για το b και 8 δυνατότητες για το a . Φυσικά και αυτό το σύστημα μπορεί με εξαντλητική αναζήτηση (brute force) να παραβιαστεί εύκολα.

Άσκηση 2.1 Πόσα διαφορετικά κλειδιά υπάρχουν στο αφφινικό κρυπτοσύστημα, αν χρησιμοποιήσουμε το αγγλικό αλφάβητο;

2.6 Κρυπτοσυστήματα μετάθεσης

Έστω $\mathcal{A}$ ένα αλφάβητο με $|\mathcal{A}|$ στοιχεία. Έστω, επίσης, ένα μήνυμα μήκους n , το οποίο το χωρίζω σε τμήματα μήκους $m < |\mathcal{A}|$. Κατόπιν, μεταθέτω με έναν συγκεκριμένο τρόπο τα σύμβολα του μηνύματος μεταξύ τους, χωρίς να προσθέσω κάποια νέα σύμβολα. Αν το m δεν διαιρεί το n , τότε συμπληρώνουμε αυθαίρετα κάποια γράμματα (pad), ώστε το νέο μήνυμα να διαιρείται από το m . Αυτά τα συστήματα ονομάζονται κρυπτοσυστήματα μετάθεσης.

Οι μεταθέσεις m -συμβόλων είναι συναρτήσεις

$$\sigma : \{1, \dots, m\} \rightarrow \{1, \dots, m\}$$

που είναι 1-1 (ένα προς ένα) και επί. Για να κρυπτογραφήσουμε με αυτό το σύστημα, επιλέγουμε μια μετάθεση του συνόλου $\{1, 2, \dots, m\}$ και χωρίζουμε το κείμενο σε τμήματα μήκους m . Μεταθέτουμε κάθε γράμμα ενός τμήματος σύμφωνα με τη μετάθεση. Το μήνυμα που στέλνουμε δεν περιέχει κενά, διότι αυτό θα αποκαλύπτει το μήκος m . Η αποκρυπτογράφηση προκύπτει, αν θεωρήσουμε την αντίστροφη μετάθεση. Π.χ. αν

$$\sigma = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}$$

τότε η

$$\sigma^{-1} = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}$$

Για παράδειγμα, ο πίνακας μετάθεσης στο διπλανό σχήμα μετασχηματίζει τη λέξη

ΛΑΘΟΣ στην ΑΘΛΣΟ.

1	2	3	4	5
3	1	2	5	4

Μεγαλύτερα κείμενα πρέπει να διαιρεθούν αρχικά σε πεντάδες και κατόπιν κάθε τμήμα να κρυπτογραφηθεί σύμφωνα με τη μετάθεση. Π.χ. η λέξη ΥΠΟΧΩΡΗΣΤΕ διαιρείται σε δύο τμήματα μήκους 5 ΥΠΟΧΩ-ΡΗΣΤΕ και κάθε τμήμα κρυπτογραφείται ανεξάρτητα ΠΟΥΩΧ-ΗΣΡΕΤ. Το κρυπτογραφημένο μήνυμα είναι ΠΟΥΩΧΗΣΡΕΤ.

Για να αποκρυπτογραφήσουμε, πρέπει να γνωρίζουμε το κλειδί (δηλαδή τη μετάθεση) και υπολογίζουμε την αντίστροφη μετάθεση που είναι η εξής:

1	2	3	4	5
2	3	1	5	4

Επομένως, το κρυπτογραφημένο μήνυμα *ΠΟΥΩΧΗΣΡΕΤ* διαιρείται σε πεντάδες (δηλαδή όσο το μήκος της μετάθεσης) *ΠΟΥΩΧ-ΗΣΡΕΤ* και κατόπιν εφαρμόζουμε τη μετάθεση, οπότε έχουμε *ΥΠΟΧΩ-ΡΗΣΤΕ*. Αν θέλαμε να γράψουμε τον τύπο κρυπτογράφησης, θα χρησιμοποιούσαμε τη συνάρτηση,

$$E_{\sigma}(x_1, x_2, x_3, x_4, x_5) = (x_{\sigma^{-1}(1)}, x_{\sigma^{-1}(2)}, x_{\sigma^{-1}(3)}, x_{\sigma^{-1}(4)}, x_{\sigma^{-1}(5)}) = (x_2, x_3, x_1, x_4, x_5)$$

ενώ η αποκρυπτογράφηση

$$D_{\sigma}(x_1, x_2, x_3, x_4, x_5) = (x_{\sigma(1)}, x_{\sigma(2)}, x_{\sigma(3)}, x_{\sigma(4)}, x_{\sigma(5)}) = (x_3, x_1, x_2, x_5, x_4) \quad ^{18}.$$

Ο παραπάνω πίνακας σ είναι το κλειδί του συστήματος. Κάθε διαφορετικό κλειδί αντιστοιχεί σε έναν διαφορετικό πίνακα μετάθεσης. Υπάρχουν $5! = 120$ διαφορετικά κλειδιά. Γενικά, για μία μετάθεση μήκους m έχουμε $m!$ κλειδιά.

Μερικές φορές η μετάθεση των συμβόλων ενός κειμένου ακολουθεί κάποιο γεωμετρικό μοτίβο. Για παράδειγμα, ο Rail Fence Cipher δουλεύει ως εξής: έστω ότι θέλουμε να κρυπτογραφήσουμε το μήνυμα: *ΥΠΟΧΩΡΗΣΤΕ*. Κατασκευάζουμε τον πίνακα κρυπτογράφησης και ξαναγράφουμε το μήνυμα ως εξής:

Υ	•	•	•	Ω	•	•	•	Τ	•
•	Π	•	Χ	•	Ρ	•	Σ	•	Ε
•	•	Ο	•	•	•	Η	•	•	•

Οπότε το μήνυμα κρυπτογραφείται : *ΥΩΤΠΧΡΣΕΟΗ*. Στο σύστημα αυτό το κλειδί είναι το $k = 3$. Έχοντας το μήκος του κλειδιού μπορούμε εύκολα να κατασκευάσουμε ξανά τον πίνακα κρυπτογράφησης ως εξής: κατασκευάζουμε έναν κενό πίνακα με διαστάσεις $k \times \ell$, όπου ℓ το μήκος του μηνύματος. Ξεκινώντας από τη θέση πάνω αριστερά γράφουμε το πρώτο γράμμα του κρυπτογραφημένου μηνύματος, δηλαδή το Υ. Ακολουθούμε το σχήμα του Rail fence και το επόμενο γράμμα το γράφουμε μόλις φτάσουμε στην πρώτη γραμμή, δηλαδή το Ω. Συνεχίζοντας, εύκολα κατασκευάζουμε τον πίνακα κρυπτογράφησης.

Μπορούμε εύκολα να κρυπταναλύσουμε τον Rail Fence. Αν έχω ένα μήνυμα μήκους ℓ , τότε το πλήθος των δυνατών κλειδιών είναι $\ell - 1$, οπότε για σχετικά μικρά μηνύματα ($\ell < 2^{40}$) μπορούμε με εξαντλητική αναζήτηση (brute force) να τον αποκρυπτογραφήσουμε.

2.7 Το σύστημα του Playfair

Σε αυτό το κρυπτοσύστημα δεν υπάρχει η αδυναμία που έχουν τα κρυπτοσυστήματα αντικατάστασης. Δηλαδή, δεν μπορούμε να επιτεθούμε με στατιστική ανάλυση. Για να κρυπτογραφήσουμε, χρειαζόμαστε μία φράση κλειδί, έστω key : matrix. Σχηματίζουμε έναν πίνακα 5×5 (key table) ως εξής:

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

Τα διπλά γράμματα παραλείπονται και το J το αντικαθιστώ με το I. Κατόπιν, χωρίζω το αρχικό κείμενο σε δυάδες γραμμάτων. Έστω ότι το κείμενο που θα κρυπτογραφήσουμε είναι *hide the gold*. Γράφουμε το κείμενο ως εξής: *HI-DE-TH-EG-OL-DX* και στο τέλος συμπληρώνουμε με ένα γράμμα, έστω το X, ώστε

¹⁸Στη βιβλιογραφία επίσης αναφέρεται ως σύστημα μετάθεσης το σύστημα με

$$E_{\sigma}(x_1, x_2, x_3, x_4, x_5) = (x_{\sigma(1)}, x_{\sigma(2)}, x_{\sigma(3)}, x_{\sigma(4)}, x_{\sigma(5)}) = (x_3, x_1, x_2, x_5, x_4)$$

και

$$D_{\sigma}(x_1, x_2, x_3, x_4, x_5) = (x_{\sigma^{-1}(1)}, x_{\sigma^{-1}(2)}, x_{\sigma^{-1}(3)}, x_{\sigma^{-1}(4)}, x_{\sigma^{-1}(5)}) = (x_2, x_3, x_1, x_4, x_5).$$

να μην περισσεύει κανένα. Εφαρμόζουμε τους παρακάτω κανόνες:

1. Αν η δυάδα του κείμενου είναι σε διαφορετικές γραμμές και στήλες, τότε σχηματίζω το ορθογώνιο που ορίζουν και αντικαθιστώ κάθε γράμμα με την απέναντι κορυφή του.

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

Έτσι, το HI κρυπτογραφείται ως LT.

2. Αν η δυάδα βρίσκεται στην ίδια γραμμή, μεταθέτω μία θέση δεξιά τα γράμματα. Έτσι το DE, γίνεται EX.

3. Αν η δυάδα βρίσκεται στην ίδια στήλη, μεταθέτω μία θέση προς τα κάτω τα γράμματα. Έτσι το TH, γίνεται CP.

4. Αν η δυάδα μας έχει ίδια γράμματα, τα χωρίζω με ένα X. Έτσι, για παράδειγμα, η λέξη corresponds θα γραφόταν πρώτα co-ix-re-sp-on-ds και κατόπιν θα εφαρμόζα τους προηγούμενους κανόνες.

Τέλος, το μήνυμα που προκύπτει είναι: LT-EX-CP-BL-SG-EB. Για να αποκρυπτογραφήσουμε, αντιστρέφουμε του κανόνες κρυπτογράφησης.

2.8 Το σύστημα του Vigenère

Το κρυπτοσύστημα του Vigenère ανακαλύφθηκε το 1553 από τον Giovan Battista Bellaso. Επίσης, ανακαλύφθηκε ανεξάρτητα από τον Vigenère το 1586. Αποτελεί γενίκευση του κρυπτοσυστήματος του Καίσαρα (σύστημα μετατόπισης). Για να κρυπτογραφήσει κάποιος με αυτό το σύστημα, χρειάζεται ένα κλειδί και το τετράγωνο του Vigenère. Ένας άλλος τρόπος είναι να χρησιμοποιήσουμε την αριθμητική mod 24 για το ελληνικό αλφάβητο και mod 26 για το αγγλικό. Με τη δεύτερη μέθοδο, αντιστοιχίζουμε κάθε γράμμα σε έναν αριθμό ξεκινώντας από το μηδέν. Δηλαδή, το γράμμα Α στο 0, το Β στο 1, κ.ο.κ. Έστω ότι το κλειδί είναι αβγδ, δηλαδή 0,1,2,3 και το μήνυμά μας είναι ΑΥΡΙΟ ΘΑ ΓΙΝΕΙ ΕΠΙΘΕΣΗ, δηλαδή 0-19-..., τότε για να κρυπτογραφήσουμε, προσθέτουμε mod 24, γράμμα προς γράμμα. Άρα, το κρυπτογραφημένο κείμενο θα ξεκινάει ως εξής: 0 – 20 – ... και θα καταλήξουμε στο κείμενο

ΑΦΤΜΟΙΓΖΙΞΗΜΕΡΑΛΕΤΙ.

Επιπρόσθετα, και εδώ χρησιμοποιούμε την κωδικοποίηση,

Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ
0	1	2	3	4	5	6	7	8	9	10	11	12	13

Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω
14	15	16	17	18	19	20	21	22	23

Πίνακας 1: Πίνακας κωδικοποίησης ελληνικού αλφαβήτου

Ας δούμε πώς γίνεται η κρυπτογράφηση με την πρώτη μέθοδο (δηλαδή με το τετράγωνο του Vigenère). Έστω ότι έχουμε το παρακάτω κλειδί και μήνυμα:

Κλειδί : ΜΕΡΑ

Μήνυμα : ΑΥΡΙΟ

Κρυπτογράφηση : Το κλειδί ΜΕΡΑ μας δείχνει με ποιον τρόπο να επιλέξω το αλφάβητο για κάθε γράμμα του μηνύματος. Το πρώτο γράμμα του κλειδιού είναι το Μ. Επομένως, το πρώτο γράμμα του μηνύματος, δηλαδή το Α, θα κρυπτογραφηθεί από το αλφάβητο που βρίσκεται στη γραμμή που ξεκινάει από το γράμμα **μ**. Για να βρούμε με ποιο γράμμα θα αντικαταστήσουμε το Α, βρίσκουμε το γράμμα που βρίσκεται στη διασταύρωση

της στήλης Α και της γραμμής Μ. Το γράμμα αυτό είναι το **μ**. Κατόπιν, το Υ θα κρυπτογραφηθεί από το αλφάβητο που ξεκινάει από το γράμμα **ε**. Συνεπώς, το δεύτερο γράμμα του μηνύματος, το Υ, θα αντικατασταθεί από το **ω**. Συνεχίζοντας καταλήγουμε στη λέξη **μωιβ**. Δηλαδή, για να βρούμε το n -οστό γράμμα του μηνύματος, βρίσκουμε το γράμμα που βρίσκεται στη διασταύρωση της γραμμής που δίνεται από το n -οστό γράμμα του κλειδιού (επαναλαμβάνουμε το κλειδί μέχρι να εξαντληθούν τα γράμματα του μηνύματος) και της στήλης που ξεκινάει από το n -οστό γράμμα του μηνύματος.

α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο	π	ρ	σ	τ	υ	φ	χ	ψ	ω
β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο	π	ρ	σ	τ	υ	φ	χ	ψ	ω	α
γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο	π	ρ	σ	τ	υ	φ	χ	ψ	ω	α	β
δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο	π	ρ	σ	τ	υ	φ	χ	ψ	ω	α	β	γ
ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο	π	ρ	σ	τ	υ	φ	χ	ψ	ω	α	β	γ	δ
ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο	π	ρ	σ	τ	υ	φ	χ	ψ	ω	α	β	γ	δ	ε
η	θ	ι	κ	λ	μ	ν	ξ	ο	π	ρ	σ	τ	υ	φ	χ	ψ	ω	α	β	γ	δ	ε	ζ
θ	ι	κ	λ	μ	ν	ξ	ο	π	ρ	σ	τ	υ	φ	χ	ψ	ω	α	β	γ	δ	ε	ζ	η
ι	κ	λ	μ	ν	ξ	ο	π	ρ	σ	τ	υ	φ	χ	ψ	ω	α	β	γ	δ	ε	ζ	η	θ
κ	λ	μ	ν	ξ	ο	π	ρ	σ	τ	υ	φ	χ	ψ	ω	α	β	γ	δ	ε	ζ	η	θ	ι
λ	μ	ν	ξ	ο	π	ρ	σ	τ	υ	φ	χ	ψ	ω	α	β	γ	δ	ε	ζ	η	θ	ι	κ
μ	ν	ξ	ο	π	ρ	σ	τ	υ	φ	χ	ψ	ω	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ
ν	ξ	ο	π	ρ	σ	τ	υ	φ	χ	ψ	ω	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ
ξ	ο	π	ρ	σ	τ	υ	φ	χ	ψ	ω	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν
ο	π	ρ	σ	τ	υ	φ	χ	ψ	ω	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ
π	ρ	σ	τ	υ	φ	χ	ψ	ω	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο
ρ	σ	τ	υ	φ	χ	ψ	ω	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο	π
σ	τ	υ	φ	χ	ψ	ω	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο	π	ρ
τ	υ	φ	χ	ψ	ω	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο	π	ρ	σ
υ	φ	χ	ψ	ω	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο	π	ρ	σ	τ
φ	χ	ψ	ω	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο	π	ρ	σ	τ	υ
χ	ψ	ω	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο	π	ρ	σ	τ	υ	φ
ψ	ω	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο	π	ρ	σ	τ	υ	φ	χ
ω	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο	π	ρ	σ	τ	υ	φ	χ	ψ

Το τετράγωνο που χρησιμοποιήσαμε ονομάζεται τετράγωνο του Vigenère. Το σύστημα του Vigenère ανήκει στην ομάδα των κρυπτοσυστημάτων πολυαλφαβητικής αντικατάστασης. Το μεγάλο πλεονέκτημα του συστήματος αυτού είναι ότι δεν δουλεύει η ανάλυση των συχνοτήτων. Αυτό συμβαίνει διότι ένα γράμμα μπορεί να κρυπτογραφείται με τόσους τρόπους όσους το μήκος του κλειδιού. Αν το μήκος του κλειδιού είναι ένα, τότε ταυτίζεται με το κρυπτοσύστημα της μετατόπισης. Για να κρυπταναλυθεί ένα κείμενο που έχει προκύψει με τον προηγούμενο τρόπο, βασιζόμαστε στην εξής παρατήρηση: το πλήθος των χαρακτήρων μεταξύ δύο επαναλαμβανόμενων λέξεων πρέπει να είναι πολλαπλάσιο του μήκους του κλειδιού. Ας πάρουμε το ακόλουθο παράδειγμα, για να καταλάβουμε την προηγούμενη (σημαντική) παρατήρηση. Αν πάρουμε τη λέξη κλειδί: **ΒΟΥΝΟ** και κρυπτογραφήσουμε το μήνυμα:

την ανατολή δεν θα επιτεθούμε δεν θα δεχτούμε συνθηκολόγηση δεν θα οπισθοχωρήσουμε

θα πάρουμε το κείμενο

υφθγβικψφετθνοζδητιεωωτετθνοετρηεφ
βωζκξχβχεμεχτθσωαχβελφθιερμηήτεωωτ

Με bold fonts είναι η κρυπτογραφημένη λέξη **δεν**. Η απόσταση d των δύο πρώτων που συμπίπτουν είναι 15. Το κλειδί έχει μήκος $d_K = 5$ και $d_K | d$ (δηλαδή το d_K διαιρεί το d). Το προηγούμενο τεστ ονομάζεται Τεστ του Kasiski. Φυσικά, υπάρχει περίπτωση μια συμβολοσειρά που εμφανίζεται στο αρχικό κείμενο δύο φορές, στο κρυπτογραφημένο κείμενο να μην επαναλαμβάνεται με παρόμοιο τρόπο. Ας δούμε το παρακάτω παράδειγμα κάνοντας χρήση του κλειδιού εδώ.

ό,τι είναι να γίνει ας γίνει γρήγορα

θα πάρουμε το κείμενο

λπκαζξφζξφωκιβκφοδεκζεωσγωπνχ

Παρατηρήστε ότι η λέξη *γίνει* έχει κρυπτογραφηθεί διαφορετικά. Η παρατήρηση του Kasiski ήταν ότι δύο συμβολοσειρές επαναλαμβάνονται στο αρχικό κείμενο και ευθυγραμμίζονται με το κλειδί με τον ίδιο ακριβώς τρόπο, όταν η απόσταση των αρχικών γραμμμάτων των συμβολοσειρών είναι πολλαπλάσιο του μήκους του κλειδιού. Όπως είπαμε, ενδέχεται κάποια λέξη στο κρυπτογραφημένο κείμενο να επαναλαμβάνεται, αλλά παρόλα αυτά να μην μας δίνει κάποια πληροφορία για το μήκος του κλειδιού. Για να πετύχει το τεστ Kasiski, πρέπει να αναζητούμε μεγάλες λέξεις που επαναλαμβάνονται (μήκους > 3) και κυρίως να επαναλαμβάνονται πολλές φορές. Σε αυτήν την περίπτωση είναι απίθανο δύο επαναλήψεις να είναι τυχαίες. Εφόσον βρούμε τα διάφορα μήκη, τότε το μήκος του κλειδιού θα διαιρεί τον μέγιστο κοινό διαιρέτη των μηκών που βρήκαμε. Με αυτόν τον τρόπο περιορίζουμε αρκετά την αναζήτησή μας για το μήκος του κλειδιού. Κατόπιν, δοκιμάζουμε με το μεγαλύτερο μήκος που βρήκαμε και χρησιμοποιούμε τον δείκτη σύμπτωσης που περιγράφουμε παρακάτω.

Η μέθοδος του Friedman χρησιμοποιεί τον δείκτη σύμπτωσης, για να ελέγξει αν το μήκος του κλειδιού που βρήκαμε είναι το σωστό. Ο δείκτης σύμπτωσης ορίζεται ως εξής:

$$IC = \sum_{l=1}^n \frac{m_l(m_l - 1)}{k(k - 1)}, \quad (2.8.1)$$

όπου n το μήκος τους αλφαβήτου μας, m_l το πλήθος εμφανίσεων του l -χαρακτήρα του αλφαβήτου μας και k το μήκος του κειμένου που εξετάζουμε. Ο δείκτης σύμπτωσης μας δίνει την πιθανότητα να διαλέξουμε (η επιλογή γίνεται ομοιόμορφα) δύο όμοια γράμματα από ένα κρυπτογραφημένο κείμενο. Αν το κείμενο είναι αγγλικό και έχουμε μονοαλφαβητική αντικατάσταση, τότε ο IC είναι κοντά στο 0.0665 (θα δούμε παρακάτω γιατί συμβαίνει αυτό). Αν έχουμε πολυαλφαβητική αντικατάσταση (όπως στου Vigenère), τότε ο δείκτης σύμπτωσης κινείται ανάμεσα στο 0.0385 και 0.0665. Επομένως, αν μας δώσουν ένα κείμενο, υπολογίζοντας τον δείκτη σύμπτωσης μπορούμε να καταλάβουμε αν προέρχεται από μονοαλφαβητική ή πολυαλφαβητική αντικατάσταση. Αν έχουμε το αγγλικό αλφάβητο, ο τύπος (2.8.1) μπορεί να γραφεί

$$IC \approx \sum_{l=1}^n \left(\frac{m_l}{k} \right)^2 = (0.0812)^2 + (0.0149)^2 + \dots \approx 0.0665.$$

Ο κάθε όρος μέσα στο άθροισμα είναι η σχετική συχνότητα εμφάνισης ενός χαρακτήρα σε ένα αγγλικό κείμενο και δίνεται από τον παρακάτω πίνακα.

A	B	C	D	E	F	G	H	I	J	K	L	M
8.12	1.49	2.71	4.32	12.02	2.3	2.03	5.92	7.31	0.1	0.69	3.92	2.61

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
6.95	7.68	1.82	0.11	6.02	6.28	9.10	2.88	1.11	2.09	0.17	2.11	0.07

Για κείμενα γραμμένα στην ελληνική γλώσσα ο δείκτης σύμπτωσης είναι κοντά στον αριθμό 0.0677. Ένας τρόπος να χρησιμοποιήσουμε τον δείκτη σύμπτωσης είναι να υποθέσουμε ότι το κλειδί είναι μήκους ρ , μετά να ομαδοποιήσουμε το κρυπτογραφημένο μήνυμα σε έναν πίνακα με ρ στήλες και τέλος, να υπολογίσουμε σε μία στήλη τον δείκτη σύμπτωσης. Αν αυτός είναι αρκετά κοντά στο 0.0665 (για αγγλικά κείμενα), τότε με μεγάλη πιθανότητα το κλειδί έχει μήκος ρ .

Αλγόριθμος 2.8.1. : Τεστ του Friedman

Είσοδος. Το κρυπτοκείμενο $c = (c_1, c_2, \dots, c_k)$ και την παράμετρο $\alpha \in \{0.0665, 0.0677\}$

Έξοδος. (Πιθανό) μήκος κλειδιού

Initialization

```

1  $\rho \leftarrow 2$ 
2  $found \leftarrow false$ 
  while  $found = false$  do
3   κατασκεύασε πίνακα με στήλες τα  $\rho$  διανύσματα
      $C_1 = [c_1, c_{1+\rho}, \dots]$ 
      $C_2 = [c_2, c_{2+\rho}, \dots]$ 
     ...
      $C_k = [c_k, c_{k+\rho}, \dots]$ 
4   for  $i = 1$  to  $\rho$  do
5     if  $IC(C_i)$  είναι κοντά στο  $\alpha$  then
6        $found \leftarrow true$ 
7       return  $\rho$ 
     end
  end
8  $\rho \leftarrow \rho + 1$ 
end

```

Άσκηση 2.2 Το επόμενο κρυπτόγραμμα έχει ληφθεί:

οκηθμφδζθγοθχυκχσφθμφμχγ

Ο αλγόριθμος κρυπτογράφησης είναι ο εξής: Κάθε γράμμα του αρχικού μας μηνύματος αντικαθίσταται από την αριθμητική του τιμή ($\alpha \rightarrow 1, \dots, \omega \rightarrow 24$). Ας είναι x_0 μία ρίζα του τριωνύμου $g(x) = x^2 + 3x + 1$. Σε κάθε αριθμό του μηνυμάτος μου προσθέτω την τιμή του πολυωνύμου $f(x) = x^5 + 3x^3 + 7x^2 + 3x^4 + 5x + 4$, στο x_0 . Αντικαθιστώ κάθε αριθμό με το αντίστοιχο γράμμα. Βρείτε το αρχικό μήνυμα.

Άσκηση 2.3 Αποκρυπτογραφήστε το κείμενο [2] [εδώ](#)¹⁹, που κρυπτογραφήθηκε με τον αλγόριθμο του Vigenère.

Υπόδ. Για την αποκρυπτογράφηση συστήνουμε να χρησιμοποιήσετε python. Για το μήκος του κλειδιού μπορείτε να χρησιμοποιήσετε είτε test Kasiski ή τη μέθοδο του Friedman.

Άσκηση 2.4 Έστω ένα μήνυμα m , 16-bits. Θεωρούμε την κυκλική κύλιση προς τα αριστερά $\ll \alpha$ κατά α bits. Έστω ότι το m κωδικοποιείται στο c σύμφωνα με τον τύπο,

$$c = m \oplus (m \ll 6) \oplus (m \ll 10).$$

Βρείτε τον τύπο αποκωδικοποίησης. Δηλαδή, γράψτε το m ως συνάρτηση του c . Υλοποιήστε κατάλληλο κώδικα, για να δείξετε ότι ο τύπος που φτιάξατε είναι σωστός.

2.9 Σημειωματάριο Μίας Φοράς

Το σημειωματάριο μίας φορές (One Time Pad - OTP) είναι το ασφαλέστερο κρυπτοσύστημα που υπάρχει. Για πρακτικούς λόγους, που θα εξηγήσουμε παρακάτω, έχει περιοριστεί η χρήση του (μπορείτε να συμβουλευτείτε και το βιβλίο [6, Ενότητα 1.4.3.]).

Ορισμός 2.9.1. Έστω n ένας φυσικός αριθμός. Ας είναι $\mathcal{K}, \mathcal{M}, \mathcal{C}$ ο χώρος κλειδιών, μηνυμάτων και κρυπτογραφημένων μηνυμάτων αντίστοιχα και $E : \mathcal{M} \times \mathcal{K} \rightarrow \mathcal{C}$, $D : \mathcal{C} \times \mathcal{K} \rightarrow \mathcal{M}$ οι συναρτήσεις κρυπτογράφησης και αποκρυπτογράφησης αντίστοιχα. Έστω $\mathcal{K} = \mathcal{M} = \mathcal{C} = \{0, 1\}^n$. Έστω m το μήνυμα

¹⁹https://github.com/drazioti/book_crypto/blob/master/symmetric_crypto/vigenere

που επιθυμούμε να κρυπτογραφήσουμε, μήκους n -bits. Διαλέγουμε το κλειδί k τυχαία²⁰ από το σύνολο $\mathcal{K}$, μήκους n -bits. Ορίζουμε το κρυπτοσύστημα OTP ,

$$E(m, k) = m \oplus k, \quad D(c, k) = c \oplus k.$$

Είναι καλά ορισμένο κρυπτοσύστημα, διότι αν $c = E(m, k)$ ισχύει:

$$D(c, k) = c \oplus k = (m \oplus k) \oplus k = m \oplus (k \oplus k) = m.$$

Ο OTP έχει πολύ γρήγορη κρυπτογράφηση και αποκρυπτογράφηση. Παρόλα αυτά, έχει μεγάλο μήκος κλειδιού και αυτό τον καθιστά μη πρακτικό. Το ερώτημα είναι αν είναι αρκετά ασφαλής, για να τον χρησιμοποιήσω. Στο ερώτημα αυτό απάντησε ο Shannon. Απέδειξε το 1949, ότι ο OTP έχει τέλεια ασφάλεια (perfect security). Τέλεια ασφάλεια σημαίνει ότι, αν κάποιος έχει το κρυπτομήνυμα c , αυτό δεν του παρέχει καμιά πληροφορία για το αρχικό μήνυμα m .

Ορισμός 2.9.2. (Τέλεια Ασφάλεια κατά Shannon) Έστω $(E, D, \mathcal{M}, \mathcal{C}, \mathcal{K})$ ένα κρυπτοσύστημα. Λέμε ότι το σύστημα έχει τέλεια ασφάλεια, αν για κάθε m_0, m_1 μηνύματα ίδιου μήκους του $\mathcal{M}$ και $c \in \mathcal{C}$ ισχύει

$$Pr(E(k, m_0) = c) = Pr(E(k, m_1) = c)$$

για κάθε $k \xleftarrow{\$} \mathcal{K}$.

Δηλαδή, αν η Εύα κλέψει το c , τότε δεν έχει ιδέα αν αυτό προήλθε από το m_0 ή από το m_1 .

Θεώρημα 2.9.1 (1ο Θεώρημα του Shannon) Ο OTP έχει τέλεια ασφάλεια.

Απόδειξη. Αρχεί να αποδείξουμε ότι για κάθε $m \in \mathcal{M}$ και $c \in \mathcal{C}$ η πιθανότητα $P(E(m, k) = c)$ (καθώς το k διατρέχει όλα τα κλειδιά στο $\mathcal{K}$) είναι ανεξάρτητη από το m . Πράγματι, τότε ικανοποιείται ο ορισμός τέλειας ασφάλειας του Shannon.

$$P(E(m, k) = c) = \frac{|\{k \in \mathcal{K} : E(m, k) = c\}|}{|\mathcal{K}|}.$$

Αλλά, $|\{k \in \mathcal{K} : E(m, k) = c\}| = 1$ και $|\mathcal{K}| = 2^n$, επομένως,

$$P(E(m, k) = c) = \frac{1}{2^n}.$$

Άρα, η πιθανότητα $P(E(m, k) = c)$ είναι ανεξάρτητη του m . □

Παρόλα αυτά, ο OTP δεν χρησιμοποιείται στην πράξη. Υπενθυμίζουμε ότι η βασική αδυναμία του OTP είναι ότι τα κλειδιά πρέπει να έχουν μήκος όσο το μήκος του μηνύματος. Μήπως υπάρχουν κρυπτοσυστήματα που έχουν τέλεια ασφάλεια, αλλά μικρού μήκους κλειδιά; Δίνουμε το παρακάτω Θεώρημα χωρίς απόδειξη.

Θεώρημα 2.9.2 (2ο Θεώρημα του Shannon). Αν ένα κρυπτοσύστημα έχει τέλεια ασφάλεια, τότε το μήκος του κλειδιού είναι τουλάχιστον όσο το μήκος του μηνύματος.

Άσκηση 2.5 Να αποδείξετε ότι, αν στο σύστημα μετατόπισης (δείτε υποενότητα 2.5) διαλέγουμε τυχαία τα κλειδιά από το σύνολο $\{0, 1, \dots, 23\}$, τότε το σύστημα έχει τέλεια ασφάλεια.

Άσκηση 2.6 Υλοποιήστε τον OTP , αφού αρχικά μετατρέψετε το μήνυμά σας σε bits με χρήση του παρακάτω πίνακα. Θα πρέπει να δουλεύουν η κρυπτογράφηση και η αποκρυπτογράφηση. Το μήνυμα δίνεται κανονικά και εσωτερικά μετατρέπεται σε bits. Το κλειδί είναι διαλεγμένο τυχαία και έχει μήκος όσο το μήκος του μηνύματός σας. Το αποτέλεσμα δίνεται όχι σε bits, αλλά με λατινικούς χαρακτήρες.

Άσκηση 2.7 Να εξηγήσετε γιατί στον OTP δεν έχει νόημα να κάνουμε εξαντλητική αναζήτηση (brute force) στο σύνολο των κλειδιών, αν έχουμε στην κατοχή μας κάποια ζευγάρια (m_i, c_i) .

²⁰ Αυτό το γράφουμε και $k \xleftarrow{\$} \mathcal{K}$

(0) A : 00000	(13) N : 01101
(1) B : 00001	(14) O : 01110
(2) C : 00010	(15) P : 01111
(3) D : 00011	(16) Q : 10000
(4) E : 00100	(17) R : 10001
(5) F : 00101	(18) S : 10010
(6) G : 00110	(19) T : 10011
(7) H : 00111	(20) U : 10100
(8) I : 01000	(21) V : 10101
(9) J : 01001	(22) W : 10110
(10) K : 01010	(23) X : 10111
(11) L : 01011	(24) Y : 11000
(12) M : 01100	(25) Z : 11001
(26) . : 11010	(27) ! : 11011
(28) ? : 11100	(29) (: 11101
(30)) : 11110	(31) - : 11111

Πίνακας 2: Κωδικοποίηση 5 bits

2.9.1 Διαχωριστές - Distinguishers

Ένας πιο ρεαλιστικός ορισμός ασφάλειας είναι η σημασιολογική ασφάλεια (semantic security) που βασίζεται στους διαχωριστές²¹. Δείτε [2, Ενότητα 3.11] και [3]. Αν διατυπώσουμε τους ορισμούς της σημασιολογικής ασφάλειας και της μη-ευδιακριτότητας (indistinguishability), αυτοί οι ορισμοί είναι διαφορετικοί. Ωστόσο, έχει αποδειχτεί ότι είναι ισοδύναμοι για την περίπτωση της CPA.

Ο ισχυρότερος ορισμός της ασφάλειας στους συμμετρικούς αλγόριθμους (αλλά και στην κρυπτογραφία δημόσιου κλειδιού) χρησιμοποιεί την έννοια του διαχωριστή-distinguisher. Ο ορισμός του διαχωριστή κάθε φορά προσαρμόζεται στο συγκεκριμένο κρυπτοσύστημα που αναφερόμαστε. Σε όλους τους ορισμούς όμως υπάρχουν κάποια κοινά στοιχεία.

(i). Ο διαχωριστής είναι μια διαδικασία που υπακούει στο υπολογιστικό μοντέλο της μηχανής Turing.

(ii). Ένας διαχωριστής $\mathcal{A}$ αλληλεπιδρά με τον κρυπταλγόριθμο ως μαύρο κουτί (black box) και επιλέγει τυχαία κλειδιά για τον κρυπταλγόριθμο.

(iii). Ο κρυπταλγόριθμος μετά την πρώτη κλήση από τον διαχωριστή απαντά με ένα τυχαίο bit b . Η Εύα, που χρησιμοποιεί τον διαχωριστή, εξάγει ένα bit, που είναι η έξοδος του $\mathcal{A}$. Ο διαχωριστής, ανάλογα με την επίθεση που υλοποιεί η Εύα, δέχεται ως είσοδο την επικοινωνία που έχει η Εύα με το σύστημα. Ο σκοπός της Εύας είναι να προσδιορίσει την τιμή του b με πιθανότητα $> 1/2$.

(iv). Η συνάρτηση πλεονεκτήματος (advantage function) δίνεται από τη σχέση

$$Adv(\mathcal{A}) = |Pr(\mathcal{A} \text{ εξάγει } 0 | b = 0) - Pr(\mathcal{A} \text{ εξάγει } 0 | b = 1)|.$$

Η $Adv(\mathcal{A}) \in [0, 1]$.

Ορισμός 2.9.3. Αν $Adv(\mathcal{A}) \gg 0$, τότε το σύστημα δεν είναι σημασιολογικά ασφαλές (*SS : Semantically Secure*).

Ισχύει το επόμενο.

Λήμμα 2.9.1. $Adv(\mathcal{A}) = |2Pr(\mathcal{A} \text{ εξάγει το } b) - 1|$.

Απόδειξη. Ισχύουν τα παρακάτω

$$Pr(\mathcal{A} \text{ εξάγει } b) = Pr(\mathcal{A} \text{ εξάγει } 0 | b = 0)Pr(b = 0) + Pr(\mathcal{A} \text{ εξάγει } 1 | b = 1)Pr(b = 1) =$$

²¹Y. Watanabe, J. Shikata, and H. Imai, Equivalence between Semantic Security and Indistinguishability against Chosen Ciphertext Attacks

$$Pr(\mathcal{A} \text{ εζάγει } 0|b = 0)/2 + Pr(\mathcal{A} \text{ εζάγει } 1|b = 1)/2.$$

Επίσης ισχύει

$$Pr(\mathcal{A} \text{ εζάγει } 0|b = 0) + Pr(\mathcal{A} \text{ εζάγει } 1|b = 0) = 1.$$

Επομένως,

$$\begin{aligned} Adv(\mathcal{A}) &= |Pr(\mathcal{A} \text{ εζάγει } 0|b = 0) - Pr(\mathcal{A} \text{ εζάγει } 0|b = 1)| = \\ &= |Pr(\mathcal{A} \text{ εζάγει } 0|b = 0) - (1 - Pr(\mathcal{A} \text{ εζάγει } 1|b = 1))| = \\ &= |Pr(\mathcal{A} \text{ εζάγει } 0|b = 0) + Pr(\mathcal{A} \text{ εζάγει } 1|b = 1) - 1| = \\ &= |2Pr(\mathcal{A} \text{ εζάγει το } b) - 1|. \end{aligned}$$

□

Υπάρχουν δύο τύποι διαχωριστών. Ο πρώτος τύπος χρησιμοποιεί τον κρυπταλγόριθμο, για να κρυπτογραφήσει μηνύματα που στέλνει ο διαχωριστής. Σε αυτήν την περίπτωση λέμε ότι υλοποιούμε μια επίθεση επιλεγμένου κειμένου (CPA: **C**hosen **P**laintext **A**ttack) και χρησιμοποιούμε τον κρυπταλγόριθμο ως μαντείο κρυπτογράφησης. Ο δεύτερος τύπος χρησιμοποιεί τον κρυπταλγόριθμο, για να αποκρυπτογραφήσει κρυπτογραφημένα μηνύματα που στέλνει ο διαχωριστής. Τότε λέμε ότι επιτελείται μια επίθεση επιλεγμένου κρυπτογραφημένου κειμένου (CCA: **C**hosen **C**iphertext **A**ttack)

Συνήθως, για να χρησιμοποιήσουμε έναν διαχωριστή, υλοποιούμε το παρακάτω παίγνιο.

Σημασιολογική ασφάλεια (Semantic Security)

- (i). Η Εύα διαλέγει δύο μηνύματα m_0, m_1 (ίδιου μήκους) και τα στέλνει στην Alice.
- (ii). Η Alice διαλέγει ένα τυχαίο bit b και ένα τυχαίο κλειδί και κρυπτογραφεί το μήνυμα $Enc(k, m_b) = c$. Τέλος, στέλνει το c στην Εύα.
- (iii). Η Εύα τώρα μαντεύει την επιλογή του b με τη βοήθεια του αλγόριθμου $\mathcal{A}$.

Σημασιολογικά ασφαλές (SS: Semantically Secure) κρυπτοσύστημα σημαίνει ότι η Εύα δεν έχει αποδοτικό αλγόριθμο που να μαντεύει σωστά την επιλογή του b με πιθανότητα σημαντικά μεγαλύτερη του $1/2$. Ισοδύναμα, αν

$$Adv_{SS}(\mathcal{A}) \gg 0,$$

τότε το σύστημα δεν είναι σημασιολογικά ασφαλές.

Σημασιολογική ασφάλεια υπό CPA-attacks (αντ. CCA-attacks)

- (i). Η Εύα (adversary) υλοποιεί μια επίθεση CPA (αντ. CCA). Αυτή η φάση είναι ανεξάρτητη από την επιλογή του b που η Εύα θέλει να προβλέψει με πιθανότητα $> 1/2$.
- (ii). Η Εύα διαλέγει κάποια μηνύματα $M_1, M_2, \dots, M_r$ (αντ. $C_1, C_2, \dots, C_r$) και ζητάει την κρυπτογράφησή τους (αντ. αποκρυπτογράφησή τους).
- (iii). Η Εύα διαλέγει δύο μηνύματα m_0, m_1 ίδιου μήκους και διαφορετικά από τα μηνύματα που επέλεξε στο προηγούμενο βήμα. Στέλνει τα μηνύματα στο κρυπτοσύστημα και δέχεται ως απάντηση ένα κρυπτογραφημένο μήνυμα $c = Enc(k, m_b)$, όπου b ένα τυχαίο bit.
- (iv). Η Εύα, αν θέλει, μπορεί και σε αυτήν τη φάση να στείλει μερικά μηνύματα για κρυπτογράφιση (ή αποκρυπτογράφιση) διαφορετικά από αυτά που έστειλε προηγουμένως.
- (v). Η Εύα τώρα μαντεύει την επιλογή του b .

SS-CPA (αντ. SS-CCA) σημαίνει ότι η Εύα δεν έχει αποδοτικό αλγόριθμο που να μαντεύει σωστά την επιλογή του b με πιθανότητα σημαντικά μεγαλύτερη του $1/2$. Ισοδύναμα

$$Adv_{CPA}(\mathcal{A}) \gg 0 \text{ (αντ. } Adv_{CCA}(\mathcal{A}) \gg 0).$$

Στην περίπτωση που η Εύα υλοποιεί CCA επίθεση και δεν επιτρέπεται να εκτελέσει το βήμα (iv), λέμε ότι εκτελεί μια CCA1 επίθεση. Μερικές φορές η επίθεση CCA, που περιγράψαμε, ονομάζεται και CCA2 επίθεση. Γενικά, από ένα κρυπτοσύστημα η ασφάλεια που απαιτούμε είναι SS-CCA δηλαδή, το κρυπτοσύστημα πρέπει να είναι σημασιολογικά ασφαλές υπό επίθεση κρυπτογραφημένου κειμένου. Μερικές φορές αυτή η ασφάλεια συμβολίζεται και IND-CCA²². Υπάρχουν περιπτώσεις που έχουμε στην πράξη ένα μαντείο αποκρυπτογράφησης, για παράδειγμα, σε μερικά DRM²³ συστήματα.

Ο ορισμός IND-CCA (ή SS-CCA) δόθηκε πρώτη φορά το 1991 από τους Dolev, Dwork και Naor. Είναι ιδιαίτερα σημαντικός, γιατί αποτελεί τη ζητούμενη ασφάλεια στα κρυπτοσυστήματα (συμμετρικά και δημόσιου κλειδιού).

²²IND-CCA: Indistinguishability of ciphertexts under chosen-ciphertext attacks

²³DRM: Digital Right Management

Βιβλιογραφία

- [1] Αντωνιάδης, Ι.; Κοντογεώργης, Α. *Πεπερασμένα Σώματα και Κρυπτογραφία*, Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. <http://dx.doi.org/10.57713/kallipos-878>, 2015.
- [2] Boneh, Dan; Shoup, V. *A Graduate Course in Applied Cryptography*, Version 0.5, Jan. 2020. <http://toc.cryptobook.us/>.
- [3] Joux, A. *Algorithmic Cryptanalysis*. Chapman & Hall/CRC, 2009.
- [4] Yaschenko, V. V. *Cryptography: An Introduction*. Student Mathematical Library, vol.18. AMS, 2002.
- [5] Μπερτσεκάς, Π. Δ.; Τσιτσικλής, Ν. Γ. *Εισαγωγή στις πιθανότητες*. Εκδ. Τζιόλας, 2010.
- [6] Ζάχος, Ε.; Παγουρτζής, Α.; Γροντάς, Π. *Υπολογιστική Κρυπτογραφία*, Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. <http://dx.doi.org/10.57713/kallipos-492>, 2015.
- [7] Κολουτζάκης, Μ.; Παπαχριστόδουλος, Χ. *Διακριτά μαθηματικά*, Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. <http://dx.doi.org/10.57713/kallipos-517>, 2015.

Κεφάλαιο 3

Κρυπτοσυστήματα Ροής (stream ciphers)

Σύνοψη - Περίληψη. Σε αυτό το κεφάλαιο παρουσιάζουμε τα κρυπτοσυστήματα ροής με έμφαση στους γραμμικούς καταχωρητές καταγραφής με μετατόπιση²⁴, LFSR. Επίσης, παρουσιάζουμε το κρυπτοσύστημα ροής RC4. Στο βιβλίο [2], στην ενότητα 3.8, παρουσιάζεται μια εφαρμογή του συστήματος LFSR στο σύστημα κρυπτογράφησης CSS των DVD και στην ενότητα 3.9 μια εφαρμογή του RC4.

²⁴Επίσης στην ελληνόγλωσση βιβλιογραφία αναφέρονται και ως, γραμμικοί καταχωρητές ολίσθησης με ανάδραση

Προαπαιτούμενη γνώση. Για αυτό το κεφάλαιο χρειάζονται βασικές γνώσεις πιθανοτήτων. Μπορείτε να συμβουλευτείτε για μια εισαγωγή στις διακριτές πιθανότητες το βιβλίο [11, Κεφ. 11] και [8]. Επίσης, για τη μαθηματική ανάπτυξη των LFSR χρειάζονται βασικές γνώσεις γραμμικής άλγεβρας και πεπερασμένων σωμάτων. Μπορείτε να ανατρέξετε στο βιβλίο [5], για τη βασική θεωρία πεπερασμένων σωμάτων και ειδικότερα στο κεφάλαιο 6. Επίσης, μπορείτε να δείτε στην ελληνόγλωσση βιβλιογραφία το [1, Κεφάλαιο 4]. Τα πεπερασμένα σώματα και ειδικότερα οι αλγεβρικές επεκτάσεις αυτών θα χρειαστούν στην ανάπτυξη του κρυπτοσυστήματος AES. Για τα συστήματα LFSR μπορείτε επίσης να δείτε στην ελληνόγλωσση βιβλιογραφία [10, υποενότητα 5.7.3], [9], [7] και στην αγγλική [2] και [3]. Όσον αφορά τη γραμμική άλγεβρα, χρειάζεται η άλγεβρα πινάκων (πολλαπλασιασμός, πρόσθεση πινάκων). Π.χ. δείτε για μια εισαγωγή στη γραμμική άλγεβρα [6, 4].

3.1 Εισαγωγή

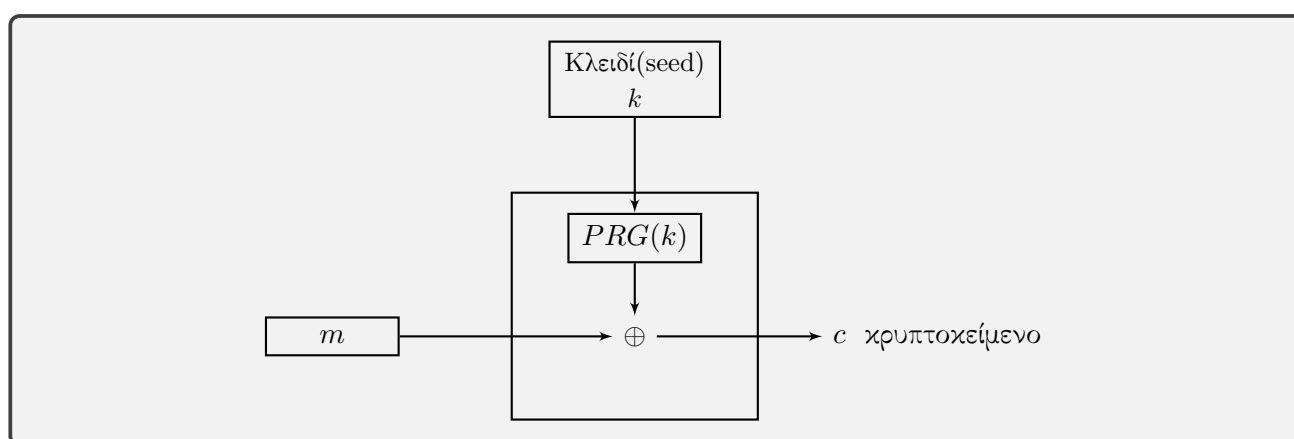
Δίνουμε αρχικά τον ακόλουθο ορισμό.

Ορισμός 3.1.1. Έστω s θετικός ακέραιος. Με $\{0,1\}^*$ συμβολίζουμε το σύνολο όλων των δυαδικών ακολουθιών οσοδήποτε μεγάλου μήκους. Ονομάζουμε ψευδοτυχαία συνάρτηση (ή γεννήτρια ψευδοτυχαίων αριθμών) και συμβολίζουμε PRG : **P**seudo **R**andom **G**enerator, μια συνάρτηση

$$PRG : \{0,1\}^s \rightarrow \{0,1\}^*,$$

όπου η έξοδος της μοιάζει τυχαία.

Θυμίζουμε ότι το βασικό μειονέκτημα του σημειωματάριου μίας φορές (OTP) είναι το μεγάλο μήκος κλειδιού. Αυτό δημιουργεί μεγάλα προβλήματα στον διαμερισμό του κλειδιού μεταξύ των οντοτήτων που θέλουν να επικοινωνήσουν. Ένας τρόπος να υπερβούμε αυτήν την αδυναμία είναι με τη χρήση κρυπτογραφικά ασφαλών PRG. Συστήματα που βασίζονται σε τέτοιες συναρτήσεις ονομάζονται κρυπτοσυστήματα ροής (stream ciphers). Η κρυπτογράφηση φαίνεται στο παρακάτω σχήμα.



Ο πιο γνωστός κρυπταλγόριθμος ροής είναι ο RC4. Σε επικοινωνίες TLS/SSL χρησιμοποιούνταν κατά 50%. Επιθέσεις που βρέθηκαν το 2013 καθιστούν απαγορευτική τη χρήση του RC4 στο SSL/TLS.

Άσκηση 3.1 Έστω ότι έχουμε ένα προβλέψιμο νόμισμα. Π.χ. η πιθανότητα να έρθει γράμμα είναι $1/2 + \delta$ για μη μηδενικό δ με $\delta \in (-1/2, 1/2)$. Μπορείτε να βρείτε ένα πείραμα ώστε να μετατρέψετε το προβλέψιμο νόμισμα σε τυχαίο νόμισμα, δηλαδή η πιθανότητα να έρθει γράμμα ή κορώνα να είναι ίδιες; Ένας άλλος τρόπος να διατυπώσουμε την ίδια ερώτηση είναι να μετατρέψουμε μια ακολουθία Bernoulli με $p \neq 1/2$ σε μια ακολουθία Bernoulli με $p = 1/2$.

Άσκηση 3.2 Έστω ότι έχουμε μια γεννήτρια ψευδοτυχαίων ακεραίων αριθμών, $RAND()$, από το 0 έως το 96. Είναι σωστός ο τρόπος παραγωγής ψευδοτυχαίων ακεραίων αριθμών από το 0 έως το 15, με τον τύπο $RAND()\%16$;

Άσκηση 3.3 Έστω $m = 2^{10}$, $c = 3$ και $a = 5$. Ας είναι $f(x) = (ax + c) \pmod{m}$. Διαλέξτε έναν ακεραίο $x_0 \neq 0$ και υπολογίστε τους 5000 πρώτους ακεραίους της ακολουθίας $x_1 = f(x_0)$ και $x_{i+1} = f(x_i)$, $i > 0$. Εξετάστε τις συχνότητες εμφάνισης της λίστας των αριθμών που βρήκατε και κάντε ένα ιστόγραμμα συχνοτήτων. Πλησιάζει την ομοιόμορφη κατανομή;

Άσκηση 3.4 Περιγράψτε το Kolmogorov-Smirnov Test. Εφαρμόστε το στη λίστα αριθμών που βρήκατε στην Άσκηση 3.3.

3.2 LFSR

Ένας τρόπος να κατασκευάσουμε κρυπτοσυστήματα ροής είναι με χρήση των Γραμμικών Συστημάτων Καταγραφής με Μετατόπιση (LFSR: **L**inear **F**eedback **S**hift **R**egisters). Σήμερα δεν χρησιμοποιούμε (ένα) LFSR ως γεννήτριες ψευδοτυχαίων δυαδικών ψηφίων, διότι έχει αποδειχτεί ότι είναι προβλέψιμες συναρτήσεις (μια έννοια που θα οριστεί παρακάτω). Στην κρυπτογραφία από μία PRG απαιτούμε δύο πράγματα: πρώτον, η έξοδος να είναι στατιστικά τυχαία και να μην είναι προβλέψιμη και δεύτερον, αν γνωρίζω μερικά bits από την έξοδο της PRG, τότε να μην μπορώ να προβλέψω το επόμενο bit με πιθανότητα $> 1/2$. Αυτές οι ψευδογεννήτριες τυχαίων αριθμών ανακαλύφθηκαν το 1950. Μας παρέχουν έναν γρήγορο τρόπο να παράγουμε ψευδοτυχαίες ακολουθίες. Τα LFSR έχουν το πλεονέκτημα ότι υλοποιούνται εύκολα σε επίπεδο Hardware.

Αν θυμηθούμε τον OTP, η συνάρτηση κρυπτογράφησης του δίνεται από τον τύπο

$$E(k, m) = c = k \oplus m.$$

Στα κρυπτοσυστήματα ροής αντί του κλειδιού k χρησιμοποιούμε την εικόνα του seed μέσω της PRG, δηλαδή

$$E(k, m) = c = PRG(seed) \oplus m.$$

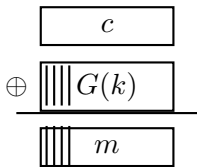
Το σύστημα που προκύπτει με αυτόν τον τρόπο, δεν έχει τέλεια ασφάλεια, διότι το κλειδί (seed) είναι μικρότερου μήκους από το μήκος του μηνύματος και δεν είναι τυχαίο. Η ασφάλεια ενός κρυπτοσυστήματος ροής εξαρτάται αποκλειστικά από τις ιδιότητες της συνάρτησης PRG.

Ορισμός 3.2.1. Μία συνάρτηση G είναι προβλέψιμη, αν και μόνο αν

$$G(k)|_{1,2,\dots,i} \rightarrow G(k)|_{k=i+1,i+2,\dots,n}.$$

Δηλαδή, αν γνωρίζουμε i bits της $G(k)$, μπορούμε να υπολογίσουμε, με έναν αποδοτικό και ντετερμινιστικό (αιτιοκρατικό) αλγόριθμο, τα επόμενα $k = i + 1, \dots, n$ bits της $G(k)$.

Αν χρησιμοποιήσουμε μια προβλέψιμη PRG, τότε το κρυπτοσύστημα ροής δεν είναι ασφαλές. Αυτό φαίνεται στο παρακάτω σχήμα.



Αν γνωρίζουμε μερικά bits, έστω πλήθους r , από την $G(k)$ (γραμμοσχιασμένο κομμάτι), τότε προσθέτουμε (XOR ή πρόσθεση mod 2) με το κρυπτογραφημένο μήνυμα c και βρίσκουμε r bits του μηνύματος m .

Η είσοδος μιας LFSR είναι μια ακολουθία από δυαδικά ψηφία. Η έξοδος είναι μία γραμμική συνάρτηση των προηγούμενων τιμών. Για παράδειγμα, έστω ότι έχουμε ως είσοδο $(1, 1, 0, 1)$. Η είσοδος αυτή ονομάζεται seed του LFSR. Για να οριστεί ένα LFSR χρειαζόμαστε τη γραμμική συνάρτηση. Έστω

$$LFSR(b'_4, b'_3, b'_2, b'_1) = (b_4, b_3, b_2, b_1) = (b'_4 \oplus b'_3, b'_4, b'_3, b'_2).$$

Θέτουμε

$$b'_4 = 1, b'_3 = 1, b'_2 = 0, b'_1 = 1.$$

Η έξοδος είναι τα bits

$$(b_4, b_3, b_2, b_1)$$

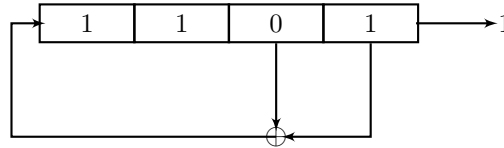
που δίνονται από τους κανόνες μετάβασης

$$b_4 = b'_4 \oplus b'_3, b_3 = b'_4, b_2 = b'_3, b_1 = b'_2.$$

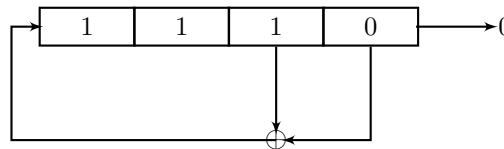
Οπότε

$$LFSR(1, 1, 0, 1) = (0, 1, 1, 0).$$

Η έξοδος του LFSR είναι το $b_4 = 0$. Αν εφαρμόσουμε ξανά την LFSR στην προηγούμενη τιμή, θα πάρουμε τέσσερα νέα bits. Ως κλειδί χρησιμοποιούμε την κλειδοροή που προκύπτει από τα bits που βρίσκονται στη θέση b_1 . Σχηματικά την προηγούμενη διαδικασία μπορούμε να την αναπαραστήσουμε ως εξής:



Στο παραπάνω σχήμα υπάρχουν όλες οι πληροφορίες για το LFSR. Υπάρχει το seed, δηλαδή η αρχική ακολουθία δυαδικών ψηφίων, το κλειδί, και υπάρχει και η συνάρτηση ανάδρασης που συνδέει τα κελιά μεταξύ τους. Από τη στιγμή που θα ξεκινήσει να δουλεύει το LFSR, θα εξάγει δυαδικά ψηφία. Στην αρχή εξάγει το 1, στην επόμενη κατάσταση (state of the LFSR) εξάγει το 0.

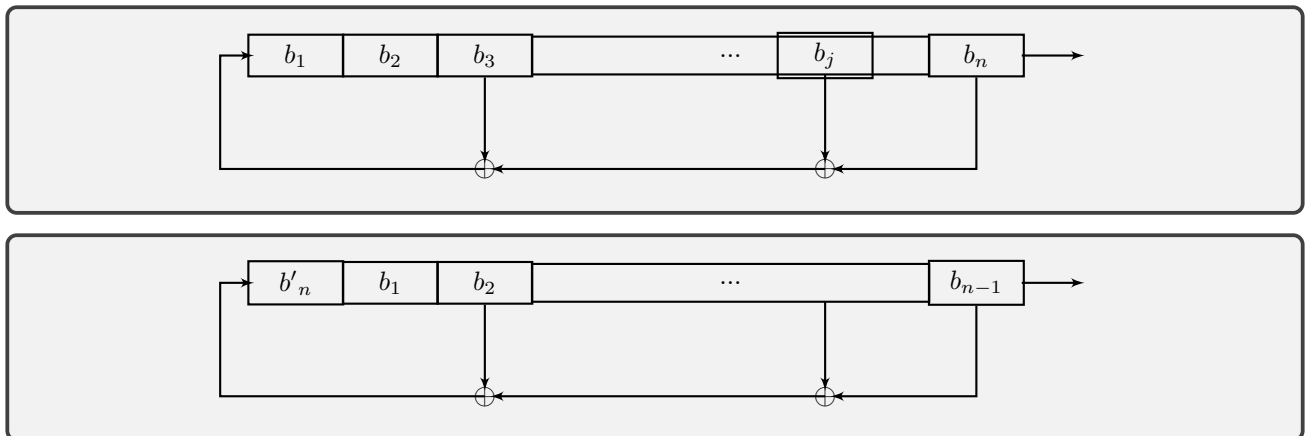


Τελικά, τα αρχικά 20 δυαδικά ψηφία που προκύπτουν είναι τα εξής:

$$1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, \mathbf{1, 1, 0, 1, 0, 1, 1, 1}.$$

Η δέκατη πέμπτη κατάσταση του LFSR ταυτίζεται με την πρώτη κατάσταση. Δηλαδή, το συγκεκριμένο LFSR έχει περίοδο 15. Τα δυαδικά ψηφία με bold είναι το seed. Πολλές φορές τη συνάρτηση ανάδρασης την αναπαριστούμε με ένα πολυώνυμο, που ονομάζεται πολυώνυμο ανάδρασης. Στο συγκεκριμένο LFSR είναι $f(x) = x^4 + x^3 + 1$.

Καθώς υπάρχει μια γραμμική σχέση των δυαδικών ψηφίων εξόδου με τα δυαδικά ψηφία εισόδου, είναι λογικό να περιμένουμε ότι η διαδικασία αυτή μπορεί να περιγραφεί από μια γραμμική απεικόνιση. Ας θεωρήσουμε τα δύο παρακάτω σχήματα όπου φαίνονται δύο διαδοχικές καταστάσεις ενός LFSR μήκους n .



Η έξοδος είναι τα bits $\{b_n, b_{n-1}, \dots, b_1, b'_n, \dots\}$. Ο κανόνας μετάβασης είναι

$$c_{n+1} = c_n \oplus c_j \oplus c_3.$$

Οι διαφορετικές καταστάσεις ενός LFSR μήκους n είναι το πολύ $2^n - 1$ (δεν συμπεριλάβαμε την κατάσταση $(0, 0, \dots, 0)$). Ας είναι $(c_1, c_2, \dots, c_n)$ τα taps, δηλαδή τα κελιά που ορίζουν τον κανόνα

μετάβασης (ή συνάρτηση ανάδρασης). Π.χ. αν $(c_1, c_2, \dots, c_n) = (1, 0, \dots, 0, 1)$, τότε ο κανόνας μετάβασης είναι $c_{n+1} = c_1 \oplus c_n$. Αν θεωρήσουμε τον πίνακα $n \times n$,

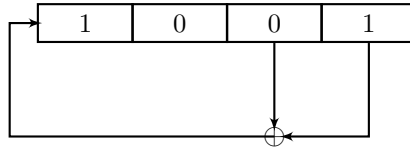
$$C = \begin{pmatrix} c_1 & c_2 & \cdots & c_{n-1} & c_n \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix},$$

τότε η επόμενη κατάσταση προκύπτει από το γινόμενο πινάκων mod 2,

$$C \cdot (s_1, s_2, \dots, s_n)^T,$$

όπου $(s_1, s_2, \dots, s_n)$ είναι το seed (ο υποπίνακας με τα μπλε στοιχεία είναι μοναδιαίος τάξης $(n-1) \times (n-1)$).

Ας θεωρήσουμε την παρακάτω κατάσταση του LFSR μήκους 4:



Ο πίνακας

$$C = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \quad (3.2.1)$$

Η επόμενη κατάσταση προκύπτει από το χοι-γινόμενο των πινάκων

$$C \cdot \mathbf{x}_0 = C \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \mathbf{x}_1 \quad (3.2.2)$$

Για να βρούμε την i -κατάσταση του LFSR υπολογίζουμε $C^i \mathbf{x}_0 = \mathbf{x}_i$. Εύκολα παρατηρούμε ότι, αν γνωρίζουμε μια κατάσταση του LFSR, μπορούμε να βρούμε το seed (αρχική κατάσταση) λύνοντας ένα γραμμικό σύστημα 4×4 (γενικά $n \times n$). Γενικότερα, αν γνωρίζουμε 4 (ή n) συνεχόμενα bits, μπορούμε να βρούμε το αρχικό seed. Η μέθοδος του Gauss θα απαιτούσε $O(n^3)$ αριθμητική πολυπλοκότητα ($O(n^5)$ bit-complexity) για ένα LFSR μήκους n . Αυτή η επίθεση απαιτεί να γνωρίζουμε τη συνάρτηση ανάδρασης.

Ο ελάχιστος φυσικός T για τον οποίο ισχύει $C^T \mathbf{x}_0 = \mathbf{x}_0$ ονομάζεται περίοδος του LFSR. Στο παράδειγμά μας μπορούμε να δούμε ότι $C^{15} \mathbf{x}_0 = \mathbf{x}_0$ και $C^i \mathbf{x}_0 \neq \mathbf{x}_0$ για $1 \leq i \leq 14$. Δηλαδή, $T = 15$. Επομένως, ο LFSR έχει τη μέγιστη δυνατή περίοδο.

Ορισμός 3.2.2. Αν αριθμήσουμε από αριστερά προς τα δεξιά όλα τα κελιά του LFSR ξεκινώντας από το 1, τότε αντιστοιχίζουμε το i -tap με τον όρο x^i . Το πολυώνυμο που προκύπτει, αν αθροίσουμε όλους τους όρους μαζί με το $x^0 = 1$, ονομάζεται πολυώνυμο ανάδρασης.

Για παράδειγμα, το πολυώνυμο ανάδρασης στο προηγούμενο LFSR είναι $x^4 + x^3 + 1$. Μια παρατήρηση που μας βοηθάει να καταλάβουμε τη σύνδεση των πολυωνύμων ανάδρασης με τα LFSR είναι η παρακάτω. Ας θεωρήσουμε τη σειρά $s(x) = \sum_{i \geq 0} x_i x^i$ με συντελεστές στο $\mathbb{F}_2[x]$. Αν αναπτύξουμε τη σειρά, προκύπτει:

$$s(x) = f(x) + x^T f(x) + x^{2T} f(x) + \cdots = f(x)(1 + x^T + x^{2T} + \cdots) =$$

$$\frac{f(x)}{x^T - 1} = \frac{f(x)}{x^T + 1}.$$

Αν ονομάσουμε τις ακολουθίες που προκύπτουν από τα LFSRs, ακολουθίες *LFSR*, τότε έχουμε το παρακάτω.

Λήμμα 3.2.1. Αν $(x_i)_i$ μια ακολουθία *LFSR* με περίοδο T και πολυώνυμο ανάδρασης $f(x)$, τότε

$$f(x)|x^T - 1$$

στο $\mathbb{F}_2[x]$.

Ισχύει και το αντίστροφο του προηγούμενου, δηλαδή, αν $s(x) = \frac{f(x)}{x^T - 1}$ και T είναι ο μικρότερος φυσικός με την ιδιότητα $f(x)|x^T - 1$ στο $\mathbb{F}_2[x]$, τότε οι συντελεστές της σειράς $s(x)$ είναι περιοδικοί, $x_{i+T} = x_i$ για κάθε $i \geq 0$. Επομένως, το T είναι περίοδος της ακολουθίας $(x_i)_i$. Συνεπώς, ισχύει το παρακάτω λήμμα.

Λήμμα 3.2.2. Αν $(x_i)_i$ μια ακολουθία *LFSR* με πολυώνυμο ανάδρασης $f(x)$, τότε ο μικρότερος φυσικός T με την ιδιότητα $f(x)|x^T - 1$ στο $\mathbb{F}_2[x]$ είναι η περίοδος της $(x_i)_i$.

Άσκηση 3.5 Στο αλγεβρικό υπολογιστικό σύστημα `sagemath`²⁵ υπάρχουν αρκετές συναρτήσεις γραμμικής άλγεβρας. Για παράδειγμα, για να ορίσετε έναν πίνακα, δίνετε την παρακάτω εντολή,

```
1 M=matrix([[1,2,3],[4,5,6],[7,8,9]])
2 show(M)
```

Ενώ στο πεπερασμένο σώμα $\mathbb{F}_2$, για να ορίσουμε έναν πίνακα 3×4 ,

```
1 M=matrix(GF(2),3,4,[ [1,0,0,1],[0,1,1,0],[1,1,1,1] ])
2 show(M)
```

Να επαληθεύσετε τον υπολογισμό (3.2.2), αφού αρχικά ορίσετε τον πίνακα (3.2.1). Χρησιμοποιήστε το `sagemath` για τους υπολογισμούς σας.

3.2.1 Επιπρόσθετη θεωρία για τα LFSR

Για την πληρότητα της ανάπτυξης της θεωρίας των LFSR δίνονται οι αποδείξεις μερικών προτάσεων που απαιτούν θεωρία πεπερασμένων σωμάτων. Για ένα εισαγωγικό βιβλίο στα πεπερασμένα σώματα συστήνουμε το [5].

Πρόταση 3.2.1 Αν το πολυώνυμο ανάδρασης $P(x)$ βαθμού n είναι ανάγωγο, τότε διαιρεί στο $\mathbb{F}_2[x]$ το πολυώνυμο $Q(x) = x^{2^n-1} - 1$.

Απόδειξη. Θεωρούμε την επέκταση $L = \mathbb{F}_2[x]/\langle P(x) \rangle$. Το πλήθος των στοιχείων της επέκτασης L είναι $|L| = 2^n$ ²⁶. Επίσης, η επέκταση περιέχει $2^n - 1$ μη μηδενικά στοιχεία. Επομένως, για κάθε $y \in L - \{0\}$, ισχύει $y^{2^n-1} = 1$. Διαιρούμε το πολυώνυμο $Q(x)$ με το $P(x)$ στον δακτύλιο πολυωνύμων $\mathbb{F}_2[x]$. Οπότε, με ευκλείδεια διαίρεση προκύπτει

$$Q(x) = A(x)P(x) + Y(x)$$

με $\deg Y < \deg P = n$. Θα δείξουμε ότι $Y(x) = 0$. Για κάθε ρίζα x_0 του $P(x)$ ισχύει

$$Q(x_0) = x_0^{2^n-1} - 1 = 0.$$

²⁵μπορείτε να το βρείτε και online <https://sagecell.sagemath.org/>

²⁶ $\mathbb{F}_2$ είναι το σώμα με δύο στοιχεία. Με $\langle P(x) \rangle$ εννοούμε το κύριο ιδεώδες του δακτυλίου $\mathbb{F}_2[x]$ που παράγεται από το $P(x)$.

Αυτό συμβαίνει διότι $x_0 \in L - \mathbb{F}_2$. Αν $x_0 = 0$ ή 1 , τότε το $P(x)$ θα ήταν σύνθετο (δηλαδή, θα είχε μη τετριμμένη παραγοντοποίηση στο $\mathbb{F}_2$). Επομένως, αν $\deg Y > 0$, τότε $Y(x_0) = 0$. Αλλά το πολυώνυμο $P(x)$ είναι ανάγωγο και ισχύει $\gcd(P(x), Y(x)) = 1$. Επομένως, υπάρχουν πολυώνυμα $a(x), b(x) \in \mathbb{F}_2[x]$ τέτοια, ώστε

$$a(x)P(x) + b(x)Y(x) = 1.$$

Αν θέσουμε $x = x_0$, τότε προκύπτει $0 = 1$, πράγμα άτοπο. Επομένως, $\deg Y = 0$. Δηλαδή, το $Y(x)$ είναι μια σταθερά του σώματος $\mathbb{F}_2$. Τέλος, επειδή $Y(x_0) = 0$, η σταθερά αυτή είναι μηδέν. Δηλαδή, το $P(x)$ διαιρεί το $Q(x) = x^{2^n-1} - 1$. $\square$

Πόρισμα 3.2.1. Η περίοδος ενός LFSR μήκους n είναι μικρότερη ή ίση από $2^n - 1$.

Προφανώς θα επιθυμούσαμε να κατασκευάσουμε LFSR μέγιστης περιόδου. Η μελέτη αυτού του προβλήματος ανάγεται φυσιολογικά στη μελέτη του πολυωνύμου ανάδρασης. Χρειαζόμαστε αρχικά την έννοια της τάξης ενός πολυωνύμου του $\mathbb{F}_2[x]$.

Λήμμα 3.2.3. Έστω $f \in \mathbb{F}_2[x]$ βαθμού $m \geq 1$ και $f(0) \neq 0$. Τότε υπάρχει ένας φυσικός αριθμός e τέτοιος, ώστε $f(x)|x^e - 1$.

Απόδειξη. Θεωρούμε τον δακτύλιο πηλίκου $L = \mathbb{F}_2[x]/\langle f(x) \rangle$. Το σύνολο L περιέχει $2^m - 1$ μη μηδενικά στοιχεία. Επίσης, το σύνολο

$$\mathcal{M} = \{x^j \pmod{f(x)} : 0 \leq j \leq 2^m - 1\} \subset L.$$

Παρατηρούμε ότι $|\mathcal{M}| = 2^m$ και $0 \notin \mathcal{M}$. Επομένως υπάρχουν i, j με $0 \leq i < j \leq 2^m - 1$, τέτοια, ώστε

$$x^i \equiv x^j \pmod{f(x)}.$$

Άρα, αν θέσουμε $e = j - i$,

$$x^e \equiv 1 \pmod{f(x)},$$

δηλαδή

$$f(x)|x^e - 1.$$

$\square$

Ορισμός 3.2.3. Έστω $f(x), g(x) \in \mathbb{F}_2[x]$ τέτοια, ώστε $f(x) = x^d g(x)$ για κάποιο $d \geq 0$. Το $g(x)$ με αυτήν την ιδιότητα είναι μοναδικό. Ονομάζουμε τάξη του $f(x)$ τον μικρότερο φυσικό αριθμό e για τον οποίο ισχύει $g(x)|x^e - 1$.

Για να κατασκευάσουμε LFSR μήκους n με μέγιστη περίοδο, δηλαδή $T = 2^n - 1$, αρκεί να βρούμε πολυώνυμα ανάδρασης με τάξη $2^n - 1$. Τα πολυώνυμα που έχουν αυτήν την ιδιότητα είναι τα πρωταρχικά πολυώνυμα βαθμού n . Δίνουμε το παρακάτω θεώρημα που τα χαρακτηρίζει.

Θεώρημα 3.2.1 Ένα πολυώνυμο $f(x)$ βαθμού n είναι πρωταρχικό, αν και μόνο αν $f(0) \neq 0$, f είναι κανονικό (δηλαδή έχει μεγιστοβάθμιο συντελεστή τη μονάδα) και έχει τάξη $2^n - 1$.

Ένας άλλος τρόπος να γράψουμε ότι ένα πολυώνυμο $f(x)$ έχει τάξη $2^n - 1$ είναι να μην ανήκει στο σύνολο

$$\{x^d - 1 : d|2^n - 1, 1 < d < 2^n - 1\}.$$

Επομένως έχουμε την παρακάτω μέθοδο για την εύρεση πρωταρχικών πολυωνύμων.

Είσοδος: Ένας φυσικός n

Έξοδος: Όλα τα πρωταρχικά πολυώνυμα βαθμού n

1. Παραγοντοποιούμε το $x^{2^n-1} - 1$ στο $\mathbb{F}_2[x]$. Έστω L_n η λίστα με όλους τους παράγοντες βαθμού n .
2. Παραγοντοποιούμε όλα τα πολυώνυμα $x^d - 1$, $d|2^n - 1$ και $1 < d < 2^n - 1$. Έστω M_n το σύνολο με όλα τα ανάγωγα πολυώνυμα βαθμού n που εμφανίζονται στην ανάλυση των $x^d - 1$.
3. Επιστρέφουμε όλα τα πολυώνυμα $L_n - L_n \cap M_n$, δηλαδή τα πολυώνυμα του συνόλου L_n που δεν ανήκουν στο σύνολο M_n .

Παράδειγμα 3.2.1. Για $n = 4$ εφαρμόζουμε τον προηγούμενο αλγόριθμο.

Βήμα 1:

$$x^{2^n-1} - 1 = x^{15} - 1 = (x+1)(x^2+x+1)(x^4+x+1)(x^4+x^3+1)(x^4+x^3+x^2+x+1). \text{ Άρα}$$

$$L_n = \{x^4 + x + 1, x^4 + x^3 + 1, x^4 + x^3 + x^2 + x + 1\}.$$

Βήμα 2:

$$x^5 - 1 = (x+1)(x^4+x^3+x^2+x+1)$$

$$x^3 + 1 = (x+1)(x^2+x+1). \text{ Άρα}$$

$$M_n = \{x^4 + x^3 + x^2 + x + 1\}.$$

Βήμα 3:

$$L_n - L_n \cap M_n = \{x^4 + x + 1, x^4 + x^3 + 1\}.$$

Παραθέτουμε τον κώδικα που χρησιμοποιήσαμε για την παραγοντοποίηση των πολυωνύμων (ο κώδικας είναι στο sagemath)²⁷.

```
1 x = PolynomialRing(GF(2), 'x').gen()
2 f = factor(x^15 - 1)
3 g = factor(x^3 - 1)
4 h = factor(x^5 - 1)
5 f, g, h
```

Για την παραγοντοποίηση των πολυωνύμων χρησιμοποιείται ο αλγόριθμος του Berlekamp.

Παρατήρηση 3.2.1. Η συνάρτηση του Euler $\phi(n)$ ορίζεται ως εξής: Αν $n = p_1^{a_1} \cdots p_k^{a_k}$ η πρωτογενής ανάλυση του φυσικού αριθμού n σε πρώτους παράγοντες, τότε

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

Μπορεί να αποδειχθεί ότι το πλήθος των πρωταρχικών πολυωνύμων (primitive polynomials) mod 2 βαθμού n είναι

$$\phi(2^n - 1)/n.$$

Για παράδειγμα ο παρακάτω κώδικας στο Sagemath υπολογίζει το πλήθος των πρωταρχικών πολυωνύμων (τρίτη στήλη) για όλα τα n με $4 \leq n \leq 20$.

```
1 def number_of_prim_poly(n):
2     return euler_phi(2^n-1)/n
3 [ [n, 2^n-1, number_of_prim_poly(n)] for n in [4..20]]
4
5 [[4, 15, 2],
6  [5, 31, 6],
7  [6, 63, 6],
8  [7, 127, 18],
9  [8, 255, 16],
10 [9, 511, 48],
11 [10, 1023, 60],
12 [11, 2047, 176],
13 [12, 4095, 144],
14 [13, 8191, 630],
15 [14, 16383, 756],
```

²⁷Μπορείτε να τον χρησιμοποιήσετε στο online <https://sagecell.sagemath.org/>

```

16 [15, 32767, 1800],
17 [16, 65535, 2048],
18 [17, 131071, 7710],
19 [18, 262143, 7776],
20 [19, 524287, 27594],
21 [20, 1048575, 24000]

```

Άσκηση 3.6 Βρείτε τα πρωταρχικά πολυώνυμα βαθμού $n = 5$.

Άσκηση 3.7 Δίνεται το πολυώνυμο

$$f(x) = x^4 + x^3 + x + 1$$

και το χρησιμοποιούμε ως πολυώνυμο ανάδρασης. Αν η είσοδος στο LFSR με πολυώνυμο ανάδρασης $f(x)$ είναι x_0, x_1, x_2, x_3 και η έξοδος, τα $y_i, i = 0, 1, 2, \dots$ βρείτε τα y_i συναρτήσει των x_0, x_1, x_2, x_3 για $i = 0, 1, 2, 3, 4, 5, 6, 7$.

Άσκηση 3.8 Αρχικά, εξοικειωθείτε με τον κώδικα για την κατασκευή των LFSR που βρίσκεται:

https://github.com/drazioti/python_scripts/blob/master/numtheory/lfsr_project.py

Θεωρούμε το LFSR με πολυώνυμο ανάδρασης $x^{10} + x^9 + x^7 + x^6 + 1$. Επίσης, δίνεται η κρυπτογράφηση του μηνύματος ab , $Enc(ab) = sq$ (δηλαδή, η εφαρμογή του LFSR με είσοδο ab , όταν μετατραπούν σε bits, σύμφωνα με τον παρακάτω πίνακα, δίνουν ως έξοδο sq). Ένα μήνυμα, που επίσης έχει κωδικοποιηθεί σύμφωνα με τον παρακάτω πίνακα, έχει δώσει την κρυπτογράφηση

i!))aiszwykqnfyc!?!secnncvch

Βρείτε το μήνυμα.

(0) A	:	00000	(13) N	:	01101
(1) B	:	00001	(14) O	:	01110
(2) C	:	00010	(15) P	:	01111
(3) D	:	00011	(16) Q	:	10000
(4) E	:	00100	(17) R	:	10001
(5) F	:	00101	(18) S	:	10010
(6) G	:	00110	(19) T	:	10011
(7) H	:	00111	(20) U	:	10100
(8) I	:	01000	(21) V	:	10101
(9) J	:	01001	(22) W	:	10110
(10) K	:	01010	(23) X	:	10111
(11) L	:	01011	(24) Y	:	11000
(12) M	:	01100	(25) Z	:	11001
(26) .	:	11010	(27) !	:	11011
(28) ?	:	11100	(29) (	:	11101
(30))	:	11110	(31) -	:	11111

Άσκηση 3.9 Αποδεικνύεται ότι το πλήθος των ανάγωγων πολυωνύμων βαθμού n στο σώμα $\mathbb{F}_2$ είναι

$$N_2(n) = \frac{1}{n} \sum_{d|n} \mu(d) 2^{n/d},$$

όπου

$$\mu(d) = \begin{cases} 1, & d = 1 \\ (-1)^k, & d = p_1 p_2 \cdots p_k \text{ (} p_i : \text{πρώτοι)} \\ 0, & \text{άλλοι} \end{cases}$$

Με το σύμβολο $d|n$ εννοούμε όλους τους θετικούς διαιρέτες του n . Π.χ. αν $n = 30$, τότε

$$\{d|n : 1 \leq d \leq n\} = \{1, 2, 3, 5, 6, 10, 15, 30\}.$$

Με χρήση του συστήματος sagemath²⁸ υπολογίστε το $N_2(10)$.

Άσκηση 3.10 Να αποδείξετε ότι

$$\sum_{d|n} \mu(d) = \begin{cases} 1, & n = 1 \\ 0, & n > 1 \end{cases},$$

όπου $\mu(d)$ η συνάρτηση που ορίστηκε στην άσκηση 3.9.

3.3 RC4 - Ron's Code

Ο Ron Rivest κατασκεύασε τον RC4 το 1987, τον οποίο κράτησε και μυστικό, έως το 1994: τότε με κάποιο τρόπο διέρρευσε στο κοινό. Μια επίθεση που βρέθηκε το 2013 (βασίζεται σε στατιστική ανωμαλία της εξόδου του RC4) δεν επιτρέπει πλέον να χρησιμοποιείται, τουλάχιστον στο πρωτόκολλο SSL. Το σύστημα αυτό χρησιμοποιήθηκε ευρέως στον SSL. Σχεδόν το 50% της κυκλοφορίας SSL γινόταν με χρήση του RC4 (πριν το 2013). Επίσης, χρησιμοποιήθηκε στο σύστημα WEP²⁹ και γενικά ήταν ένα πετυχημένο συμμετρικό κρυπτοσύστημα.

Αλγόριθμος 3.3.1. : Κατασκευή μετάθεσης S

Είσοδος. seed (40 με 256 bits)

Έξοδος. αρχική τιμή της μετάθεσης S

```

1 seedlen = length(seed)
2 for i = 0 to 255 do
3   | S[i] ← i
   end
4 j = 0
5 for i = 0 to 255 do
6   | j ← (j + S[i] + seed[i mod seedlen]) mod 256
7   | S[i] ↔ S[j]
   end
```

Στο επόμενο βήμα θα χρησιμοποιήσουμε τη μετάθεση S , για να εξαγάγουμε τα bytes του κλειδιού K . Η μετάθεση S έχει μήκος 256 bytes. Στον επόμενο αλγόριθμο δεν θα ξαναχρησιμοποιηθεί το κλειδί seed.

²⁸Για ευκολία μπορείτε να χρησιμοποιήσετε το sagecell <https://sagecell.sagemath.org/>

²⁹Wired Equivalent Privacy

Αλγόριθμος 3.3.2. : Κλειδοροή του RC4

Είσοδος. S **Έξοδος.** το κλειδί K

```
1  $i = 0, j = 0$ 
2  $plen = \text{length}(\text{plaintext})$  // το μήκος του κλειδιού θα είναι ίσο με το μήκος του μηνύματος  $plen$ 
3 while  $plen > 0$  do
4    $i \leftarrow (i + 1) \bmod 256$ 
5    $j \leftarrow (j + S[i]) \bmod 256$ 
6    $S[i] \leftrightarrow S[j]$ 
7    $K \leftarrow S[(S[i] + S[j]) \bmod 256]$ 
8   return  $K$ 
9    $plen \leftarrow plen - 1$ 
end
```

Σε κάθε κύκλο εξάγεται και 1 byte του κλειδιού, που γίνεται xor με 1 byte του μηνύματος, έως ότου τελειώσουν τα bytes του μηνύματος. Παρατηρήστε ότι η μετάθεση S αλλάζει με τον χρόνο.

Άσκηση 3.11 Υλοποιήστε τον RC4. Χρησιμοποιώντας το κλειδί HOUSE κρυπτογραφήστε το μήνυμα (ξεαναγράψτε το κείμενο χωρίς κενά):

MISTAKES ARE AS SERIOUS AS THE RESULTS THEY CAUSE

Η υλοποίησή σας πρέπει και να αποκρυπτογραφεί σωστά. Χρησιμοποιήστε τον πίνακα της άσκησης 3.8.

Βιβλιογραφία

- [1] Αντωνιάδης, Ι.; Κοντογεώργης, Α. *Πεπερασμένα Σώματα και Κρυπτογραφία*, Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. <http://dx.doi.org/10.57713/kallipos-878>, 2015.
- [2] Boneh, Dan; Shoup, V. *A Graduate Course in Applied Cryptography*, Version 0.5, Jan. 2020. <http://toc.cryptobook.us/>.
- [3] Cozzens, M.; Miller, S. J. *The Mathematics of Encryption-An Elementary Introduction*. Mathematical World, 1st edition, vol. 29, 2013. AMS.
- [4] Halmos, P. *Finite-Dimensional Vector Spaces*. Springer, 1993.
- [5] Mullen, G. L.; Mummert, C. *Finite fields and Applications*. AMS, 2007.
- [6] Strang, G. *Introduction to Linear Algebra*. Wellesley-Cambridge Press, 2016.
- [7] Κάτος, Β.; Στεφανίδης, Γ. *Τεχνικές Κρυπτογραφίας και Κρυπτανάλυσης*, Εκδ. Ζυγός, 2003.
- [8] Μπερτσεκάς, Π. Δ.; Τσιτσικλής, Ν. Γ. *Εισαγωγή στις πιθανότητες*. Εκδ. Τζιόλας, 2010.
- [9] Πουλάκης, Δ. *Κρυπτογραφία*. Εκδ. Ζήτη, 2009.
- [10] Ζάχος, Ε.; Παγουρτζής, Α.; Γροντάς, Π. *Υπολογιστική Κρυπτογραφία*, Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. <http://dx.doi.org/10.57713/kallipos-492>, 2015.
- [11] Κολουντζάκης, Μ.; Παπαχριστόδουλος, Χ. *Διακριτά μαθηματικά*, Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. <http://dx.doi.org/10.57713/kallipos-517>, 2015.

Κεφάλαιο 4

Κρυπτοσυστήματα τμήματος (block ciphers)

Σύνοψη - Περίληψη. Σε αυτό το κεφάλαιο παρουσιάζονται τα κρυπτοσυστήματα τμήματος (block ciphers). Αυτά τα συστήματα χρησιμοποιούνται για την κρυπτογράφηση ενός μηνύματος στο διαδίκτυο. Υπάρχουν δύο βασικές αρχές σχεδιασμού, το σχήμα του Feistel που χρησιμοποιήθηκε στη σχεδίαση του κρυπτοσυστήματος DES: Data Encryption Standard και το σχήμα SPN: Substitution and Permutation Network που χρησιμοποιήθηκε στη σχεδίαση του AES: Advanced Encryption Standard. Επίσης, θα παραθέσουμε κάποια βασικά συστήματα καταστάσεων λειτουργίας (modes of operation).

Προαπαιτούμενη γνώση. Για αυτό το κεφάλαιο χρειάζονται μερικές γνώσεις για τις συναρτήσεις Bool (δείτε την αναφορά [6]). Το βιβλίο των Dan Boneh και Victor Shoup [1] είναι πολύ αναλυτικό, όσον αφορά τους αλγορίθμους τμήματος. Επίσης, θα χρειαστούν κάποια βασικά στοιχεία αλγεβρικών δομών στην παρουσίαση του AES. Ειδικότερα, χρειάζονται κάποιες πολύ βασικές γνώσεις για πεπερασμένα σώματα, καθώς και επεκτάσεις αυτών, π.χ. δείτε [5].

4.1 Εισαγωγή

Ξεκινάμε με τον ακόλουθο ορισμό.

Ορισμός 4.1.1. Ονομάζουμε κρυπτοσύστημα τμήματος έναν ντετερμινιστικό αλγόριθμο που λειτουργεί επί μηνυμάτων συγκεκριμένου μήκους (blocks).

Αποτελείται από μια τριάδα αλγορίθμων (K, E, D) όπου K είναι ο αλγόριθμος παραγωγής κλειδιών, E ο αλγόριθμος κρυπτογράφησης και D ο αλγόριθμος αποκρυπτογράφησης. Έστω $\mathcal{M}, \mathcal{C}, \mathcal{K}$ οι χώροι μηνυμάτων, κρυπτογραφημένων μηνυμάτων και κλειδιών, αντίστοιχα. Για τις ανάγκες αυτού του κεφαλαίου θέτουμε

$$\mathcal{M} = \mathcal{C} = \{0, 1\}^n, \mathcal{K} = \{0, 1\}^k$$

για κάποιο φυσικό n και k . Επομένως, οι συναρτήσεις κρυπτογράφησης και αποκρυπτογράφησης ορίζονται ως εξής:

$$E : \{0, 1\}^k \times \{0, 1\}^n \rightarrow \{0, 1\}^n, \quad D : \{0, 1\}^k \times \{0, 1\}^n \rightarrow \{0, 1\}^n$$

τέτοιες, ώστε

$$D(k, E(k, m)) = m$$

για κάθε $m \in \mathcal{M}$ και $k \in \mathcal{K}$. Εδώ το n είναι το μήκος του τμήματος (length of the block). Επίσης, τα στοιχεία του συνόλου $\mathcal{M}$ καλούνται *τμήματα* (blocks). Να παρατηρήσουμε ότι, αν σταθεροποιήσουμε σε κάποια συγκεκριμένη τιμή το κλειδί k , τότε η συνάρτηση

$$E(k, m) = E_k(m) : \mathcal{M} = \{0, 1\}^n \rightarrow \mathcal{C} = \{0, 1\}^n$$

είναι 1-1. Επιπλέον, επειδή το σύνολο $\mathcal{M}$ είναι πεπερασμένο, είναι και επί, άρα η συνάρτηση $E_k(x)$ είναι μια μετάθεση (permutation). Η αντίστροφη συνάρτηση είναι $E_k^{-1}(x) = D_k(x)$.

4.1.1 Ασφάλεια ενός κρυπτοσυστήματος τμήματος

Τώρα που ορίσαμε τι είναι ένα κρυπτοσύστημα τμήματος, μπορούμε να ορίσουμε και τι σημαίνει ασφαλές κρυπτοσύστημα τμήματος.

Ορισμός 4.1.2. Ένα ασφαλές κρυπτοσύστημα τμήματος είναι υπολογιστικά μη ευδιάκριτο (computationally indistinguishable) από μία τυχαία μετάθεση με πεδίο ορισμού το $\mathcal{M}$.

Με απλά λόγια, θα μπορούσαμε να περιγράψουμε τον προηγούμενο ορισμό ως εξής (ονομάζουμε αυτόν τον νέο ισοδύναμο ορισμό black-box-security-definition). Έχει δοθεί στην Εύα ένα μαύρο κουτί το οποίο περιέχει μια μετάθεση P με πεδίο ορισμού το σύνολο $\{0, 1\}^n$. Αυτή η μετάθεση έχει παραχθεί με τυχαίο τρόπο με μία από τις δύο παρακάτω διαδικασίες:

- $P(x) = E_k(x)$ για $k \xleftarrow{\$} \mathcal{K}$ ή
- P μια τυχαία μετάθεση από το σύνολο όλων των μεταθέσεων με πεδίο ορισμού το $\mathcal{M} = \{0, 1\}^n$.

Η Εύα δεν μπορεί να συμπεράνει με σιγουριά πώς έχει παραχθεί η P , ακόμη και αν της δώσουμε τη δυνατότητα να υπολογίσει κάποιες τιμές $y_i = P(x_i)$, για x_i που διαλέγει η ίδια.

Άσκηση 4.1 Στον AES-128 το κλειδί έχει μήκος 128 bits και τα μηνύματα επίσης 128 bits, δηλαδή $\mathcal{M} = \mathcal{C} = \mathcal{K} = \{0, 1\}^{128}$. Σύμφωνα με τον ορισμό ασφάλειας (black-box-security-definition) η μετάθεση στο μαύρο κουτί έχει παραχθεί είτε:

δ₁. ως E_k για τυχαίο κλειδί k , είτε ως:

δ₂. μια τυχαία μετάθεση με πεδίο ορισμού το $\mathcal{M}$.

i. Πόσα στοιχεία περιέχει το σύνολο, που παράγει στοιχεία με τον τρόπο δ₁, και πόσα το σύνολο που δημιουργείται από τη διαδικασία δ₂; Υπολογίστε, κατά προσέγγιση, το πλήθος των στοιχείων σε bits.

ii. Να απαντήσετε στο ίδιο ερώτημα, αλλά για τον AES-192, που ισχύει $\mathcal{M} = \mathcal{C} = \{0, 1\}^{128}$ και $\mathcal{K} = \{0, 1\}^{192}$.

4.2 DES

Το συμμετρικό σύστημα DES: Data Encryption Standard βασίζεται στο σχήμα Feistel, το οποίο και περιγράφουμε στο σχήμα 4.1. Ένα βασικό στοιχείο του DES, αλλά και γενικά των συμμετρικών συστημάτων, είναι τα S-boxes. Αυτά προσφέρουν μη γραμμικότητα στον αλγόριθμό μας. Έστω ένα μήνυμα m μήκους $2d$ -bits (στο σχήμα $d = 32$). Ας είναι $F : K \times \{0, 1\}^d \rightarrow \{0, 1\}^d$ μία συνάρτηση, την οποία θα ορίσουμε αργότερα, που δέχεται τιμές από τον χώρο κλειδιών, και μία λέξη d -bits και πεδίο τιμών οι λέξεις μήκους d -bits. Έστω $K = \{K_1, \dots, K_n\}$ και K_i είναι μήκους κ -bits. Το αρχικό μήνυμα, αφού εφαρμοστεί σε αυτό μία μετάθεση, χωρίζεται σε δύο ισομήκη τμήματα L_1, R_1 . Το σύστημα αυτό εκτελεί n -φορές την επόμενη διαδικασία:

$$L_{i+1} \leftarrow R_i, R_{i+1} \leftarrow F(K_i, R_i) \oplus L_i, i = 1, 2, \dots, n. \quad (4.2.1)$$

Το αποτέλεσμα που θα προκύψει, αφού υποστεί ακόμη μία μεταβολή από την αντίστροφη της αρχικής μετάθεσης, είναι η κρυπτογράφηση του αρχικού μηνύματος m . Επομένως, για να οριστεί ένα σχήμα Feistel, χρειαζόμαστε μια εξάδα

$$(d, n, \kappa, K, F, IP).$$

Για το DES έχουμε ορίσει $d = 32, n = 16, \kappa = 48$. Το $2d$ είναι ένα χαρακτηριστικό των κρυπτοσυστημάτων τύπου Feistel και μας καθορίζει το μήκος του τμήματος. Άλλα συστήματα που χρησιμοποιούν το σχήμα Feistel είναι οι IDEA, RC5, αλλά όχι ο AES. Για να είναι πλήρης η παρουσίαση του DES, χρειαζόμαστε τον αλγόριθμο που παράγει το σύνολο K από ένα αρχικό κλειδί K_0 , την περιγραφή της συνάρτησης F , καθώς και τη μετάθεση IP που δρα πάνω στο αρχικό μήνυμα m . Το μήκος του κλειδιού K_0 για το DES είναι 64-bits και από αυτά χρησιμοποιούνται μόνο 56. Επίσης, μπορούμε να δούμε ότι η προηγούμενη διαδικασία εύκολα αντιστρέφεται (αποκρυπτογράφηση, δείτε το σχήμα 4.2), αρκεί να εφαρμόσουμε τους κανόνες

$$R_i \leftarrow L_{i+1}, L_i \leftarrow F(K_i, L_{i+1}) \oplus R_{i+1}, i = n, n-1, \dots, 1. \quad (4.2.2)$$

Πράγματι, $F(K_i, L_{i+1}) \oplus R_{i+1} = F(K_i, R_i) \oplus R_{i+1}$ (από την πρώτη σχέση του κανόνα (4.2.1), $L_{i+1} = R_i$). Αλλά $F(K_i, R_i) \oplus R_{i+1} = L_i$ (από τη δεύτερη σχέση του κανόνα (4.2.1), xoring και τα δύο μέλη με το $F(K_i, R_i)$). Άρα, $F(K_i, L_{i+1}) \oplus R_{i+1} = L_i$.

4.2.1 Η συνάρτηση Feistel F

Στο σχήμα 4.3 δίνεται η συνάρτηση F που υλοποιείται με τα S -κιβώτια (Substitution Box). Η ασφάλεια του DES βασίζεται στα S -κιβώτια. Η συνάρτηση $E : \{0, 1\}^{32} \rightarrow \{0, 1\}^{48}$ και η έξοδος της γίνονται xor με το i -οστό session key K_i . Μετά το xor τα bit που προκύπτουν γίνονται 8 εξάδες και κάθε εξάδα εισέρχεται σε ένα S -κιβώτιο. Το S -κιβώτιο εξάγει 4-bits, οπότε συνολικά $4 \times 8 = 32$ bits που εισέρχονται σε μια μετάθεση P και τελικά η έξοδος της P είναι η έξοδος της συνάρτησης F .

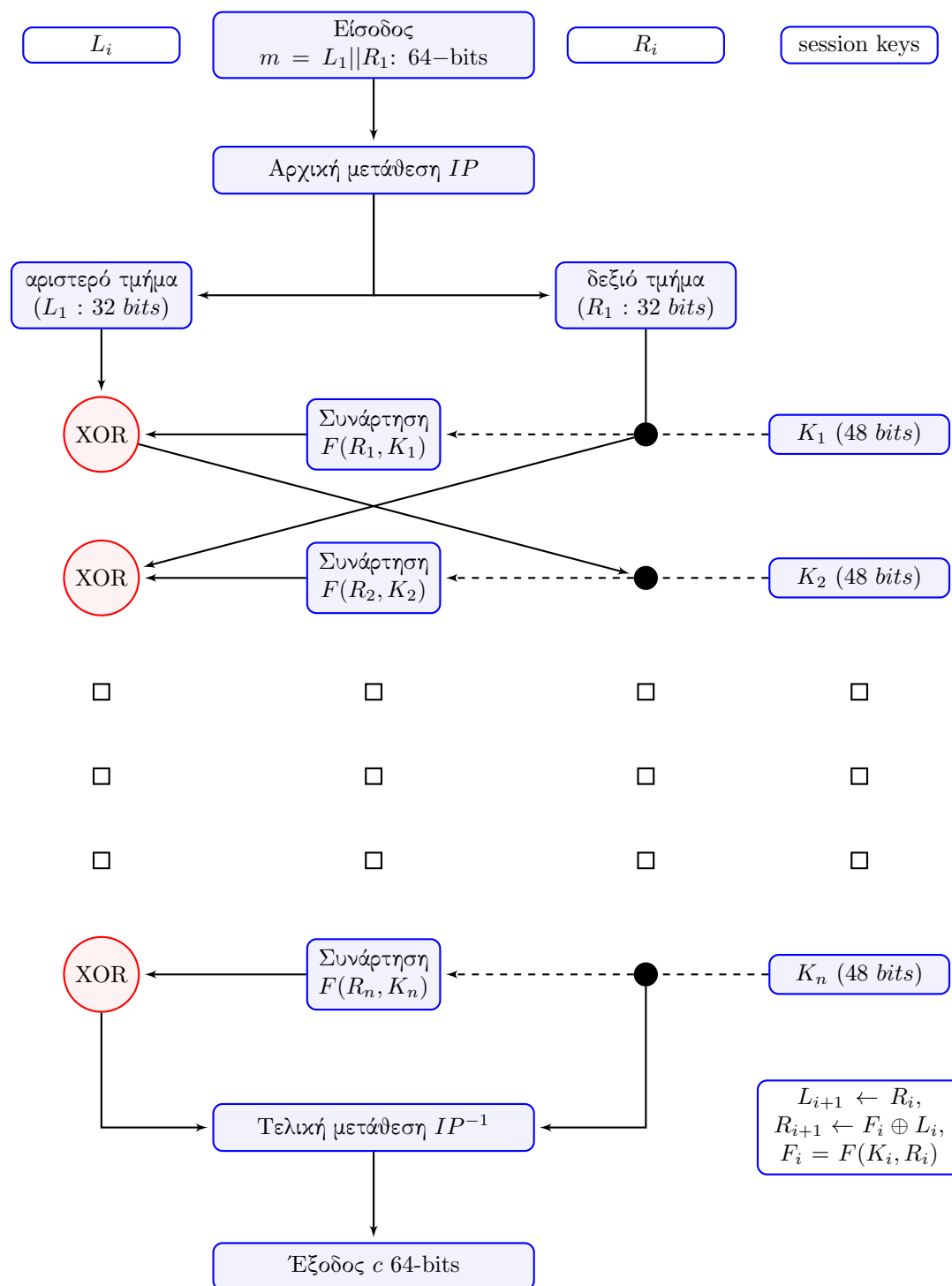
Τα S -κιβώτια είναι 4×16 πίνακες που έχουν ως στοιχεία τους θετικούς ακέραιους αριθμούς μικρότερους των 16-bits, δηλαδή αριθμούς ανάμεσα στο 0 και το 15. Για παράδειγμα, ένα S -κιβώτιο είναι της μορφής

$$S_i = \begin{bmatrix} 0 & 2 & 3 & 7 & 9 & 12 & 15 & 7 & 6 & 15 & 15 & 17 & 3 & 1 & 0 \\ 1 & 5 & 6 & 13 & 4 & 1 & 5 & 11 & 7 & 8 & 7 & 1 & 13 & 2 & 13 \\ 5 & 3 & 5 & 12 & 11 & 1 & 1 & 5 & 13 & 0 & 15 & 7 & 2 & 2 & 13 & 0 \\ 3 & 12 & 3 & 11 & 2 & 2 & 2 & 4 & 6 & 5 & 5 & 0 & 4 & 3 & 1 & 0 \end{bmatrix} \quad (4.2.3)$$

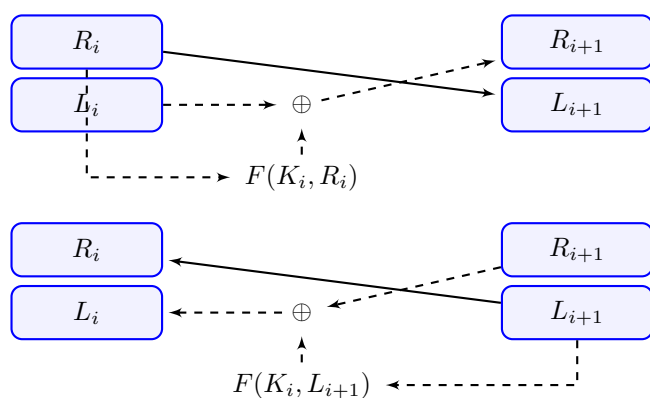
Έστω ότι δέχεται ως είσοδο την εξάδα 110101.

1. Υπολογίζουμε σε ποιον δεκαδικό αντιστοιχούν τα δύο ακραία bits, $(11)_2 = (3)_{10}$.
2. Υπολογίζουμε σε ποιον δεκαδικό αριθμό αντιστοιχούν τα υπόλοιπα bits, $(1010)_2 = (10)_{10}$.
3. Αριθμούμε στον προηγούμενο πίνακα τις γραμμές και τις στήλες ξεκινώντας από το 0.
4. Βρίσκουμε το στοιχείο στη θέση $(3, 10)$ που είναι το 5.
5. Γράφουμε σε δυαδική μορφή το 5, άρα (0101).

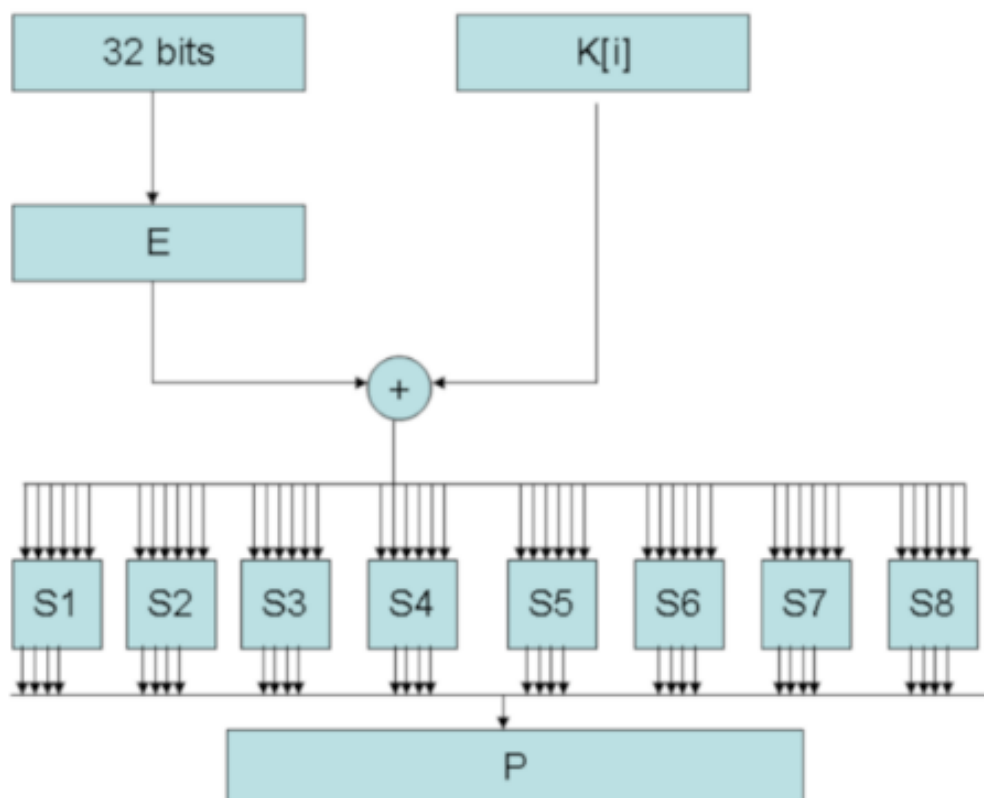
Τότε $S_i(110101) = (0101)$.



Σχήμα 4.1: Σχήμα Feistel.



Σχήμα 4.2: ο i -οστός γύρος της F κατά την κρυπτογράφηση και αποκρυπτογράφηση (κάτω σχήμα).



Σχήμα 4.3: Στιγμιότυπο της συνάρτησης $F : K \times \{0, 1\}^{32} \rightarrow \{0, 1\}^{32}$.

Άσκηση 4.2 Κάνοντας χρήση του S-box (4.2.3), υπολογίστε τα $S(111111), S(000000)$. Κατόπιν, βρείτε, εφόσον υπάρχουν, δύο εξάδες $\mathbf{x}, \mathbf{y} \in \{0, 1\}^6$ τέτοιες, ώστε $S(\mathbf{x}) = S(\mathbf{y})$.

Άσκηση 4.3 Αν Σ ένα σύνολο με $|\Sigma|$ συμβολίζουμε το πλήθος των στοιχείων του, υπολογίστε τη διαφορική ομοιομορφία (differential uniformity³⁰) του S-box (4.2.3),

$$\text{Diff}(S) = \max_{\mathbf{x} \in \{0,1\}^6 - \{\mathbf{0}\}, \mathbf{y} \in \{0,1\}^4} \left| \left\{ \mathbf{z} \in \{0,1\}^6 : S(\mathbf{z} \oplus \mathbf{x}) \oplus S(\mathbf{z}) = \mathbf{y} \right\} \right|$$

Γενικά, για S-boxes:

$$S : \{0, 1\}^n \rightarrow \{0, 1\}^m$$

ο προηγούμενος ορισμός γράφεται:

$$\text{Diff}(S) = \max_{\mathbf{x} \in \{0,1\}^n - \{\mathbf{0}\}, \mathbf{y} \in \{0,1\}^m} \left| \left\{ \mathbf{z} \in \{0,1\}^n : S(\mathbf{z} \oplus \mathbf{x}) \oplus S(\mathbf{z}) = \mathbf{y} \right\} \right|$$

και ισχύει:

$$\text{Diff}(S) \geq \max\{2, 2^{n-m}\}.$$

Όσο μικρότερη είναι αυτή η ποσότητα, τόσο πιο ανθεκτικό είναι το S-Box στη διαφορική κρυπτανάλυση.

Άσκηση 4.4 Έστω, $S : \{0, 1\}^m \rightarrow \{0, 1\}^m$ ένα S-κιβώτιο. Για κάθε $\alpha, \beta \in \mathbb{F}_2^m$ ορίζουμε τον Correlation Coefficient του S-κιβωτίου S στο (α, β) :

$$C_S(\alpha, \beta) = \sum_{\mathbf{x} \in \mathbb{F}_2^m} (-1)^{\beta \cdot S(\mathbf{x}) + \alpha \cdot \mathbf{x}}.$$

Ορίζουμε επίσης τον Linear Branch Number του S-κιβωτίου S την ποσότητα:

$$LBN(S) = \min_{\alpha, \beta \in \mathbb{F}_2^m - \{\mathbf{0}\}, C_S(\alpha, \beta) \neq 0} \{wt(\alpha) + wt(\beta)\}.$$

Όπου $wt(\alpha)$ το Hamming weight της δυαδικής λέξης α .

Θεωρούμε το S-κιβώτιο $S : \{0, 1\}^5 \rightarrow \{0, 1\}^5$ που παράγεται από τον τύπο $S(x) = (x^2 + 3) \bmod 32$, όπου x είναι η δεκαδική παράσταση του δυαδικού. Π.χ.

$$S(01000) = S(8) = (8^2 + 3) \bmod 32 = 3 = (00011).$$

Να υπολογιστεί ο Linear Branch Number του προηγούμενου S-κιβωτίου. Κατόπιν, βρείτε ένα S-κιβώτιο με $LBN(S) > 4$ χρησιμοποιώντας κάποια παρόμοια συνάρτηση.

Υψηλή τιμή του LBN δείχνει υψηλή σύγχυση (confusion) του S-κιβωτίου. Η μεγαλύτερη τιμή που μπορεί να έχει είναι $m + 1$.

4.2.2 Παραγωγή υποκλειδιών K_i

- (Parity Check) Αρχικά, έχουμε ένα κλειδί 64 bits. Αφαιρούμε τα bits που βρίσκονται στις θέσεις 8, 16, 24, 32, 40, 48, 56, 64 (τα bits σε αυτές τις θέσεις ονομάζονται parity bits). Καθένα από αυτά πρέπει να ισούται με το xor των επτά προηγούμενων. Επομένως, (πρακτικά) μένουμε με ένα κλειδί 56- bits. Η χρησιμότητα αυτού του βήματος, που μειώνει το κλειδί κατά 8-bits, αμφισβητήθηκε από πολλούς ειδικούς.
- Θεωρούμε τον παρακάτω πίνακα 8×7 με στοιχεία από το σύνολο $\{1, \dots, 64\} - \{\text{parity bits}\}$. Έστω

$$PC1 = \begin{bmatrix} 57 & 49 & 41 & 33 & 25 & 17 & 9 \\ 1 & 58 & 50 & 42 & 34 & 26 & 18 \\ 10 & 2 & 59 & 51 & 43 & 35 & 27 \\ 19 & 11 & 3 & 60 & 52 & 44 & 36 \\ 63 & 55 & 47 & 39 & 31 & 23 & 15 \\ 7 & 62 & 54 & 46 & 38 & 30 & 22 \\ 14 & 6 & 61 & 53 & 45 & 37 & 29 \\ 21 & 13 & 5 & 28 & 20 & 12 & 4 \end{bmatrix}$$

³⁰K. Nyberg, S-boxes and round functions with controllable linearity and differential uniformity

Επομένως, το αρχικό κλειδί μας, των 64-bits, αλλάζει ως εξής:

Το bit στη θέση 57 γίνεται πρώτο, το bit στη θέση 49 γίνεται δεύτερο, κ.ο.κ.

3. Το κλειδί που θα προκύψει το διαιρούμε σε δύο ισομεγέθη τμήματα, έστω $C[0], D[0]$.

4. Υπολογίζουμε τα 16 υποκλειδιά ως εξής. Ξεκινάμε με $i = 1$.

4.1 Κάνουμε μία ή δύο κυκλικές κυλίσσεις προς τα αριστερά των $C[i - 1], D[i - 1]$ σύμφωνα με τον πίνακα

$$\left(\begin{array}{l} \text{Iteration: } 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15 \ 16 \\ \text{Left shift: } 1 \ 1 \ 2 \ 2 \ 2 \ 2 \ 2 \ 2 \ 1 \ 2 \ 2 \ 2 \ 2 \ 2 \ 2 \ 1 \end{array} \right)$$

Η έξοδος είναι $C[i], D[i]$. Δηλαδή, αν $i = 1$, θα κάνουμε μία κυκλική κύλιση προς τα αριστερά των τμημάτων $C[0], D[0]$.

4.2 Σχημάτισε τη λέξη $C[i] \parallel D[i]$ ($\parallel$: ένωση των δύο λέξεων) και εφάρμοσε την παρακάτω μετάθεση που ορίζεται από έναν πίνακα 8×6 με στοιχεία από το σύνολο $\{1, \dots, 56\}$. Θα μας δώσει μια έξοδο 48-bits.

$$PC2 = \begin{bmatrix} 14 & 17 & 11 & 24 & 1 & 5 \\ 3 & 28 & 15 & 6 & 21 & 10 \\ 23 & 19 & 12 & 4 & 26 & 8 \\ 16 & 7 & 27 & 20 & 13 & 2 \\ 41 & 52 & 31 & 37 & 47 & 55 \\ 30 & 40 & 51 & 45 & 33 & 48 \\ 44 & 49 & 39 & 56 & 34 & 53 \\ 46 & 42 & 50 & 36 & 29 & 32 \end{bmatrix}$$

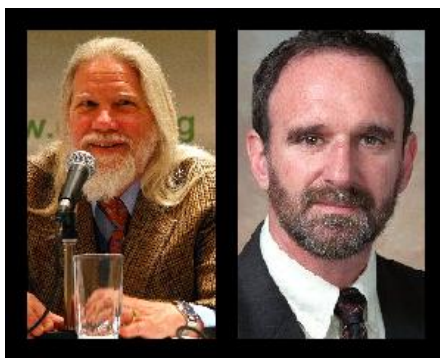
Η έξοδος είναι το υποκλειδί $K[i]$

5. $i \leftarrow i + 1$ και επανάλαβε από το βήμα 4. Συνέχισε έως ότου το i γίνει ίσο με 16.

Για μια αρκετά αναλυτική παρουσίαση (με παρουσίαση παραδείγματος) του DES μπορείτε να δείτε το [7]. Επίσης, μπορείτε να δείτε την επίθεση της διαφορικής κρυπτανάλυσης (στην ελληνόγλωσση βιβλιογραφία) [8].

4.2.3 Σχόλια για το σύστημα DES

Πρώτοι οι ερευνητές του Stanford, Whit Diffie και Martin Hellman, έθεσαν τον προβληματισμό τους για τη χρήση τόσο μικρού κλειδιού στον DES. Δείτε και το βιβλίο [2].



Σχήμα 4.4: Whit Diffie and Martin Hellman. Turing Medal 2016.

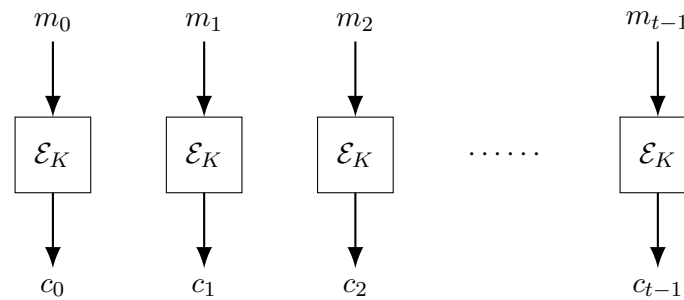
CC BY-SA 2.0/CC BY-SA 3.0, δείτε

https://commons.wikimedia.org/wiki/File:Whit_Diffie_at_CFP_2007.jpg,

<https://commons.wikimedia.org/wiki/File:Martin-Hellman.jpg>.

Υπολόγισαν ότι με 20 εκατ. δολάρια θα μπορούσαν να σπάσουν το σύστημα μέσα σε μία μέρα, ποσό, που μεγάλες εταιρείες και κυβερνήσεις εύκολα θα διέθεταν για έναν τέτοιο σκοπό ³¹ (αργότερα ο M. Wiener το 1993 πρότεινε μια σχεδίαση ενός παράλληλου συστήματος υπολογιστών, με ικανότητα σε 3.5 ώρες να είναι εφικτή η εύρεση του κλειδιού).

³¹https://en.wikipedia.org/wiki/EFF_DES_cracker ανάκτηση 14-9-2015



Σχήμα 4.5: ECB mode (κρυπτογράφηση)

Ας πάρουμε τα πράγματα από την αρχή. Το 1972 προτάθηκε διαγωνισμός από την NBS³² (μετέπειτα NIST) για την καθιέρωση ενός προτύπου κρυπτογράφησης. Ο συμμετρικός αλγόριθμος DES αναπτύχθηκε από την IBM και επιλέχθηκε ως ο βασικός αλγόριθμος κρυπτογράφησης από την Αμερικανική κυβέρνηση το 1976. Το σύστημα DES ήταν η τροποποίηση ενός άλλου συστήματος, του LUCIFER³³, που ήταν ο πρώτος που χρησιμοποιούσε το σχήμα Feistel. Ο LUCIFER χρησιμοποιούσε κλειδί 128-bits. Καθώς η NBS δεν μπορούσε να αξιολογήσει την ασφάλεια του συστήματος, πρότεινε στην NSA να το κάνει. Η NSA έκανε δύο πράγματα στο σύστημα DES. Αρχικά, μείωσε το μήκος του κλειδιού σε 56 bits και κατόπιν άλλαξε τις σταθερές των S -κιβωτίων³⁴³⁵. Το νέο σύστημα που προέκυψε δημοσιεύτηκε το 1975. Το 1990 οι Adi Shamir και Eli Biher ανακάλυψαν μια πολύ ισχυρή μέθοδο κρυπτανάλυσης του DES, τη διαφορική κρυπτανάλυση, και έδειξαν ότι η κατασκευή των S -κιβωτίων απέτρεπε αυτήν την επίθεση. Αργότερα έγινε δημόσια γνωστό ότι η IBM γνώριζε (15 χρόνια πριν) για αυτήν την επίθεση και αυτός ήταν ο λόγος που η επίθεση απέτυχε. Αυτό το παράδειγμα προβλημάτισε τους ειδικούς και μέχρι σήμερα πολλοί αναρωτιούνται ποια είναι πραγματικά τα μέσα που έχει ένας επιτιθέμενος. Ερώτηση πολύ επίκαιρη, ειδικά μετά τις αποκαλύψεις του Ed Snowden³⁶.

4.2.4 Καταστάσεις λειτουργίας συμμετρικού συστήματος τμήματος

Όπως είδαμε, ένας συμμετρικός αλγόριθμος τμήματος κρυπτογραφεί ένα συγκεκριμένου μήκους μήνυμα. Π.χ. για τον DES είναι 64-bits. Τι γίνεται στην περίπτωση που το μήνυμά μας είναι μεγαλύτερο από 64-bits; Χρειαζόμαστε, λοιπόν, μία μέθοδο που να περιγράφει με ποιον τρόπο θα δουλεύει ο αλγόριθμος, για να επεξεργαστεί μεγάλα μηνύματα. Η πιο απλή κατάσταση λειτουργίας ενός κρυπτοσυστήματος τμήματος είναι η ECB (Electronic Codebook Mode) (σχήμα 4.5). Χωρίζουμε το αρχικό μας μήνυμα σε τμήματα των 64-bits. Για το τελευταίο τμήμα, αν δεν έχει μήκος 64-bits, προσθέτουμε μηδενικά (ή το δεκαεξαδικό, hex 80). Κατόπιν, κάθε τμήμα κρυπτογραφείται ανεξάρτητα.

Η επόμενη κατάσταση λειτουργίας CBC: Cipher Block Chaining χρησιμοποιεί Initial Vector, IV που γίνεται XOR με το μήνυμα πριν κρυπτογραφηθεί (σχήμα 4.6). Κατόπιν, το κρυπτογραφημένο μήνυμα που προκύπτει είναι το IV του δεύτερου τμήματος, κ.ο.κ. έως ότου τελειώσουν όλα τα τμήματα.

Λήμμα 4.2.1. Η κατάσταση λειτουργίας ECB δεν είναι *SS-secure* υπό επίθεση επιλεγμένου κειμένου (CPA).

Απόδειξη. Η Εύα εκτελεί επίθεση επιλεγμένου κειμένου. Στέλνει ένα μήνυμα (μήκους ενός block) M στο κρυπτοσύστημα που χρησιμοποιεί την κατάσταση λειτουργίας ECB και λαμβάνει ένα μήνυμα C που αποτελεί την κρυπτογράφηση του M . Κατόπιν, στέλνει δύο μηνύματα $m_0 = M || m$ και $m_1 = m' || m''$ ίδιου μήκους αποτελούμενα από δύο block. Τα m, m', m'' είναι τυχαία block. Τα $m_0, m_1 \neq M$. Το κρυπτοσύστημα (που δρα ως μαντείο), αφού διαλέξει ένα τυχαίο bit b , στέλνει το κρυπτογραφημένο μήνυμα $c = Enc(k, m_b)$. Ο

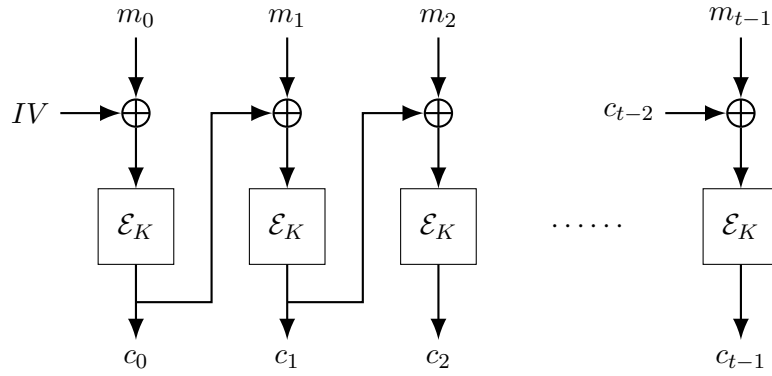
³²National Bureau of Standards Publications

³³[https://en.wikipedia.org/wiki/Lucifer_\(cipher\)](https://en.wikipedia.org/wiki/Lucifer_(cipher))

³⁴<https://www.schneier.com/crypto-gram/archives/2000/0615.html#DES>

³⁵https://en.wikipedia.org/wiki/Data_Encryption_Standard ανάκτηση 14-9-2015

³⁶https://en.wikipedia.org/wiki/Edward_Snowden



Σχήμα 4.6: CBC mode (κρυπτογράφηση) [4]

διαχωριστής $\mathcal{A}$, που χρησιμοποιεί η Εύα, λειτουργεί ως εξής. Συγκρίνει το πρώτο block του c με το C που ήδη κατέχει. Αν είναι ίσα, τότε επιλέγει $b = 0$, διαφορετικά επιλέγει $b = 1$. Επομένως,

$$Adv_{CPA}(\mathcal{A}) = |2Pr(\mathcal{A} \text{ εξάγει } b) - 1| = |2 \cdot 1 - 1| = 1.$$

□

Για να αποφύγουμε την προηγούμενη επίθεση, πρέπει η κρυπτογράφηση να μην είναι ντετερμινιστική. Ένας τρόπος να το πετύχουμε αυτό είναι με χρήση των διανυσμάτων αρχικοποίησης (IV), όπως στην κατάσταση λειτουργίας CBC.

Τέλος, υπάρχουν επιθέσεις σε κρυπτοσυστήματα τμήματος και με άλλες καταστάσεις λειτουργίας. Ένα τέτοιο παράδειγμα είναι η χρήση του AES με κατάσταση λειτουργίας OCB³⁷. Αυτή η επίθεση παρουσιάστηκε το 2018 και το OCB2 παρουσιάστηκε το 2004, και έγινε πρότυπο το 2009.

4.3 AES

Στις 26 Μαΐου 2002 εγκρίθηκε από το Αμερικανικό Υπουργείο Εμπορίου, έπειτα από επιλογή ανάμεσα σε 16 συμμετρικά κρυπτοσυστήματα, ο συμμετρικός αλγόριθμος τμήματος Rijndael³⁸, δείτε [3]. Ο δεύτερος καλύτερος ήταν το σύστημα Serpent³⁹ και ο τρίτος καλύτερος το σύστημα Twofish. Το όνομα αυτό προέκυψε από τη συνένωση των ονομάτων των δύο εφευρετών του, Vincent **Rijmen** και Joan **Daemen**. Μετά την επιλογή του από την αμερικανική κυβέρνηση, η οποία τον τροποποίησε, πλέον ονομάζεται AES: **A**dvanced **E**ncryption **S**tandard⁴⁰. Αντί του σχήματος Feistel χρησιμοποιεί το σχήμα SPN: **S**ubstitution and **P**ermutation **N**etwork. Το μήκος των τμημάτων είναι 128 bits και τα μήκη των κλειδών 128, 192 και 256 bits. Η βασική διαφορά των σχημάτων Feistel και SPN είναι ότι στο σχήμα Feistel το αρχικό μήνυμα χωρίζεται σε δύο τμήματα τα οποία κατά την εκτέλεση του αλγορίθμου Feistel αλληλεπιδρούν, ενώ στο σχήμα SPN το αρχικό τμήμα χωρίζεται σε πολλά μικρά τμήματα, όπου το καθένα εισέρχεται στα S-boxes και κατόπιν εφαρμόζεται μια μετάθεση (mix) στην κάθε έξοδο. Για παράδειγμα, δείτε έναν γύρο ενός σχήματος Feistel (σχήμα 4.2) όπου ένα τμήμα του αρχικού τμήματος μένει αμετάβλητο, ώστε να μπορεί να γίνει η αποκρυπτογράφηση. Αυτό το πράγμα δεν συμβαίνει στο σχήμα SPN, όπου σε κάθε επανάληψη όλα τα bits του τμήματος αλλάζουν.

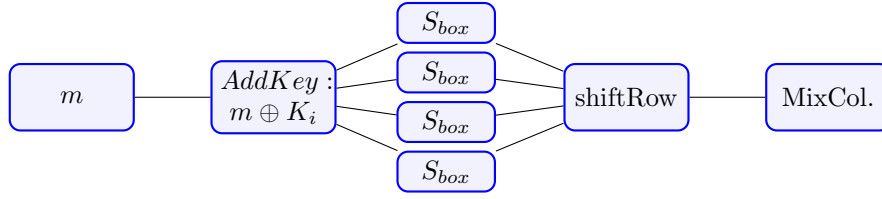
Στον AES, ανάλογα με το μήκος του κλειδιού, 128, 192 ή 256, εκτελούνται 10, 12 ή 14 επαναλήψεις, αντίστοιχα. Αποτελείται από τρεις διαφορετικές ρουτίνες: Key Expansion, Initial Round(IR) και SPN. Η πρώτη ρουτίνα από το αρχικό μας κλειδί παράγει τα απαραίτητα υποκλειδιά (round-keys) που

³⁷<https://eprint.iacr.org/2019/311.pdf>

³⁸Για μια εισαγωγή στον Rijndael, δείτε την πτυχιακή εργασία, *Το πρόβλημα MQ: Επιθέσεις στον AES και το πρόβλημα διακριτού λογαρίθμου επί ελλειπτικών καμπυλών*, του Β. Τζάτζιου. <https://ikee.lib.auth.gr/record/294644/files/GRI-2017-20399.pdf>

³⁹Ο Serpent είχε την καλύτερη υλοποίηση σε επίπεδο hardware

⁴⁰<http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>



Σχήμα 4.7: Μια τυπική επανάληψη του mini-AES.

χρησιμοποιούνται σε κάθε επανάληψη. Η δεύτερη εφαρμόζει και στο αρχικό τμήμα του μηνύματος με το κλειδί. Στην τρίτη, που είναι ο πυρήνας του συστήματος, εφαρμόζεται το σχήμα SPN που υλοποιείται στα τέσσερα επίπεδα: SubBytes, ShiftRow, MixColumn, AddRoundkey. Στην τελευταία επανάληψη δεν εκτελείται ο αλγόριθμος MixColumn. Ο πρώτος αλγόριθμος, ο SubBytes, υλοποιείται με (τα γνωστά) S-boxes, ενώ οι επόμενοι δύο με τα P-boxes. Στο σχήμα 4.7, το μήνυμα, m , το οποίο έχει προκύψει από την προηγούμενη επανάληψη, γίνεται και με το κλειδί K_i και κατόπιν διαχωρίζεται το μήνυμα σε 4-ισομεγέθη blocks μήκους 1 nibble (στην περίπτωση του AES-128 έχουμε 8—superblocks με μήκος 1—byte) και εισέρχονται στο S-box. Τέλος, η έξοδος εισέρχεται διαδοχικά από τους αλγόριθμους ShiftRow, MixColumn. Στο S-box εισέρχεται 1-nibble και εξέρχεται 1-nibble (στον AES-128 το S-box είναι 8-bits). Στον τελευταίο γύρο ο αλγόριθμος MixColumn δεν εκτελείται.

Για να καταλάβουμε πώς ακριβώς δουλεύει ο AES θα χρησιμοποιήσουμε μια απλοποίησή του, τον mini AES του Raphael Chung-Wei Phan, που δημοσιεύτηκε στο περιοδικό Cryptologia. Στον mini AES το κλειδί και το μήνυμα έχουν μήκος 16 bits, και εκτελούνται μόνο δύο επαναλήψεις. Ο mini AES, για την ακρίβεια, αποτελεί απλοποίηση του AES όταν το κλειδί έχει μήκος 128 bits. Στην περίπτωση αυτήν το κλειδί μήκους 128 bits αναπαρίσταται με έναν πίνακα 4×4 , όπου κάθε στοιχείο του πίνακα είναι 1 byte. Στην περίπτωση του mini AES το κλειδί αναπαρίσταται με έναν 2×2 πίνακα, όπου κάθε στοιχείο του είναι 1 nibble, δηλαδή 4 bits. Επίσης, το μήκος τμήματος (blocksize) στον AES-128 είναι 128-bits, ενώ στον mini-AES είναι 8-bits. Για να κάνουμε πράξεις μεταξύ των bytes (ή αντίστοιχα των nibbles), δουλεύουμε στο πεπερασμένο σώμα $\mathbb{F}_{2^8}$ ($\mathbb{F}_{2^4}$ αντίστοιχα για το mini AES). Αντιστοιχίζουμε κάθε byte ($b_7, \dots, b_0$) με ένα πολυώνυμο $\sum_{i=0}^7 b_i x^i$, $b_i \in \{0, 1\}$. Στην περίπτωση του AES το σώμα ορίζεται από το πολυώνυμο $x^8 + x^4 + x^3 + x + 1$ (ενώ στο mini AES από το πολυώνυμο $x^4 + x + 1$). Θα δείξουμε πώς προσθέτουμε και πολλαπλασιάζουμε nibbles στον mini AES.

4.3.1 Πρόσθεση και πολλαπλασιασμός nibbles στον mini AES

Η πρόσθεση δύο nibbles είναι η πρόσθεση mod 2. Ας δούμε τον πολλαπλασιασμό με ένα παράδειγμα:

$$(0110) \cdot (1011) = (x^2 + x)(x^3 + x + 1) =$$

$$x^5 + x^3 + x^2 + x^4 + x^2 + x =$$

$$x^5 + x^4 + x^3 + 2x^2 + x = x^5 + x^4 + x^3 + x,$$

διότι $2x^2 = 0$ στο σώμα $\mathbf{F} = \mathbb{F}_2[x]/(x^4 + x + 1)$. Για να πέσει ο βαθμός του γινομένου, δουλεύουμε mod $(x^4 + x + 1)$. Τότε το $x^4 = -1 - x \equiv 1 + x$, άρα

$$x^5 + x^4 + x^3 + x = x(1 + x) + (1 + x) + x^3 + x = x^3 + x^2 + x + 1 \rightarrow (1111)$$

(η ισότητα εννοείται στο σώμα $\mathbf{F}$).

Άσκηση 4.5 Να γίνει ο πολλαπλασιασμός, $(1101) \cdot (1111)$ στο προηγούμενο σώμα $\mathbf{F}$.

Άσκηση 4.6 Να γίνουν οι παρακάτω υπολογισμοί στο σώμα $\mathbf{F} = \mathbb{F}_2[x]/(x^4 + x^3 + 1)$.

- (i). $(1101) \cdot ((1110) + (0001))$
- (ii). $((1101) + (0101)) \cdot ((1110) + (0001))$
- (iii). $((1101))^2 + ((0101))^2$

4.3.2 Κρυπτογράφηση στο mini AES

Η συνάρτηση κρυπτογράφησης του mini AES ενός μηνύματος m μήκους 16 bits είναι

$$E(m, K) = (\sigma_2 \circ \Pi \circ \gamma) \circ (\sigma_1 \circ \text{mix} \circ \Pi \circ \gamma \circ \sigma_0(m)).$$

Όπως παρατηρούμε, έχουμε δύο επαναλήψεις ($n = 2$) και χρησιμοποιούμε 3 κλειδιά, το αρχικό K_0 και τα δύο που παράγονται από την Key expansion. Στην αρχή και στο τέλος κάθε επανάληψης γίνεται xor με ένα κλειδί $(\sigma_0, \sigma_1, \sigma_2)$. Στην τελευταία επανάληψη δεν εκτελείται ο αλγόριθμος mix column. Να αναφέρουμε εδώ ότι και σε περίπτωση που εκτελεστεί αυτός ο αλγόριθμος στην τελευταία επανάληψη, οι δύο εκδοχές συνδέονται με έναν γραμμικό μετασχηματισμό. Δηλαδή, η επιλογή του αν θα εκτελεστεί ή όχι στην τελευταία επανάληψη ο mix column δεν επηρεάζει την ασφάλεια του AES. Η συνάρτηση σ_i ($i = 0, 1, 2$) δέχεται δύο εισόδους, το μήνυμα m εκφρασμένο σε 2×2 πίνακα (με στοιχεία nibbles) και το κλειδί K_i επίσης εκφρασμένο σε πίνακα 2×2 . Π.χ. $\sigma_0(m) = m \oplus K_0$. Η συνάρτηση $\gamma : \{0, 1\}^4 \rightarrow \{0, 1\}^4$ είναι ο πίνακας του S -box. Στην περίπτωση του mini AES είναι ο παρακάτω:

Input	Output	Input	Output
0000	1110	1000	0011
0001	0100	1001	1010
0010	1101	1010	0110
0011	0001	1011	1100
0100	0010	1100	0101
0101	1111	1101	1001
0110	1011	1110	0000
0111	1000	1111	0111

Η συνάρτηση Π εναλλάσσει τα στοιχεία της δεύτερης γραμμής (shiftRow), ενώ η συνάρτηση mix (που υλοποιεί το βήμα MixColumn) είναι πολλαπλασιασμός από αριστερά με τον πίνακα

$$A = \begin{pmatrix} 0011 & 0010 \\ 0010 & 0011 \end{pmatrix}.$$

Δηλαδή, $\text{mix}(m) = Am$. Τέλος, εφαρμόζουμε την AddRoundKey, που είναι xor με το κλειδί του i -γύρου.

Όλοι οι προηγούμενοι αλγόριθμοι Subbytes, ShiftRow, Mixcolumn μπορούν να αντιστραφούν. Δηλαδή ο πίνακας που χρησιμοποιείται στον MixColumn είναι αντιστρέψιμος στο $\mathbb{F}_2$. Επίσης, η διαδικασία ShiftRow αντιστρέφεται.

Άσκηση 4.7 Υλοποιήστε τον αλγόριθμο κρυπτογράφησης του mini-AES και κρυπτογραφήστε ένα μήνυμα 16-bits.

Παρατήρηση 4.3.1. Είναι πολύ σημαντικό να παρατηρήσουμε ότι η προηγούμενη απλοποίηση του AES παραλείπει μια σημαντική κατασκευή του AES. Στον κανονικό AES η κατασκευή του S-Box είναι αλγεβρική (στον AES έχουμε ένα μόνο S-Box που χρησιμοποιείται σε όλους τους γύρους), ενώ στον mini-AES του Phan δεν είναι αλγεβρική η κατασκευή αυτή. Απλά, στο αρχικό του paper ο Phan πήρε ένα S-box από ένα κομμάτι των S-boxes του DES. Στον αλγόριθμο Rijndael η κατασκευή των S-boxes είναι μια διαδικασία τριών βημάτων. Αρχικά, γίνεται η αντιστροφή μέσα στο σώμα (το μη-γραμμικό κομμάτι του AES), κατόπιν η εφαρμογή μιας γραμμικής απεικόνισης και τέλος η πρόσθεση (στο σώμα) με έναν σταθερό αριθμό. Δηλαδή,

$$S - \text{Box} : \text{Add} \circ \text{Aff.} \circ \text{inv}(m).$$

Κάνοντας τη σύμβαση $\text{inv}(0) = 0$, η προηγούμενη συνάρτηση έχει νόημα για όλα τα στοιχεία του σώματος. Αυτή ακριβώς η κατασκευή επιτρέπει αλγεβρικές επιθέσεις στον AES. Λόγω αυτής της αλγεβρικής κατασκευής του αλγορίθμου, μπορούμε να περιγράψουμε τον AES από ένα σύστημα δευτεροβάθμιων εξισώσεων πολλών μεταβλητών, του οποίου η επίλυση δίνει το κλειδί (η επίλυση τέτοιων συστημάτων είναι πρόβλημα της κλάσης NP-hard). Το σύστημα αυτό μέχρι σήμερα δεν μπορεί να λυθεί αποδοτικά. Μια απλοποιημένη εκδοχή του AES που διατηρεί την αλγεβρική κατασκευή των S-boxes είναι ο Babbage-AES⁴¹ του Clifford Bergman, καθώς και οι Small Scale Variants⁴² $SR(n, r, c, e)$ και $SR^*(n, r, c, e)$. Η Nyberg έχει προτείνει διάφορες μεθόδους κατασκευής S-Boxes⁴³. Μία από αυτές εφαρμόστηκε στο AES.

Άσκηση 4.8 Εξετάστε αν ισχύει το avalanche effect στον AES-128. Αναλυτικότερα, φτιάξτε αρκετά ζευγάρια (> 30) μηνυμάτων (m_1, m_2) που να διαφέρουν σε ένα bit. Εξετάστε σε πόσα bits διαφέρουν τα αντίστοιχα κρυπτομηνύματα. Δοκιμάστε με δύο καταστάσεις λειτουργίας: ECB και CBC (η δεύτερη θέλει και IV block). Τα μήκη των μηνυμάτων που θα χρησιμοποιήσετε να έχουν μήκος διπλάσιο του μήκους ενός block. Δηλαδή για τον AES-128, να είναι μήκους 256 bits.

⁴¹E.Kleiman, The XL and XSL attacks on Baby Rijndael. Master Thesis.

⁴²<https://www.iacr.org/archive/fse2005/35570143/35570143.pdf>

⁴³Differentially uniform mapping for cryptography. Eurocrypt'93.

Βιβλιογραφία

- [1] Boneh, Dan; Shoup, V. *A Graduate Course in Applied Cryptography*, Version 0.5, Jan. 2020. <http://toc.cryptobook.us/>.
- [2] Curtin, M. *Brute force. Cracking the Data Encryption Standard*. Copernicus Books, 2005.
- [3] Daemen, J.; Rijmen, V. *The design of Rijndael. AES : Advanced Encryption Standard*. Springer, 2002.
- [4] Jérémy Jean. *TikZ for Cryptographers*. 2016.
- [5] Mullen, G. L.; Mummert, C. *Finite fields and Applications*. AMS, 2007.
- [6] Vasilenko, O. N. *Number-Theoretic Algorithms in Cryptography*. Translations of Mathematical Monographs, vol. 232. AMS, 2006.
- [7] Πομπόρτσας, Α.; Παπαδημητρίου, Γ. *Ασφάλεια δικτύων υπολογιστών*. Εκδ. Τζιόλας, 2003.
- [8] Νάστου, Π. Ε.; Σπυράκης, Π. Γ.; Σταματίου, Γ. Κ. *Σύγχρονη Κρυπτογραφία*. Εκδ. Ελληνικά γράμματα, 2003.

Κεφάλαιο 5

Συναρτήσεις Κατακερματισμού & Κώδικες αυθεντικοποίησης

Σύνοψη - Περίληψη. Σε αυτό το κεφάλαιο παρουσιάζουμε τις συναρτήσεις κατακερματισμού (hash functions) και τους κώδικες αυθεντικοποίησης (Message Authentication Codes: MAC). Τα δύο αυτά σχήματα κρυπτογραφίας συνδέονται. Πολλές φορές οι MAC ονομάζονται και συναρτήσεις κατακερματισμού με κλειδί (Hash keyed functions). Είναι συμμετρικά συστήματα και έχουν πάρα πολλές χρήσεις στην κρυπτογραφία και όχι μόνο. Π.χ. οι συναρτήσεις κατακερματισμού χρησιμοποιούνται για την ακεραιότητα των αρχείων ή για την επαλήθευση κάποιας διαδικασίας. Οι συναρτήσεις MAC χρησιμοποιούνται για την καταπολέμηση της πειρατείας των ηλεκτρονικών παιχνιδιών. Τέλος, στην αυθεντικοποιημένη κρυπτογράφηση εφαρμόζεται μια MAC πριν ή μετά την κρυπτογράφηση του μηνύματος, δείτε [4, Κεφ. 1].

Προαπαιτούμενη γνώση. Για αυτό το κεφάλαιο χρειάζονται βασικές γνώσεις πιθανοτήτων. Μια βασική επίθεση που γίνεται στις συναρτήσεις κατακερματισμού είναι η επίθεση των γενεθλίων (Birthday attack). Αυτή βασίζεται στο γνωστό παράδοξο των γενεθλίων που συνήθως μαθαίνουμε στις πιθανότητες. Επίσης, κάποιες βασικές γνώσεις ανάλυσης χρειάζονται στην απόδειξη κάποιων αποτελεσμάτων που αφορούν το παράδοξο των γενεθλίων. Για τις πιθανότητες μπορείτε να δείτε το βιβλίο [8]. Για τις συναρτήσεις κατακερματισμού μπορείτε να δείτε [5, Ενότητα 1.9], [7, Ενότητα 2.4]. Για το πρόβλημα των γενεθλίων μπορείτε να δείτε [4, Κεφ. 6]. Για τη συνάρτηση κατακερματισμού sha1, δείτε [6, Appendix F].

5.1 Εισαγωγή

Ξεκινάμε με τον ορισμό της συνάρτησης κατακερματισμού (Hash Function).

Ορισμός 5.1.1. Ονομάζουμε συνάρτηση κατακερματισμού έναν αλγόριθμο που δέχεται ένα σύνολο δεδομένων (μεταβαλλόμενου μήκους) και ως έξοδο δίνει ένα σύνολο δεδομένων συγκεκριμένου μήκους.

Επομένως, $h : \{0,1\}^* \rightarrow \{0,1\}^n$, όπου το n συνήθως είναι μεταξύ του 128 και 512. Ο λόγος που συμβαίνει αυτό θα φανεί παρακάτω, όταν θα μιλήσουμε για το παράδοξο των γενεθλίων. Είναι πολύ βασικό η συνάρτηση αυτή να είναι αποδοτική, τόσο για software όσο και για υλοποιήσεις σε hardware. Οι συναρτήσεις κατακερματισμού, όταν συνδυαστούν με ένα κλειδί, μπορούν να χρησιμοποιηθούν για έλεγχο αυθεντικοποίησης ενός μηνύματος (τότε ονομάζονται MAC, όπως θα δούμε στην επόμενη ενότητα). Οι χρήσεις τους είναι πάρα πολλές. Για παράδειγμα, τις χρησιμοποιούμε, για να διαπιστώσουμε αν δύο τμήματα σε δύο σύνολα δεδομένων είναι ίδια και για την ακεραιότητα δεδομένων. Ωστόσο, για το τελευταίο μπορούμε να χρησιμοποιήσουμε συστήματα που ονομάζονται αποτυπώματα (fingerprints), όπως το αποτύπωμα του Rabin. Επίσης, αν ενδιαφερόμαστε για τυχόν λάθη, που μπορούν να συμβούν κατά τη μετάδοση του μηνυμάτος μας, μπορούμε να χρησιμοποιήσουμε κώδικες διορθωτές λαθών, όπως τον CRC. Πολλές φορές θα έχετε δει ότι ένα αρχείο που κατεβάζετε από το διαδίκτυο συνοδεύεται και από κάποιο αλφαριθμητικό συγκεκριμένου μήκους, το οποίο συνήθως είναι ένα md5 ή SHA-1 hash. Επομένως, όταν κατεβάσουμε το αρχείο, υπολογίζουμε στον υπολογιστή μας την κατάλληλη hash επί του αρχείου και το συγκρίνουμε με το hash που συνοδεύεται.

Μια πολύ σημαντική εφαρμογή των συναρτήσεων κατακερματισμού είναι στις βάσεις δεδομένων, στο πρόβλημα αναζήτησης σε ένα λεξικό (hash tables). Αν, για παράδειγμα, έχουμε ένα ευρετήριο τηλεφώνων της μορφής $\{name : phone\}$, τότε κατασκευάζουμε μια λίστα $L = [(T[i]), i = 1, 2, \dots, m]$ τέτοια, ώστε $T[i] = hash(name[i]) \in \{1, 2, \dots, m\}$. Αν υποθέσουμε ότι δεν έχουμε διενέξεις και αν αναζητούμε π.χ. το τηλέφωνο του John Baker, τότε υπολογίζουμε το $hash(John Baker)$, το οποίο μας δίνει τον δείκτη της λίστας L , όπου είναι αποθηκευμένο το τηλέφωνο. Επομένως, αν δεν υπάρχουν διενέξεις, ο χρόνος εύρεσης είναι $O(1)$. Επίσης, οι συναρτήσεις κατακερματισμού χρησιμοποιούνται σε συστήματα απόδειξης - εργασίας (proof-of-work), όπως είναι το Bitcoin.

Στην κρυπτογραφία απαιτούμε κάποιες ιδιότητες από τις συναρτήσεις κατακερματισμού, ώστε να θεωρούνται ασφαλείς. Ας είναι $h : X \rightarrow Y$ ($|X| > |Y|$) μία συνάρτηση κατακερματισμού.

- i. **Αντίσταση στις διενέξεις (collision resistance).** Είναι υπολογιστικά αδύνατο να βρούμε δύο τιμές x, x' με $x \neq x'$ τέτοιες, ώστε $h(x) = h(x')$.
 - ii. **Αντίσταση στην αντίστροφη εικόνα (first pre-image resistance ή one-wayness).** Δοθέντος $y \in Y$, είναι υπολογιστικά αδύνατο να βρούμε $x \in X$ τέτοιο, ώστε $h(x) = y$.
 - iii. **Αντίσταση στη δεύτερη αντίστροφη εικόνα (second pre-image resistance).** Δοθέντος $x \in X$, είναι υπολογιστικά αδύνατο να βρούμε $x' \in X$ με $x' \neq x$, τέτοιο, ώστε $h(x') = h(x)$.
- Όταν η h ικανοποιεί την απαίτηση (ii), λέμε ότι είναι **συνάρτηση μίας φοράς (one way function)**.

Ορισμός 5.1.2. Λέμε ότι μία συνάρτηση κατακερματισμού είναι κρυπτογραφικά ασφαλής όταν έχει αντίσταση στις διενέξεις (δηλαδή ικανοποιεί την απαίτηση (i)).

Μερικοί συγγραφείς απαιτούν για μια κρυπτογραφική συνάρτηση κατακερματισμού να είναι μίας φοράς. Όπως θα δούμε στο παρακάτω λήμμα, αν η συνάρτηση έχει αντίσταση στις διενέξεις, τότε είναι και μίας φοράς.

- Λήμμα 5.1.1.** (i). Αν μια συνάρτηση κατακερματισμού έχει αντίσταση στις διενέξεις, τότε έχει και αντίσταση στη δεύτερη αντίστροφη εικόνα.
 (ii). Μία συνάρτηση κατακερματισμού, που έχει αντίσταση στη δεύτερη αντίστροφη εικόνα, τότε είναι μίας φοράς.

Απόδειξη. (i). Υποθέτουμε ότι η h δεν έχει αντίσταση στη δεύτερη αντίστροφη εικόνα. Διαλέγουμε ένα τυχαίο $x \in X$ και υπολογίζουμε μία αντίστροφη εικόνα του $h(x)$. Έστω x' , με $x' \neq x$. Τότε το ζευγάρι (x, x') αποτελεί μία διένεξη της h .

(ii). [συνοπτική]. Υποθέτουμε ότι η h δεν είναι μίας φορές. Επομένως, υπάρχει πιθανοτικός αλγόριθμος $\mathcal{A}$ ο οποίος δέχεται ως είσοδο ένα στοιχείο $y \in Y$ και υπολογίζει ένα $x \in X$ με $h(x) = y$. Θα αποδείξουμε ότι η h δεν έχει αντίσταση στη δεύτερη αντίστροφη εικόνα. Διαλέγουμε ένα τυχαίο $x \in X$. Εκτελούμε τον $\mathcal{A}$ με είσοδο $y = h(x) \in Y$ και ως είναι x' η έξοδος του αλγορίθμου. Αποδεικνύεται ότι με μεγάλη πιθανότητα το $x' \neq x$. Άρα $h(x) = h(x')$ και $x \neq x'$. Δηλαδή, δεν έχει αντίσταση στη δεύτερη αντίστροφη εικόνα. $\square$

Πόρισμα 5.1.1. *Collision resistance $\Rightarrow$ One way.*

Παρατήρηση 5.1.1. Στο πρότυπο FIPS202⁴⁴ ορίστηκε η νέα οικογένεια συναρτήσεων κατακερματισμού SHA3 οι οποίες βασίστηκαν στη συνάρτηση κατακερματισμού Keccak. Στο FIPS202 επίσης ορίστηκε μία συνάρτηση *extendable-output functions (XOFs)*⁴⁵ η οποία έχει μήκος εξόδου που ορίζουμε εμείς. Δεν είναι συναρτήσεις κατακερματισμού, αλλά μπορούν να χρησιμοποιηθούν με παρόμοιο τρόπο.

Άσκηση 5.1 Ο παρακάτω κώδικας (python 3) υπολογίζει την md5 ενός κειμένου και επιστρέφει τη δεκαεξαδική ισοδύναμη μορφή του.

```
1 import hashlib
2 result = hashlib.md5(b'Crypto')
3 print(result.hexdigest())
```

Υπολογίστε την md5 των λέξεων, crypto, CRYPTO, CRyPTo. Κατόπιν, μετατρέψτε τις εξόδους στο ισοδύναμο δυαδικό του και μετρήστε σε πόσα bits διαφέρουν.

Άσκηση 5.2 Ο παρακάτω κώδικας (python 3) υπολογίζει τις sha512, sha224, sha1 ενός κειμένου και επιστρέφει τη δεκαεξαδική ισοδύναμη μορφή του.

```
1 import hashlib
2 result_sha512 = hashlib.sha512(b'Crypto')
3 result_sha224 = hashlib.sha224(b'Crypto')
4 result_sha1 = hashlib.sha1(b'Crypto')
5 print('sha512:', result_sha512.hexdigest())
6 print('sha224:', result_sha224.hexdigest())
7 print('sha1:', result_sha1.hexdigest())
```

Υπολογίστε τις προηγούμενες hashes στις λέξεις, crypto, CRYPTO, CRyPTo.

Άσκηση 5.3 Θεωρήστε τα παρακάτω δύο (δεκαεξαδικά) block⁴⁶.

```
M1=b'0e306561559aa787d00bc6f70bbdfe3404cf03659e704f8534c00ffb659c4
c8740cc942feb2da115a3f4155cbb8607497386656d7d1f34a42059d78f5a8dd1ef'

M2=b'0e306561559aa787d00bc6f70bbdfe3404cf03659e744f8534c00ffb659c4
c8740cc942feb2da115a3f415dcbb8607497386656d7d1f34a42059d78f5a8dd1ef'
```

Υπολογίστε την md5 hash των δύο μηνυμάτων. Τι παρατηρείτε;

Υπόδ. Εδώ η είσοδος είναι hexstrings, οπότε πρέπει να χρησιμοποιήσετε την `binascii.unhexlify` πριν τα εισάγετε στην md5 hash.

⁴⁴<https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf>

⁴⁵SHAKE128, SHAKE256

⁴⁶Tao Xie and Dengguo Feng, Construct MD5 Collisions Using Just A Single Block Of Message

5.2 Εύρεση διενέξεων-Επίθεση των γενεθλίων (birthday attack)

Πρώτοι οι Paul Camion και Jacques Patarin έδωσαν την ονομασία birthday problem⁴⁷. Το παράδοξο των γενεθλίων συνήθως δίνεται όπως παρακάτω.

Θεώρημα 5.2.1 (Παράδοξο των γενεθλίων). Η πιθανότητα δύο άτομα σε μία τάξη m -μαθητών να έχουν γενέθλια την ίδια ημέρα είναι μεγαλύτερη από $1/2$, όταν $m \geq 23$.

Απόδειξη. Έστω P_j η πιθανότητα j -άτομα να έχουν γενέθλια την ίδια ημέρα. Τότε, $P_1 + P_2 = 1$. Άρα, αρκεί να υπολογίσω την πιθανότητα P_1 , δηλαδή, την πιθανότητα όλοι οι μαθητές να έχουν γενέθλια διαφορετικές ημέρες. Υποθέτουμε ότι οι γεννήσεις ακολουθούν ομοιόμορφη κατανομή. Τότε η πιθανότητα ο δεύτερος μαθητής να έχει γεννηθεί διαφορετική ημέρα από τον πρώτο είναι $\frac{364}{365}$. Η πιθανότητα ο τρίτος μαθητής να έχει γεννηθεί διαφορετική ημέρα από τον πρώτο και δεύτερο μαθητή είναι $\frac{363}{365}$. Συνεχίζοντας με αυτόν τον τρόπο καταλήγουμε ως εξής:

$$P_1 = \frac{364}{365} \cdot \frac{363}{365} \cdots \frac{365 - (m - 1)}{365} = \frac{\prod_{j=1}^{m-1} (365 - j)}{365^{m-1}}.$$

Επομένως, $P_2 = 1 - P_1$. Θα λύσουμε την ανίσωση $P_2 \geq 1/2$, ως προς m ή ισοδύναμα $P_2 \leq 1/2$. Μπορούμε εύκολα να σχηματίσουμε έναν πίνακα με τιμές $(i, P(i))$ με $i = 2, 3, \dots$

[2, 0.0027397260274]	[3, 0.0082041658847]	[4, 0.0163559124666]
[5, 0.0271355736998]	[6, 0.0404624836491]	[7, 0.0562357030960]
[8, 0.0743352923517]	[9, 0.0946238338892]	[10, 0.116948177711]
[11, 0.141141378322]	[12, 0.167024788838]	[13, 0.194410275232]
[14, 0.223102512005]	[15, 0.252901319764]	[16, 0.283604005253]
[17, 0.315007665297]	[18, 0.346911417872]	[19, 0.379118526032]
[20, 0.411438383581]	[21, 0.443688335165]	[22, 0.475695307663]
[23, 0.507297234324].		

Παρατηρούμε ότι για $m = 23$, η πιθανότητα είναι $\approx 0.507 > 1/2$. □

Παρατήρηση 5.2.1. Αξίζει να σημειωθεί ότι οι γεννήσεις, στην πραγματικότητα, δεν κατανέμονται ομοιόμορφα κατά τη διάρκεια του έτους, επομένως θα ανέμενε κανείς κάποια απόκλιση από το κλασικό Παράδοξο των Γενεθλίων. Ωστόσο, μελέτες σε πραγματικά δεδομένα έχουν δείξει ότι στην πράξη οι πιθανότητες παραμένουν πολύ κοντά στο θεωρητικό μοντέλο. Για παράδειγμα, σε ανάλυση γεννήσεων στην Ιαπωνία για την περίοδο 1900-1988, παρατηρήθηκε ότι η πιθανότητα σύμπτωσης υπερβαίνει το 50% για $n = 27$, αντί για $n = 23$ όπως προκύπτει υπό την υπόθεση της ομοιόμορφης κατανομής⁴⁸.

Παρατήρηση 5.2.2. Ο κυκλικός αλγόριθμος του Floyd μπορεί να χρησιμοποιηθεί για την εύρεση διενέξεων με λίγη μνήμη, δείτε [εδώ](#).

Μια πιο γενική εκδοχή του παράδοξου των γενεθλίων είναι η παρακάτω:

(Παράδοξο των γενεθλίων-2η εκδοχή). Έχουμε το παρακάτω πείραμα: Ένα κουτί περιέχει n -αριθμημένες μπάλες με αρίθμηση από το 1 έως το n . Κάνουμε δεγματοληψία m -μπαλών ($1 < m < n$) με επανατοποθέτηση. Ζητάμε την πιθανότητα ℓ -μπάλες ($1 \leq \ell \leq m$) να είναι διαφορετικές.

Δουλεύοντας, όπως προηγούμενως, προκύπτει η πιθανότητα:

$$P(n, m, \ell) = \frac{1}{\ell!} \sum_{j=0}^{\ell} (-1)^{\ell-j} \binom{\ell}{j} j^m P_1(n, m),$$

⁴⁷P. Camion, J. Patarin, The Knapsack Hash Function proposed at Crypto '89 can be broken, Advances in Cryptology, EUROCRYPT '91

⁴⁸Mase, Shigeru (1992). Approximations to the Birthday Problem with Unequal Occurrence Probabilities and Their Application to the Surname Problem in Japan. Annals of the Institute of Statistical Mathematics, 44(3).

όπου

$$P_1(n, m) = \frac{\prod_{j=1}^{m-1} (n - j)}{n^{m-1}}$$

είναι η πιθανότητα όλες οι m -μπάλες να είναι διαφορετικές. Για $n = 365$, $m = 23$, $\ell = 1$ και υπολογίζοντας την πιθανότητα $1 - P(365, 23, 1) = 1 - P_1(365, 23)$, έχουμε την πρώτη εκδοχή του παραδόξου των γενεθλίων.

Έστω τώρα $h : X \rightarrow Y$ μια συνάρτηση κατακερματισμού, με $|X| > |Y| = n = 2^k$. Η πιθανότητα η h να μην έχει διενέξεις, υπολογίζοντας m -τιμές $h(x_i)$, $i = 1, 2, \dots, m$, είναι $P_1(n, m)$. Επομένως, η πιθανότητα να έχουμε τουλάχιστον μία διένεξη είναι $p_c = 1 - P_1(n, m)$. Έχουμε

$$P_1(n, m) = \frac{\prod_{j=1}^{m-1} (n - j)}{n^{m-1}} = \prod_{j=1}^{m-1} \left(1 - \frac{j}{n}\right).$$

Αλλά $1 - x \approx e^{-x}$ για αρκετά μικρά x . Επομένως, προσεγγίζουμε την $1 - j/n$ με την $e^{-j/n}$ για αρκετά μεγάλα n . Έτσι, έχουμε

$$P_1(n, m) \approx \prod_{j=1}^{m-1} e^{-j/n} = e^{-\frac{\sum_{j=1}^{m-1} j}{n}} = e^{-\frac{m(m-1)}{2n}}.$$

Επομένως, η πιθανότητα να έχουμε τουλάχιστον μία διένεξη, μετά από υπολογισμό m -hashes, είναι:

$$p_c \approx 1 - e^{-m(m-1)/2n}.$$

Γράφουμε την τελευταία εξίσωση ως $e^{-m(m-1)/2n} \approx 1 - p_c$ και λύνουμε ως προς m .

$$-m(m-1)/2n \approx \ln(1 - p_c),$$

οπότε

$$m^2 - m \approx -2n \ln(1 - p_c)$$

και για αρκετά μεγάλα m αγνοούμε τον όρο m στην παράσταση $m^2 - m$. Επομένως,

$$m \approx \sqrt{2n \ln(1/(1 - p_c))}.$$

Θέτουμε $p_c = 1/2$ και έχουμε $m \approx 1.17\sqrt{n}$. Δηλαδή, αν υπολογίσουμε περίπου $2^{k/2}$ **τυχαία** hashes της h , τότε με πιθανότητα $1/2$ θα βρούμε μία τουλάχιστον διένεξη. Αν οι είσοδοι στην hash δεν ακολουθούν ομοιόμορφη κατανομή, τότε το προηγούμενο δεν ισχύει.

Για παράδειγμα, η συνάρτηση κατακερματισμού md5 έχει $n = 2^{128}$ πιθανές εξόδους

$$\text{md5} : \{0, 1\}^* \rightarrow \{0, 1\}^{128}.$$

Επομένως, για να βρούμε μια διένεξη, χρειαζόμαστε κατά μέσο όρο, τουλάχιστον $m = 2^{64}$ τυχαία μηνύματα $x_1, \dots, x_m$. Τότε, με πιθανότητα $1/2$ θα βρούμε μία διένεξη ανάμεσα σε αυτά. Το όριο 2^{64} για την κρυπτογραφία δεν θεωρείται ασφαλές. Εκτός αυτού, για τη συγκεκριμένη συνάρτηση βρέθηκε επίθεση (δηλαδή εύρεση διένεξης)⁴⁹ σε χρόνο $O(2^{18})$. Για αυτό και **δεν** τη χρησιμοποιούμε στην κρυπτογραφία. Το ίδιο ισχύει για τη συνάρτηση κατακερματισμού SHA-1 η οποία δίνει έξοδο 160-bits. Αν και δεν έχουμε πρακτική επίθεση για την τελευταία, οι ειδικοί συστήνουν να μην χρησιμοποιείται στην κρυπτογραφία, δείτε [1] και [3, 6].

Ένα διαφορετικό ερώτημα θα ήταν: πόσοι μαθητές χρειάζονται σε μια τάξη, ώστε να περιμένουμε κατά μέσο όρο δύο μαθητές να έχουν γενέθλια την ίδια ημέρα; Για να απαντήσουμε σε αυτό το ερώτημα, θα χρησιμοποιήσουμε χαρακτηριστικές τυχαίες μεταβλητές (indicator random variables). Έστω

$$X_{ij} = \begin{cases} 1, & \text{αν οι μαθητές } i, j \text{ έχουν γενέθλια την ίδια ημέρα} \\ 0, & \text{διαφορετικά} \end{cases}$$

⁴⁹<https://eprint.iacr.org/2013/170.pdf>

Αν b_i η ημερομηνία ο i μαθητής να έχει γενέθλια (π.χ. $b_i = 1$ αν γεννήθηκε την 1η Ιανουαρίου), τότε

$$Pr(b_i = b_j) = \sum_{r=1}^{n=365} Pr(b_i = r)Pr(b_j = r) = \sum_{r=1}^n 1/n^2 = 1/n.$$

Επομένως, $E[X_{ij}] = Pr(X_{ij}) = 1/n$. Χρησιμοποιήσαμε το παρακάτω.

Λήμμα 5.2.1. Έστω A ένα ενδεχόμενο του δειγματοχώρου Ω και

$$X_A = \begin{cases} 1, & \text{αν το } A \text{ συμβαίνει} \\ 0, & \text{διαφορετικά.} \end{cases}$$

Τότε $E[X_A] = Pr(A)$.

Απόδειξη.

$$E[X_A] = \sum_{Y \in \Omega} I(X_Y)Pr(Y) = 1 \cdot Pr(A) + 0 \cdot Pr(\bar{A}) = Pr(A).$$

□

Επίσης, υποθέτουμε ότι η τάξη έχει k μαθητές. Ας είναι $X = \sum_{i=1}^k \sum_{j=1}^k X_{ij}$ η τυχαία μεταβλητή που μετράει πόσα ζευγάρια μαθητών (i, j) έχουν γενέθλια την ίδια ημέρα. Τότε,

$$E[X] = \sum_{i=1}^k \sum_{j=1}^k E[X_{ij}] = \sum_{i=1}^k \sum_{j=1}^k \frac{1}{n} = \binom{k}{2} \frac{1}{n} = \frac{k(k-1)}{2n}.$$

Η απαίτηση του προβλήματος τώρα γράφεται $E[X] \geq 1$, επομένως $k = \sqrt{2n} + 1$ είναι ο ζητούμενος αριθμός των μαθητών.

Αν πάμε πίσω στις συναρτήσεις κατακερματισμού τώρα, αυτό μεταφράζεται ως εξής: αν θεωρήσω (π.χ. στην md5), τουλάχιστον $2^{64.5}$ τυχαία hashes, τότε αναμένουμε τουλάχιστον μία διένεξη. Δείτε, επίσης, την αναφορά [2]. Η υλοποίηση της προηγούμενης ιδέας χρειάζεται αρκετή μνήμη. Χρησιμοποιούμε τον κυκλικό αλγόριθμο του Floyd για την εύρεση διενέξεων, με μνήμη $O(1)$.

Άσκηση 5.4 Διαλέγουμε 20 άτομα στην τύχη. Ποια είναι η πιθανότητα τουλάχιστον δύο άτομα να έχουν γενέθλια την ίδια εβδομάδα;

Άσκηση 5.5 Διαλέγουμε n άτομα στην τύχη. Ποια είναι η πιθανότητα $\mathcal{P}$, τουλάχιστον m από αυτά ($m < n$) να έχουν γενέθλια την ίδια μέρα; Ποια είναι η μικρότερη τιμή n , ώστε $\mathcal{P} > 1/2$;

Άσκηση 5.6 Πόσα τουλάχιστον άτομα χρειάζονται, ώστε, όταν επιλέγουν τυχαία γράμματα από το αγγλικό αλφάβητο, να έχουν πιθανότητα ≈ 0.95 να διαλέξουν το ίδιο γράμμα;

Άσκηση 5.7 Θεωρήστε την hash function

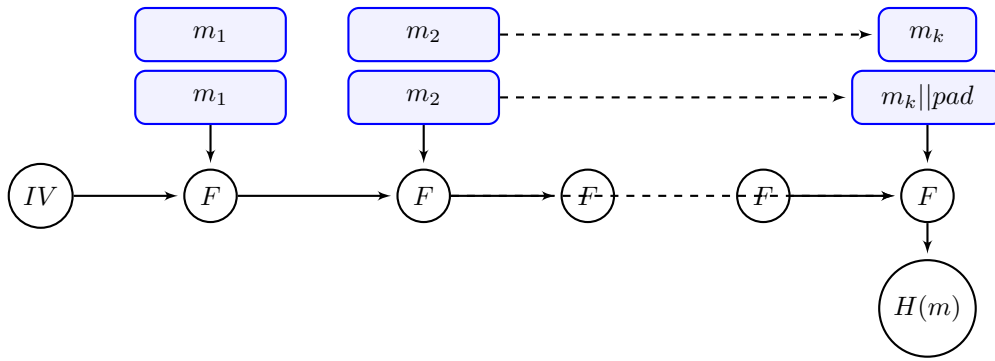
$$H(x) = x \bmod n,$$

για κάποιο n φυσικό αριθμό. Το αποτέλεσμα αυτής της συνάρτησης είναι ένας ακέραιος στο σύνολο $\{0, 1, 2, \dots, n-1\}$. Να εξετάσετε αν είναι collision free.

Άσκηση 5.8 Θεωρήστε την hash function

$$H(x_1, x_2) = x_1 \oplus x_2$$

όπου τα x_1, x_2 είναι δύο bytes. Να εξετάσετε αν είναι collision free.



Σχήμα 5.1: Σχήμα των Merkle-Damgård

Άσκηση 5.9 Έστω $x \in \{0, 1, \dots, 2^n - 1\}$ και $1 < m < n$. Θεωρήστε την hash function

$$H(x) = (2x^2 + x + 1) \bmod 2^n \bmod 2^m.$$

Το πρόβλημα της δεύτερης αντίστροφης εικόνας είναι το εξής: δοθέντος x , βρείτε $x' \neq x$ τέτοιο, ώστε $H(x) = H(x')$. Να εξετάσετε αν η προηγούμενη hash είναι ασφαλής στη δεύτερη αντίστροφη εικόνα.

Άσκηση 5.10 (Leftover Hash Lemma) Αρχικά εξοικειωθείτε με την εκφώνηση του Leftover Hash Lemma. Έχεις μια τυχαία πηγή X με ελάχιστη εντροπία $H_\infty(X) = 80$ και θέλεις να εξάγεις μια σχεδόν ομοιόμορφη ακολουθία τυχαίων bit χρησιμοποιώντας μια καθολική συνάρτηση κατακερματισμού H .

(i). Ποιο είναι το μέγιστο μήκος εξόδου m που μπορείς να εξάγεις διατηρώντας τη στατιστική απόσταση μικρότερη από $\varepsilon = 2^{-20}$;

(ii). Τι θα συμβεί αν προσπαθήσεις να εξάγεις 50 bits με την παρεχόμενη ελάχιστη εντροπία H_∞ ;

5.3 Σχήμα των Merkle-Damgård

Μια μέθοδος κατασκευής για συναρτήσεις κατακερματισμού δόθηκε από τους Merkle-Damgård (δείτε σχήμα 5.1). Βασίζεται στις συναρτήσεις συμπίεσης (ή συστολικές συναρτήσεις - compression functions). Μια συστολική συνάρτηση ορίζεται

$$F : X \times Y \rightarrow Y \text{ με } |Y| \ll |X|.$$

Έστω $X = \{0, 1\}^s$, $Y = \{0, 1\}^\ell$ με $\ell \ll s$. Οπότε, η κατασκευή συναρτήσεων κατακερματισμού μπορεί να αναχθεί σε κατασκευή συναρτήσεων συμπίεσης. Αν οι συναρτήσεις συμπίεσης δεν έχουν διενέξεις, μπορούμε να κατασκευάσουμε συναρτήσεις κατακερματισμού χωρίς διενέξεις.

Θεώρημα 5.3.1 Αν η συνάρτηση κατακερματισμού F δεν έχει διενέξεις, τότε και η συνάρτηση κατακερματισμού που προκύπτει από το σχήμα Merkle-Damgård δεν έχει διενέξεις.

Για να αποδειχθεί αυτό το θεώρημα, πρέπει να διαλέξουμε ένα padding που να ικανοποιεί τις παρακάτω ιδιότητες.

1. Το μήνυμα M είναι επίθημα (suffix) του $\text{Pad}(M)$.
2. Αν $|M_1| = |M_2|$, τότε $\text{Pad}(M_1) = \text{Pad}(M_2)$.
3. Αν $|M_1| \neq |M_2|$, τότε το τελευταίο block του $\text{Pad}(M_1)$ είναι διαφορετικό από το τελευταίο block του $\text{Pad}(M_2)$.

Τότε, κάθε διένεξη της συνάρτησης κατακερματισμού οδηγεί σε διένεξη της συνάρτησης συμπίεσης και αντίστροφα. Οι συναρτήσεις κατακερματισμού MD5, SHA1 και SHA2 βασίζονται σε αυτό το σχήμα.

Άσκηση 5.11 Να δείξετε ότι το αποτέλεσμα του θεωρήματος 5.3.1 δεν ισχύει, αν ως pad θεωρήσουμε μια σταθερή επέκταση του μηνύματος m_k .

5.4 Εφαρμογές

5.4.1 Το σύστημα αποθήκευσης κωδικών σε μηχανές linux

Μια άλλη βασική εφαρμογή είναι στην αποθήκευση κωδικών. Για παράδειγμα, στα συστήματα linux οι κωδικοί αποθηκεύονται στο αρχείο `/etc/shadow`. Αν έχουμε κατάλληλα δικαιώματα, μπορούμε να δούμε ότι το αρχείο περιέχει (μεταξύ άλλων) αλφαριθμητικά της μορφής

```
test:$6$ANrWqWm8$ALD1gQkIyB3/I7Zcqohd2t147EBHagFE2.GHFy.zP5eAHxHbuJjnCMLJvrWFqMo6
LZ5g5.5eMu61tebZ/djLM.:16496:0:99999:7:::
```

Με αυτήν τη μορφή αποθηκεύονται οι κωδικοί μας σε μια μηχανή linux. Η λέξη `test` είναι το όνομα του χρήστη. Το `$6$` σημαίνει ότι χρησιμοποιήθηκε η συνάρτηση κατακερματισμού SHA-512, ενώ η ακολουθία `ANrWqWm8` ονομάζεται `salt` και είναι ένα τυχαίος δεκαεξαδικός αριθμός. Η εφαρμογή της SHA-512 γίνεται (επαναληπτικά) στο `salt||password` και αυτό που προκύπτει είναι η δεκαεξαδική ακολουθία

```
ALD1gQkIyB3/I7Zcqohd2t147EBHagFE2.GHFy...djLM.
```

Με αυτόν τον τρόπο προστατεύονται οι κωδικοί μας⁵⁰. Κάποιος που θα είχε στην κατοχή του αυτό το αρχείο θα μπορούσε (αν είχε αρκετή υπομονή και οι κωδικοί ήταν αρκετά μικροί) να βρει τους κωδικούς όλων των χρηστών (και του admin). Αν μετρήσουμε τα δεκαεξαδικά ψηφία της εξόδου, θα διαπιστώσουμε ότι είναι ακριβώς 86 ($86 \times 4 = 344$ -bits). Η SHA-512 παράγει πάντα έξοδο 512 bits και από αυτά διαλέγουμε τα 344. Τέλος, η χρήση των συναρτήσεων κατακερματισμού γίνεται σε όλα τα συστήματα αυθεντικοποιημένης κρυπτογράφησης όπως στο SSL/TLS, IPSEC και SSH.

5.4.2 Blockchain και Bitcoin

*Crypto means Cryptography
not cryptocurrency.*

Το Blockchain⁵¹ είναι ένα ψηφιακό δημόσιο λογιστικό βιβλίο⁵², που περιέχει όλες τις συναλλαγές ενός ψηφιακού νομίσματος (π.χ. του Bitcoin), οι οποίες έχουν εκτελεστεί μέχρι αυτήν τη στιγμή. Το μέγεθός του αυξάνεται συνεχώς, διότι περίπου ανά 10 λεπτά οι εξορύκτες (miners) του ψηφιακού νομίσματος προσθέτουν νέα τμήματα στο ήδη υπάρχον λογιστικό βιβλίο (blockchain). Οι εξορύκτες είναι ένα κατανεμημένο δίκτυο υπολογιστών (pool) αφιερωμένο στο να επιλύει ένα πολύ συγκεκριμένο μαθηματικό πρόβλημα που απαιτεί αρκετή υπολογιστική ισχύ. Όταν λυθεί αυτό το πρόβλημα, τότε παράγεται ένα νέο block. Το 2015 το μέγεθος του blockchain ήταν περίπου 25GB. Οι υπόλοιποι χρήστες του Bitcoin επαληθεύουν αν το νέο τμήμα που προστέθηκε στο blockchain είναι σωστό, υπολογίζοντας την τιμή της SHA-256 στα προηγούμενα δεδομένα. Η επαλήθευση γίνεται πολύ γρήγορα, αλλά, όπως είπαμε, η εξόρυξη⁵³ είναι αρκετά δύσκολη. Η δυσκολία αυτή στο πρωτόκολλο του Bitcoin εκφράζεται από την απαίτηση στους εξορύκτες να βρουν έναν αριθμό (nonce) που να λύνει μία ανίσωση (της μορφής)

$$\text{sha256}(\text{sha256}(\text{previous_data}||\text{nonce})) < 0000\dots000xyz\dots \quad (5.4.1)$$

Για παράδειγμα, τον Φεβρουάριο του 2015, για να προστεθεί ένα νέο τμήμα στο blockchain, απαιτούνταν εργασία πολυπλοκότητας $2^{67.5}$ bits. Η δυσκολία (difficulty) αναπροσαρμόζεται (σύμφωνα με το πρωτόκολλο του Bitcoin) ανά 2016 τμήματα και δίνεται με το πλήθος των μηδενικών που υπάρχουν στο δεύτερο μέλος της (5.4.1). Οπότε, κάθε τμήμα, που προστίθεται στο ήδη υπάρχον blockchain, είναι το αποτέλεσμα μιας

⁵⁰Οι κωδικοί που χρησιμοποιούμε γενικά, πρέπει να είναι τουλάχιστον 9 χαρακτήρες. Ένας κωδικός με 8 χαρακτήρες μπορεί να βρεθεί σε μερικές ώρες.

⁵¹Ένα από τα πρώτα mail που έστειλε ο/η Satoshi Nakamoto, <http://satoshinakamoto.me/2009/01/12/bitcoin-list-bitcoin-v0-1-alpha-release-notes/#selection-107.5-107.56>

⁵²καθολικό

⁵³δηλαδή η λύση του μαθηματικού παζλ

υπολογιστικά δαπανηρές εργασίας και βασίζεται στη συνάρτηση κατακερματισμού SHA-256. Λέμε ότι το Bitcoin χρησιμοποιεί, για την παραγωγή ενός νέου τμήματος, την *απόδειξη-εργασίας* (proof-of-work) για την SHA-256. Επειδή ανά ταχτά χρονικά διαστήματα (περίπου 10 λεπτά) προστίθεται ένα τμήμα (block) από τους εξορύκτες, πρέπει να περιμένουμε τουλάχιστον 10 λεπτά για την επαλήθευση μίας συναλλαγής μας (για μεγάλα ποσά θα χρειαστούν περισσότερες επαληθεύσεις από μία). Κάποιος που θα είχε αρκετά μεγάλη υπολογιστική ισχύ (τουλάχιστον 51% της συνολικής υπολογιστικής ισχύος του δικτύου) θα μπορούσε να πάρει τον έλεγχο του δημόσιου λογιστικού βιβλίου (blockchain) και να καταθέσει στον λογαριασμό του Bitcoins που ήδη έχουν καταναλωθεί από τον ίδιο. Αυτή η επίθεση ονομάζεται 51% – *attack*. Αυτό είναι το πρόβλημα της *διπλής κατανάλωσης* ενός Bitcoin (double spending problem). Όπως είπαμε, το πρόβλημα αυτό αντιμετωπίζεται με την *απόδειξη εργασίας* της SHA-256 και τη δυσκολία που θέτει το πρωτόκολλο του Bitcoin.

Για να ξεκινήσει κάποιος να συναλλάσσεται με αυτό το νόμισμα, χρειάζεται ένα δημόσιο και ένα ιδιωτικό κλειδί. Το ζευγάρι αυτών των κλειδιών παράγεται από το πρωτόκολλο της ψηφιακής υπογραφής EC-DSA (Elliptic Curve DSA). Υπολογίζοντας την RIPEMD-160 (μια άλλη συνάρτηση κατακερματισμού) επί του δημόσιου κλειδιού, έχουμε τη διεύθυνσή μας, στην οποία μπορούν να μας στέλνουν Bitcoins. Όταν θέλουμε να στείλουμε ένα Bitcoin, χρειάζεται να γνωρίζουμε τη διεύθυνση του αποστολέα (δηλαδή το δημόσιο κλειδί του). Η συναλλαγή μας υπογράφεται με το ιδιωτικό μας κλειδί. Τα δημόσια κλειδιά των κατόχων Bitcoins υπάρχουν στην αλυσίδα blockchain, οπότε οποιοσδήποτε μπορεί να δει πόσα Bitcoins αντιστοιχούν σε ένα δημόσιο κλειδί. Όμως, δεν ξέρει ποιος είναι ο κάτοχος. Οπότε, το Bitcoin, μας παρέχει (σχετική) **ανωνυμία**. Επίσης, δεν χρειάζεται να εγγραφούμε σε κάποια εταιρεία, για να ανοίξουμε έναν λογαριασμό Bitcoin. Χρειαζόμαστε μόνο ένα ζευγάρι, δημόσιο και ιδιωτικό κλειδί. Αυτός θα είναι ο λογαριασμός μας.

Για παράδειγμα το Genesis Block στο blockchain του bitcoin είναι το παρακάτω,

```

1  "blocks": [
2      {
3          "bits": 486604799,
4          "block_index": 0,
5          "fee": 0,
6          "hash": "000000000019d6689c085ae1658...a6c172b3f1b60a8ce26f",
7          "height": 0,
8          "main_chain": true,
9          "mrkl_root": "4a5e1e4baab89f3a3251...673e2cc77ab2127b7afdeda33b",
10         "n_tx": 1,
11         "next_block": [
12             "00000000839a8e6886ab5951d76f411475428afc90947ee320161bbf18eb6048"
13         ],
14         "nonce": 2083236893,
15         "prev_block": "000000000000000000000000...00000000000000000000",
16         "size": 285,
17         "time": 1231006505,
18         "tx": [
19             {
20                 "block_height": 0,
21                 "block_index": 0,
22                 "double_spend": false,
23                 "fee": 0,
24                 "hash": "4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7
afdeda33b",
25                 "inputs": [
26                     {
27                         "index": 0,
28                         "prev_out": null,
29                         "script": "04ffff001d010445...722062616e6b73",
30                         "sequence": 4294967295,
31                         "witness": ""
32                     }
33                 ],
34                 "lock_time": 0,
35                 "out": [

```



```

36         {
37             "addr": "1A1zP1eP5QGefi2DMPTfTL5SLmv7DivfNa",
38             "n": 0,
39             "script": "4104678afdb0f...b6bf11d5fac",
40             "spending_outpoints": [],
41             "spent": false,
42             "tx_index": 2098408272645986,
43             "type": 0,
44             "value": 50000000000
45         }
46     ],
47     "relayed_by": "0.0.0.0",
48     "size": 204,
49     "time": 1231006505,
50     "tx_index": 2098408272645986,
51     "ver": 1,
52     "vin_sz": 1,
53     "vout_sz": 1,
54     "weight": 816
55 }
56 ],
57 "ver": 1,
58 "weight": 1140
59 }
60 ]

```

Παρατηρούμε ότι η χρονοσφραγίδα (timestamp) βεβαιώνει ότι το Genesis Block έχει παραχθεί (σε unix format) 1231006505 δηλαδή, *Σάββατο 03 Ιανουαρίου, 2009, 08:15:05 μ.μ. ΕΕΕ*. Το Block height είναι 0, διότι πριν από αυτό το block δεν υπάρχουν άλλα blocks. Αυτή η συναλλαγή έστειλε 50 BTC⁵⁴ στη διεύθυνση 1A1zP1eP5QGefi2DMPTfTL5SLmv7DivfNa. Σήμερα⁵⁵, η αμοιβή για την παραγωγή ενός νέου block είναι 6.25 BTC.

Τέλος, το σύστημα αυτό είναι αποκεντρωμένο⁵⁶, δεν χρειάζεται κάποια τρίτη οντότητα να μεσολαβήσει, για να γίνει η συναλλαγή (όπως συμβαίνει με το συμβατικό νόμισμα που χρησιμοποιούμε ή το σύστημα των πιστωτικών καρτών). Χρειάζεται, όμως, να εμπιστευόμαστε το δίκτυο του Bitcoin, αλλά όχι απαραίτητα αυτόν με τον οποίο συναλλασσόμαστε. Δηλαδή, το σύστημα είναι **αποκεντρωμένο** (decentralized). Η ιδέα του δημόσιου ψηφιακού λογιστικού βιβλίου έχει και άλλες εφαρμογές εκτός της συναλλαγής με ψηφιακά νομίσματα.

Άσκηση 5.12 Μπορείτε να δείτε την τρέχουσα τιμή του bitcoin κάνοντας χρήση του κώδικα:

```

1 # pip3 install cryptocompare
2 import cryptocompare
3 price = cryptocompare.get_price('BTC', 'EUR')
4 print(price)

```

Ενώ ο παρακάτω κώδικας:

```

1 #pip3 install datetime
2 import datetime
3 date = datetime.date.today()
4 cryptocompare.get_historical_price('BTC', 'EUR', datetime.datetime(2017,6,6))

```

μας εκτυπώνει παρελθοντικές τιμές του bitcoin.

(i). Σχεδιάστε με τη matplotlib την εξέλιξη των τιμών του bitcoin το 2020.

(ii). Κατόπιν, με χρήση του παρακάτω κώδικα

⁵⁴παρόλα αυτά, αυτή η συναλλαγή ουδέποτε εισήχθη στις συναλλαγές του blockchain

⁵⁵2021

⁵⁶υπάρχουν περίπου 15 pools, οπότε το κατά πόσο είναι αποκεντρωμένο το bitcoin είναι ένα σχετικό ερώτημα

```

1 # pip3 install numpy
2 # pip3 install pandas
3 import numpy as np
4 import pandas as pd
5 import cryptocompare as cc
6 coin_list = cc.get_coin_list()
7 coins = sorted(list(coin_list.keys()))
8 coins

```

επαναλάβετε ό,τι και προηγουμένως, αλλά για κάποιο διαφορετικό κρυπτονόμισμα.

Άσκηση 5.13 Βρείτε στη βιβλιογραφία δύο εφαρμογές του blockchain εκτός των κρυπτονομισμάτων.

Άσκηση 5.14 Ο παρακάτω κώδικας επιστρέφει τη χρονοσφραγίδα του block με αριθμό x , όπου ξεκινάμε από το $x = 0$ το οποίο αντιπροσωπεύει το Genesis Block.

```

1 def print_bitcoin_time_block(x):
2     from datetime import datetime
3     import requests
4     r = requests.get('https://blockchain.info/block-height/' + str(x) + '?format=json')
5     data = r.json()
6     time = data['blocks'][0]['time']
7     return datetime.utcfromtimestamp(time).strftime('%Y-%m-%d %H:%M:%S')

```

Βρείτε τις ημερομηνίες που παρήχθησαν τα πρώτα πέντε blocks. Γνωρίζουμε ότι τα blocks πρέπει να παράγονται μέσα σε ένα δεκάλεπτο. Τι παρατηρείτε;

Άσκηση 5.15 Ο παρακάτω κώδικας για να δουλέψει χρησιμοποιεί το api του etherscan⁵⁷. Σε αυτήν την άσκηση θα αντλήσουμε στοιχεία από το Ethereum blockchain.

```

1 def print_ethereum_block(x):
2     import requests
3     r = requests.get('https://api.etherscan.io/api?module=block&action=getblockreward&
4     blockno='+str(x)+'&apikey=your_api_key')
5     data = r.json()
6     return data

```

Βρείτε την ημερομηνία που έχει παραχθεί το genesis block⁵⁸.

5.5 Κώδικες αυθεντικοποίησης: MAC

Οι συμμετρικοί αλγόριθμοι, όπως είδαμε, χρησιμοποιούνται για την *εμπιστευτικότητα* ενός μηνύματος. Δηλαδή, αν η Εύα κλέψει το μήνυμά μας, δεν θα καταλάβει τι περιέχει. Μπορεί, όμως, να αλλάξει μερικά bits. Επομένως, οι συμμετρικοί αλγόριθμοι δεν μας προστατεύουν από την πιθανή αλλαγή του κρυπτομηνύματος. Δηλαδή, δεν μας παρέχουν *ακεραιότητα*.

Οι συναρτήσεις MAC: **M**essage **A**uthentication **C**odes χρησιμοποιούνται, για να πετύχουμε *ακεραιότητα* και *αυθεντικοποίηση* ενός μηνύματος. Η ακεραιότητα μας βεβαιώνει ότι το μήνυμα δεν άλλαξε κατά τη μεταφορά και η αυθεντικοποίηση μας βεβαιώνει ότι η πηγή που το έστειλε είναι η πραγματική. Μερικές φορές ονομάζουμε τις MAC συναρτήσεις, συναρτήσεις κατακερματισμού με κλειδί. Ο όρος αυτός είναι εν μέρει σωστός, διότι οι MAC μπορούν να κατασκευαστούν και χωρίς να χρησιμοποιήσουμε συναρτήσεις κατακερματισμού. Το αποτέλεσμα μιας συνάρτησης MAC ονομάζεται ετικέτα (tag). Επομένως,

$$MAC : \mathcal{K} \times \mathcal{M} \rightarrow \text{tag space}.$$

⁵⁷<https://etherscan.io/>

⁵⁸Για άλλες μεθόδους δείτε <https://docs.etherscan.io/api-endpoints/>

Όπου $\mathcal{K}$ είναι ο χώρος των κλειδιών και $\mathcal{M}$ ο χώρος των μηνυμάτων. Αυτό που **δεν** μας παρέχουν οι MAC είναι η ιδιότητα της “αδυναμία αποκήρυξης” (non-repudiation). Δηλαδή, αν η Alice μας στείλει την $MAC(k, m)$ (όπου k το μυστικό κλειδί που μοιραζόμαστε), τότε **δεν** μπορεί να αποδείξει (μαθηματικά και όχι νομικά!) ότι έστειλε το μήνυμα. Με τον όρο αδυναμία αποκήρυξης εννοούμε εκείνη την υπηρεσία που μας παρέχει απόδειξη ότι ένα μήνυμα προήλθε από συγκεκριμένο αποστολέα και ότι το μήνυμα είναι αυθεντικό (δεν έχει αλλάξει κατά την αποστολή του). Ο όρος αδυναμίας αποκήρυξης είναι νομικός όρος και υπάρχει έντονη συζήτηση κατά πόσο μπορεί να επιτευχθεί με τη χρήση της κρυπτογραφίας (π.χ. με ψηφιακές υπογραφές). Ας δούμε ένα παράδειγμα, για να καταλάβουμε τη διαφορά αυθεντικοποίησης και αδυναμίας αποκήρυξης. Στο πρωτόκολλο SSL/TLS έχουμε αυθεντικοποιημένη κρυπτογράφηση. Δηλαδή, ο χρήστης, π.χ. η Alice, που συνδέεται με έναν διακομιστή είναι σίγουρη ότι συνδέθηκε με τον σωστό διακομιστή, έστω X , και όχι με κάποιον άλλον. Ωστόσο, η Alice μπορεί να καταγράψει το αρχείο επικοινωνίας τους και να το παρουσιάσει ως νομικό στοιχείο, ότι δηλαδή πράγματι επικοινωνήσε με τον διακομιστή X , διότι εύκολα μπορεί να κατασκευάσει ένα, χωρίς να έχει επικοινωνήσει. Άρα, το SSL/TLS δεν παρέχει (κρυπτογραφική) αδυναμία αποκήρυξης. Επομένως, από εδώ και πέρα, όταν χρησιμοποιούμε τον όρο αδυναμία αποκήρυξης, θα εννοούμε κρυπτογραφική αδυναμία αποκήρυξης και όχι νομική αδυναμία αποκήρυξης⁵⁹. Θα δούμε ότι οι ψηφιακές υπογραφές παρέχουν την υπηρεσία της αδυναμίας αποκήρυξης, διότι η Alice για παράδειγμα, δεν μπορεί να αρνηθεί ότι ένα μήνυμά της υπογράφηκε με το ιδιωτικό κλειδί της. Ωστόσο, μπορεί να ισχυριστεί ότι το ιδιωτικό της κλειδί εκλάπη.

Οι συναρτήσεις MAC μπορούν να κατασκευαστούν είτε από συμμετρικούς αλγόριθμους τμήματος είτε από συναρτήσεις κατακερματισμού. Στην πρώτη περίπτωση λέμε ότι έχουμε μία CMAC: Cipher MAC, ενώ στη δεύτερη περίπτωση λέμε ότι έχουμε μια HMAC: Hash MAC. Το πρώτο παράδειγμα HMAC δόθηκε από τους Mihir Bellare, Ran Canetti και Hugo Krawczyk⁶⁰ και βασίζεται στον τύπο

$$HMAC(m, k) = \text{hash}((k \oplus \text{opad}) || (\text{hash}(k \oplus \text{ipad}) || m),$$

όπου k το κλειδί, m το μήνυμα, hash η συνάρτηση κατακερματισμού που χρησιμοποιούμε, opad η δεκαεξαδική λέξη μήκους όσο και το μήκος του κλειδιού

$$0 \times 5c5c5c...$$

και τέλος ipad η δεκαεξαδική λέξη

$$0 \times 363636...,$$

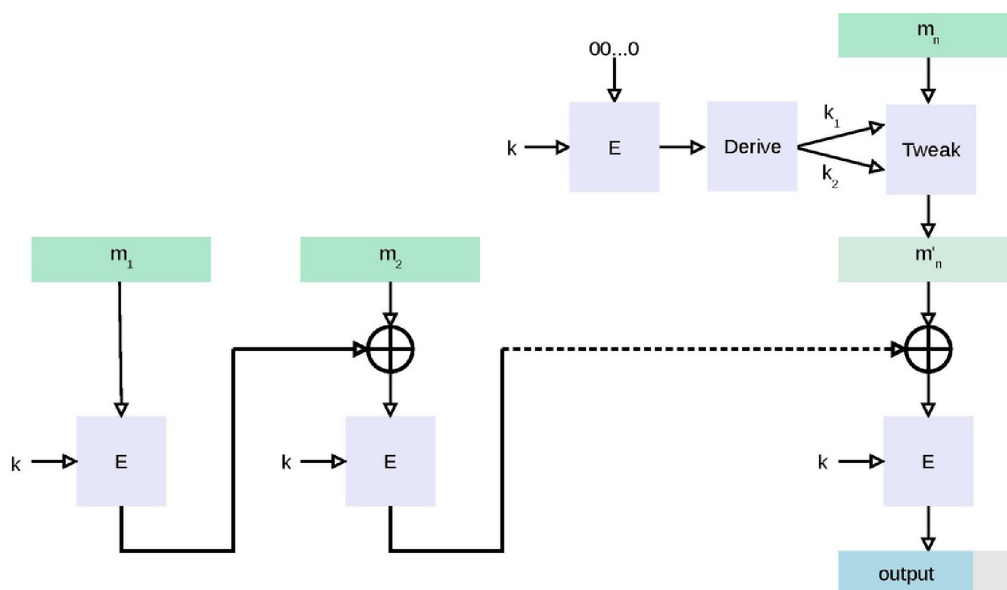
μήκους όσο και το μήκος του κλειδιού επίσης. Τα ipad και opad δεν είναι τυχαία επιλεγμένα. Π.χ. αν $\text{ipad} \approx \text{opad}$ τότε υπάρχει επίθεση⁶¹ (δηλαδή το μήνυμα m μπορεί να πλαστογραφηθεί).

Για την περίπτωση των CMAC χρησιμοποιούμε την κατασκευή που δόθηκε από τους Black και Rogaway (δείτε σχήμα 5.2).

⁵⁹δείτε, <http://world.std.com/~cme/non-repudiation.htm>, ανάκτηση 21-10-2015.

⁶⁰<https://cseweb.ucsd.edu/~mihir/papers/kmd5.pdf>

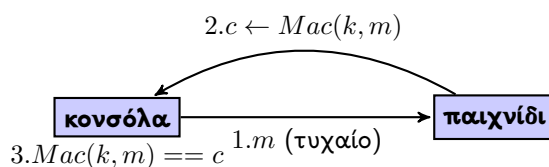
⁶¹<https://eprint.iacr.org/2012/684.pdf>



Σχήμα 5.2: CMAC

Εικόνα από wikipedia commons, άδεια: CC BY-SA 3.0.
https://en.wikipedia.org/wiki/File:CMAC_-_Cipher-based_Message_Authentication_Code.pdf

Οι συναρτήσεις MAC χρησιμοποιήθηκαν από τη γνωστή εταιρεία NINTENDO, για να ελέγχει την παραγωγή παιχνιδιών για τις κονσόλες της. Όλοι οι κατασκευαστές παιχνιδιών, που προορίζονταν για Nintendo, έπρεπε να έχουν ενσωματωμένο ένα τσιπάκι (10NES chip), το οποίο είχε αποθηκευμένο ένα κλειδί k . Το κλειδί αυτό, επίσης, υπήρχε στις κονσόλες (στο ίδιο τσιπάκι). Οπότε, κάθε φορά που εισερχόταν ένα παιχνίδι στην κονσόλα, γινόταν η παρακάτω επικοινωνία (σχήμα 5.3). Συνεπώς οι κατασκευαστές παιχνιδιών έπρεπε να αγοράζουν το τσιπάκι από την κεντρική εταιρεία, διότι μόνο αυτή κατείχε το σωστό κλειδί.



Σχήμα 5.3: Κονσόλες που ελέγχουν το λογισμικό με MAC

Στο προηγούμενο σχήμα, στο βήμα 3, γίνεται έλεγχος αν $Mac(k, m) = c$. Αν το αποτέλεσμα είναι true, τότε η κονσόλα αποδέχεται το παιχνίδι (κώδικας) ως γνήσιο. Στις κονσόλες Nintendo το παιχνίδι αποτελούνταν και από hardware που ήταν αποθηκευμένο το κλειδί, καθώς και η δυνατότητα υπολογισμού της MAC.

Σήμερα, οι εταιρείες χρησιμοποιούν ψηφιακές υπογραφές, π.χ. η SONY, για την υπογραφή του κώδικά τους (δηλαδή το παιχνίδι). Βέβαια, όπως θα δούμε, μερικές φορές οι υλοποιήσεις τους είναι λανθασμένες.

Τα μαθηματικά της κρυπτογραφίας δουλεύουν. Συνήθως τα προβλήματα παρουσιάζονται στις υλοποιήσεις.

Βιβλιογραφία

- [1] Burmester, M.; Γκρίτζαλης, Σ.; Κάτσικας, Σ.; Χρυσικόπουλος, Β. *Σύγχρονη Κρυπτογραφία. Θεωρία και Εφαρμογές*. Εκδ. Παπασωτηρίου, 2011.
- [2] Cormen, T. H.; Leiserson C. E.; Rivest E. L.; Stein C. *Introduction To Algorithms*, (Third edition), MIT Press, 2009.
- [3] Delfs, H.; Knebl, H. *Introduction to Cryptography. Principles and Applications*. Information Security and Cryptography (Texts and Monographs). Springer, 2007.
- [4] Joux, A. *Algorithmic Cryptanalysis*. Chapman & Hall/CRC, 2009.
- [5] Menezes, A.; van Oorschot, P.; Vanstone, S. *Handbook of Applied Cryptography*, CRC Press. <https://cacr.uwaterloo.ca/hac/>, 1996.
- [6] Mollin, R. *An Introduction to Cryptography*. 2nd edition, Series : Discrete mathematics and its applications. Taylor and Francis Group, 2007.
- [7] Schneier, B. *Applied Cryptography, Protocols, Algorithms, and Source Code in C (cloth)*. 2nd Edition, Wiley, 1996.
- [8] Μπερτσεκάς, Π. Δ.; Τσιτσικλής, Ν. Γ. *Εισαγωγή στις πιθανότητες*. Εκδ. Τζιόλας, 2010.

Κεφάλαιο 6

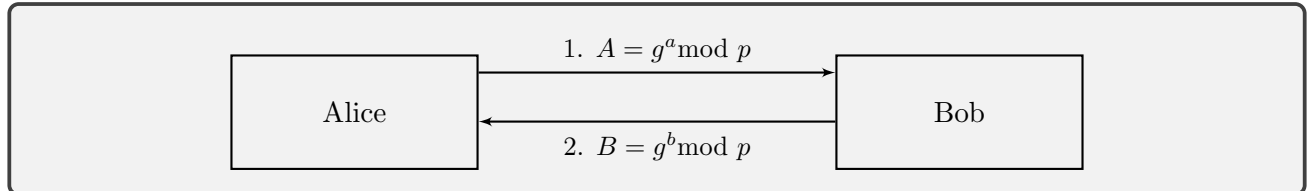
Το πρόβλημα ανταλλαγής κλειδιού

Σύνοψη - Περίληψη. Σε αυτό το κεφάλαιο παρουσιάζουμε το πρωτόκολλο ανταλλαγής των Diffie-Hellman.

Προαπαιτούμενη γνώση. Για αυτό το κεφάλαιο χρειάζεται απλά να γνωρίζετε τι είναι modulo και πρώτος αριθμός. Για τις ασκήσεις θα χρειαστεί να υλοποιήσετε τον αλγόριθμο της υποενότητας (7.2.3). Συνήθως αυτό το πρωτόκολλο παρουσιάζεται σε όλα τα εισαγωγικά βιβλία κρυπτογραφίας, μπορείτε να δείτε [1, 3, 5, 6] και για την ελληνόγλωσση βιβλιογραφία [2, 7, 8, 9].

6.1 Το Πρωτόκολλο Diffie-Hellman

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 τη μεταξύ τους επικοινωνία, αλλά δεν μπορεί να εισάγει δεδομένα στη γραμμή επικοινωνίας. Η Alice και ο Bob συμφωνούν σε έναν πρώτο αριθμό p και έναν αριθμό g από το σύνολο $\{1, \dots, p-1\}$. Κατόπιν, η Alice διαλέγει έναν τυχαίο αριθμό a από το σύνολο $\{1, \dots, p-1\}$ και ο Bob διαλέγει τον b από το ίδιο σύνολο. Οι αριθμοί a, b που επιλέξαντε 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 υποενότητα 7.2.3. 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* [4]. Όπως είδαμε, η Alice και ο Bob μπορούν να καταλήξουν γρήγορα στο ίδιο κλειδί (όπως θα δούμε στις παρακάτω 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 τη 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:

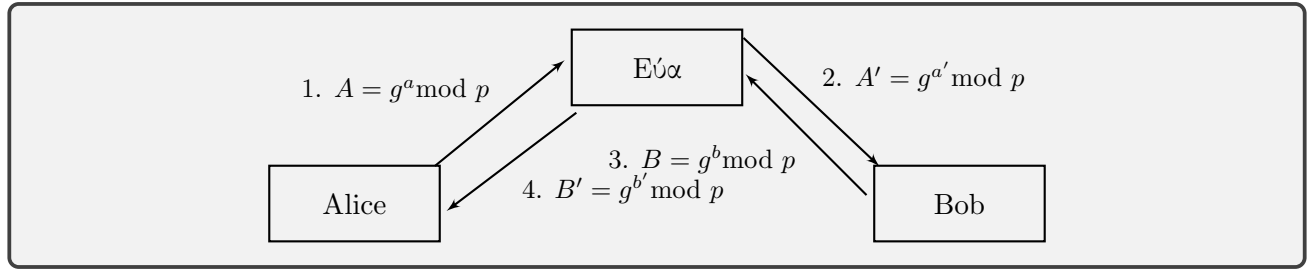
$$\exp \left(\left(\sqrt[3]{\frac{64}{9}} + o(1) \right) (\ln p)^{1/3} (\ln \ln p)^{2/3} \right). \quad (6.1.1)$$

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

Αν το σώμα που λύνουμε το πρόβλημα του διακριτού λογαρίθμου είναι της μορφής $GF(2^k)$, τότε η πολυπλοκότητα είναι σχεδόν πολυωνυμική (quasi polynomial), δηλαδή,

$$2^{O((\ln k)^2)}.$$

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 τη 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 στη 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 στη 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 αυτήν την επίθεση, είναι η χρήση των ψηφιακών πιστοποιητικών. Το πρωτόκολλο αυτό μαζί με τη χρήση ψηφιακών πιστοποιητικών (για αυθεντικοποίηση) χρησιμοποιείται στο πρωτόκολλο SSL/TLS, που έχει ευρύτατη χρήση στο ηλεκτρονικό εμπόριο και στις 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 fields).

6.2 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 για το οποίο έχει συμφωνήσει με 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 :

$$(g^r, E_{sym}(K, m) = c),$$

για κάποιο συμμετρικό σύστημα (E_{sym}, D_{sym}) . Η Alice υπολογίζει τον αριθμό:

$$(g^r)^x = g^{rx} = g^{xr} = K.$$

Τέλος, υπολογίζει:

$$D_{sym}(K, c) = m.$$

Παρατήρηση 6.2.1. Όσον αφορά τον πρώτο αριθμό p αυτός πρέπει να είναι safe, δηλαδή ο $(p-1)/2$ να είναι πρώτος. Ο λόγος είναι ότι αν ο $p-1$ έχει ανάλυση σε πρώτους που δεν είναι μεγάλοι, αυτό μπορεί να βελτιώσει επιθέσεις ωμής βίας. Για παράδειγμα μπορεί να εφαρμοστεί ο αλγόριθμος Pohlig-Hellman (Παρατήρηση 10.2.3). Αποδεικνύεται ότι αν

$$p-1 = q_1^{e_1} q_2^{e_2} \cdots q_r^{e_r},$$

τότε η εύρεση του διακριτού λογάριθμου στην ομάδα $\mathbb{Z}_p^*$ κοστίζει

$$O\left(\sum_{i=1}^r (e_i(\ln p + \sqrt{p_i}))\right).$$

6.3 Άλλα ζητήματα ασφάλειας

Έστω n το δυαδικό μήκος του πρώτου αριθμού p και $k = \lceil \sqrt{n} \rceil + \lceil \log_2 n \rceil$. Επίσης, ας είναι $g^{ab} \bmod p$ το κοινό κλειδί. Έχει αποδειχθεί⁶² ότι, αν έχουμε στην κατοχή μας έναν αποδοτικό αλγόριθμο που με είσοδο $(g^x \bmod p, g^y \bmod p, g^{xy} \bmod p)$ εξάγει τα k πιο σημαντικά bits της ποσότητας $g^{xy} \bmod p$, τότε μπορούμε να υπολογίσουμε το κοινό κλειδί $g^{ab} \bmod p$. Π.χ. αν ο πρώτος p έχει 1024 bits, τότε, αν η Eve έχει έναν αποδοτικό αλγόριθμο, που εξάγει $k = 42$ bits, μπορεί σε πολυωνυμικό χρόνο να υπολογίσει το κλειδί $g^{ab} \bmod p$. Δηλαδή, ο υπολογισμός της συνάρτησης DH_p για p 1024 bits, είναι τόσο δύσκολος όσο και ο υπολογισμός των 42 πιο σημαντικών bits του κοινού κλειδιού.

Άσκηση 6.1 Εφαρμόστε το πρωτόκολλο ανταλλαγής Diffie-Hellman για $g = 3$, $p = 101$, $a = 77$, $b = 91$. Δηλαδή, πρέπει να υπολογίσετε το κοινό κλειδί. Θα χρειαστεί να υλοποιήσετε τον αλγόριθμο της υποενότητας (7.2.3).

Άσκηση 6.2 Ο Ανδρέας θέλει να χρησιμοποιήσει το πρωτόκολλο Diffie-Hellman, για να καταλήξει σε ένα κοινό κλειδί με τη Σοφία. Ο Ανδρέας δημοσιεύει τον πρώτο αριθμό $p = 197$ και τη βάση $g = 13$ καθώς και το $g^b = 86$ για κάποιο $1 < b < p$. Η Σοφία διαλέγει τον μυστικό εκθέτη $a = 31$. Ποιο είναι το μυστικό κλειδί;

Άσκηση 6.3 Σας δίνεται ότι το $g^{ab} \bmod p = 1$ και $g = 13$, $p = 677$. Υπολογίστε τα υποψήφια a, b . Θα χρειαστεί να υλοποιήσετε τον αλγόριθμο της υποενότητας (7.2.3).

Άσκηση 6.4 Έστω $p = 101$. Βρείτε όλα τα στοιχεία g τέτοια, ώστε $2 \leq g \leq p - 2$ που έχουν την εξής ιδιότητα: Οι δυνάμεις $\{g^i \bmod p : i = 0, 1, 2, \dots, p - 1\}$ παράγουν όλα τα στοιχεία του συνόλου $\mathbb{Z}_{101}^* = \{1, 2, \dots, p - 1\}$. Δηλαδή, ζητάμε να βρείτε τους γεννήτορες της ομάδας $\mathbb{Z}_{101}^*$. Θα χρειαστεί να υλοποιήσετε κάποιο πρόγραμμα στον υπολογιστή σας.

Άσκηση 6.5 Για έναν πρώτο αριθμό p , 1024 bits, υπολογίστε ως δύναμη του 2, κατά προσέγγιση, την τιμή της ποσότητας (6.1.1).

Άσκηση 6.6 Θεωρήστε τον πρώτο $p = 1013$ και $g = 3$. Σχεδιάστε τα σημεία (x, g^x) για 200 τυχαίες τιμές του x από το σύνολο $\{2, 3, \dots, p - 2\}$. Υπάρχει κάποια κανονικότητα στο σχήμα που προκύπτει ή φαίνεται να είναι τυχαία τα σημεία; Αναζητήστε στη βιβλιογραφία κάποιο θεωρητικό αποτέλεσμα που να υποστηρίζει αυτό που πιστεύετε.

⁶²Boneh, Venkatesan, Hardness of Computing the Most Significant Bits of Secret Keys in Diffie-Hellman and Related Scheme, 1996.

Βιβλιογραφία

- [1] Boneh, Dan; Shoup, V. *A Graduate Course in Applied Cryptography*, Version 0.5, Jan. 2020. <http://toc.cryptobook.us/>.
- [2] Burmester, M.; Γκρίτζαλης, Σ.; Κάτσικας, Σ.; Χρυσικόπουλος, Β. *Σύγχρονη Κρυπτογραφία. Θεωρία και Εφαρμογές*. Εκδ. Παπασωτηρίου, 2011.
- [3] Cozzens, M.; Miller, S. J. *The Mathematics of Encryption-An Elementary Introduction*. Mathematical World, 1st edition, vol. 29, 2013. AMS.
- [4] Diffie, W.; Hellman, M. *New Directions in Cryptography*. IEEE Transactions on Information Theory, vol. **22**, no.6. 1976.
- [5] Joux, A. *Algorithmic Cryptanalysis*. Chapman & Hall/CRC, 2009.
- [6] Menezes, A.; van Oorschot, P.; Vanstone, S. *Handbook of Applied Cryptography*, CRC Press. <https://cacr.uwaterloo.ca/hac/>, 1996.
- [7] Κάτος, Β.; Στεφανίδης, Γ. *Τεχνικές Κρυπτογραφίας και Κρυπτανάλυσης*, Εκδ. Ζυγός, 2003.
- [8] Πουλάκης, Δ. *Κρυπτογραφία*. Εκδ. Ζήτη, 2009.
- [9] Ζάχος, Ε.; Παγουρτζής, Α.; Γροντάς, Π. *Υπολογιστική Κρυπτογραφία*, Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. <http://dx.doi.org/10.57713/kallipos-492>, 2015.

Κεφάλαιο 7

Πολυπλοκότητα & Μηχανές Turing

Σύνοψη - Περίληψη. Σε αυτό το κεφάλαιο παρουσιάζουμε τις βασικές έννοιες της θεωρίας πολυπλοκότητας. Ξεκινάμε με τον ορισμό της μηχανής Turing και κατόπιν δίνουμε τους ορισμούς των βασικών κλάσεων πολυπλοκότητας P, NP, NP-complete, PSPACE κ.ά. Επίσης, διατυπώνουμε ένα βασικό πρόβλημα που έχει μεγάλο ενδιαφέρον στην κρυπτογραφία: το πρόβλημα της παραγοντοποίησης. Όπως θα δούμε, αυτό το πρόβλημα ανήκει στην τομή των κλάσεων P και NP. Στο κλασικό μοντέλο υπολογισμού, δηλαδή στη μηχανή Turing, πιστεύουμε ότι το πρόβλημα είναι δύσκολο κατά μέσο όρο. Αν όμως έχουμε στην κατοχή μας έναν κβαντικό υπολογιστή, με αρκετά μεγάλη μνήμη, τότε ο αλγόριθμος του Shor μπορεί να λύσει το πρόβλημα της παραγοντοποίησης σε πολυωνυμικό χρόνο.

Προαπαιτούμενη γνώση. Για αυτό το κεφάλαιο χρειάζονται κάποιες βασικές γνώσεις ανάλυσης. Ειδικότερα, είναι απαραίτητη η γνώση των συμβόλων του Landau. Επίσης, χρειάζονται βασικές γνώσεις αλγόριθμων και θεωρίας γραφημάτων. Ένα εισαγωγικό βιβλίο για τις μηχανές Turing είναι [8, Κεφ. 6]. Για τα γραφήματα μπορείτε να δείτε [5, Κεφ. 6], [9, Κεφ. 5] και [2, Κεφ. 11]. Για θεωρία πολυπλοκότητας [2, Κεφ. 10].

7.1 Εισαγωγή

Για να μελετήσουμε το πρωτόκολλο Diffie-Hellman, το σύστημα RSA, την ψηφιακή υπογραφή DSA και άλλα συστήματα, πρέπει να έχουμε κάποιο βασικό μαθηματικό υπόβαθρο το οποίο αποτελείται κυρίως από θέματα της θεωρίας αριθμών: πρώτοι αριθμοί, ισотиμίες, πεπερασμένα σώματα, διακριτό λογάριθμο και άλλα. Για μια εκτενή ανάλυση της θεωρίας αριθμών προτείνουμε από την ελληνόγλωσση βιβλιογραφία [7, 4, 3, 6]. Αλλά αυτά θα τα δούμε πιο αναλυτικά στο επόμενο κεφάλαιο. Σε αυτό το κεφάλαιο θα παρουσιάσουμε βασικές έννοιες της θεωρίας πολυπλοκότητας που θα χρειαστούμε στην ανάλυση αλγορίθμων. Για την ανάλυση των αλγορίθμων μπορείτε να συμβουλευτείτε το κλασικό βιβλίο του Donald Knuth [1]. Επομένως, θα χρειαστούμε κάποιους ορισμούς όσον αφορά την πολυπλοκότητα των αλγορίθμων.

7.2 Πολυπλοκότητα

Πολυπλοκότητα (complexity) θα ονομάζουμε το μέτρο της ποσότητας του χρόνου (ταχύτητα) ή/και του χώρου (μνήμη) που απαιτείται από έναν αλγόριθμο να τερματίσει, 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)⁶³. Στην κρυπτολογία παίζει μεγάλο ρόλο η θεωρία πολυπλοκότητας. Τα πρώτα κρυπτοσυστήματα δημόσιου κλειδιού βασίζονταν σε προβλήματα NP-πλήρη (NP: Non deterministic Polynomial), όπως για παράδειγμα το πρόβλημα του γυλιού (subset sum problem).

Πρόβλημα του γυλιού - πρόβλημα απόφασης (NP-πλήρες)

Δίνονται ένα σύνολο A από n θετικούς ακераίους και ένας θετικός ακέραιος s . Υπάρχει υποσύνολο του A που έχει άθροισμα s ;

Πρόβλημα του γυλιού (NP-hard)

Δίνονται ένα σύνολο A από n θετικούς ακεραίους και ένας θετικός ακέραιος s . Βρείτε (εφόσον υπάρχει) υποσύνολο του A 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). Ωστόσο, 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 πολυωνυμικός χρόνος 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 τη μυστική πληροφορία. Η κλάση των NP-hard προβλημάτων δεν είναι καλή για την κρυπτογραφία, διότι αυτά τα προβλήματα είναι δύσκολα στη χειρότερη περίπτωση. Αλλά ακόμη και αν υπήρχαν προβλήματα σε αυτήν την κλάση, που ήταν δύσκολα κατά μέσο όρο (hard on average), τότε θα έπρεπε αυτά να είναι εύκολα αν είχαμε στη διάθεσή μας μια βοηθητική πληροφορία. Διαφορετικά, δηλαδή αν δεν υπήρχε η βοηθητική πληροφορία, τότε και ο νόμιμος χρήστης δεν θα μπορούσε να αποκρυπτογραφεί γρήγορα. Επομένως, τα ασφαλή κρυπτοσυστήματα (δημόσιου κλειδιού) πρέπει να έχουν τις εξής δύο ιδιότητες:

(i). Είναι εύκολο να λύνω αυτά τα προβλήματα, αν γνωρίζω μία μυστική πληροφορία,

⁶³<https://turingmaschine.klickagent.ch/einband>. Δίνεται μία προσομοίωση της μηχανής Turing.

(ii). Είναι δύσκολο (κατά μέσο όρο) να λύνω αυτά τα προβλήματα, όταν δεν γνωρίζω τη 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 κατασκευή του RSA.

Το πρόβλημα της χρήσης προβλημάτων, που είναι δύσκολα στη χειρότερη περίπτωση, επανήλθε από τον Miklos Ajtai, ο οποίος πρότεινε χρήση προβλημάτων που είναι δύσκολα στη χειρότερη περίπτωση, κατασκευάζοντας, όμως, μια αναγωγή 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 τη 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.

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

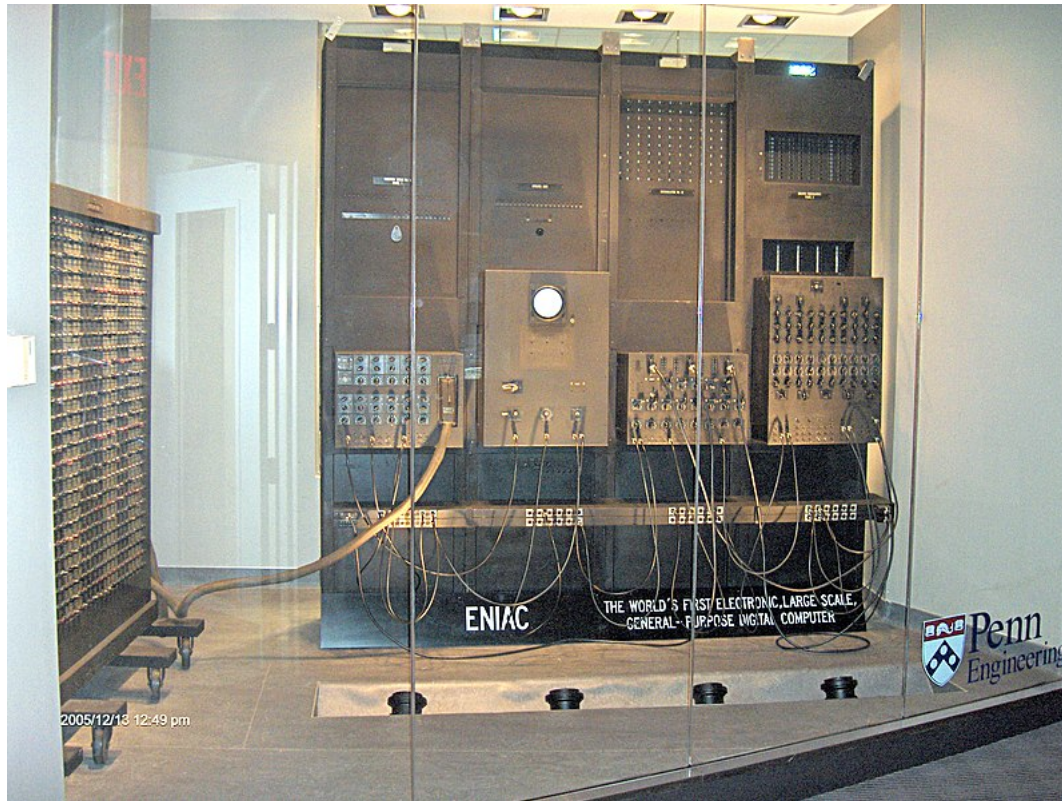
Το παραπάνω σύνολο είναι το software μίας μηχανής Turing. Το Hardware αποτελείται από μία ταινία άπειρου μήκους από δεξιά, η οποία είναι χωρισμένη σε κελιά, όπου κάθε κελί μπορεί να περιέχει ένα στοιχείο από το αλφάβητο S . Επίσης, περιέχει μία κεφαλή που κινείται κατά μήκος της ταινίας (ή ισοδύναμα 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 τη μετάβαση της μηχανής Turing (MT) στη 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 τη 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 Shannon, που απέδειξε ότι τα (relay) circuits και η άλγεβρα του Boole (Boolean algebra) είναι ισοδύναμα ⁶⁴ (π.χ. ο ENIAC (1943-1945) ήταν δεκαδικός υπολογιστής (decimal computer) και όχι δυαδικός (digital computer)) ⁶⁵.



Σχήμα 7.1: Η μηχανή ENIAC.

Άδεια CC BY-SA 3.0

https://commons.wikimedia.org/wiki/File:ENIAC_Penn1.jpg.

Η κατάσταση a.

Περιγράφεται από τις επόμενες τρεις τετράδες: $(\sqcup, \text{Print } \sqcup, \Delta_p = -1, b)$. Δηλαδή, αν η κεφαλή είναι πάνω σε ένα κενό κελί, τότε άφησε το κελί κενό ($\text{Print } \sqcup$). Κατόπιν, μετακίνησε την κεφαλή αριστερά και τέλος πήγαινε στην κατάσταση *b*. Η συνάρτηση μετάβασης τότε δίνεται από τον τύπο $\delta(a, \sqcup) = (b, \sqcup, -1)$. Οι άλλες δύο τετράδες που περιγράφουν τη συνάρτηση μετάβασης για την κατάσταση *a* είναι : $(0, \text{Print } 0, \Delta_p = 1, a)$, $(1, \text{Print } 1, \Delta_p = 1, a)$.

Η κατάσταση b περιγράφεται με τις τετράδες :

$(\sqcup, \text{Print } 1, \Delta_p = 1, c)$, $(0, \text{Print } 1, \Delta_p = -1, c)$, $(1, \text{Print } 0, \Delta_p = -1, b)$.

Η κατάσταση c περιγράφεται με τις τετράδες :

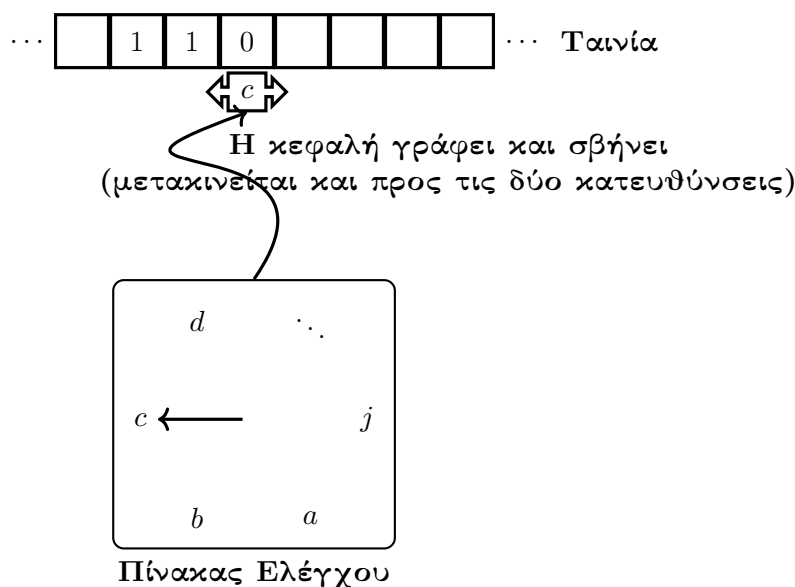
$(\sqcup, \text{Print } \sqcup, \Delta_p = -1, \text{halt})$, $(0, \text{Print } 0, \Delta_p = 1, c)$, $(1, \text{Print } 1, \Delta_p = 1, c)$.

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 μία θέση αριστερά και παραμένει στην κατάσταση *b*. Το τρέχον κελί είναι 0,

⁶⁴A Symbolic Analysis of Relay and Switching Circuits, Master thesis, MIT (1937)

⁶⁵The Annotated Turing, C.Petzold (Wiley)

οπότε τυπώνει 1, η κεφαλή μετακινείται μία θέση αριστερά και μεταβαίνει στην κατάσταση c . Το τρέχον κελί τώρα έχει την τιμή 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).



Σχήμα 7.2: Μια μηχανή Turing.

Το σχήμα είναι παραλλαγή του σχήματος του Sebastian Sardina και έχει άδεια CC-BY 2.5.
<http://www.texample.net/tikz/examples/turing-machine-2>

Στο σχήμα βλέπουμε τον πίνακα ελέγχου (software), καθώς και μία ταινία άπειρου μήκους, με μία κεφαλή που έχει τη δυνατότητα εγγραφής και σβησίματος (hardware), καθώς και μετακίνησής της επί της ταινίας.

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	$\square$	Print 0, $\Delta_p = 1$	c
c	$\square$	Print $\square$, $\Delta_p = 1$	e
e	$\square$	Print 1, $\Delta_p = 1$	f
f	$\square$	Print $\square$, $\Delta_p = 1$	b

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

Κλάσεις Πολυπλοκότητας

Η κλάση πολυπλοκότητας P αποτελείται από όλα τα προβλήματα απόφασης (δηλαδή 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 7.2.1. (απλός ορισμός) Ένα πρόβλημα απόφασης ανήκει στην κλάση 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 ελέγξουμε τις λύσεις, είναι εξίσου εύκολο να υπολογίσουμε τις λύσεις. Το πρόβλημα αυτό μέχρι σήμερα είναι άλυτο, δηλαδή, δεν γνωρίζουμε αν ισχύει $P = NP$.

κλάση P : Μπορώ να βρω αποδοτικό αλγόριθμο που να λύνει τα προβλήματα. Συνοπτικά θα λέγαμε ότι σε αυτήν την κλάση ανήκουν όλα τα προβλήματα που είναι quick solvable.
κλάση NP : Μπορώ να βρω αποδοτικό αλγόριθμο που να επαληθεύει μια λύση. Ανάλογα θα χαρακτηρίζαμε τα προβλήματα αυτής της κλάσης quick checkable.

NDTM

Οι NDTM σε μια συγκεκριμένη κατάσταση μπορεί να έχουν πολλές δυνατότητες επιλογών και όχι μόνο μία, όπως συμβαίνει στις κλασικές (δηλαδή 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 NAI, τότε λέμε ότι το αρχικό μας πρόβλημα έχει απάντηση NAI. Αν όλα τα μονοπάτια οδηγούν στην απάντηση OXI, τότε η απάντηση στο αρχικό μας πρόβλημα είναι OXI. Με αυτόν τον τρόπο προβλήματα εκθετικού χρόνου μια NDTM μπορεί να τα λύσει σε πολυωνυμικό χρόνο.

Ορισμός 7.2.2. Ένα πρόβλημα λέγεται NP , αν υπάρχει ένα μη ντετερμινιστικό πρόγραμμα (δηλαδή περιέχει την εντολή goto both line 1,2), που δίνει απάντηση NAI, αν και μόνο αν υπάρχει ένα μονοπάτι που δίνει απάντηση NAI.

Μπορούμε να προσομοιάσουμε το μη ντετερμινιστικό πρόγραμμα με ένα ντετερμινιστικό πρόγραμμα, αλλά με ένα επιπλέον στοιχείο, το οποίο θα λέει ποιο δρόμο να ακολουθήσει το πρόγραμμα κάθε φορά που συναντά 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 προβλήματός μας x που έχει απάντηση NAI και το πιστοποιητικό w , τότε σε πολυωνυμικό χρόνο απαντά NAI. Αντιστρόφως, αν υπάρχει ένα πιστοποιητικό w τέτοιο, ώστε για το τυχαίο στιγμιότυπο x του προβλήματος να έχω $C(x, w) = \text{NAI}$, τότε το x έχει απάντηση NAI. Μπορούμε να διατυπώσουμε τον προηγούμενο ορισμό κάνοντας χρήση των πιστοποιητικών.

Ορισμός 7.2.3. (2ος ορισμός) Ένα πρόβλημα καλείται NP , αν υπάρχει πολυωνυμικού χρόνου πιστοποιητής.

Ακόμη πιο φορμαλιστικά μπορούμε να γράψουμε:

$X \in NP$, αν υπάρχει πιστοποιητικό $w = \text{poly}(|x|)$ με $C(x, w) = \text{NAI}$, όπου C είναι στην κλάση P .

Ένα παράδειγμα προβλήματος που ανήκει στην κλάση NP είναι το πρόβλημα εύρεσης ενός δρόμου σε έναν γράφο, που να περνάει μόνο μία φορά από κάθε κόμβο (το πρόβλημα αυτό ονομάζεται 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 ισχύει $P = NP$ (Godel 1956 & Cook 1971).

Αν θεωρήσουμε τα προβλήματα NP των οποίων η απάντηση είναι OXI 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. Έστω ότι έχουμε ένα γράφημα G , όπου πρέπει να απαντήσουμε με

ένα NAI/OXI στο εξής πρόβλημα:
Το G δεν έχει Hamiltonian μονοπάτι;

Εύκολα μπορούμε να απαντήσουμε σε αυτήν την ερώτηση, αλλά δεν ισχύει το ίδιο για το πρόβλημα: *Το G έχει Hamiltonian μονοπάτι;*⁶⁶ Πιστεύουμε ότι $\text{co-NP} \neq \text{NP}$.

Η κλάση NP και η co-NP περιέχουν 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. Τέλος, και το επόμενο πρόβλημα είναι σημαντικό για την κρυπτογραφία.

FACTOR (πρόβλημα απόφασης)

Δίνονται δύο θετικοί ακέραιοι n, k ($n > k$). Υπάρχει διαιρέτης του n , μεγαλύτερος του 1, που είναι μικρότερος του k ;

Το πρόβλημα αυτό είναι NP, διότι ένας θετικός ακέραιος p , $1 < p < k$, και p διαιρέτης του n αποτελεί ένα πιστοποιητικό για τα NAI-στιγμιότυπα. Επίσης, είναι co-NP , διότι η παραγοντοποίηση του n σε πρώτους παράγοντες είναι ένα πιστοποιητικό για τα OXI-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 αποδειχτεί ότι είναι στην κλάση P θα κατέστρεφε ένα πολύ βασικό κρυπτογραφικό σύστημα, το RSA. Επίσης, θα έλυνε μια σειρά από δύσκολα προβλήματα της θεωρίας αριθμών, όπως για παράδειγμα, την εύρεση τετραγωνικών ριζών mod N , για κάποιον θετικό ακέραιο N .

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

Υπάρχουν προβλήματα στην κλάση NP που είναι δύσκολα και τα ονομάζουμε NP-πλήρη (NP-complete), με την έννοια ότι όλα τα προβλήματα της κλάσης NP ανάγονται πολυωνυμικά σε αυτό. Τα προβλήματα που είναι τουλάχιστον τόσο δύσκολα όσο τα NP-complete ορίζουν μια νέα κλάση, που την ονομάζουμε NP-hard. Σε αυτήν 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.

Ένας ισοδύναμος ορισμός της κλάσης NP-hard είναι ο εξής: Ένα πρόβλημα Λ καλείται NP-hard, αν έχω ένα μαντείο⁶⁷ (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 $\text{NP-hard} \cap \text{NP}$. Πιο φορμαλιστικά μπορούμε να δώσουμε τον ακόλουθο ορισμό για τα NP-πλήρη προβλήματα. Χρειαζόμαστε τον παρακάτω ορισμό.

Ορισμός 7.2.4. 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 7.2.5. Ένα πρόβλημα απόφασης L λέγεται NP-πλήρες αν

- i. $L \in \text{NP}$
- ii. $L_1 \leq_p L$ για κάθε $L_1 \in \text{NP}$.

Το πρόβλημα απόφασης του γυλιού (knapsack problem) είναι NP-πλήρες. 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.

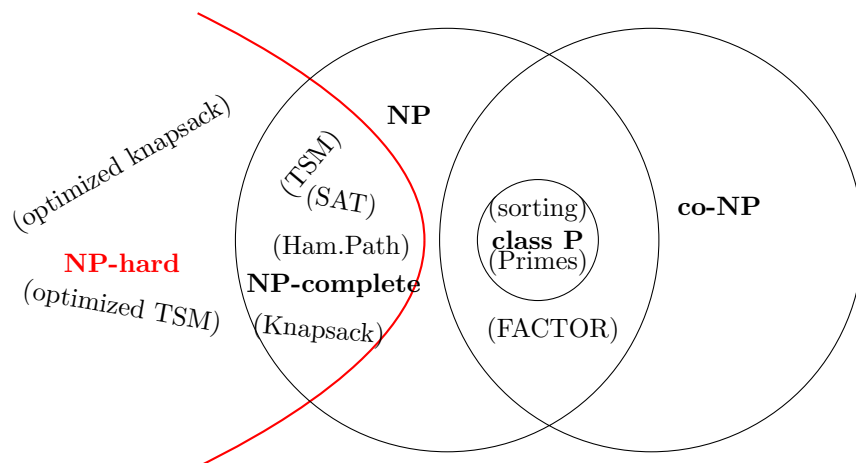
⁶⁶ Δηλαδή, υπάρχει απλός κύκλος στον γράφο που να επισκέπτεται κάθε κόμβος;

⁶⁷ με τον όρο μαντείο εννοούμε μια ντετερμινιστική μηχανή, ενώ ο όρος τυχαίο μαντείο (random oracle) είναι μια μηχανή που σε μία είσοδο απαντά με μια τυχαία ακολουθία χαρακτήρων. Με την ίδια είσοδο, το τυχαίο μαντείο ή μαντείο, πάντα απαντά με την ίδια έξοδο.

Πράγματι, ας υποθέσουμε ότι έχουμε ένα μαντείο που δέχεται ως είσοδο ένα σύνολο $A = \{a_1, a_2, \dots, a_n\}$ και έναν 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).

Ας αναφέρουμε και την κλάση PSPACE, που αποτελείται από όλα τα προβλήματα απόφασης που για να απαντηθούν από μία ντετερμινιστική μηχανή Turing, χρησιμοποιούν $\text{poly}(n)$ 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 συμβαίνει, 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 7.3: Υποθέσαμε ότι $NP \neq co-NP$

7.2.2 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 $O(n^2)$ και ήταν γνωστός στους αρχαίους πολιτισμούς (Αιγυπτίους, Βαβυλώνιους κ.). Υπάρχουν και υποτετραγωνικού χρόνου αλγόριθμοι. Η ιδέα που χρησιμοποιούμε είναι να ανάγουμε τον πολλαπλασιασμό των ακεραίων σε πολλαπλασιασμό δύο ακεραίων πολυωνύμων. 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, έχουμε 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 7.2). Έτσι, το

$$z_1 = (17 + 82) \cdot (16 + 59) - z_2 - z_0 = 99 \cdot 75 - 272 - 4838 = 2315.$$

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

$$x = 10^{(\rho+1)/2}x_2 + x_1, \quad y = 10^{(\rho+1)/2}y_2 + y_1.$$

Αλγόριθμος 7.2.1. : Πολλαπλασιασμός του Karatsuba

Είσοδος. a, b ακέραιοι

Έξοδος. $a \cdot b$

```

1 def karatsuba(a, b)
2   if a < 100 or b < 100 then
3     return a * b
4   end
5   m = max(log10(a), log10(b))
6   m2 = floor(m/2)
7   high(a) = take the first m2 decimal digits of a
8   low(a) = take the last m2 decimal digits of a
9   high(b) = take the first m2 decimal digits of b
10  low(b) = take the last m2 decimal digits of b
11  z0 = karatsuba(low(a), low(b))
12  z1 = karatsuba((low(a) + high(a)), (low(b) + high(b)))
13  z2 = karatsuba(high(a), high(b))
14  print (z2 * 102m2 + (z1 - z2 - z0) * 10m2 + 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 τη διαίρεση με υπόλοιπο, απαιτείται τετραγωνικός χρόνος $O(n^2)$. Γραμμικός χρόνος απαιτείται για την πρόσθεση και αφαίρεση. Μια υλοποίηση στην python θα μπορούσε να είναι η παρακάτω,

```

1 def karatsuba(x,y,B):
2     if len(str(x)) == 1 or len(str(y)) == 1:
3         return x*y
4     else:
5         m = max(len(str(x)), len(str(y)))
6         m2 = m // 2
7         a = x // B**(m2)
8         b = x % B**(m2)
9         c = y // B**(m2)
10        d = y % B**(m2)
11
12        z0 = karatsuba(b,d,B)
13        z1 = karatsuba((a+b), (c+d), B)
14        z2 = karatsuba(a,c,B)
15        return ( z2 * B**(2*m2) ) + ( (z1 - z2 - z0) * B**(m2) ) + z0

```

Άσκηση 7.1 Να πολλαπλασιάσετε (με το χέρι) τους αριθμούς 1234, 5678 με χρήση του πολλαπλασιασμού Karatsuba. Κατόπιν, πολλαπλασιάστε τους αριθμούς,

$$123456789, 987654321.$$

με χρήση Η/Υ, αφού χρησιμοποιήσετε την προηγούμενη υλοποίηση σε python.

Άσκηση 7.2 Έστω, $a > b > 1$ και η αναγωγική ακολουθία $T(n) = aT(n/b) + Cn$. Αποδείξτε επαγωγικά ότι υπάρχουν δύο σταθερές A, B που εξαρτώνται από το $T(1)$ και C τέτοιες, ώστε

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

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

7.2.3 Γρήγορη ύψωση σε δύναμη

Πολλές φορές στην κρυπτογραφία απαιτείται να μπορούμε να υπολογίζουμε δυνάμεις mod m γρήγορα, δηλαδή, σε πολυωνυμικό χρόνο. Ένας γρήγορος αλγόριθμος ύψωσης σε δύναμη βασίζεται στην εξής παρατήρηση: Έστω ότι θέλουμε να υπολογίσουμε τη 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 τη δύναμη

$$3^7 = 3^{2^2+2^1+2^0} = 3^4 \cdot 3^2 \cdot 3.$$

Τέλος, υπολογίζουμε τις δυνάμεις $3^2 = 9$, $3^4 = (3^2)^2 = 9^2 = 81 \equiv 9 \pmod{12}$. Το αποτέλεσμα είναι $9 \cdot 9 \cdot 3 \equiv 3 \pmod{12}$. Μερικές φορές η μέθοδος αυτή ονομάζεται από τα δεξιά προς τα αριστερά δυαδική μέθοδος.

Αλγόριθμος 7.2.2. : Γρήγορη ύψωση σε δύναμη mod m

Είσοδος. b (base), e (exponent), m (modulus) ($b < m$)

Έξοδος. $b^e \pmod{m}$

```

1  $B \leftarrow b$ 
2  $result \leftarrow 1$ 
3  $M \leftarrow m$ 
4  $E \leftarrow e$ 
  while  $E > 0$  do
5   if  $E \bmod 2 = 1$  then
6      $result \leftarrow result * B \bmod M$ 
   end
7    $E \leftarrow \lfloor E/2 \rfloor$ 
8    $B \leftarrow B^2 \bmod M$ 
  end
```

Στη γραμμή 7 υπολογίζουμε το ακέραιο μέρος του $E/2$, το οποίο μας δίνει το πηλίκο της διαίρεσης του E με το 2 (Θεώρημα 8.2.1). Το δυαδικό ανάπτυγμα του εκθέτη e υπολογίζεται στη γραμμή 5. Αν είναι 0, τότε δεν υπολογίζουμε τίποτα, διαφορετικά υπολογίζουμε το $b^{2^i} \pmod{m}$, όπου i είναι η θέση ενός 1 στο δυαδικό ανάπτυγμα του e .

Παράδειγμα 7.2.1. Έστω ότι θέλουμε να υπολογίσουμε τη δύναμη $2^{53} \pmod{157}$ (όλες οι πράξεις στη δεύτερη και τρίτη στήλη είναι mod 157).

α/α	B	$result$	E
initialization	$b = 2$	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, $2^{53} \equiv 2 \pmod{157}$. Η πολυπλοκότητα για τον υπολογισμό του $b^e \pmod{157}$ 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 7.3 Υλοποιήστε σε όποια γλώσσα προγραμματισμού θέλετε τον αλγόριθμο 7.2.2 και κατόπιν υπολογίστε τη δύναμη $5^{77} \pmod{19}$.

Άσκηση 7.4 Υλοποιήστε σε όποια γλώσσα προγραμματισμού θέλετε τον αλγόριθμο 7.2.2 και κατόπιν υπολογίστε το γινόμενο

$$2^{1000} \cdot 3^{101} \cdot 5^{47} \pmod{2^{107} - 1}$$

κάνοντας χρήση και του αλγορίθμου του Karatsuba.

Βιβλιογραφία

- [1] Knuth, Donald E. *The Art of Computer Programming*. vol.2, Seminumerical Algorithms, 3rd edition, Addison-Wesley, 1998.
- [2] Τσίχλας, Κ.; Μανωλόπουλος, Ι.; Γούναρης, Α. *Σχεδίαση και Ανάλυση Αλγορίθμων*, Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. <http://dx.doi.org/10.57713/kallipos-613>, 2015.
- [3] Πουλάκης, Δ. *Θεωρία Αριθμών*, Εκδ. Ζήτη, 2009.
- [4] Πουλάκης, Δ. *Υπολογιστική Θεωρία Αριθμών*, Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. <http://dx.doi.org/10.57713/kallipos-323>, 2015.
- [5] Ζάχος, Ε.; Παγουρτζής, Α.; Σούλιου, Θ. *Θεμελίωση επιστήμης υπολογιστών*, Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. <http://dx.doi.org/10.57713/kallipos-491>, 2015.
- [6] Τζανάκης, Νίκος. *Θεμελιώδης Θεωρία Αριθμών*, Σημειώσεις, Πανεπιστήμιο Κρήτης, 2012.
- [7] Ζάχος, Ε.; Παγουρτζής, Α.; Γροντάς, Π. *Υπολογιστική Κρυπτογραφία*, Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. <http://dx.doi.org/10.57713/kallipos-492>, 2015.
- [8] Κατσαρός, Π. *Θεωρία Υπολογισμού και Εφαρμογές*. Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών, 2015.
- [9] Κολουντζάκης, Μ.; Παπαχριστόδουλος, Χ. *Διακριτά μαθηματικά*, Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. <http://dx.doi.org/10.57713/kallipos-517>, 2015.

Κεφάλαιο 8

Εισαγωγή στη Θεωρία αριθμών

Σύνοψη - Περίληψη. Σε αυτό το κεφάλαιο παρουσιάζουμε την απαραίτητη θεωρία αριθμών που χρειαζόμαστε, για να καταλάβουμε το βασικό κρυπτόςυστημα RSA, καθώς και μερικές επιθέσεις σε αυτό. Επίσης, το κεφάλαιο αυτό είναι απαραίτητο για αυτούς που επιθυμούν να υλοποιήσουν κρυπτογραφικά πρωτόκολλα, διότι θα πρέπει να γνωρίζουν όλες τις λεπτομέρειες (μαθηματικές και μη) για την υλοποίηση κρυπτοσυστημάτων δημόσιου κλειδιού. Μια αβλεψία στην υλοποίηση μπορεί να έχει καταστροφικά αποτελέσματα γι' αυτόν που χρησιμοποιεί τη συγκεκριμένη υλοποίηση.

Η θεωρία αριθμών υπάρχει ανεξάρτητα από την κρυπτογραφία και πολύ πριν την ανακάλυψή της. Πρόκειται για έναν κλάδο των καθαρών μαθηματικών (pure mathematics) που μέχρι σήμερα γνωρίζει μεγάλη ανάπτυξη ανεξάρτητα από την κρυπτογραφία. Ούτως ή άλλως, ένα πολύ μικρό κομμάτι της θεωρίας αριθμών έχει εφαρμογές στην κρυπτογραφία. Ο αναγνώστης που θέλει να δει πώς διακλαδίζεται όλο αυτό το πεδίο, μπορεί να επισκεφτεί τον ιστότοπο της Αμερικάνικης Μαθηματικής Εταιρείας⁶⁸ και να αναζητήσει το subject classification : 11. Θα παρατηρήσει ότι από τις 20 υποπεριοχές της θεωρίας αριθμών, μία μόνο πραγματικά αφορά την κρυπτογραφία, η υποπεριοχή με κωδικό 11Yxx : Computational Number Theory. Ειδικότερα, και από αυτήν την υποπεριοχή ένα κομμάτι της αφορά στην κρυπτογραφία δημόσιου κλειδιού. Βέβαια, και οι υποπεριοχές 11Txx : Finite Fields and commutative rings καθώς και η 11Hxx : Geometry of Numbers έχουν περιεχόμενο που αφορά την κρυπτογραφία. Ο αναγνώστης που θα έρθει σε επαφή με αυτόν τον κλάδο μελετώντας κρυπτογραφία, ίσως βρει ένα κίνητρο να ασχοληθεί και ανεξάρτητα με αυτόν τον όμορφο κλάδο των καθαρών μαθηματικών. Οπότε ο τίτλος *Εισαγωγή στη θεωρία αριθμών* είναι παραπλανητικός. Αυτό το οποίο θα παρουσιάσουμε είναι η απαραίτητη θεωρία αριθμών που χρειάζεται για να καταλάβουμε βασικά το RSA.

Στο κεφάλαιο αυτό, μερικές αποδείξεις παραλείπονται, αλλά συστήνουμε στον αναγνώστη να αναζητήσει τις αποδείξεις σε κάποιο από τα βιβλία που προτείνονται ως βιβλιογραφία. Τα επόμενα δύο κεφάλαια που έχουν τίτλο *Τεστ Πιστοποίησης πρώτων αριθμών* και *Παραγοντοποίηση & Διακριτός λογάριθμος*, επίσης αφορούν στοιχεία της θεωρίας αριθμών που είναι απαραίτητα για την κατανόηση της κρυπτογραφίας δημόσιου κλειδιού.

⁶⁸<https://mathscinet.ams.org/mathscinet/msc/msc2020.html?t=11-XX&s=&btn=Search&ls=s>

Προαπαιτούμενη γνώση. Πολλές από τις έννοιες που παρουσιάζονται σε αυτό το κεφάλαιο θα είναι γνωστές στους αναγνώστες, ειδικότερα αν έχουν διδαχθεί ένα εισαγωγικό μάθημα διακριτών μαθηματικών ή ακόμη και από τα λυκειακά μαθηματικά. Βασικές έννοιες της αριθμητικής είναι απαραίτητες, π.χ. μπορείτε να δείτε εισαγωγικά βιβλία διακριτών μαθηματικών, όπως [17, Κεφάλαια 1 και 2] και το βιβλίο του Rosen [8]. Αν ο αναγνώστης επιθυμεί να δει και κάποιο βιβλίο θεωρίας αριθμών στην ελληνόγλωσση βιβλιογραφία, μπορεί να μελετήσει τα [1, 13, 15] ή και το [3, Κεφ. 31] που παρουσιάζει με έμφαση στους αλγόριθμους στοιχειώδεις έννοιες της θεωρίας αριθμών. Στην αγγλική βιβλιογραφία υπάρχουν πολλές επιλογές για παράδειγμα, ενδεικτικά αναφέρουμε [6, 12] και το online βιβλίο των Boneh-Shoup [2, Παράρτημα 1]. Στην υποενότητα των ισοτιμιών (8.4), θα ήταν χρήσιμο ο αναγνώστης να γνωρίζει βασικές έννοιες αλγεβρικών δομών (αντιμεταθετικός δακτύλιος, ακέραια περιοχή, σώμα). Επίσης, στην υποενότητα που παρουσιάζουμε το κινεζικό θεώρημα υπολοίπων χρησιμοποιούμε τον όρο *ισομορφισμός δακτυλίων*. Μπορείτε να δείτε τα βασικά για τις αλγεβρικές δομές στο βιβλίο του J. B. Frayleigh, [4, Κεφάλαιο 4]. Στην υποενότητα του θεωρήματος των πρώτων αριθμών απαιτούνται βασικές γνώσεις ανάλυσης μίας μεταβλητής (ορισμένο ολοκλήρωμα και όρια), π.χ. μπορείτε να δείτε το βιβλίο του Spivak, [10, Κεφάλαιο 13].

8.1 Εισαγωγή

*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 H/Y , σε ένα μεγάλο κομμάτι της θεωρίας αριθμών η ανάγκη αυτή αποδεδειγμένα υπάρχει. Για παράδειγμα, η διάσημη εικασία των Birch και Swinnerton-Dyer που συνδέει τη 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 H/Y . Π.χ. αν $\lambda(n) = (-1)^r$, όπου r το πλήθος των πρώτων διαιρετών του n και $\Lambda(x) = \sum_{n \leq x} \lambda(n)$, ο Polya διατύπωσε την εικασία ότι $\Lambda(x) \leq 0$. Ο Lehman το 1960 με τη βοήθεια H/Y υπολόγισε ότι $\Lambda(906180359) = 1$. Επίσης, ο Mertens, διατύπωσε την εικασία ότι $|M(x)| = |\sum_{n \leq x} \mu(n)| \leq \sqrt{x}$, όπου $\mu(x)$ η συνάρτηση του Möbius. Οι Odlyzko και Riele το 1985 βρήκαν ένα αντιπαραδείγμα βασισμένοι σε πολύπλοκους υπολογισμούς που έγιναν σε H/Y .

Σε αυτό το κεφάλαιο θα ξεκινήσουμε με εισαγωγικές έννοιες, όπως είναι η διαιρετότητα και ο μέγιστος κοινός διαιρέτης, 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 mod 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.

8.2 Διαιρετότητα

Ορισμός 8.2.1. 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 $a \nmid b$.

Για παράδειγμα, $2|8$, $3|9$, $22963|45926$, ενώ $17 \nmid 18$.

Πρόταση 8.2.1 (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 8.2.1. 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 8.2.1 (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 με τον b .

Απόδειξη. Πρώτα θα αποδείξουμε ότι το $q = \lfloor \frac{a}{b} \rfloor$ ικανοποιεί την απαιτούμενη ιδιότητα και κατόπιν τη 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, $b || r - r' \Rightarrow b \leq |r - r'|$. Αυτό είναι άτοπο, διότι $0 \leq r, r' < b$ δηλαδή, $|r - r'| < b$. $\square$

Το προηγούμενο θεώρημα ισχύει και γενικότερα.

Θεώρημα 8.2.2 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 8.2.2. 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 διαίρεση του a με τον b με πηλίκο q κοστίζει $O(\text{len}_2(a)\text{len}_2(b))$. Όλες οι προηγούμενες πράξεις χρειάζονται $O(\text{len}_2(a) + \text{len}_2(b))$ χώρο. Επομένως, η εύρεση του q καθώς και του υπολοίπου r κοστίζει $O(\text{len}_2(a)\text{len}_2(b))$.

Παρατήρηση 8.2.3. Στην κρυπτογραφία συνήθως η λέξη αποδοτικός ταυτίζεται με τη λέξη πολυωνυμικός. Εξηγούμε τι ακριβώς εννοούμε με αυτό. Έστω ότι έχουμε έναν αλγόριθμο με είσοδο τους θετικούς ακέραιους, $a_1, \dots, a_n$. Λέμε ότι ο αλγόριθμος είναι πολυωνυμικός, αν εκτελείται σε

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

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

Άσκηση 8.1 Ν.α.ο. αριθμοί της μορφής $4n + 3$ δεν είναι τέλεια τετράγωνα. Κατόπιν, ν.α.ό. κανέναν από τους αριθμούς

$$11, 111, \dots, 111 \dots 111, \dots$$

δεν είναι τέλειο τετράγωνο.

Άσκηση 8.2 cross+roads=danger. Αν κάθε γράμμα αντιστοιχεί σε ένα μη μηδενικό ψηφίο, βρείτε την αριθμητική τιμή της λέξης danger.

Άσκηση 8.3 (**) Θεωρούμε την ακόλουθη διαδικασία. Έστω x ένας ακέραιος. Αν x άρτιος, τότε διαίρεσέ τον με το 2, διαφορετικά τον πολλαπλασιάζουμε με τον 3 και προσθέτουμε 1. Αν καταλήξουμε στον αριθμό 1, σταματάμε. Επαναλαμβάνουμε τη 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 8.4 Ν.δ.ό. το 2^m ($m \geq 2$) δεν είναι άθροισμα διαδοχικών θετικών ακεραίων.

Άσκηση 8.5 Να βρείτε έναν αποδοτικό αλγόριθμο που με είσοδο δύο λίστες $P = [p_1, \dots, p_n]$ (p_i πρώτοι) και $e = [e_1, \dots, e_n]$ (e_i θετικοί ακέραιοι) να εξάγει όλους τους διαιρέτες του $N = p_1^{e_1} \dots p_n^{e_n}$. Να γίνει μελέτη όσον αφορά τις απαιτήσεις σε μνήμη.

Άσκηση 8.6 Ν.α.ο. $n^5 + 1 | n^{10} - 1$ για κάθε φυσικό n .

Άσκηση 8.7 Να αποδείξετε την Πρόταση 8.2.1.

Άσκηση 8.8 Ν.α.ό. το γινόμενο n διαδοχικών φυσικών αριθμών διαιρείται από το $n!$

Άσκηση 8.9 Να αποδείξετε ότι ο ακέραιος $n \geq 2$ διαιρεί ακριβώς έναν αριθμό, από ένα σύνολο n διαδοχικών ακεραίων.

Άσκηση 8.10 (*) Έστω $\mathcal{A} = \{1, 2, \dots, 50\}$. Ας είναι $\mathcal{B}_k \subset \mathcal{A}$ όπου $|\mathcal{B}_k| = k$. Θεωρούμε όλα τα υποσύνολα $\mathcal{B}_k$ τέτοια, ώστε, να έχουν τουλάχιστον ένα ζευγάρι (a, b) με $a + b | ab$ και $a \neq b$. Ν.δ.ό. η μικρότερη τιμή του k είναι 39. Μπορείτε να το αποδείξετε χωρίς τη χρήση υπολογιστή;

Άσκηση 8.11 Βρείτε του θετικούς διαιρέτες των αριθμών 220 και 284. Ας είναι $\sigma(220)$ και $\sigma(284)$ τα αθροίσματα των θετικών διαιρετών του 220 και 284, αντίστοιχα. Παρατηρήστε ότι $\sigma(220) - 220 = 284$, $\sigma(284) - 284 = 220$. Βρείτε το επόμενο ζευγάρι ακεραίων (n, m) , ώστε $\sigma(n) = \sigma(m) = n + m$. Οι αριθμοί αυτοί ονομάζονται amicable numbers (δείτε και την αναφορά [1, ενότητα 3.1]).

Άσκηση 8.12 Αν $H_n = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$ (n θετικός ακέραιος), ν.α.ό. υπάρχει το

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

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

Άσκηση 8.13 Να κάνετε τα γραφήματα των ακολουθιών⁶⁹

$$a_n = \ln n + 2\gamma - 1$$

και

$$b_n = \frac{1}{n} \sum_{r=1}^n \tau(r),$$

όπου $\gamma \approx 0.577$ η σταθερά των Euler-Mascheroni⁷⁰ και $\tau(n)$ το πλήθος των θετικών διαιρετών του n , π.χ. $\tau(12) = 6$. Ο Lejeune Dirichlet το 1838 απέδειξε ότι ο μέσος όρος των διαιρετών του n πλησιάζει στο $\ln n + 2\gamma - 1$.

Άσκηση 8.14 (**) 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

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

για αυθαίρετο φυσικό k .

Άσκηση 8.15 (**) (Erdős-Borwein) Ν.α.ό. ο αριθμός

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

είναι ρητός.

⁶⁹Π.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.

⁷⁰Julian Havil, γ : Exploring Euler's constant, 2003, Princeton University Press.

8.2.1 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 τη μονάδα και τον εαυτό του. Διαφορετικά ονομάζεται σύνθετος. Το σύνολο των πρώτων αριθμών είναι:

$$\{2, 3, 5, \dots, 101, \dots, 571, \dots, 6997, \dots, 2^{57885161} - 1, \dots, 2^{82589933} - 1, \dots\}. \quad (8.2.1)$$

Κάθε αριθμός > 1 έχει έναν πρώτο διαιρέτη.

Λήμμα 8.2.1. 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 . Δηλαδή, $p|a$ επομένως, $p|ax$ για κάθε ακέραιο x . Για $x = b$, προκύπτει $p|ab = n$. Επομένως, ισχύει και για $k = n$. $\square$

Αμέσως τίθεται το ερώτημα για το πού σταματάει το σύνολο των πρώτων αριθμών.

Θεώρημα 8.2.3 (Ευκλείδης) Υπάρχουν άπειροι πρώτοι αριθμοί.

Απόδειξη. Υποθέτουμε ότι υπάρχουν πεπερασμένου πλήθους πρώτοι αριθμοί

$$p_1, p_2, \dots, p_n.$$

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 έναν πρώτο διαιρέτη. Επειδή όλοι οι πρώτοι είναι οι $p_1, \dots, p_n$, ο πρώτος διαιρέτης θα είναι κάποιος από τους αριθμούς $p_1, \dots, p_n$. Ας είναι ο p_i για κάποιο $i \in \{1, 2, \dots, n\}$. Αν διαιρέσουμε τον m με το p_i , έχουμε υπόλοιπο 1, ενώ θα έπρεπε το υπόλοιπο να είναι 0. Άτοπο. Άρα, υπάρχουν άπειροι το πλήθος πρώτοι. $\square$

Παρατήρηση 8.2.4. Εύκολα προκύπτει από την απόδειξη ότι ο ακέραιος

$$k = \prod_{i=1}^n p_i + 1 \quad (p_1 < p_2 < \cdots < p_n)$$

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 8.2.5. 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 $ab = n > n$. Αυτό όμως είναι άτοπο.

Στο σύνολο όλων των πρώτων αριθμών (8.2.1) συμπεριλάβουμε και τον αριθμό 101. Μπορεί κάποιος δοκιμάζοντας να βρει όλους τους πρώτους διαιρέτες του 101 και να διαπιστώσει σχετικά γρήγορα ότι δεν υπάρχουν άλλοι, εκτός του 1 και του 101. Π.χ. ελέγχουμε έναν-έναν τους αριθμούς

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

αν είναι διαιρέτες του 101. Αν κανείς από αυτούς δεν είναι, τότε είναι πρώτος. Τι γίνεται όμως με τον 6997; Επίσης, πώς θα ελέγχαμε ότι ο αριθμός 14220533 είναι πράγματι πρώτος; Ακόμη χειρότερα, πώς θα μπορούσαμε να αποδείξουμε ότι ο αριθμός

$$2^{82589933} - 1$$

είναι πρώτος⁷¹ αριθμός; Αυτό είναι το πρόβλημα της πιστοποίησης των πρώτων αριθμών και έχει λυθεί με την έννοια ότι υπάρχει πολυωνυμικός ντετερμινιστικός αλγόριθμος που να αποφασίζεται 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' αυτόν τον λόγο υπάρχει το καταναμεμημένο δίκτυο "Great Internet Mersenne Prime Search" (GIMPS). Το επόμενο πρόβλημα στη θεωρία των πρώτων αριθμών είναι το πρόβλημα της παραγοντοποίησης ενός αριθμού στους πρώτους παράγοντές του. Ένα σημαντικό θεώρημα της θεωρίας αριθμών είναι το παρακάτω.

Θεώρημα 8.2.4 (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 αυτήν τη δυσκολία πιστεύουμε ότι βασίζεται η ασφάλεια του RSA. Για τη μελέτη των πρώτων αριθμών συστήνουμε το βιβλίο των Pomerance και Grandall [6]. 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 λογάριθμου 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 8.2.6. Ένας λόγος που ο 1 δεν θεωρείται πρώτος, είναι για να έχω τη μοναδικότητα του θεωρήματος 8.2.4. Αν $p_1 = 1$, τότε δεν υπάρχει μοναδικός ακέραιος a_1 .

Με χρήση του θεωρήματος 8.2.4, μπορούμε να δώσουμε ακόμη μια απόδειξη ότι υπάρχουν άπειροι πρώτοι αριθμοί.

Λήμμα 8.2.2 (2η απόδειξη ότι υπάρχουν άπειροι πρώτοι). Το σύνολο των πρώτων είναι άπειρο.

Απόδειξη. Ας υποθέσουμε ότι υπάρχουν πεπερασμένοι πρώτοι αριθμοί, $p_1, p_2, \dots, p_r$. Διαλέγουμε αυθαίρετα ένα $n \geq 2$. Ας είναι ένας ακέραιος s με $1 \leq s \leq n$. Από την υπόθεσή 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 τη συνάρτηση,

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

⁷¹Ο συγκεκριμένος αριθμός το 2018 αποδείχθηκε ότι είναι ένας πρώτος αριθμός του Mersenne

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 πλήθος των s είναι n επομένως και το πλήθος των διαφορετικών τιμών του $\Psi(s)$ είναι n . Καταλήγουμε ότι

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

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

Παρατήρηση 8.2.7. Η παρακάτω εντολή παράγει πρώτους αριθμούς με τη βοήθεια του openssl. Στο συγκεκριμένο παράδειγμα παράγεται ένας πρώτος με 512 bits.

```
1 $openssl prime -generate -bits 512
```

Άσκηση 8.16 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 8.17 Ν.α.ό. ο n -οστός πρώτος p_n είναι μεγαλύτερος ή ίσος από το $n + 1$.

Άσκηση 8.18 Να αποδείξετε ότι

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

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

Άσκηση 8.19 Να αποδείξετε ότι ο n -οστός πρώτος p_n ικανοποιεί την ανισότητα

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

Υπόδ. Χρησιμοποιήστε επαγωγή σε συνδυασμό με την παρατήρηση 8.2.4.

Άσκηση 8.20 Ο αριθμός $H_n = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$ ονομάζεται n -οστός αρμονικός αριθμός. Ν.δ.ό. $\lim_{n \rightarrow \infty} H_n = \infty$. Η ακολουθία αυτή πλησιάζει το άπειρο πολύ αργά. Αν περιορίσουμε το n να παίρνει τιμές στους πρώτους, τότε έχουμε τη σειρά,

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

Τι μπορείτε να πείτε για τη σύγκλιση αυτής της σειράς;

Υπόδειξη.

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

Ένας άλλος τρόπος θα ήταν να υπολογιστεί το ολοκλήρωμα

$$\int_{-\infty}^0 \frac{e^x}{1 - e^x} dx = \dots = \lim_{n \rightarrow \infty} H_n.$$

Άσκηση 8.21 (Το κρυμμένο θεώρημα του Πλάτωνα). Έστω p, q , διαδοχικοί πρώτοι αριθμοί τέτοιοι, ώστε $3 < p < q$. Τότε, κάθε ακέραιος $r < q$ διαιρεί το $p!$

Σχόλια. Για την ενδιαφέρουσα ιστορία αυτού του θεωρήματός δείτε [14, ενότητα 3.2.5].

Άσκηση 8.22 Αν $n > 5$ ακέραιος, τότε το $n!$ διαιρείται από όλους τους θετικούς ακέραιους $\leq n$ καθώς και από όλους τους σύνθετους αριθμούς του συνόλου $\{n+1, \dots, 2n\}$.

Σχόλια. Το συγκεκριμένο θεώρημα συνδέεται με το κρυμμένο θεώρημα του Πλάτωνα⁷².

Άσκηση 8.23 Με τη βοήθεια H/Y μπορείτε να επαληθεύσετε (για αρκετά μεγάλα x και p πρώτος)

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

Άσκηση 8.24 (Euler's formula). Ν.α.ό.

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

Αυτή η ισότητα αποδεικνύει ότι υπάρχουν άπειροι πρώτοι. Γιατί;

Άσκηση 8.25 Έστω p πρώτος. Με τη βοήθεια H/Y μπορείτε να επαληθεύσετε (για αρκετά μεγάλα x)

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

Επίσης, να δώσετε και μία μαθηματική απόδειξη.

Υπόδειξη.

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

Αυτή η σταθερά ισούται με $\zeta(2)$, όπου $\zeta(s)$ η συνάρτηση του Riemann (δείτε και άσκηση 8.67).

Άσκηση 8.26 Αν p πρώτος ν.α.ό. ο $\sqrt{p}$ είναι άρρητος.

8.3 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 αυτήν 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 $x^{-1} \pmod{N}$ (εφόσον υπάρχει). Η σύνδεση αυτή θα γίνει πιο ξεκάθαρη στο επόμενο βασικό θεώρημα.

Θεώρημα 8.3.1 (Ταυτότητα Bézout). Αν x, y ακέραιοι αριθμοί, όχι και οι δύο μηδέν, και $d = \gcd(x, y)$, τότε υπάρχουν ακέραιοι αριθμοί a, b τέτοιοι, ώστε $d = ax + by$.

⁷²A. Vardoulakis and C. Pugh, Plato's hidden theorem on the distribution of primes. The Mathematical Intelligencer Vol. **30**(3), 2008.

Απόδειξη. Έστω

$$L_{x,y} = \{n \in \mathbb{N} - \{0\} : n = ax + by \text{ για κάποια } a, b \in \mathbb{Z}\}.$$

Αρχικά, παρατηρούμε ότι $L_{x,y} \neq \emptyset$. Δηλαδή, υπάρχει ακέραιος θετικός αριθμός που γράφεται στη 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,

$$d \leq g. \quad (8.3.1)$$

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 θεώρημα 8.2.1, 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 τη σχέση (8.3.1). Δηλαδή, $g = d$. $\square$

Πόρισμα 8.3.1. Για κάθε ακέραιο a και b , αν $k|a$ και $k|b$ τότε $k|\gcd(a, b)$.

Απόδειξη. Θέτουμε $d = \gcd(a, b)$. Από την ταυτότητα του Bézout, υπάρχουν ακέραιοι x, y τέτοιοι ώστε $ax + by = d$. Εφόσον, $k|a$, $k|b$, τότε $k|ax + by$ δηλαδή $k|d$. $\square$

Παρατήρηση 8.3.1. Αν $d = 1$, τότε γράφεται $ax + by = 1$. Μερικές φορές την ονομάζουμε ταυτότητα Bachet.

Παρατήρηση 8.3.2. 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 8.3.1 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 $x = abz_2$ δηλαδή $ab|x$. $\square$

Επίσης, έχουμε το παρακάτω χρήσιμο αποτέλεσμα.

Λήμμα 8.3.1 (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 αποδείξουμε ότι $p|b$. Από την ταυτότητα του Bézout (λήμμα (8.3.1)), υπάρχουν ακέραιοι x, y τέτοιοι, ώστε $1 = xp + ya$. Οπότε, $b = bxp + bya$. Αλλά, $p|bxp$ και από την υπόθεση $p|y(ab)$. Επομένως, $p|bxp + bya$ δηλαδή, $p|b$. $\square$

Υπάρχει η εξής γενίκευση:

Λήμμα 8.3.2. Αν $n|ab$ και $\gcd(n, a) = 1$, τότε $n|b$.

Από την απόδειξη της ταυτότητας Βézout έχουμε το εξής: ένας κοινός διαιρέτης των a, b διαιρεί τον μέγιστο κοινό διαιρέτη των a, b (Πόρισμα 8.3.1). Μπορούμε να διατυπώσουμε και τον παρακάτω ισοδύναμο ορισμό του $\gcd$.

Ορισμός 8.3.1. 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 . Δηλαδή, $x = \Pi y + Y$, όπου $Y = x \bmod y$ το υπόλοιπο του x διαιρούμενο με το y .

Λήμμα 8.3.3. Έστω x, y θετικοί ακέραιοι με $x \geq y$. Τότε, $\gcd(x, y) = \gcd(y, x \bmod y)$.

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 8.3.1. : 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 αλγόριθμος αυτός παρουσιάστηκε πρώτη φορά στο βιβλίο *Στοιχεία* του Ευκλείδη το 300 π .Χ. (βιβλίο VII). Ίσως είναι ο παλαιότερος αλγόριθμος που επιβίωσε μέχρι τη σημερινή εποχή. Παρά την απλότητα του αλγορίθμου, δεν είναι καθόλου εύκολο να υπολογίσουμε μετά από πόσες επαναλήψεις τερματίζει (δηλαδή, την πολυπλοκότητά του). Έστω n το πλήθος των επαναλήψεων του αλγορίθμου (8.3.1).

Πρόταση 8.3.2 (Reynaud, 1811) $n < y$.

Πρόταση 8.3.3 (Reynaud, 1821) $n < y/2$.

Πρόταση 8.3.4 (Finck, 1841) $n < 2 \log_2 y + 1$.

Απόδειξη. Θα αποδείξουμε ότι $x \bmod y < x/2$. Θυμίζουμε ότι με τον συμβολισμό $x \bmod y$ εννοούμε το υπόλοιπο της διαίρεσης x διά του y . Διακρίνουμε δύο περιπτώσεις:

- Αν $y \leq x/2$, τότε το υπόλοιπο της διαίρεσης του x με το 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 ή 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. $\square$

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 8.3.2. Η bit πολυπλοκότητα του αλγορίθμου (8.3.1) είναι $O((\log_2 N)^3)$.

Επίσης, βελτιώθηκε το προηγούμενο φράγμα για το n .

Πρόταση 8.3.5 (Finck, 1844) $n < 5 \log_{10} y$.

Με μια πιο προσεκτική ανάλυση του Ευκλείδειου αλγορίθμου αποδεικνύεται ότι η bit-complexity είναι $O((\log_2 N)^2)$ [6, Άσκηση 2.6]. Τέλος, ισχύει το παρακάτω βασικό θεώρημα:

Θεώρημα 8.3.2 (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

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

Παράδειγμα 8.3.1. 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.

Επεκτεταμένος Ευκλείδειος Αλγόριθμος.

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

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

Από τα προηγούμενα προκύπτει $\gcd(x, y) = r_N$. Ακολουθώντας την αντίστροφη πορεία μπορούμε να βρούμε και τους συντελεστές Bézout. Έτσι, στο παράδειγμα 8.3.1 έχουμε

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

Κάθε φορά αντικαθιστώ τα προηγούμενα υπόλοιπα.

Αλγόριθμος 8.3.2. : 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 8.3.1 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

$$q = 3, w = 1769 \pmod{551} = 116, U = a - qu = 1 - 3 \cdot 0 = 1,$$

$$V = b - qv = 0 - 3 \cdot 1 = -3, (a, b, u, v) = (0, 1, 1, -3) \text{ κ.ο.κ.}$$

Συνοψίζουμε στον παρακάτω πίνακα:

α/α	a	b	u	v	g	w	U	V	q
initialization	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 8.3.3 Ας είναι $x \geq y > 0$ δύο ακέραιοι. Θέτουμε

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

Έστω οι ακολουθίες

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

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

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} με το r_{i-1} και ισχύει $r_0 \geq r_1 > r_2 > \dots$ οπότε αργά ή γρήγορα για κάποιο δείκτη, έστω $n+1$, θα έχουμε $r_{n+1} = 0$. Ειδικότερα, το n φράσσεται, όπως είδαμε στο Θεώρημα 8.3.2, 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 θεώρημά μας προκύπτει αν θέσουμε $i = n$. Παρατηρούμε ότι για $i = 0$ ισχύει. Έστω ότι ισχύει για κάποιο δείκτη $i \in \{1, 2, \dots, n\}$. Θα αποδείξουμε ότι ισχύει για $i+1$. Δηλαδή, θα αποδείξουμε ότι ισχύει

$$r_n = s_{i+1} r_{n-i} + s_i r_{n-i-1}. \quad (8.3.2)$$

Αντικαθιστούμε στο δεξιό μέλος της εξίσωσης (8.3.2) τις σχέσεις

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

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 8.3.3. : 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 τη συνάρτηση $N(x, y)$ η οποία επιστρέφει το πλήθος των επαναλήψεων που χρειάζεται ο ευκλείδειος αλγόριθμος, για να τερματίσει. Αυτό θα χρησιμοποιηθεί στη ρουτίνα για τον υπολογισμό της s_i που δέχεται ως εισόδους (n, x, y, N) . Τέλος, ορίζουμε τη 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. $(a, b, d = \gcd(x, y))$ 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 < [√5 ln(x)] do
| if r(j, x, y) = 0 then
| | print j - 1
| | break
| else
| | j = j + 1
| end
end
```

end

Computation of q_i

```
def q(n,x,y)
return [r(n - 2, x, y)/r(n - 1, x, y)]
```

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) · 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 · s(N - 1, x, y, N) + y · 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 8.3.2 για τα ίδια x, y επέστρεψε την τριάδα $(60, -469, 7)$ (δείτε και την αναφορά [7]). 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.

Άσκηση 8.27 Να αποδείξετε το λήμμα 8.3.2.

Άσκηση 8.28 Έστω N, a, b, c θετικοί ακέραιοι με $\gcd(a, b) = \gcd(b, c) = \gcd(a, c) = 1$. Έστω

$$n = |\{x \in \mathbb{Z} : a \nmid x, b \nmid x, c \nmid x, 1 \leq x \leq N\}|.$$

N.α.ο. ο αριθμός

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

είναι μια καλή προσέγγιση του n .

Άσκηση 8.29 Να βρείτε τον μκδ των αριθμών 540, 315. Κατόπιν, αν $d = \gcd(540, 315)$, βρείτε ακέραιους α, β τέτοιους, ώστε $540\alpha + 315\beta = d$ (κάντε τους υπολογισμούς με το χέρι).

Άσκηση 8.30 (*) Αν (x_0, y_0) οι συντελεστές Bézout των ακεραίων (a, b) , με $\gcd(a, b) = d$ και $a, b > 0$, που έχουν προκύψει με τον Ευκλείδειο αλγόριθμο, ν.α.ό. $\|(x_0, y_0)\| \leq \frac{1}{2d}\|(a, b)\|$.
Υπόδ. $\|(x_0, y_0)\| = \sqrt{x_0^2 + y_0^2}$.

Άσκηση 8.31 Έστω k, n δύο θετικοί ακέραιοι. Υπολογίστε το παρακάτω άθροισμα

$$2 \sum_{j=1}^{k-1} \left\lfloor \frac{nj}{k} \right\rfloor - kn + k + n.$$

για τουλάχιστον 100 τιμές των k, n . Τι παρατηρείτε;⁷³

Άσκηση 8.32 Να αποδείξετε ότι οι αριθμοί 1456, 1717 είναι πρώτοι μεταξύ τους.

Άσκηση 8.33 Έστω a, b θετικοί ακέραιοι. 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.ό. αν a, b περιττοί θετικοί ακέραιοι, τότε $\gcd(a+b, a-b) = 2$.
- (iii). $\gcd(M_p, M_q) = 1$, όπου $M_p = 2^p - 1, M_q = 2^q - 1$, Mersenne ακέραιοι (p, q πρώτοι με $p \neq q$).

Άσκηση 8.34 Έστω ένας ακέραιος της μορφής $P = k2^n + 1$, για $0 < k < 2^n$. Ακέραιοι αυτής της μορφής ονομάζονται Proth numbers. Ισχύει η πρόταση, αν $a^{(P-1)/2} \equiv -1 \pmod{P}$ για κάποιον ακέραιο a , τότε ο P είναι πρώτος. Βρείτε 10 πενταψήφιους πρώτους αριθμούς του Proth με την προηγούμενη πρόταση.

Άσκηση 8.35 (**) Έστω n θετικός ακέραιος, αν $\sigma(n)$ το άθροισμα των διαιρετών του n και

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

ν.α.ό.

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

Άσκηση 8.36 Να υλοποιήσετε τον Ευκλείδειο αλγόριθμο και κατόπιν μετρήστε πόσα βήματα απαιτούνται για τον υπολογισμό του $\gcd(121393, 75025)$.

Άσκηση 8.37 Έστω $x > y > 0$ ακέραιοι αριθμοί. Υποθέτουμε ότι ο $\gcd(x, y)$ υπολογίζεται με τον αλγόριθμο του Ευκλείδη μετά από n -βήματα. Τότε, οι μικρότερες τιμές των x, y ικανοποιούν $x \geq F_{n+2}, y = F_{n+1}$. Όπου F_n η ακολουθία Fibonacci,

$$F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2} \quad (n \geq 2).$$

Αυτό αποδεικνύει ότι η χειρότερη περίπτωση για τον αλγόριθμο του Ευκλείδη είναι να έχουμε είσοδο δύο αριθμούς Fibonacci.

Υπόδ. Χρησιμοποιήστε επαγωγή.

⁷³Δείτε και τη διατριβή του Τσαγκάρη Παναγιώτη, Πρώτοι αριθμοί και κυκλοτομία - Πρώτοι αριθμοί της μορφής $x^2 + (x+1)^2$
<https://thesis.ekt.gr/thesisBookReader/id/0150?lang=el#page/48/mode/2up>

Άσκηση 8.38 Αν m, n είναι ακέραιοι πρώτοι μεταξύ τους ν.α.ό. $m^{-1} \bmod n = \frac{1+n(-n^{-1} \bmod m)}{m}$.

Άσκηση 8.39 Να αποδείξετε ότι το ελάχιστο πλήθος διαιρετών του B είναι 12 αν το πλήθος διαιρετών του AB είναι 105 και του A είναι 24.

(Υπόδ. Το πλήθος των διαιρετών ενός ακεραίου $n = p_1^{a_1} p_2^{a_2} \cdots p_n^{a_n}$ είναι $\tau(n) = (a_1 + 1)(a_2 + 1) \cdots (a_n + 1)$. Γράψτε κώδικα στο Sagemath για όλες τις δυνατές περιπτώσεις, βάζοντας τους κατάλληλους περιορισμούς.)

Άσκηση 8.40 Αν $F_n = 2^{2^n} + 1$, ν.α.ό. $\gcd(F_i, F_j) = 1$ για κάθε $i \neq j$. Κατόπιν ν.δ.ό. υπάρχουν άπειροι πρώτοι.

Άσκηση 8.41 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 8.42 Έστω p πρώτος αριθμός και

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

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

Άσκηση 8.43 Έστω a, b, c ακέραιοι αριθμοί και $\delta = a^2 - 4bc^2 \neq 0$. Ν.δ.ό. $\gcd(\delta, 4c^2)$ είναι τετράγωνο.

Άσκηση 8.44 Η πιθανότητα δύο ακέραιοι αριθμοί από το σύνολο $\{1, 2, \dots, N\}$ (για μεγάλο N) να έχουν μέγιστο κοινό διαιρέτη $\leq B$ ισούται με

$$\frac{6}{\pi^2} \sum_{k=1}^B \frac{1}{k^2}.$$

Ελέγξτε υπολογιστικά με χρήση κατάλληλου αλγόριθμου την προηγούμενη πρόταση.

8.4 Ισοτιμίες (ή Ισοδυναμίες)

Η έννοια της ισοτιμίας οφείλεται στον Gauss, δείτε [5]. Αν m ακέραιος > 1 , καλούμε τους ακέραιους x, y ισότιμους modulo m , αν η διαφορά $x - y$ διαιρείται από το m . Δηλαδή, $m | x - y$, και γράφουμε $x \equiv y \pmod{m}$. Τότε, ισχύει το εξής:

Λήμμα 8.4.1. $m | x - y \Leftrightarrow$ οι x, y έχουν ίδιο υπόλοιπο διαιρούμενοι με τον m .

Απόδειξη. ($\Rightarrow$) Έστω $0 \leq v_1, v_2 < m$ τα υπόλοιπα των διαιρέσεων x, y με το m αντίστοιχα. Τότε, $x = mq_1 + v_1$, $y = mq_2 + v_2$. Θα δείξουμε ότι $v_1 = v_2$. Έχουμε

$$x - y = m(q_1 - q_2) + v_1 - v_2. \quad (8.4.1)$$

Άρα,

$$|v_1 - v_2| = |m(q_2 - q_1) + x - y|.$$

Αλλά $0 \leq v_1, v_2 < m$ οπότε $|v_1 - v_2| < m$. Επομένως,

$$0 \leq |v_1 - v_2| = |m(q_2 - q_1) + x - y| < m.$$

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

$$x - y = mq, \quad (8.4.2)$$

άρα,

$$0 \leq |v_1 - v_2| = m|q_2 - q_1 + q| < m.$$

Εφόσον $|q_2 - q_1 + q|$ είναι μη-αρνητικός ακέραιος, αναγκαστικά, $|q_2 - q_1 + q| = 0$. Οπότε, $|v_1 - v_2| = 0$ ισοδύναμα, $v_1 = v_2$.

($\Leftarrow$) Αν $v_1 = v_2$, τότε $x - y = m(q_1 - q_2)$, άρα $m|a - b$. $\square$

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

$$\bar{a} = \{x \in \mathbb{Z} : x \equiv a \pmod{m}\}.$$

Επομένως, από το προηγούμενο λήμμα, κάθε κλάση modulo m αναπαρίσταται με ένα στοιχείο του συνόλου $\{\bar{0}, \bar{1}, \dots, \overline{m-1}\}$. 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. Δηλαδή, υπάρχουν διαιρέτες του μηδενός. Π.χ. αν $m = 6$, τότε $\bar{2} \cdot \bar{3} = \bar{0}$. Αν ο $m = p$ είναι πρώτος, τότε μερικές φορές συμβολίζεται και ως $\mathbf{F}_p$. Σε αυτήν την περίπτωση είναι ακέραια περιοχή (ειδικότερα, είναι σώμα (field)). Επιπλέον, ισχύουν οι εξής βασικές ιδιότητες:

Λήμμα 8.4.2. (i). $a \equiv a \pmod{m}$.

(ii). Αν $a \equiv b \pmod{m}$, τότε $b \equiv a \pmod{m}$.

(iii). Αν $a \equiv b \pmod{m}$ και $b \equiv c \pmod{m}$, τότε $a \equiv c \pmod{m}$.

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 8.4.1 (i). Έστω $a \equiv b \pmod{m}$, $c \equiv d \pmod{m}$, τότε $a \pm c \equiv b \pm d \pmod{m}$.

(ii). Έστω $a \equiv b \pmod{m}$, τότε $a^n \equiv b^n \pmod{m}$, n θετικός ακέραιος.

(iii). Έστω $a \equiv b \pmod{m}$, τότε $na \equiv nb \pmod{m}$, n ακέραιος.

(iv). Έστω $a \equiv b \pmod{m}$, και $d|m$, τότε $a \equiv b \pmod{d}$.

(v). Έστω d κοινός διαιρέτης των a, b με $a \equiv b \pmod{m}$. Αν $\gcd(d, m) = 1$, τότε

$$a/d \equiv b/d \pmod{m}.$$

(vi). Έστω $a \equiv b \pmod{m}$ και $a \equiv b \pmod{n}$. Αν $\gcd(m, n) = 1$, τότε

$$a \equiv b \pmod{mn}.$$

Απόδειξη. (vi). Ισοδύναμα, $m|a - b$, $n|a - b$. Επειδή $\gcd(m, n) = 1$, έχουμε από την πρόταση 8.3.1 ότι $mn|a - b$. Το ζητούμενο έπεται. $\square$

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

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

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

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

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 $\bar{1} \in \mathbb{Z}_m^*$ και $\phi(1) = 1$. Πολλές φορές παραλείπουμε την “μπάρα” πάνω από τους αριθμούς, και γράφουμε $1 \in \mathbb{Z}_m^*$. Δηλαδή, αντί του $\bar{1}$ θεωρούμε το στοιχείο 1 της κλάσης $\bar{1}$. Μία μέθοδος υπολογισμού του αντιστρόφου ενός στοιχείου $\text{mod } m$ δίνεται από τον επεκτεταμένο Ευκλείδειο αλγόριθμο.

Παρατήρηση 8.4.1. Το σύνολο $\mathbb{Z}_m^*$ με πράξη πολλαπλασιασμού τον πολλαπλασιασμό $\text{mod } m$ είναι αβελιανή ομάδα. Ειδικότερα, αν $m = 2, 4, p^e$ ή $2p^e$ για p περιττό πρώτο αριθμό και e φυσικό ≥ 1 , είναι κυκλική ομάδα ([3, Ενότητα 31.6, Θεώρημα 31.32]). Αυτό σημαίνει ότι υπάρχει ένα στοιχείο g ώστε, κάθε στοιχείο x της ομάδας $\mathbb{Z}_m^*$ να γράφεται $x = g^t$ για κάποιον φυσικό t . Το στοιχείο g ονομάζεται γεννήτορας της ομάδας ή αρχική ρίζα $\text{mod } m$. Για παράδειγμα το 3 είναι μια αρχική ρίζα $\text{mod } 7$ ενώ το 2 όχι.

Παράδειγμα 8.4.1. Έστω ότι θέλουμε να υπολογίσουμε το $4^{-1} \pmod{15}$. Εφαρμόζουμε τον επεκτεταμένο Ευκλείδειο αλγόριθμο και 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 $\text{mod } 15$ και στα δύο μέλη και έχουμε

$$1 \equiv 4 \cdot 4 \pmod{15} \Rightarrow 4^{-1} \equiv 4 \pmod{15}.$$

Στο προηγούμενο παράδειγμα χρησιμοποιήσαμε το εξής:

Λήμμα 8.4.3. $\gcd(x, y) = 1$, 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 $\text{mod } N$. Επομένως, η εύρεση αντιστρόφου μπορεί να γίνει σε πολυωνυμικό χρόνο (bit complexity).

Παράδειγμα 8.4.2. 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 8.4.1 Αν $n = p_1^{a_1} \cdots p_k^{a_k}$ η πρωτογενής ανάλυση του φυσικού αριθμού n σε πρώτους παράγοντες, τότε

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

Παρατήρηση 8.4.2. Αν γνωρίζουμε όλους τους πρώτους διαιρέτες του n και όχι την πρωτογενή του ανάλυση⁷⁴, επίσης μπορούμε να χρησιμοποιήσουμε το προηγούμενο θεώρημα για να υπολογίσουμε το $\phi(n)$.

Πόρισμα 8.4.1. Αν $\gcd(m, n) = 1$, τότε $\phi(mn) = \phi(m)\phi(n)$.

Απόδειξη. Άσκηση για τον αναγνώστη. Μια συνάρτηση με αυτήν την ιδιότητα ονομάζεται πολλαπλασιαστική. $\square$

Παρατήρηση 8.4.3. 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 ασφάλειά του στη δυσκολία υπολογισμού του $\phi(n)$, όταν δεν γνωρίζουμε την παραγοντοποίηση του n . Για περισσότερα δείτε την επόμενη υποενότητα 8.4.1.

Άσκηση 8.45 Έστω m θετικός ακέραιος και $\tau(m)$ το πλήθος των θετικών διαιρετών του m . Αν $m = p_1^{a_1} p_2^{a_2} \cdots p_k^{a_k}$ η πρωτογενής ανάλυση του m , ν.δ.ό.

$$\tau(m) = (a_1 + 1)(a_2 + 1) \cdots (a_k + 1).$$

Κατόπιν, ν.α.ό. είναι πολλαπλασιαστική. Δηλαδή, αν $\gcd(n, m) = 1$ (για n, m θετικούς ακέραιους), τότε $\tau(nm) = \tau(n)\tau(m)$.

Άσκηση 8.46 Να αποδείξετε ότι ο αριθμός 5040 είναι highly composite⁷⁵, δηλαδή, για κάθε θετικό ακέραιο n με $n < 5040$ ισχύει $\tau(n) < \tau(5040)$ (για τον ορισμό της συνάρτησης $\tau(n)$ δείτε την άσκηση 8.45).

Άσκηση 8.47 Έστω m θετικός ακέραιος και $\sigma(m)$ το άθροισμα των θετικών διαιρετών του m . Αν $m = p_1^{a_1} p_2^{a_2} \cdots p_k^{a_k}$ η πρωτογενής ανάλυση του m , ν.α.ό.

$$\sigma(m) = \prod_{i=1}^k \frac{p_i^{a_i+1} - 1}{p_i - 1}.$$

Κατόπιν, ν.α.ό. είναι πολλαπλασιαστική. Δηλαδή, αν $\gcd(n, m) = 1$ (για n, m θετικούς ακέραιους), τότε $\sigma(nm) = \sigma(n)\sigma(m)$. Τέλος, βρείτε δύο θετικούς ακέραιους που να έχουν την ιδιότητα $\sigma(n) = 2n$.

Άσκηση 8.48 Έστω m θετικός ακέραιος και $\sigma(m)$ το άθροισμα των θετικών διαιρετών του m και $\tau(m)$ το πλήθος των (θετικών) διαιρετών του m . Ν.α.ό.

$$\sum_{d|m} \left(\frac{m}{d} - 1\right) = \sigma(m) - \tau(m).$$

⁷⁴ Δηλαδή, να γνωρίζουμε τους πρώτους διαιρέτες του n , αλλά όχι τους εκθέτες a_i .

⁷⁵ Ο αριθμός 5040 μελετήθηκε αρχικά από τον Πλάτωνα.

Άσκηση 8.49 Έστω m θετικός ακέραιος και $\sigma(m)$ το άθροισμα των θετικών διαιρετών του m . Ν.α.ό. ο μεγαλύτερος ακέραιος n μέσα στο διάστημα $[2, 10^7]$ με την ιδιότητα

$$\sigma(n) > e^\gamma n \ln(\ln n),$$

είναι ο 5040. Η σταθερά γ του Euler είναι $\gamma \approx 0.577...$

Υπόδ. Μπορείτε να χρησιμοποιήσετε κατάλληλο πρόγραμμα που να ελέγχει μία-μία όλες τις περιπτώσεις.

Σχόλιο. Ο αριθμός 5040 ονομάζεται και αριθμός του Πλάτωνα και εμφανίζεται πρώτη φορά στο 5ο βιβλίο των *Νόμων* του Πλάτωνα, ως ο ιδανικός αριθμός οικογενειών μιας ιδανικής πόλης⁷⁶. Επίσης ο Robin έχει αποδείξει την εξής πρόταση : Η υπόθεση του Riemann είναι ισοδύναμη με την πρόταση

$$\frac{\sigma(n)}{n} < e^\gamma \ln \ln n$$

για όλα τα $n > 5040$.

Άσκηση 8.50 Επαληθεύστε πειραματικά⁷⁷ ότι για όλους τους περιττούς ακέραιους $< 2^{20}$ ισχύει

$$\frac{\sigma(n)}{n} < \frac{e^\gamma}{2} \ln \ln n + \frac{0.74}{\ln \ln n}.$$

Άσκηση 8.51 Αν m θετικός ακέραιος, να αποδείξετε ότι

$$\phi(m) = m \sum_{d|m} \frac{\mu(d)}{d},$$

όπου

$$\mu(d) = \begin{cases} 1, & d = 1 \\ (-1)^k, & d = p_1 p_2 \cdots p_k \quad (p_i : \text{πρώτοι}) \\ 0, & \text{αλλού} \end{cases}$$

Με το σύμβολο $d|m$ εννοούμε όλους τους θετικούς διαιρέτες του m . Π.χ. αν $m = 30$, τότε

$$\{d|m : 1 \leq d \leq m\} = \{1, 2, 3, 5, 6, 10, 15, 30\}.$$

Δηλαδή, το σύνολο θετικών διαιρετών του m .

Άσκηση 8.52 Να αποδείξετε ότι το πλήθος των πρωταρχικών (primitive) πολυωνύμων βαθμού n είναι

$$\phi(2^n - 1)/n.$$

Δουλεύουμε στο σώμα με δύο στοιχεία $\{0, 1\}$, δηλαδή, mod 2. Θυμίζουμε ότι η θεωρία των πρωταρχικών πολυωνύμων παίζει ρόλο στην εύρεση LFSR που έχουν μέγιστη περίοδο.

Άσκηση 8.53 Αν m θετικός ακέραιος, να αποδείξετε ότι

$$\phi(m) \geq 2^{\omega(m)-1},$$

όπου $\omega(m)$ είναι το πλήθος των διαφορετικών πρώτων διαιρετών του m . Π.χ. $\omega(2) = \omega(2^2) = 1$, $\omega(10) = 2$.

⁷⁶Δήμητρα Ρετσινά, Η πολιτική φιλοσοφία του Πλάτωνα και του Αριστοτέλη, 2008, <http://ikee.lib.auth.gr/record/110139/files/gri-2009-2032.pdf>

⁷⁷δηλαδή, με κατάλληλη υλοποίηση σε Η/Υ

Άσκηση 8.54 Να αποδείξετε ότι, αν $a \equiv 1 \pmod{5}$, τότε $a^2 \equiv 1 \pmod{5}$.

Άσκηση 8.55 Να αποδείξετε ότι $a^3 \equiv 0$ ή 1 ή $8 \pmod{9}$.

Άσκηση 8.56 Αν p πρώτος και a, b θετικοί ακέραιοι, να αποδείξετε ότι $(a + b)^p \equiv a^p + b^p \pmod{p}$.

Άσκηση 8.57 (*) 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 $a^2 + b^2 = c^2$ και $\frac{ab}{2} = n$. 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 Άραβι. Ενώ φαίνεται απλό, είναι αρκετά σύνθετο και εμπλέκει θεωρία των ελλειπτικών καμπυλών.

Άσκηση 8.58 Ν.α.ο. **δεν** υπάρχει θετικός ακέραιος n τέτοιος, ώστε

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

Άσκηση 8.59 Να αποδείξετε ότι, αν $a \equiv 1 \pmod{2}$, τότε $a^4 \equiv 1 \pmod{16}$.

Άσκηση 8.60 Να αποδείξετε το λήμμα 8.4.2.

Άσκηση 8.61 Να υπολογίσετε το $\phi(1134)$ και το $\phi(2457)$.

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

Το πρόβλημα της παραγοντοποίησης και το πρόβλημα υπολογισμού της $\phi(n)$ είναι **(πιθανοτικά) ισοδύναμα** προβλήματα (δείτε [9, Ενότητα 10.4]). Δηλαδή, αν έχω έναν πολυωνυμικό αλγόριθμο που λύνει το ένα πρόβλημα, τότε μπορούμε να λύσουμε σε πολυωνυμικό χρόνο και το άλλο, με μεγάλη πιθανότητα. Αν έχουμε την παραγοντοποίηση του n από το θεώρημα 8.4.1 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 $n = p \times q$, 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 $\phi(n) = (p - 1)(q - 1)$. Η προηγούμενη μέθοδος είναι ντετερμινιστική. Αν ο $n = pqr$ γινόμενο τριών πρώτων, επίσης υπάρχει ντετερμινιστικός αλγόριθμος⁷⁸ που με είσοδο το $\phi(n)$ επιστρέφει την παραγοντοποίηση του n .

Άσκηση 8.62 Δίνεται $(\phi(N), N) = (137678184, 137703491)$. Να βρείτε τους πρώτους διαιρέτες του N .

Άσκηση 8.63 Το ίδιο με την προηγούμενη άσκηση, όπου

$$\phi(n) = 4516338132025103032106450322755616278189697800986956966 \\ 29409713682134644105212900399132945840693276541563285514439736000$$

και

$$n = 451633813202510303210645032278038248611927777174629568254195974 \\ 587431642559706866792720921199279008765236326990270821389.$$

⁷⁸J. Urozz, Factorization, malleability and equivalent problems

8.5 Το Θεώρημα των Πρώτων Αριθμών

Τσως το πιο διάσημο άλυτο πρόβλημα των μαθηματικών, η υπόθεση του *Riemann* μας απαντάει στο πρόβλημα της κατανομής των πρώτων αριθμών. Οι πρώτοι αριθμοί είναι οι φυσικοί > 1 που διαιρούνται μόνο με τη μονάδα και τον εαυτό τους. Το σύνολο των πρώτων $\{2, 3, 5, 7, 11, \dots, 101, \dots\}$ είναι άπειρο. Όμως οι πρώτοι αριθμοί είναι διεσπαρμένοι μέσα στους φυσικούς με ακανόνιστο τρόπο. Όσο μελετώνται, παρατηρούμε ότι εμφανίζουν και μία κανονικότητα. Για παράδειγμα, μία εκδοχή του θεωρήματος των πρώτων αριθμών μάς 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 :

$$\frac{x}{\ln(x)} < \pi(x) < \int_2^x \frac{dt}{\ln t}. \quad (8.5.1)$$

Η τελευταία συνάρτηση στην ανισότητα (8.5.1) συμβολίζεται με $Li(x)$ (offset Logarithmic integral). Οπότε έχουμε

$$Li(x) = \int_2^x \frac{dt}{\ln t} \quad (x \geq 2).$$

Το θεώρημα 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,

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

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

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

Ο Havil το 2003 απέδειξε ότι

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

Η υπόθεση 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 υπόθεση του Riemann μας δίνει βελτιωμένο error term.

Παρατήρηση 8.5.1. Ο Yuri Matiyasevich το 2020 απέδειξε ότι η υπόθεση του Riemann είναι αληθής αν και μόνο αν το παρακάτω πρόγραμμα δεν τερματίζει!

```
1 import math
2 from math import gcd
3 d=0;m=0;p=0;
4 f=1;g=1;h=1;n=1;q=1;
5 while p**2 * (m-f)<h:
6     d = 2*n*d-(-1)**n * g
7     n+=1
```

```

8      c=math.gcd(n,q)
9      q = n*q//c
10     if c==1:p+=1
11     m = 0
12     r = q
13     while r>1:
14         r=r//2
15         m+=d
16     g = 2*f
17     f = 2*n*f
18     h = (2*n+3)*h

```

Άσκηση 8.64 Να αποδείξετε ότι $\pi(x) \geq \ln x - 1$.

Υπόδ. Αρχικά ν.δ.ό. 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 8.65 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 τιμές.

Άσκηση 8.66 Με το θεώρημα των πρώτων αριθμών, ν.δ.ό. $p_n \sim n \ln n$, όπου p_n ο n -οστός πρώτος.

Άσκηση 8.67 Η συνάρτηση ζήτα του Riemann είναι

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

Ν.α.ό.

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

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 $-2, -4, \dots$).

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

Λήμμα 8.5.1.

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

όπου e η βάση των νεπέριων λογαρίθμων.

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 $Pr \approx \frac{2 \log_2 e}{R}$. $\square$

Σύμφωνα με τα προηγούμενα, για να κατασκευάσω έναν 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 έναν 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 [16, 14]. Για το πρόβλημα της παραγοντοποίησης έχουμε μέχρι στιγμής μόνο υποεκθετικού χρόνου αλγορίθμους [11]. Πρέπει 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 8.68 Ν.δ.ό. δεν υπάρχει πρώτος στο διάστημα $[n! + 2, n! + n]$ για $n \geq 2$.

Άσκηση 8.69 Έστω

$$R(p, n) = \max \left\{ r : p^r \mid \binom{2n}{n} \right\}.$$

Ν.α.ό.

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

Άσκηση 8.71 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 αριθμοί 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 8.72 Ν.α.ό. αν $p \mid F_n = 2^{2^n} + 1$ ($n \geq 2$), όπου p πρώτος, τότε $p = 2^{n+1}k + 1$ για κάποιο φυσικό k . Ο Euler το 1732 έδειξε ότι $641 \mid F_5$, ενώ ο Laundry το 1880 ότι $274177 \mid F_6$. Βρείτε έναν πρώτο διαιρέτη του F_7 . Το 2004 βρέθηκε ένας πρώτος διαιρέτης του $F_{3329780}$.

Άσκηση 8.73 (*) Βρείτε έναν πρώτο διαιρέτη του $F_9 = 2^{512} + 1$. Ο Western το 1903 βρήκε έναν πρώτο διαιρέτη του.

Άσκηση 8.74 Ν.α.ό. αν $q \mid M_p$, 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 8.75 Οι πρώτοι αριθμοί του Mersenne (ασκ. 8.71) εμπλέκονται και με τους τέλειους αριθμούς⁷⁹. 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 απέδειξε ότι, αν M_q είναι πρώτος, τότε ο αριθμός $2^{q-1}M_q$ είναι τέλειος. Για $q = 7$ προκύπτει τέλειος αριθμός;

Άσκηση 8.76 Να βρείτε (με χρήση κατάλληλου κώδικα) τον επόμενο τέλειο αριθμό που είναι > 8128 (δείτε την ασκ. 8.75 για τον ορισμό του τέλειου αριθμού.). Πόσο χρόνο χρειάστηκε ο κώδικάς σας για το αποτέλεσμα;

Άσκηση 8.77 Έστω $p > 3$ πρώτος. Να αποδείξετε ότι υπάρχει μόνο μια τριάδα $(p, p+2, p+4)$, όπου και οι τρεις αριθμοί είναι πρώτοι.

Άσκηση 8.78 Προσεγγίστε τη σταθερά Landau-Ramanujan,

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

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

Άσκηση 8.79 Επαληθεύστε ότι για αρκετά μεγάλα $n \geq 1$,

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

Μπορείτε να το αποδείξετε;

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

Άσκηση 8.80 Ν.α.ό. 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 8.81 (*) Για κάθε θετικό ακέραιο n να υπολογίσετε τον $\gcd(n! + 1, (n+1)!)$.

Υπόδ. Θα χρειαστείτε και το θεώρημα του Wilson, δείτε την άσκηση 8.80.

Άσκηση 8.82 Ν.α.ό. 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 που ισούται με $2^{2^n} + 1$).

8.6 e -οστές ρίζες mod p

Έστω p πρώτος αριθμός.

Ορισμός 8.6.1. 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 $6^3 \equiv 7 \pmod{11}$. 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 $\mathbb{Z}_p$.

⁷⁹https://en.wikipedia.org/wiki/List_of_Mersenne_primes_and_perfect_numbers,
<https://oeis.org/A000396/list>

Λήμμα 8.6.1. 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 mod 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 mod p γίνεται σε πολυωνυμικό χρόνο. $\square$

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

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

Ορισμός 8.6.3. Έστω n ένας θετικός ακέραιος. Το $x \in \mathbb{Z}_n$ ονομάζεται τετραγωνικό υπόλοιπο (Τ.Υ.) mod 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 mod 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 8.6.2. $\#\{T.Y. \text{ mod } p\} = \frac{p-1}{2} + 1$.

Το πλήθος των Τ.Υ. mod $2p$ είναι,

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

το οποίο ισούται με $\#\{y : x^2 \equiv y \pmod{p}\} \times 2 - 1 = p$ (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 8.6.1 $x \in \mathbb{Z}_p^*$ 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 έναν αλγόριθμο υπολογισμού των τετραγωνικών ριζών. Για την περίπτωση όπου $p \equiv 1 \pmod{4}$, μπορούμε να αποδείξουμε το παρακάτω λήμμα.

Λήμμα 8.6.3. 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 $\mathbb{Z}_p^*$). $\square$

Για την περίπτωση $p \equiv 3 \pmod{4}$ 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 9.1.1

$$c^{(p-1)/2} = c^k = c^{2^s t} \equiv 1 \pmod{p}, \quad (8.6.1)$$

Ας είναι N ένα μη-τετραγωνικό υπόλοιπο mod p , δηλαδή $N^{(p-1)/2} = N^{2k} = N^{2^s t} \equiv -1 \pmod{p}$. Εξάγουμε τετραγωνικές ρίζες από τα δύο μέλη της ισοτιμίας (8.6.1). Οπότε προκύπτει

$$c^{2^{s-1}t} \equiv \pm 1 \pmod{p}.$$

Διακρίνουμε δύο περιπτώσεις. Αν $c^{2^{s-1}t} \equiv 1 \pmod{p}$, 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 ιστιμιά της μορφής $c^t N^{2^\ell} \equiv 1 \pmod{p}$, άρα $x \equiv \pm c^{(t+1)/2} N^\ell \pmod{p}$ (οπότε $x^2 \equiv c \pmod{p}$) και x είναι μία τετραγωνικά ρίζα $\pmod{p}$ του c).

Γενικά, μπορούμε να χρησιμοποιήσουμε τον πιθανοτικό αλγόριθμο (πολυωνυμικού χρόνου) του Tonelli [6, Algorithm 2.3.8], που είναι πιο αποδοτικός από τον προηγούμενο. Δεν γνωρίζουμε κάποιον ντετερμινιστικό αλγόριθμο για τον υπολογισμό τετραγωνικών ριζών $\pmod{p}$, 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 $\pmod{N}$ ($e > 2$), όταν δεν γνωρίζουμε την παραγοντοποίηση του N . Είναι ανοιχτό πρόβλημα, αν το RSA-problem είναι **ισοδύναμο** με το πρόβλημα της παραγοντοποίησης.

Άσκηση 8.83 Αν a τετραγωνικό υπόλοιπο $\pmod{p}$, τότε ν.α.ό. η εξίσωση $x^2 \equiv a \pmod{p}$ έχει δύο λύσεις.

Υπόδειξη.

Αν x, y λύσεις της εξίσωσης, ν.α.ό. $x \equiv \pm y \pmod{p}$.

Άσκηση 8.84 Έστω q πρώτος με $q \equiv 5 \pmod{8}$. Ας είναι $x \in \mathbb{Z}_q$. Ν.δ.ό. αν $x^{(q-1)/4} \equiv 1 \pmod{q}$ τότε μια τετραγωνική του ρίζα είναι η $x^{(q+3)/8} \pmod{q}$. Ενώ, αν $x^{(q-1)/4} \equiv -1 \pmod{q}$, τότε μια τετραγωνική του ρίζα είναι η

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

Άσκηση 8.85 Να βρείτε το πλήθος των Τ.Υ. $\pmod{N}$ αν $N = p_1 p_2 \cdots p_n$ όπου p_i περιττοί διακεκριμένοι πρώτοι αριθμοί.

8.7 Κινεζικό Θεώρημα Υπολοίπων (CRT)

Το κινεζικό θεώρημα υπολοίπων (CRT : Chinese Remainder Theorem) 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 8.7.1 Έστω $m_1, m_2, \dots, m_r$ 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 $n_i \pmod{m_i}$. Τότε το σύστημα ισοδυναμιών και η ανισότητα

$$x \equiv n_i \pmod{m_i}, \quad 0 \leq x < M$$

έχουν μοναδική λύση. Επιπλέον, αυτή η λύση δίνεται με το ελάχιστο θετικό υπόλοιπο $\pmod{M}$,

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

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

Παράδειγμα 8.7.1. 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,

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

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

Παρατήρηση 8.7.1. Δίνεται $N = n_1 n_2 \cdots n_k$, (n_i, n_j) πρώτοι μεταξύ τους για $i \neq j$. Έστω η απεικόνιση

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

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

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 11.2.3).

Πόρισμα 8.7.1. Έστω $m_1, m_2, \dots, m_r$ θετικοί ακέραιοι > 1 , πρώτοι μεταξύ τους ανά δύο, με γινόμενο $M = m_1 \cdot m_2 \cdots m_r$. Τότε,

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

Απόδειξη. ($\Rightarrow$) Έστω $z = b \sum_{i=1}^r v_i M_i$. Εφόσον το a είναι μια λύση του συστήματος

$$x \equiv b \pmod{m_i}, \quad 1 \leq i \leq r,$$

από CRT προκύπτει $z \equiv a \pmod{M}$. Επομένως,

$$a \equiv b \sum_{i=1}^r v_i M_i \pmod{M}.$$

Άρα, από τον ορισμό των v_i, M_i προκύπτει $a \equiv b \pmod{m_i}$. Πράγματι, όλοι αριθμοί

$$M_1, \dots, M_{i-1}, M_{i+1}, \dots, M_r$$

διαιρούνται από το m_i , επομένως

$$M_j \equiv 0 \pmod{m_i}, \quad j \neq i.$$

Ενώ, $v_i M_i \equiv 1 \pmod{m_i}$. Επομένως, $a \equiv b \pmod{m_i}$ για $i = 1, 2, \dots, r$.

($\Leftarrow$) Προκύπτει επαγωγικά από την πρόταση 8.4.1 (vi). □

Παρατήρηση 8.7.2. 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 ισοτιμία $x^2 \equiv 1 \pmod{m}$ είναι ισοδύναμη με το σύστημα $x^2 \equiv 1 \pmod{p_i^{a_i}}$ για $1 \leq i \leq r$. Καθεμία από αυτές τις εξισώσεις έχει δύο λύσεις, επομένως έχουμε συνολικά 2^r γραμμικά συστήματα ισοτιμιών όπου καθένα 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 10.1.5.

Άσκηση 8.86 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.

Άσκηση 8.87 Να λυθεί το γραμμικό σύστημα ισοτιμιών

$$\begin{cases} 2x \equiv 1 \pmod{5} \\ 3x \equiv 9 \pmod{6} \\ 4x \equiv 1 \pmod{7} \\ 5x \equiv 9 \pmod{11} \end{cases}.$$

Βιβλιογραφία

- [1] Αντωνιάδης, Ι.; Κοντογεώργης, Α. *Θεωρία αριθμών και εφαρμογές*, Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. <http://dx.doi.org/10.57713/kallipos-882>, 2015.
- [2] Boneh, Dan; Shoup, V. *A Graduate Course in Applied Cryptography*, Version 0.5, Jan. 2020. <http://toc.cryptobook.us/>.
- [3] Cormen, T. H.; Leiserson C. E.; Rivest E. L.; Stein C. *Εισαγωγή στους αλγορίθμους*, (Τόμος ΙΙ), ΠΕΚ, 2003.
- [4] Frayleigh, John B. *Εισαγωγή στην άλγεβρα*. Π.Ε.Κ., 1994.
- [5] Gauss, C. F. *Disquisitiones Arithmeticae. Revised by William C. Waterhouse (Translated by Arthur A. Clarke)*. Springer, 1966.
- [6] Grandall, R.; Pommerance, C. *Prime Numbers. A Computational Perspective*. 2nd edition, pp. 596, Springer, 2005.
- [7] Knuth, Donald E. *The Art of Computer Programming*. vol.2, Seminumerical Algorithms, 3rd edition, Addison-Wesley, 1998.
- [8] Rosen, Kenneth. *Διακριτά μαθηματικά και εφαρμογές*. Εκδ. Τζιόλας, 2018.
- [9] Shoup, Victor. *A Computational Introduction to Number Theory and Algebra (2nd edition)*, Cambridge University Press. <https://shoup.net/ntb> , 2008.
- [10] Spivak, M. *Διαφορικός και ολοκληρωτικός λογισμός*, (Εκδ. Π.Ε.Κ.). <https://www.cup.gr/book/diaforikos-ke-oloklirotikos-logism/>, 2020.
- [11] Stein, W. *Elementary Number Theory : Primes, Congruences, and Secrets*. <http://wstein.org/ent/>, 2011.
- [12] Wagstaff, Samuel. *The joy of factoring*. Student Mathematical Library, vol. 68, 2013. AMS.
- [13] Πουλάκης, Δ. *Θεωρία Αριθμών*, Εκδ. Ζήτη, 2009.
- [14] Πουλάκης, Δ. *Υπολογιστική Θεωρία Αριθμών*, Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. <http://dx.doi.org/10.57713/kallipos-323>, 2015.
- [15] Τζανάκης, Νίκος. *Θεμελιώδης Θεωρία Αριθμών*, Σημειώσεις, Πανεπιστήμιο Κρήτης, 2012.
- [16] Ζάχος, Ε.; Παγουρτζής, Α.; Γροντάς, Π. *Υπολογιστική Κρυπτογραφία*, Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. <http://dx.doi.org/10.57713/kallipos-492>, 2015.
- [17] Κολουντζάκης, Μ.; Παπαχριστόδουλος, Χ. *Διακριτά μαθηματικά*, Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. <http://dx.doi.org/10.57713/kallipos-517>, 2015.

Κεφάλαιο 9

Τεστ Πιστοποίησης πρώτων αριθμών

Σύνοψη - Περίληψη. Σε αυτό το κεφάλαιο παρουσιάζουμε κάποιους αλγορίθμους που χρησιμοποιούμε, για να ελέγχουμε αν ένας φυσικός αριθμός είναι πρώτος. Οι αλγόριθμοι αυτοί είναι πιθανοτικοί και αποφαινόμενοι με κάποια πιθανότητα για το αποτέλεσμα. Αυτοί οι αλγόριθμοι είναι απαραίτητοι, γιατί πάρα πολλά κρυπτοσυστήματα δημόσιου κλειδιού απαιτούν να γνωρίζουμε μεγάλους πρώτους. Συνήθως μεγάλος πρώτος στην κρυπτογραφία θεωρείται ένας πρώτος αριθμός με τουλάχιστον 2048 δυαδικά ψηφία. Θα δούμε το τεστ του Fermat και το τεστ των Miller-Rabin. Το δεύτερο χρησιμοποιείται και σήμερα κατά την παραγωγή πρώτων αριθμών, για παράδειγμα, από το openssl. Όλα τα κριτήρια που χρησιμοποιούμε στην πράξη είναι πιθανοτικά. Ένας ντετερμινιστικός αλγόριθμος παρουσιάστηκε το 1983 από τους Adleman, Pomerance και Rumely. Ο αλγόριθμος αυτός για μεγάλους ακέραιους n είναι χρόνου $(\ln n)^{c \ln \ln \ln n}$. Αρκετά αργότερα, ένα πολύ ισχυρό αποτέλεσμα των Agrawal, Kayal, Saxena, λέει ότι υπάρχει ντετερμινιστικός αλγόριθμος πολυωνυμικού χρόνου, για να ελέγξουμε αν ένας φυσικός αριθμός είναι πρώτος⁸⁰. Στην πράξη αυτός ο αλγόριθμος είναι αρκετά αργός, παρότι είναι πολυωνυμικού χρόνου.

⁸⁰ Στην ελληνόγλωσση βιβλιογραφία υπάρχει η διπλωματική : <https://apothesis.eap.gr/handle/repo/39035> όπου παρουσιάζεται η απόδειξη των AKS.

Προαπαιτούμενη γνώση. Σε αυτό το κεφάλαιο θα χρειαστούμε βασικές γνώσεις, όπως μέγιστο κοινό διαιρέτη, ισοτιμίες, πρώτους αριθμούς (δείτε το προηγούμενο κεφάλαιο, καθώς και την αναφορά [3]), και βασικές γνώσεις για διακριτές πιθανότητες ([6, Κεφ. 11] και [4]).

9.1 Τεστ Πιστοποίησης του Fermat και τάξη ακεραίου

Ο Euler έχει αποδείξει ότι:

Θεώρημα 9.1.1 (Euler) Για κάθε θετικό αέριο m πρώτο προς τον αέριο 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 μια μέθοδο υπολογισμού αντίστροφων στοιχείων $\pmod{m}$.

Άσκηση 9.1 Έστω a, m θετικοί αέριοι. Αν $\gcd(a, m) = 1$ και $n_1 \equiv n_2 \pmod{\phi(m)}$, τότε

$$a^{n_1} \equiv a^{n_2} \pmod{m}.$$

Υπόδ. Εφαρμογή του Θεωρήματος του Euler.

Ειδικότερα, για $m = p$ πρώτο και $p \nmid a$ προκύπτει

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

Ισοδύναμα γράφεται

$$a^p \equiv a \pmod{p}, \tag{9.1.1}$$

για κάθε αέριο a και πρώτο p .

Απόδειξη του Μικρού θεωρήματος του Fermat. Παρατηρούμε ότι το σύνολο

$$\mathcal{A} = \{ja \in \mathbb{Z}_p : j = 1, 2, \dots, p-1\}$$

αποτελείται από διακριτά στοιχεία. Πράγματι, αν $ja \equiv ia \pmod{p}$, προκύπτει $j \equiv i \pmod{p}$. Αλλά $i, j < p$ όποτε, $p \mid i - j$ 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 9.2 Το μικρό θεώρημα του Fermat μας λέει ότι $p \mid 2^{p-1} - 1$ (για p περιττό πρώτο). Μερικές φορές το $p^2 \mid 2^{p-1} - 1$. Οι πρώτοι αριθμοί με αυτήν την ιδιότητα ονομάζονται Wieferich primes. Να βρείτε όλους τους Wieferich πρώτους που είναι < 2000 . Είναι ανοιχτό πρόβλημα στη θεωρία αριθμών το αν υπάρχουν άπειροι Wieferich πρώτοι αριθμοί.

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

Ορισμός 9.1.1. Αν $\gcd(a, m) = 1$, ονομάζουμε τάξη του $a \pmod{m}$ τον ελάχιστο φυσικό αριθμό n που ικανοποιεί την ιστιμία

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

Την τάξη του στοιχείου $a \pmod{m}$ τη συμβολίζουμε $\text{ord}_m(a)$.

Άσκηση 9.3 Αν $a \equiv b \pmod{m}$, τότε $\text{ord}_m(a) = \text{ord}_m(b)$.

Ισχύει $\text{ord}_m(a) | \phi(m)$. Πράγματι, αν θέσουμε $d = \text{ord}_m(a)$, 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,

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

Λόγω 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, $\text{ord}_p(a) | p - 1$. Αν

$$\text{ord}_m(a) = \phi(m),$$

τότε ο a καλείται *αρχική ρίζα* (primitive root) modulo m (δείτε και την Παρατήρηση 8.4.1).

Ένα εύλογο ερώτημα που προκύπτει είναι πώς υπολογίζουμε την τάξη ενός ακεραίου mod n . 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 κάποιο φυσικό αριθμό i . 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 $g^{18} \equiv 1 \pmod{p}$ και να το υψώνουμε σε δυνάμεις. Αν παράγει όλους τους αριθμούς της ομάδας, τότε είναι το ζητούμενο. Ας δοκιμάσουμε το 2. Τότε η

$$\{2^i : i = 1, 2, \dots, 18\} = \{1, 2, 3, \dots, 18\}.$$

Αν αντί του 19 είχαμε έναν πρώτο 100— bits, τότε η προηγούμενη διαδικασία θα ήταν πολύ χρονοβόρα. Δίνουμε χωρίς απόδειξη το παρακάτω λήμμα.

Λήμμα 9.1.1. Το g έχει τάξη n , αν και μόνο αν $g^n \equiv 1 \pmod{p}$ και $g^{n/p} \not\equiv 1 \pmod{p}$ για κάθε πρώτο p διαιρέτη του n .

Άσκηση 9.4 Υλοποιήστε τον αλγόριθμο 5.3 του [5], για την εύρεση αρχικών ριζών mod p , όπου p πρώτος. Αν για κάποιον πρώτο p βρείτε π.χ. 2 τότε θα πρέπει να ισχύει $\{2^i : i = 1, \dots, p - 1\} = \{1, 2, \dots, p - 1\}$. Επαληθεύστε την προηγούμενη ισότητα συνόλων για τις αρχικές ρίζες που βρήκατε. Τι πολυπλοκότητα έχει ο αλγόριθμός σας;

Άσκηση 9.5 Ν.α.ό. για θετικούς ακεραίους n, a ισχύει $n | \phi(a^n - 1)$.

Υπόδ. Αρκεί $\text{ord}_{a^n-1}(a) = n$.

Άσκηση 9.6 Ν.α.ό. για πρώτο p και φυσικό $n \nmid (a - 1)$ και $a^p \equiv 1 \pmod{n}$, τότε $\text{ord}_n(a) = p$.

Υπόδ. $\text{ord}_n(a) = 1$ ή $p \dots$

Άσκηση 9.7 Έστω p πρώτος. Ν.α.ό. $\text{ord}_p(ab) = \text{ord}_p(a) + \text{ord}_p(b)$ για όλους τους ακεραίους a, b .

Άσκηση 9.8 Έστω p πρώτος με $p | M_n$, $M_n = 2^n - 1$. Ν.α.ό. $\text{ord}_p(M_{kn}) = \text{ord}_p(M_n) + \text{ord}_p(k)$.

Άσκηση 9.9 Ν.α.ό. $\gcd(M_n, M_m) = M_{\gcd(n, m)}$. ($M_n = 2^n - 1$).

Άσκηση 9.10 Ν.α.ό. $a^x \equiv a^y \pmod{n} \Rightarrow x \equiv y \pmod{\text{ord}_n(a)}$.

Άσκηση 9.11 Ν.δ.ό. δεν υπάρχει ακέραιος $n \geq 2$ με $n | 2^n - 1$.

Άσκηση 9.12 (*) Ν.α.ό. 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 9.1.1. : Τεστ του Fermat

Είσοδος. Τυχαίο ακέραιο n

Έξοδος. Αν είναι σύνθετος, εξάγει True, διαφορετικά Probable prime with respect a

```

1  $a \xleftarrow{\$} \{2, \dots, n-1\}$ 
2 if  $\gcd(a, n) > 1$  then
3   | return True
  end
4 if  $a^{n-1} \equiv 1 \pmod{n}$  then
5   | return Probable prime with respect  $a$ :  $n$  πιθανός πρώτος
  else
6   | return True
  end
```

Στη γραμμή 2, εφόσον ο $a < n$ και $\gcd(a, n) > 1$, προκύπτει ότι ο n είναι σύνθετος. Στη γραμμή 4 του αλγορίθμου υπάρχει μια πιθανότητα το αποτέλεσμα να είναι λάθος, δηλαδή ο αριθμός να είναι σύνθετος. Γι' αυτόν τον λόγο τους σύνθετους αριθμούς που ικανοποιούν την ισοτιμία (9.1.1) τους ονομάζουμε *ψευδοπρώτους αριθμούς του Fermat* ως προς τη βάση a (*Fermat pseudoprimes with respect a*). Γενικά, η έξοδος του προηγούμενου αλγορίθμου είναι ένας πιθανός πρώτος (*probable prime*).

9.2 Ψευδοπρώτοι και αριθμοί Carmichael

Ορισμός 9.2.1. Ένας σύνθετος αριθμός n που ικανοποιεί την ισοτιμία

$$a^n \equiv a \pmod{n} \quad (9.2.1)$$

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

Ορισμός 9.2.2. Ορίζουμε στη 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 $\text{ord}_m(a) | \lambda(m)$. Ο $\lambda(m)$ είναι η μεγαλύτερη τάξη $\text{mod } m$ που μπορούμε να έχουμε. Αν $n = p_1 p_2 \cdots p_r$, τότε

$$\lambda(n) = \text{lcm}(p_1 - 1, p_2 - 1, \dots, p_r - 1).$$

Άσκηση 9.13 Να υπολογιστούν τα $\phi(65520)$, $\lambda(65520)$.

Για παράδειγμα, ο 91 είναι ψευδοπρώτος ως προς τη βάση 3, διότι $3^{91} \equiv 3 \pmod{91}$ και $91 = 7 \cdot 13$, σύνθετος. Επίσης, ο αριθμός 341 είναι ψευδοπρώτος αριθμός του Fermat ως προς τη βάση 2. Γενικά, υπάρχουν άπειροι ακέραιοι που είναι ψευδοπρώτοι αριθμοί του Fermat ως προς μια βάση a (για $a \geq 2$)⁸¹. Η περίπτωση $a = 1$ είναι αδιάφορη, διότι όλοι οι θετικοί ακέραιοι είναι ψευδοπρώτοι ως προς τη βάση $a = 1$. Αν $\gcd(a, n) = 1$, τότε μπορούμε να απλοποιήσουμε το a στη σχέση (9.2.1). Τώρα θα αποδείξουμε ότι υπάρχουν άπειροι ψευδοπρώτοι αριθμοί του Fermat ως προς μια βάση $a \geq 2$.

⁸¹https://de.wikibooks.org/wiki/Pseudoprimezahlen:_Tabelle_Fermatsche_Pseudoprimezahlen

Θεώρημα 9.2.1 Για κάθε ακέραιο $a \geq 2$ υπάρχουν άπειροι ψευδοπρώτοι ως προς τη βάση a .

Χρειαζόμαστε ένα βοηθητικό λήμμα.

Λήμμα 9.2.1. (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 $p \nmid a^2 - 1$, 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

$$\begin{aligned} N &= \frac{a^2}{a^2-1}(a^{2p-2} - 1) = \\ &= \frac{a^2}{a^2-1}(a^2 - 1)(a^{2(p-2)} + a^{2(p-3)} + \dots + 1) = \\ &= a^{2p-2} + a^{2p-4} + \dots + a^2. \end{aligned}$$

Άρα, ο αριθμός N είναι άρτιος ακέραιος, 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 $p \nmid a^2 - 1$. Το ζητούμενο έπεται. □

Απόδειξη του θεωρήματος 9.2.1. Έστω $a \geq 2$. Ας είναι p περιττός πρώτος τέτοιος ώστε $p \nmid a^2 - 1$. 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

$$n = A \cdot B,$$

όπου

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

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 9.2.1 (ii) έχουμε

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

Από λήμμα 9.2.1 (iii) επίσης έχουμε

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

Επομένως, από την Πρόταση 8.3.1, $2p | n - 1$. 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,

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

□

Οπότε, πρέπει να εκτελέσουμε τον αλγόριθμο 9.1.1 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. Στο επόμενο θεώρημα θα μελετήσουμε την πιθανότητα αποτυχίας του Test.

9.2.1 Πιθανότητα αποτυχίας του Test του Fermat

Δίνουμε το παρακάτω θεώρημα (δείτε [2]).

Θεώρημα 9.2.2 Έστω $n \geq 2$ και υπάρχει b με $\gcd(b, n) = 1$ και $b^{n-1} \not\equiv 1 \pmod{n}$. Τότε,

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

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 την 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 ανά δύο μεταξύ τους και ισχύει

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

Επίσης, $|\mathcal{A}| \leq |\mathcal{B}|$. Πράγματι η απεικόνιση

$$\Phi_b : \mathcal{A} \rightarrow \mathcal{B},$$

$$\Phi_b(x) = bx \bmod n$$

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 (9.1.1) στο βήμα 3 να επιστρέψει έναν probable prime, ενώ είναι σύνθετος, είναι, $< \frac{1}{2^k}$. Μπορούμε να μετατρέψουμε τον αλγόριθμο (9.1.1) σ' έναν Monte Carlo αλγόριθμο, ως εξής:

Αλγόριθμος 9.2.1. : 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  $i \leq k$  do
4       $a \xleftarrow{\$} \{2, \dots, n-1\}$ 
5      if  $\gcd(a, n) > 1$  then
6          return  $n$  : Σύνθετος
7      end
8      if  $a^{n-1} \equiv 1 \pmod{n}$  then
9           $j = j + 1$ 
10         if  $j = k$  then
11             return  $n$  : Πρώτος με πιθανότητα  $1 - 2^{-k}$ 
12         end
13     end
14      $i \leftarrow i + 1$ 
15 end
16 return  $n$  : Σύνθετος

```

Παρατήρηση 9.2.1. Με λίγη θεωρία ομάδων, επίσης μπορούμε να αποδείξουμε το προηγούμενο θεώρημα. Έστω $\mathbb{Z}_n^*$ η ομάδα όλων των αντιστρέψιμων στοιχείων $\text{mod } n$. Ας είναι $G = \{x \in \mathbb{Z}_n^* : x^{n-1} \equiv 1 \pmod{n}\}$. Η G είναι υποομάδα της $\mathbb{Z}_n^*$. Από το Θεώρημα του Lagrange ([1, Θεώρημα 31.15])

$$|G| \mid |\mathbb{Z}_n^*|$$

διότι 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,

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

9.2.2 Αριθμοί Carmichael

Υπάρχει η περίπτωση ο προηγούμενος αλγόριθμος να δώσει έξοδο πρώτος, αλλά ο n να είναι σύνθετος, όσο και αν αυξήσουμε το k (δηλαδή τις επαναλήψεις). Άρα, υπάρχει περίπτωση ένας αριθμός n να είναι ψευδοπρώτος αριθμός του Fermat ως προς όλες τις βάσεις $a \in \{2, 3, \dots, n-1\}$.

Ορισμός 9.2.3. (Carmichael). Ένας σύνθετος αριθμός $n > 1$ που ικανοποιεί την ιστιμία,

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

για κάθε θετικό ακέραιο a , ονομάζεται αριθμός του Carmichael⁸².

Αν $\gcd(a, n) = 1$, τότε $a^{n-1} \equiv 1 \pmod{n}$. Δηλαδή, 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 9.16). Με άλλα 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, από τον οποίο και πήραν το όνομά τους αυτοί οι αριθμοί. Αυτός είναι ο πρώτος αριθμός Carmichael που βρέθηκε.

⁸²Μερικές φορές τους ονομάζουμε και absolute Fermat Pseudoprimes.

Μερικοί ακόμη αριθμοί του Carmichael είναι οι $\{1105, 1729, 2465, 2821, 6601, 8911\}$ ⁸³. Ο αριθμός 1729 είναι γνωστός και ως taxicab number ή αριθμός του Ramanujan. Υπάρχει περίπτωση ο αλγόριθμος 9.2.1 να δώσει ως έξοδο σύνθετος, για την περίπτωση ενός Carmichael, στην περίπτωση όπου $\gcd(a, n) > 1$.

Άσκηση 9.14 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 H/T, 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 τρόπους⁸⁴. 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 9.2.3 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. [3, 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 (δηλαδή, θα αποτύχει για κάθε $a \in \{2, \dots, n-1\}$).

Παρατήρηση 9.2.2. Ο Erdős το 1956 απέδειξε ότι για μεγάλα X ισχύει

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

Παρατήρηση 9.2.3. 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 πως οι αριθμοί Carmichael είναι σπάνιοι, αλλά πειράματα που έχουν γίνει μας επιτρέπουν να ισχυριστούμε ότι είναι σπάνιοι⁸⁵.

Άσκηση 9.15 (*) (Korselt, 1899). Ν.α.ο. ο $N > 2$ είναι ένας αριθμός του Carmichael, αν και μόνο αν ο N είναι σύνθετος, ελεύθερος τετραγώνων⁸⁶ και για κάθε πρώτο $p|N$, ο $p-1|N-1$.

Άσκηση 9.16 (*) Έστω Λ ένας θετικός ακέραιος και

$$\mathcal{P}_\Lambda = \{p : p \text{ πρώτος } p-1|\Lambda, p \nmid \Lambda\}.$$

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

Άσκηση 9.17 Ν.α.ο. ο 341 είναι ψευδοπρώτος αριθμός του Fermat ως προς τη βάση $a = 2$. Επίσης, ν.α.ό. δεν είναι Carmichael. Ο αριθμός 3914864773 είναι ψευδοπρώτος αριθμός του Fermat ως προς τη βάση 2;

⁸³<https://oeis.org/A002997>

⁸⁴<https://oeis.org/A011541>

⁸⁵<http://www.cecm.sfu.ca/Pseudoprimes/index-2-to-64.html> περιέχει όλους τους Carmichael αριθμούς που είναι $< 2^{64}$

⁸⁶δηλαδή, είναι της μορφής $N = p_1 p_2 \cdots p_r$, όπου p_i πρώτοι, όλοι διαφορετικοί μεταξύ τους. Γενικά, για τον έλεγχο αν ένας ακέραιος είναι ελεύθερος τετραγώνων, χρησιμοποιούμε τον αλγόριθμο των Pollard-Strassen ο οποίος έχει πολυπλοκότητα $O(N^{1/6} \ln N)$. Πρόσφατα βελτιώθηκε σε $O(N^{1/8} \ln N)$ από τους Harvey-Hittmeir.

Άσκηση 9.18 Ν.α.ο. οι αριθμοί 9999109081, 6553130926752006031481761 είναι αριθμοί του Carmichael. Μπορείτε να βρείτε κάποιον μεγαλύτερο;

Υποδ. Να γίνει χρήση του κριτηρίου Korselt, δείτε την Άσκηση 9.15.

Άσκηση 9.19 Ν.α.ο. ο 561 είναι ο μικρότερος αριθμός Carmichael με τρεις πρώτους παράγοντες. Βρείτε τον μικρότερο Carmichael με τέσσερις πρώτους παράγοντες.

Άσκηση 9.20 Με το κριτήριο του Korselt να αποδείξετε ότι, αν N είναι ένας αριθμός του Carmichael, τότε είναι περιττός.

Άσκηση 9.21 Με το κριτήριο του Korselt να αποδείξετε ότι, αν N είναι ένας αριθμός του Carmichael, τότε έχει τουλάχιστον τρεις πρώτους διαιρέτες.

Άσκηση 9.22 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 9.23 Βρείτε έναν ψευδοπρώτο ως προς τη βάση 10 που είναι μεγαλύτερος από το 10000.

Άσκηση 9.24 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. $X = 100, 200, \dots, 10000$) να βρείτε το $|\Sigma(X)|$ με χρήση του Τεστ του Fermat. Η θεωρία λέει ότι $|\Sigma(X)| = O(X/\ln(X))$.

Άσκηση 9.25 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 τη βάση 2 και 3.

Άσκηση 9.26 Να δείξετε ότι ο αριθμός 8911 είναι αριθμός Carmichael και ισχυρός ψευδοπρώτος ως προς τη βάση 3.

Άσκηση 9.27 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, δηλαδή, 8281...504948...10987654321.

Άσκηση 9.28 Ικανοποιούν το Τεστ του Fermat οι αριθμοί

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

Παρουσιάστε κώδικα, καθώς και το αποτέλεσμα του κώδικα μαζί με τον χρόνο που χρειάστηκε. Ζευγάρια πρώτων αριθμών της μορφής $(p, p + 2)$ ονομάζονται δίδυμοι πρώτοι. Είναι ανοιχτό πρόβλημα αν υπάρχουν άπειροι δίδυμοι πρώτοι. Έχει αποδειχτεί από τον 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

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

Ο Zhang απέδειξε ότι

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

Αργότερα ο Maynard βελτίωσε το άνω φράγμα στο 600 και το polymath project⁸⁷ στο 246.

⁸⁷DHJ Polymath, Variants of the Selberg sieve, and bounded intervals containing many primes

Άσκηση 9.29 Ικανοποιεί το Τεστ του Fermat ο αριθμός

$$3714089895285 \cdot 2^{60000} - 1;$$

Ν.α.ό. είναι Sophie Germain prime. Δηλαδή, και ο $2p + 1$ είναι επίσης πιθανός πρώτος. Πόσο χρόνο χρειάστηκε ο κώδικας σας;

Άσκηση 9.30 Ο Brun απέδειξε ότι η σειρά

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

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

Άσκηση 9.31 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.

Άσκηση 9.32 Αποδείξτε (υπολογιστικά) ότι όλοι οι πρώτοι μεταξύ 1000 και 2000 είναι άθροισμα τριών πρώτων (όχι κατά 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). Το 2013 ο H. A. Helfgott ανέβασε στο arxiv⁸⁸ μία απόδειξη ότι όλοι οι περιττοί αριθμοί $> 10^{30}$ είναι άθροισμα τριών πρώτων (η πρόταση ότι κάθε περιττός μεγαλύτερος του 5 είναι άθροισμα τριών πρώτων καλείται weak Goldbach conjecture). Η απόδειξη ότι και όλοι οι περιττοί $\leq 10^{30}$ είναι άθροισμα τριών πρώτων είναι πολύ απλή υπολογιστική άσκηση. Σήμερα όλοι αποδέχονται ως σωστή την απόδειξη του H. A. Helfgott. Δηλ. η weak Goldbach conjecture αποδείχτηκε.

Άσκηση 9.33 Υπολογίστε τις τιμές του πολυωνύμου $f(x) = x^2 + x + 41$ για $x = 0, 1, \dots, 40$. 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 9.34 Χρησιμοποιώντας αριθμούς της μορφής

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

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

Άσκηση 9.35 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 $\equiv 1 \pmod{n}$.

Άσκηση 9.36 Ν.α.ό.

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

⁸⁸<https://arxiv.org/pdf/1312.7748.pdf>, <https://arxiv.org/pdf/1501.05438.pdf>

Άσκηση 9.37 Όταν ήταν 14 χρονών ο Gauss, διατύπωσε την εικασία

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

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

Άσκηση 9.38 Βρείτε 5 πρώτους της μορφής $a^2 + b^4$. Έχει αποδειχτεί ότι υπάρχουν άπειροι πρώτοι αυτής της μορφής (Friedlander & Iwaniec, 1998).

Άσκηση 9.39 Αποδείξτε ότι, αν $a|b$, τότε $\lambda(a)|\lambda(b)$ (όπου $\lambda(n)$ η συνάρτηση του Carmichael, ορισμός 9.2.2).

Άσκηση 9.40 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 9.41 (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.ό. $M_p = 2^p - 1$ 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 τη χρήση υπολογιστή).

9.3 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, τη θεωρούμε αληθή (το πιθανότερο να είναι). Κατόπιν, ο Rabin αφαίρεσε αυτήν 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 θεωρήματος μας δίνει μια συνθήκη για φυσικούς n που ικανοποιούν την ισοτιμία $a^n \equiv a \pmod{n}$, να είναι σύνθετοι.

Θεώρημα 9.3.1 Έστω ότι ο n είναι ένας περιττός πρώτος και $n - 1 = 2^s t$, όπου t περιττός και s θετικός ακέραιος. Έστω ένας ακέραιος $a \in \{1, 2, \dots, n - 1\}$ με $a \not\equiv 1 \pmod{n}$. Θέτουμε $b \equiv a^t \pmod{n}$. Τότε ισχύει το παρακάτω:

$$b \equiv 1 \pmod{n} \text{ ή } b^{2^i} \equiv -1 \pmod{n} \quad (9.3.1)$$

για κάποιο i με $0 \leq i \leq s - 1$.

Ορισμός 9.3.1. Ένας περιττός ακέραιος αριθμός $n > 3$ λέγεται ισχυρός πιθανός πρώτος ως προς τη βάση a με $1 < a < n - 1$ (*strong probable prime*), αν ισχύει η συνθήκη (9.3.1). Αν είναι σύνθετος και ικανοποιεί την (9.3.1), καλείται ισχυρώς ψευδοπρώτος αριθμός του Fermat ως προς τη βάση a (*strong Fermat pseudoprime to base a*).

Για παράδειγμα, ο 341, που είναι ψευδοπρώτος ως προς τη βάση 2, δεν είναι ισχυρός ψευδοπρώτος ως προς τη βάση 2. Άρα, ένας αλγόριθμος που χρησιμοποιεί τους ισχυρούς ψευδοπρώτους και όχι ψευδοπρώτους αριθμούς του *Fermat*, στην περίπτωση του 341 θα εξάγει: σύνθετος για τη βάση $a = 2$. Ο ακέραιος 2047 είναι ισχυρός ψευδοπρώτος ως προς τη βάση 2. Πράγματι, $n = 2047$ είναι σύνθετος, διότι $n = 23 \times 89$ και $n - 1 = 2 \times 1023$, άρα $s = 1, t = 1023$. Υπολογίζουμε το $b = a^t \equiv 1 \pmod n$. Δηλ. ικανοποιεί το Τεστ των Miller-Rabin για την βάση $a = 2$ και είναι σύνθετος.

Αλγόριθμος 9.3.1. 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  $i \leq k$  do
5       $a \xleftarrow{\$} \{2, \dots, n - 1\}$ 
6       $b \leftarrow a^t \pmod n$ 
7      if  $b \equiv 1 \pmod n$  then
8           $j = j + 1$ 
9          if  $j = k$  then
10             return  $n$  : strong probable prime
11             end
12         end
13     for  $0 \leq r \leq s - 1$  do
14         if  $b \equiv -1 \pmod n$  then
15              $j = j + 1$ 
16             if  $j = k$  then
17                 return  $n$  : strong probable prime
18                 end
19             end
20          $b \leftarrow b^2 \pmod n$ 
21     end
22      $i \leftarrow i + 1$ 
23 end
24 return  $n$  : Σύνθετος

```

Στις γραμμές 7 και 12, ο αλγόριθμος ελέγχει τις υποθέσεις του προηγούμενου θεωρήματος. Αν ο αλγόριθμος εισέλθει στη γραμμή 8, τότε δεν θα εισέλθει στη 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 (9.3.1) k -φορές, τότε ο αλγόριθμος εξάγει strong probable prime. Για να αναλύσουμε την πιθανότητα επιτυχίας του αλγορίθμου χρειαζόμαστε το παρακάτω θεώρημα.

Θεώρημα 9.3.2 (Monier και Rabin). Έστω n περιττός φυσικός $\neq 9$. Τότε, το πλήθος των $a \in \{1, 2, \dots, n - 1\}$ με $\gcd(a, n) = 1$, για τα οποία ο n είναι ισχυρός πιθανός πρώτος ως προς τη βάση a είναι $\leq \frac{\phi(n)}{4}$.

Από το θεώρημα έπεται ότι η πιθανότητα ένας σύνθετος να περάσει το τεστ των Miller-Rabin είναι

$$\leq \frac{\text{number of bad } a\text{'s}}{\text{all possible } a\text{'s}} = \frac{\phi(n)/4}{\phi(n)} = \frac{1}{4}.$$

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 $\leq 2^{-80}$.

Παρατήρηση 9.3.1. Μερικές φορές είναι χρήσιμο να κατασκευάσουμε έναν πρώτο αριθμό p τέτοιον, ώστε ο $2p + 1$ να είναι και αυτός πρώτος. Ο p ονομάζεται πρώτος της Germain⁸⁹ και ο $2p + 1$ ονομάζεται ασφαλής πρώτος (safe prime).

⁸⁹Sophie Germain, (1776-1831) Γαλλίδα μαθηματικός, φυσικός και φιλόσοφος, https://en.wikipedia.org/wiki/Sophie_Germain.

Υπάρχει η εικασία ότι υπάρχουν άπειροι πρώτοι της Germain. Μέχρι σήμερα το πρόβλημα αυτό είναι ανοιχτό. Ο μεγαλύτερος πρώτος αριθμός της Germain (μέχρι σήμερα) είναι ο

$$2618163402417 \times 2^{1290000} - 1.$$

Μια ευρετική εκτίμηση για το πλήθος των πρώτων αριθμών της Germain που είναι $\leq x$,

$$1.32 \times \frac{x}{(\ln x)^2}. \quad (9.3.2)$$

Άσκηση 9.42 Γράψτε κώδικα που να υπολογίζει το πλήθος των πρώτων αριθμών της Germain που είναι $\leq x$ για

$$x = 100, 200, 300, \dots, 10000.$$

Συγκρίνετε με τον τύπο (9.3.2) και σχεδιάστε ένα κατάλληλο γράφημα.

Άσκηση 9.43 Ν.α.ό. αν

$$p = 1068669447 \times 2^{211088} - 1,$$

τότε ο p και ο $2p+1$ περνάνε με επιτυχία το Τεστ των Miller-Rabin (ως επιτυχία θεωρούμε ότι είναι πρώτοι αριθμοί).

Άσκηση 9.44 Βρείτε τον δεύτερο μικρότερο ισχυρό ψευδοπρώτο ως προς τη βάση 32.

Άσκηση 9.45 Να εξετάσετε αν ο αριθμός Fibonacci F_{104911} είναι ισχυρός πιθανός πρώτος (δείτε άσκηση 8.37).

(Υπόδ. Ο αριθμός αυτός έχει 21925 ψηφία. Θα χρειαστείτε έναν κατάλληλο αλγόριθμο που να μπορεί να υπολογίζει δυνάμεις mod p για πολύ μεγάλους ακέραιους.)

Άσκηση 9.46 (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 τους αριθμούς του Fermat υπάρχει ντετερμινιστικός πολυωνυμικός αλγόριθμος για το αν είναι πρώτος ή όχι.

Άσκηση 9.47 Ένας πρώτος p καλείται πρώτος αριθμός της Germain, αν ο $2p+1$ είναι επίσης πρώτος. Βρείτε όλους τους πρώτους αριθμούς της Germain που είναι < 1000 .

Άσκηση 9.48 Αν διαλέξουμε στην τύχη έναν ακέραιο από το σύνολο $\{1, 2, 3, \dots, N\}$ για κάποιο φυσικό N , ποια είναι η πιθανότητα να είναι πρώτος αριθμός της Germain; Αρχικά υπολογίστε αυτήν την πιθανότητα πειραματικά και κατόπιν δείτε το πρόβλημα θεωρητικά.

Άσκηση 9.49 (Αριθμοί του Proth). Οι πρώτοι αριθμοί της μορφής $k2^n + 1$ λέγονται αριθμοί του Proth. Έχουμε, το εξής θεώρημα: Αν υπάρχει ακέραιος a τέτοιος ώστε $a^{(p-1)/2} \equiv -1 \pmod{p}$ όπου $p = k2^n + 1$ (k περιττός και $k < 2^n$), τότε ο p είναι πρώτος. Βρείτε 100 πρώτους αριθμούς του Proth που είναι μεγαλύτεροι από 10000.

9.4 Κατασκευή μεγάλων πρώτων

Η πιθανότητα να διαλέξω έναν περιττό θετικό ακέραιο με 2048-bits και αυτός να είναι πρώτος, είναι περίπου $1/1024$ (δείτε το λήμμα 8.5.1). 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 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.

Άσκηση 9.50 (i). Να κατασκευάσετε με τη μέθοδο Fermat έναν πρώτο αριθμό με 2048 bits.
(ii). Να κατασκευάσετε με τη μέθοδο Miller-Rabin έναν πρώτο αριθμό με 1024 bits.

Άσκηση 9.51 Οι αριθμοί της μορφής $n2^n - 1$ ονομάζονται Woodall numbers. Κατασκευάστε έναν πρώτο αριθμό Woodall με 100-bits, με το τεστ των Miller-Rabin. Κατόπιν εξετάστε αν ο αριθμός $2^{521} - 1$ είναι Woodall prime.

Άσκηση 9.52 Να υλοποιήσετε το τεστ των Miller-Rabin, για να παραγάγετε δύο πρώτους αριθμούς p, q , όπου ο p να έχει 2048 bits και ο q να είναι της μορφής $2p + 1$. Σε πόσο χρόνο τερμάτισε ο αλγόριθμός σας; Συγκρίνετε με τον χρόνο που κάνει το openssl (δείτε την επόμενη παρατήρηση).

Παρατήρηση 9.4.1. Ο παρακάτω κώδικας παράγει, με τη βοήθεια του openssl, Sophie Germain primes μήκους 2048 bits. Δείτε την παρατήρηση (9.3.1).

```
$time openssl dhparam -out keyfile 2048 2> /dev/null && openssl asn1parse -in keyfile
> foo && awk 'NR==2 {print substr($0,49,570)}' foo > foo_2 && rm foo && for n in $(
  (cat foo_2 | tr 'a-f' 'A-F'));do echo "obase=10;ibase=16; $n"|bc;done;rm keyfile
foo_2

real    0m16,263s
user    0m16,252s
sys     0m0,000s
21670429833515164564621687007735376210810420181017333319621605812720\
30086062442767218496371341755776841759203928647355930953373201543715\
53841361594298082017946637729215167101702985255755149788175969946630\
24228526910354179866058939886550303410445670331911984842172440763625\
22385477435224742180666757267932473068929892798527463542757847838541\
10724288126592446606631672499203186629007864357785645983789201505636\
04758968759239998655785555493663351890204103341066529298419277492906\
03832525994213432467886696017683709493269516880569185040412310875772\
38513139987707644739872111451748549747176745793159559350721192475410\
23683
```

Ο προηγούμενος αριθμός, έστω p , είναι πρώτος και ο $2p + 1$ είναι επίσης πρώτος. Ο p ονομάζεται πρώτος αριθμός της Germain και ο $2p + 1$ ασφαλής πρώτος (safe prime).

Βιβλιογραφία

- [1] Cormen, T. H; Leiserson C. E.; Rivest E. L.; Stein C. *Εισαγωγή στους αλγορίθμους*, (Τόμος II), ΠΕΚ, 2003.
- [2] Dasgupta, S; Papadimitriou, C.; Vazirani, Υ. *Algorithms*. McGraw-Hill Education, 2006.
- [3] Grandall, R.; Pommerance, C. *Prime Numbers. A Computational Perspective*. 2nd edition, pp. 596, Springer, 2005.
- [4] Μπερτσεκάς, Π. Δ.; Τσιτσικλής, Ν. Γ. *Εισαγωγή στις πιθανότητες*. Εκδ. Τζιόλας, 2010.
- [5] Πουλάκης, Δ. *Υπολογιστική Θεωρία Αριθμών*, Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. <http://dx.doi.org/10.57713/kallipos-323>, 2015.
- [6] Κολουντζάκης, Μ.; Παπαχριστόδουλος, Χ. *Διακριτά μαθηματικά*, Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. <http://dx.doi.org/10.57713/kallipos-517>, 2015.

Κεφάλαιο 10

Παραγοντοποίηση & Διακριτός λογάριθμος

Σύνοψη - Περίληψη. Σε αυτό το κεφάλαιο παρουσιάζουμε αλγορίθμους παραγοντοποίησης και εύρεσης διακριτού λογάριθμου. Όσον αφορά την παραγοντοποίηση, ξεκινάμε με τον απλό αλγόριθμο της δοκιμαστικής διαίρεσης (trial division), συνεχίζουμε με τον αλγόριθμο παραγοντοποίησης του Fermat όπου θέτουμε τις βάσεις για τον υποεκθετικό αλγόριθμο Quadratic Sieve. Αναλύουμε εκτενώς τον αλγόριθμο αυτόν, διότι είναι αρκετά σημαντικός μέχρι σήμερα. Είναι ο καλύτερος αλγόριθμος παραγοντοποίησης για ακραίους από 50 μέχρι 100 δεκαδικά ψηφία. Επίσης, είναι απλούστερος από τον Number field sieve που είναι ο καλύτερος αλγόριθμος παραγοντοποίησης που έχουμε μέχρι σήμερα.

Όσον αφορά τον διακριτό λογάριθμο, παρουσιάζουμε τον αλγόριθμο του Shanks και τον Pollard-ρ. Ο δεύτερος αποτελεί μια βελτίωση του πρώτου όσον αφορά τη μνήμη. Επίσης, στην παρουσίαση του αλγορίθμου του Pollard στην άσκηση 10.21 δίνουμε και την παραλλαγή του αλγορίθμου για παραγοντοποίηση.

Προαπαιτούμενη γνώση. Για αυτό το κεφάλαιο απαιτείται πολύ καλή γνώση θεωρίας αριθμών (δείτε, στην αγγλική βιβλιογραφία [1, 8] και στην ελληνόγλωσση [10, 12]), ειδικότερα σε ό,τι αφορά τις ισοτιμίες, τους πρώτους αριθμούς, τα συνεχή κλάσματα, τα τετραγωνικά υπόλοιπα. Επιπλέον, είναι απαραίτητες και κάποιες βασικές γνώσεις θεωρίας πολυπλοκότητας. Για τον διακριτό λογάριθμο θα χρειαστούν κάποιες γνώσεις από κυκλικές ομάδες και ειδικότερα από την πολλαπλασιαστική ομάδα ακεραίων $\text{mod } p$ δηλαδή, $\mathbb{Z}_p^*$. Επίσης, τα σύμβολα του Landau, $O()$, $o()$ (big oh, little oh) χρειάζονται, για να δώσουμε άνω φράγματα για την πολυπλοκότητα των αλγόριθμων που θα παρουσιαστούν. Όσον αφορά τον αλγόριθμο ρ του Pollard προτείνουμε το [2, Κεφ. 14].

*A mathematician is a machine
for turning coffee into theorems.*
– Paul Erdős

10.1 Παραγοντοποίηση

Υπάρχουν αρκετοί αλγόριθμοι παραγοντοποίησης. Μπορούμε να τους χωρίσουμε σε εκθετικούς και υποεκθετικούς. Επίσης, τους υποεκθετικούς μπορούμε να τους χωρίσουμε σε ευρετικούς και μη-ευρετικούς (rigorous). Οι ευρετικοί χρησιμοποιούν κάποια υπόθεση που δεν έχει αποδειχτεί, αλλά υπάρχουν ενδείξεις ότι ισχύει. Για παράδειγμα, οι ακόλουθοι αλγόριθμοι: δοκιμαστικής διαίρεσης (trial division), του Fermat, του Legendre, του $p - 1$ Pollard, του Pollard ρ και του Coppersmith είναι εκθετικού χρόνου. Αυτοί οι αλγόριθμοι αναπτύσσονται στο [11]. Όμως, οι αλγόριθμοι των συνεχών κλασμάτων των Morrison-Brillhart, του Dixon⁹⁰, του Pomerance Quadratic Sieve (QS)⁹¹, 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 του Dixon είναι μη-ευρετικός πιθανοτικός αλγόριθμος, ενώ ο αλγόριθμος συνεχών κλασμάτων είναι ευρετικός πιθανοτικός αλγόριθμος. Ο αλγόριθμος NFS προήλθε από μια ιδέα του John Pollard το 1988 που προτάθηκε για την επίλυση του διακριτού λογαρίθμου. Δηλαδή, ο NFS δουλεύει και για τα δύο προβλήματα : παραγοντοποίηση και διακριτό λογάριθμο. Θα εξηγήσουμε αναλυτικά τι σημαίνει υπο-εκθετική πολυπλοκότητα λίγο παρακάτω. Ο αλγόριθμος που είναι ο πιο αποδοτικός σήμερα, είναι ο GNFS : General Number Field Sieve και αποτελεί μία παραλλαγή του NFS.

Στην επόμενη ενότητα ξεκινάμε με τον πιο απλό αλγόριθμο παραγοντοποίησης (trial division) και συνεχίζουμε με τη μέθοδο του Fermat που αποτελεί τη βάση για πολλούς σύγχρονους αλγόριθμους παραγοντοποίησης. Κλείνουμε με τον πολύ βασικό αλγόριθμο QS, υποεκθετικού χρόνου. Τέλος, για τη μελέτη αλγορίθμων παραγοντοποίησης συστήνουμε το βιβλίο των Grandall & Pomerance : Primes [3].

10.1.1 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 $n/2$, 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 (αυτήν τη 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.

⁹⁰J. D. Dixon (1981). Asymptotically fast factorization of integers, Math. Comp. 36 (153), p. 255–260. <https://tinyurl.com/y7hlsnyf>

⁹¹C. Pomerance, The Quadratic Sieve Algorithm, <https://math.dartmouth.edu/~carlp/PDF/paper52.pdf>

Αλγόριθμος 10.1.1. : trial division
Είσοδος. n φυσικός
Έξοδος. όλοι οι πρώτοι παράγοντες του n (και την πολλαπλότητα του κάθε πρώτου)

```

1  $L = [ ]$  // is the list that we shall add all the divisors of  $n$ 
2 while  $n \equiv 0 \pmod{2}$  do
3   | Append  $L$  with 2
4   |  $n \leftarrow n/2$ 
5 end
6  $f \leftarrow 3$  // a possible divisor of  $n$ , also called trial divisor.
7 while  $f^2 \leq n$  do
8   | if  $n \equiv 0 \pmod{f}$  then
9     | Append  $L$  with  $f$ 
9     |  $n \leftarrow n/f$ 
10  | else
10  |   |  $f \leftarrow f + 2$ 
10  | end
11 end
12 if  $n \neq 1$  then
12 | Append  $L$  with  $n$ 
12 end
13 return  $L$ 

```

Η πολυπλοκότητα του αλγορίθμου με είσοδο έναν φυσικό n είναι $O(\sqrt{n}/\ln n)$ 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 γενικά αν ο μικρότερος πρώτος διαιρέτης του n είναι αρκετά μεγάλος (για τους σημερινούς υπολογιστές, ας πούμε μεγαλύτερος από 100 bits). Οπότε, ο αλγόριθμος αυτός είναι ακατάλληλος για RSA modulus.

Άσκηση 10.1 Να υλοποιήσετε (σε όποια γλώσσα προγραμματισμού επιθυμείτε) τον αλγόριθμο δοκιμαστικής διαίρεσης και να παραγοντοποιήσετε τους αριθμούς $2^{62} - 1, 2^{102} - 1$.

Άσκηση 10.2 Να υλοποιήσετε (σε όποια γλώσσα προγραμματισμού επιθυμείτε) τον αλγόριθμο δοκιμαστικής διαίρεσης και να παραγοντοποιήσετε τον αριθμό $2^{61} - 1$. Γιατί αργεί τόσο πολύ; Πώς θα μπορούσατε να βελτιώσετε τον αλγόριθμο δοκιμαστικής διαίρεσης;

Άσκηση 10.3 Να υλοποιήσετε (σε όποια γλώσσα προγραμματισμού επιθυμείτε) τον αλγόριθμο δοκιμαστικής διαίρεσης. Κατόπιν, διαλέξτε έναν τυχαίο ακέραιο με 100 bits. Θεωρούμε επιτυχία, αν ο αλγόριθμός σας παραγοντοποιήσει τον τυχαίο ακέραιο σε λιγότερο από 10 δευτερόλεπτα στον υπολογιστή σας. Εκτελέστε το προηγούμενο πείραμα 1000 φορές. Ποιο είναι το ποσοστό επιτυχίας;

Άσκηση 10.4 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) $n! = a!b!$

10.1.2 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, χωρίς βλάβη της γενικότητας υποθέτουμε ότι ο n είναι περιττός. 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

$$a_1 = (x + 1)^2 - n, a_2 = (x + 2)^2 - n, \dots, a_k = (x + k)^2 - n,$$

όπου το

$$k = \frac{A + B}{2} - \lceil \sqrt{n} \rceil.$$

Σταματάμε μόλις βρούμε κάποιο a_i ($i \in \{1, 2, \dots, k\}$) που είναι τετράγωνο. Δηλαδή, $a_i = b^2$, για κάποιον ακέραιο b και για κάποιον δείκτη $i \in \{1, 2, \dots, k\}$. Τότε, $(x + i)^2 - n = b^2$, και επομένως το

$$n = (x + i - b)(x + i + b).$$

Αν $x + i - b$ ή $x + i + b \neq 1$, τότε βρήκαμε κάποιους μη τετριμμένους διαιρέτες του n .

Ένα άνω φράγμα για το k είναι: $k < n - \lceil \sqrt{n} \rceil$. Οπότε στη χειρότερη περίπτωση απαιτούνται $O(n)$ τιμές των a_i . Δηλαδή, η bit πολυπλοκότητα του αλγορίθμου είναι εκθετική. Η μέθοδος αυτή είναι αποδοτική (δηλαδή, 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 $\sqrt{n}$. Δηλαδή, οι A και B είναι κοντά, επομένως η διαφορά $A - B$ είναι κοντά στο μηδέν. Σε αυτήν 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 γνησίως αύξουσα).

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

Αλγόριθμος 10.1.2. : Η μέθοδος του Fermat

Είσοδος. n θετικός περιττός ακέραιος

Έξοδος. Ένας μη τετριμμένος διαιρέτης του n

```

for  $\lceil \sqrt{n} \rceil \leq a \leq \lfloor (n+9)/6 \rfloor$  do
1    $b \leftarrow \sqrt{a^2 - n}$ 
2   if  $b$  είναι ακέραιος then
3     return gcd( $a - b, n$ )
   end
end
```

Ένας άλλος τρόπος να περιγράψουμε τη μέθοδο του Fermat είναι ο ακόλουθος: θεωρούμε 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 αυτήν τη 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($x + ay - z, n$).

Άσκηση 10.5 Να αποδείξετε ότι, αν $n = pq$ όπου p, q πρώτοι, η μέθοδος του Fermat χρειάζεται στη χειρότερη περίπτωση $\frac{(p-q)^2}{8\sqrt{n}} + 1$ επαναλήψεις, για να τερματίσει. Πόσο κοντά πρέπει να είναι τα p, q , ώστε ο αλγόριθμος του Fermat να χρειαστεί μόνο μία επανάληψη;

Άσκηση 10.6 Ο Euler παρατήρησε ότι, αν ένας αριθμός είναι της μορφής

$$n = x^2 + ay^2 = z^2 + aw^2,$$

τότε

$$(xw)^2 \equiv (zy)^2 \pmod{n}.$$

Αποδείξτε την τελευταία ισοτιμία. Αν $xw \not\equiv \pm zy \pmod{n}$, τότε $\gcd(xw - zy, n) > 1$. Τέλος, παραγοντοποιήστε τον ακέραιο 1000009.

Άσκηση 10.7 Αν γνωρίζατε ότι ένας ακέραιος n έχει έναν διαιρέτη κοντά στο $\sqrt[3]{n}$, πώς θα τροποποιούσατε τον αλγόριθμο του Fermat;

10.1.3 Η εκδοχή του Lehman

Ο Lehman⁹² βελτίωσε τον αλγόριθμο του Fermat από $O(n^{1/2})$ σε $O(n^{1/3} \ln(\ln n))$ χρονική πολυπλοκότητα^{93 94}. Ο Lehman αναζητάει 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 $O(n^{1/3})$.

Αλγόριθμος 10.1.3. : Η μέθοδος του Lehman

Είσοδος. n θετικός περιττός ακέραιος

Έξοδος. Ένας μη τετριμμένος διαιρέτης του n

```

1 Με δοκιμαστική διαίρεση βρες διαιρέτη  $d$  του  $n$  με  $d < n^{1/3}$ 
  for  $1 \leq k \leq \lfloor \sqrt[3]{n} \rfloor$  do
    for  $\lceil \sqrt{4kn} \rceil \leq a \leq \lfloor \sqrt{4kn} + \frac{n^{1/6}}{4\sqrt{k}} \rfloor$  do
2       $b \leftarrow \sqrt{a^2 - 4kn}$ 
3      if  $b$  είναι ακέραιος then
4        return gcd( $a - b, n$ )
      end
    end
  end
end
```

Άσκηση 10.8 Να υλοποιήσετε (σε όποια γλώσσα προγραμματισμού επιθυμείτε) τον αλγόριθμο του Lehman. Κατόπιν, διαλέξτε έναν τυχαίο ακέραιο με 100 bits. Θεωρούμε επιτυχία, αν ο αλγόριθμός σας παραγοντοποιήσει τον τυχαίο ακέραιο σε λιγότερο από 10 δευτερόλεπτα στον υπολογιστή σας. Εκτελέστε το προηγούμενο πείραμα 1000 φορές. Ποιο είναι το ποσοστό επιτυχίας;

Υπόδ. Ο προηγούμενος ψευδοκώδικας δίνει έναν διαιρέτη του n , ενώ η άσκηση ζητάει παραγοντοποίηση του n . Θα χρειαστεί να τροποποιήσετε τον ψευδοκώδικα, πριν τον υλοποιήσετε.

10.1.4 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

⁹²Lehman, *Factoring Large integers*, Math. Comp. 28 (126), 1974, <https://www.ams.org/journals/mcom/1974-28-126/S0025-5718-1974-0340163-2/S0025-5718-1974-0340163-2.pdf>

⁹³Επίσης, υπάρχει και η εξής παραλλαγή του αλγορίθμου του S. Lehman : W. B. A. Hart, *A one line factoring algorithm*. Journal of the Australian Mathematical Society, Volume 92 (1), 2012 http://wrap.warwick.ac.uk/54707/1/WRAP_Hart_S1446788712000146a.pdf

⁹⁴Να σημειώσουμε ότι έχει βρεθεί ντετερμινιστικός αλγόριθμος παραγοντοποίησης με πολυπλοκότητα $\tilde{O}(N^{1/5})$ <https://arxiv.org/pdf/2105.11105.pdf>

ιδέα να χρησιμοποιήσουμε ισοτιμίες $x^2 \equiv y^2 \pmod n$ αντί την ισότητα $n = x^2 - y^2$ είναι του Kraitchik⁹⁵ 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 $x^2 \equiv y^2 \pmod n$, κυριαρχεί σε όλους τους σύγχρονους αλγορίθμους παραγοντοποίησης. Βέβαια, στην περίπτωση που βρούμε δύο ακεραίους x, y με την προηγούμενη επιθυμητή ιδιότητα, μπορεί να μην καταφέρουμε να παραγοντοποιήσουμε το n . Αυτή η ανεπιθυμητή περίπτωση θα συμβεί, όταν $x \equiv \pm y \pmod n$. Στην περίπτωση όπου $x \equiv \pm y \pmod n$, τότε υπάρχει περίπτωση ο $\gcd(n, x - y) = 1$.

Επομένως, μας ενδιαφέρει μόνο η περίπτωση που $x \not\equiv \pm y \pmod n$. Τότε ο $\gcd(n, x - y)$ είναι $< n$, οπότε θα προκύψει ένας μη τετριμμένος διαιρέτης του n . Για να το δούμε αυτό, αποδεικνύουμε το παρακάτω λήμμα.

Λήμμα 10.1.1. Έστω ότι το n είναι RSA modulus με $n = pq$. Αν $x^2 \equiv y^2 \pmod n$ και $x \not\equiv \pm y \pmod n$, τότε $(\gcd(x - y, n), \gcd(x + y, n)) = (p, q)$ ή (q, p) .

Απόδειξη. Αρχικά παρατηρούμε ότι $pq|a$, αν και μόνο αν $p|a$ και $q|a$. Οπότε

$$n \nmid x + y, \text{ αν και μόνο αν } p \nmid x + y \text{ ή } q \nmid x + y.$$

Παρόμοια,

$$n \nmid x - y, \text{ αν και μόνο αν } p \nmid x - y \text{ ή } q \nmid x - y.$$

Από την υπόθεση του θεωρήματος το $n|(x + y)(x - y)$, αλλά το $n \nmid x \pm y$. Οπότε, ένας από τους πρώτους p, q θα διαιρεί το $x + y$ αλλά όχι και οι δύο. Το ζητούμενο έπεται. $\square$

Υπάρχει η περίπτωση, όμως, $\gcd(n, x - y) = 1$. Π.χ. αν $n = 609$, τότε οι $x = 176, y = 25$ είναι τέτοιοι, ώστε $x^2 \equiv y^2 \pmod n$ και $x \not\equiv y \pmod n$. Αλλά, $\gcd(x - y, n) = 1$. Αν οι x, y είναι τυχαίοι, τότε με πιθανότητα $1/2$ θα προκύψει μη τετριμμένη παραγοντοποίηση. Το προηγούμενο γενικεύεται για n που έχει τουλάχιστον δύο πρώτους παράγοντες.

Λήμμα 10.1.2. Έστω ότι το n έχει τουλάχιστον δύο πρώτους παράγοντες. Αν $x^2 \equiv y^2 \pmod n$ και $x \not\equiv \pm y \pmod n$, τότε

$$(\gcd(x - y, n), \gcd(x + y, n)) = (n_1, n_2)$$

για κάποιους (μη τετριμμένους) διαιρέτες n_1, n_2 του n .

Ένα εύλογο ερώτημα που προκύπτει είναι αν πάντα υπάρχουν x, y με

$$x^2 \equiv y^2 \pmod n, \quad x \not\equiv y \pmod n.$$

Ισχύει η παρακάτω πρόταση,

Πρόταση 10.1.1 Αν $n = ab$, $a, b > 1$ και $\gcd(a, b) = 1$, τότε υπάρχουν ακεραίοι x, y με

$$x^2 \equiv y^2 \pmod n, \quad x \not\equiv y \pmod n.$$

Απόδειξη. [11, Πρόταση 7.3] $\square$

⁹⁵οι ρίζες της ιδέας αυτής ανάγονται στον Gauss και στον Seelhoff. Ο Donald Knuth στον δεύτερο τόμο του διασημού Art of Computing Programming αποδίδει αυτήν τη μέθοδο στον Legendre. Παρόλα αυτά, αυτό δεν είναι σωστό, όπως εξηγούν στην ενότητα 6, οι Morrison και Brillhart, στην εργασία τους: A Method of Factoring and the Factorization of F_7 .

Θέτουμε

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

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 ένα RSA modulus, τότε $|\mathcal{B}_n| = |\mathcal{A}_n|/2$ (δείτε την άσκηση 10.10). Επομένως, τουλάχιστον οι μισοί αριθμοί που ικανοποιούν την $x^2 \equiv y^2 \pmod{n}$, μας οδηγούν στην παραγοντοποίηση του n με πιθανότητα $1/2$, αν υποθέσουμε ότι διαλέγουμε τυχαία x, y από το $\mathcal{A}_n$.

Η δεύτερη ιδέα του Kraitchik. Βρήκε έναν έξυπνο τρόπο να ψάχνει για αριθμούς x, y , με την ιδιότητα:

$$x^2 \equiv y^2 \pmod{n}.$$

Η μέθοδος του Fermat ξεκινάει από τον μικρότερο ακέραιο k που είναι μεγαλύτερος από τον $\sqrt{n}$ και προσπαθεί να βρει τετράγωνο της μορφής $k^2 - n$. Ο Kraitchik υπολόγιζε πολλούς τέτοιους αριθμούς a_i με $x_i^2 \equiv a_i \pmod{n}$ με τελικό στόχο να βρει ένα υποσύνολο των a_i τέτοιο, ώστε

$$a_{i_1} \cdots a_{i_r} \equiv U^2 \pmod{n}.$$

Αλλά,

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

Τέλος, υπολογίζουμε τον $\gcd(U - V, n)$ και, αν βγει ένας μη τετριμμένος διαιρέτης του n , συνεχίζουμε την παραγοντοποίηση στο πηλίκο που θα προκύψει. Αν όχι, βρίσκουμε ένα νέο ζευγάρι (U, V) . Αυτές οι δύο ιδέες αναπτύχθηκαν αργότερα από τους εξής: Lehmer, Powers, Morrison, Brillhart, Dixon, Pomerance, Lenstra, Coppersmith, Pollard, Odlyzko και άλλους. Τέλος, οι ιδέες του Kraitchik οδήγησαν στους υποεκθετικούς αλγόριθμους των συνεχών κλασμάτων (η αρχή έγινε το 1931 από τους Lehmer-Powers⁹⁶) και στον quadratic sieve που θα δούμε στην υποενότητα 10.1.5.

Άσκηση 10.9 Να αποδείξετε ότι, αν p πρώτος αριθμός, τότε $|\mathcal{B}_p| = 0$.

Άσκηση 10.10 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.ό. $|\mathcal{A}_n| = 4\phi(n)$.

Άσκηση 10.11 (*) 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 10.12 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. Εκτός της γενίκευσης, που δόθηκε από τον Lehman το 1974, ο Shanks (2004) θεώρησε την εξής γενίκευση που ονομάστηκε SQUFOF : SQUare FORMs Factorization. Πρότεινε έναν νέο τρόπο, για να ψάχνει για ακραίους x, y με $x^2 \equiv y^2 \pmod{n}$ που βασίζεται σε συνεχή κλάσματα (αλλά μπορεί να παρουσιαστεί και με χρήση τετραγωνικών μορφών). Η ευρετική πολυπλοκότητά του είναι $O(n^{1/4})$. Επίσης, υπάρχει ο ντετερμινιστικός αλγόριθμος του Coppersmith που βασίζεται σε πλέγματα με πολυπλοκότητα $O(n^{1/4})$.

⁹⁶Lehmer, D.H.; Powers, R.E. (1931). *On Factoring Large Numbers*. Bulletin of the American Mathematical Society. 37 (10).

10.1.5 Αλγόριθμος του Dixon/Quadratic Sieve

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

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

Αλγόριθμοι με πολυπλοκότητα $O(L_n[c, d])$ και $0 < c < 1$ καλούνται υποεκθετικοί. Αν $c = 0$, έχουμε πολυωνυμική bit πολυπλοκότητα, $L_n[0, d] = (\ln n)^d$. Όμως, αν $c = 1$, δηλαδή, $L_n[1, d] = n^d$, έχουμε εκθετική πολυπλοκότητα. Ο πρώτος αλγόριθμος παραγοντοποίησης υποεκθετικής πολυπλοκότητας βασίζεται σε συνεχή κλάσματα και παρουσιάστηκε στις αρχές του 1970 (Morrison-Brillhart⁹⁷). Οι αρχικές ιδέες αυτού του αλγορίθμου πρώτη φορά δόθηκαν από τους Lehmer και Powers το 1931. Ο αλγόριθμος αυτός βρήκε την παραγοντοποίηση του αριθμού $F_7 = 2^{2^7} + 1 = 2^{128} + 1$, το 1975 από τους Morrison και Brillhart. Η ευρετική bit πολυπλοκότητά του είναι

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

Πριν από αυτόν τον αλγόριθμο, δηλαδή, πριν τη δεκαετία του 1970, δεν μπορούσαν να παραγοντοποιήσουν αριθμούς μεγαλύτερους από 20 ψηφία. Ο αλγόριθμος των Morrison-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 υποεκθετική πολυπλοκότητα,

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

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

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

Παρουσιάστηκε από τον Pomerance το 1981 και αρκετά αργότερα από τους Lenstra-Pomerance δόθηκε (αποδεδειγμένα) η πολυπλοκότητά του⁹⁸. 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 δεκαδικά ψηφία (περίπου 332 bits), 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 [3, 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 υπολογίσουμε

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

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 χρήση τέτοιων αριθμών στην 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.

⁹⁷Morrison, Michael A.; Brillhart, John (January 1975). *A Method of Factoring and the Factorization of F_7* . Mathematics of Computation. American Mathematical Society. 29 (129)

⁹⁸H. W. Lenstra, Jr. and C. Pomerance. *A rigorous time bound for factoring integers*. Journal of the AMS, 4, p. 483–516, 1992.

Για να καταλάβουμε πώς 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 στη μεταβλητή $x = N, N + 1, \dots$ και υπολογίζουμε τα $x^2 \bmod n$. Έτσι έχουμε

$$41^2 \equiv 32 \bmod n, 42^2 \equiv 115 \bmod n, 43^2 \equiv 200 \bmod n, 44^2 \equiv 287 \bmod n,$$

$$45^2 \equiv 376 \bmod n, 46^2 \equiv 467 \bmod n, 47^2 \equiv 560 \bmod n, 48^2 \equiv 655 \bmod n,$$

$$49^2 \equiv 752 \bmod n, 50^2 \equiv 851 \bmod n.$$

Παρατηρούμε ότι $32 \cdot 376 \cdot 752 \equiv 1 \bmod n$. Επομένως, αν $x = 41 \cdot 45 \cdot 49$, προκύπτει ότι $x^2 \equiv 1 \bmod n$ 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 στην τύχη.

Επίσης, παρατηρούμε ότι $752 \cdot 655 \cdot 467 \cdot 115 \equiv 4 \bmod n$ και, αν θέσουμε $y = 49 \cdot 48 \cdot 46 \cdot 42$, έχουμε $y^2 \equiv 4 \bmod n$ και $\gcd(y - 2, n) = 97$. Γενικά, αναζητούμε συνδυασμούς, ώστε το γινόμενο να μας δίνει τετράγωνο $\bmod n$ (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 τη γραμμική άλγεβρα μας επιτρέπει να εντοπίσουμε τετράγωνα $\bmod n$ 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 10.13 Παραγοντοποιήστε τον αριθμό 8051 το πολύ σε 5 λεπτά, χωρίς υπολογιστή.

10.1.6 B-smooth numbers

Ορισμός 10.1.1. 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 απέδειξε ότι για σταθερό $u > 0$,

$$\psi(x, x^{1/u}) \sim \rho(u)x, \quad (x \rightarrow \infty), \quad (10.1.1)$$

όπου⁹⁹ η συνάρτηση $\rho(u)$ ονομάζεται συνάρτηση του Dickman (δείτε το σχήμα 10.1). Η συνάρτηση αυτή ορίζεται για $u \geq 0$ και δεν είναι γνωστή κάποια κλειστή μορφή της για $u > 2$. Αλλά για $1 < u < 2$ η $\rho(u) = 1 - \ln u$ και για $0 \leq u < 1$ η $\rho(u) = 1$. Δείτε [3, ενότητα 1.4.5]. Από τον Hildebrand το 1986 αποδείχτηκε ότι

$$\psi(x, x^{1/u}) = \rho(u)x \left(1 + O\left(\frac{\ln(u+1)}{\ln y} \right) \right) \quad (10.1.2)$$

για

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

⁹⁹ $f(n) \sim g(n) \Leftrightarrow \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 1$.

Δηλαδή, μπορούμε να χρησιμοποιήσουμε την εκτίμηση (10.1.1) 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 (10.1.2) για το διάστημα

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

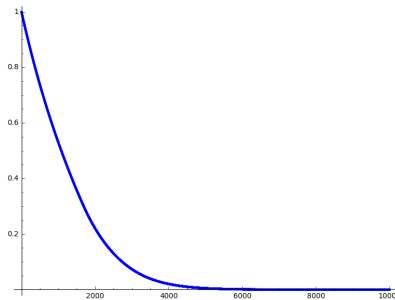
Το αποτέλεσμα του Dickman (10.1.1) μπορεί να χρησιμοποιηθεί, για να υπολογίσουμε την $\psi(x, y)$ για $x, y \rightarrow \infty$ και $u = \frac{\ln x}{\ln y}$ φραγμένο άνω ή σταθερό. Δεν μας βοηθάει όμως στον υπολογισμό, για παράδειγμα, της $\psi(x, x^{1/\ln x})$. Αυτό, διότι το $u = \frac{1}{1/\ln x} = \ln x$ δεν είναι φραγμένο. Επομένως,

$$\psi(x, y) \sim \rho\left(\frac{\ln x}{\ln y}\right)x, \quad (x, y \rightarrow \infty). \quad (10.1.3)$$

Μια ασυμπτωτική εκτίμηση που δόθηκε από τους Canfield, Erdős, Pomerance για τη συνάρτηση Dickman είναι

$$\rho(u) = u^{-u+o(u)}. \quad (10.1.4)$$

Αυτή η εκτίμηση ισχύει για μεγάλα u που ικανοποιούν την $u < (1 - \varepsilon) \ln x / \ln \ln x$.

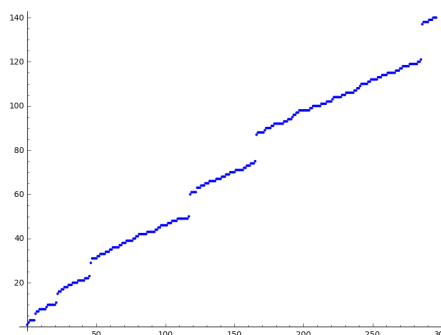


Σχήμα 10.1: Η συνάρτηση του Dickman $\rho(u)$. Παρατηρήστε ότι $\rho(u) \leq 1$ και τείνει στο μηδέν πολύ γρήγορα.

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

Πίνακας 3: $\psi(y, \sqrt{y})$

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



Σχήμα 10.2: Στον κάθετο άξονα είναι οι τιμές της συνάρτησης $\psi(y, \sqrt{y})$ και στον οριζόντιο άξονα το y

Η θεωρία προτείνει για μεγάλα y , από την (10.1.1) και (10.1.4), ότι

$$\Pr\left(x : x \text{ είναι } y^{1/2}\text{-ομαλός και } x \leftarrow^{\$} \{1, 2, \dots, y\}\right) \sim \frac{\rho(2)y}{y} \approx 0.3. \quad (10.1.5)$$

Άσκηση 10.14 Έστω x θετικός ακέραιος και $B = \sqrt[3]{x}$. Να υπολογίσετε πόσοι B -ομαλοί ακέραιοι υπάρχουν στο διάστημα $[1, x]$ για $x = 100, 200, \dots, 1000$. Υπολογίστε το ποσοστό αυτών και βρείτε τον μέσο όρο.

Άσκηση 10.15 Να επεκτείνετε τον προηγούμενο πίνακα 3, για $y = 1000, 1100, \dots, 2000$.

Άσκηση 10.16 Για μεγάλα x ισχύει $\psi(x, 4) \sim c(\ln x)^2$. 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

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

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

Υπάρχουν και μη ασυμπτωτικά κάτω φράγματα,

Πρόταση 10.1.2 (Pomerance, Konyagin, 1997). Έστω $u \in \mathbb{R}$. Για όλα τα $x \geq 4$ και $2 \leq x^{1/u} \leq x$ έχουμε

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

Μερικοί λόγοι που χρησιμοποιούμε B -ομαλούς ακέραιους είναι οι εξής:

- (i). Έχουν απλή πολλαπλασιαστική δομή (το γινόμενό τους είναι επίσης B -ομαλός).
- (ii). Δεν είναι σπάνιοι (δείτε την (10.1.5)).
- (iii). Είναι εύκολο να τους αναγνωρίσουμε. Για παράδειγμα, με δοκιμαστική διαίρεση μπορούμε γρήγορα να αναγνωρίσουμε αν ένας ακέραιος είναι B -ομαλός.

10.1.7 Ο αλγόριθμος Quadratic Sieve

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

Ο Αλγόριθμος Quadratic Sieve

Είσοδος : $n > 2$ ακέραιος που δεν είναι δύναμη πρώτου.

Έξοδος : Ένας μη τετριμμένος διαιρέτης του n .

[Initialization Step]

1. $B \leftarrow \lfloor L(n)^{1/2} \rfloor = \lfloor L[1/2, 1/2] \rfloor$, $S \leftarrow \{-1, p_1 = 2, p_2, p_3, \dots, p_t\}$ όπου $p_2 < \dots < p_t$ πρώτοι αριθμοί (p_i είναι ο i -οστός πρώτος) που είναι B -ομαλοί και το n είναι τετραγωνικό υπόλοιπο mod p_i , για κάθε $i > 1$.

[Sieving Step]

2. Για κάθε αριθμό x_i στο σύνολο $\{\lceil \sqrt{n} \rceil, \lceil \sqrt{n} \rceil \pm 1, \dots\}$ υπολογίζουμε τα $c_i = x_i^2 - n$, ώστε όλοι οι διαιρέτες του να είναι στη 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

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

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

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 $(t+1) \times (t+2)$ 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 γραμμικά 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 $\mathbf{e}(c_{i_1}) + \dots + \mathbf{e}(c_{i_r}) = \mathbf{0} \pmod{2}$ θέτουμε $x^2 = c_{i_1} \cdots c_{i_r} \pmod{n}$ και

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

Τότε, $x^2 \equiv z^2 \pmod{n}$.

7. $d \leftarrow \gcd(x - z, n)$.

8. Αν $d > 1$, επιστρέφουμε τον αριθμό $d = \gcd(x - z, n)$, διαφορετικά δηλαδή, αν $d = 1$ επιστρέψε αποτυχία.

Μερικές παρατηρήσεις.

Στο Βήμα 1.

- Απαιτούμε ο αριθμός n να μην είναι δύναμη πρώτου, ώστε να διαιρείται από τουλάχιστον δύο πρώτους.
- Από τη σχέση (10.1.5), 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 $\{1, 2, \dots, L(n)\}$. Ο λόγος που διαλέγουμε έτσι το B , δηλαδή ίσο με $L(n)^{1/2}$, είναι αρκετά πιο πολύπλοκος.

Αρχικά μπορούμε να υποθέσουμε ότι

$$Pr(1 \leq x \leq n : x^2 \pmod{n} \text{ είναι } B\text{-smooth}) \approx \psi(n, B) \sim u^{-u} \quad \left(\text{όπου } u = \frac{\ln n}{\ln B} \right).$$

Αυτή είναι μια ευρετική υπόθεση, διότι η προηγούμενη πιθανότητα είναι σωστή, αν θεωρήσουμε όλα τα x να παίρνουν τιμές σε όλο το σύνολο $\{1, 2, \dots, n\}$ και όχι στο υποσύνολό του

$$\{1 \leq x \leq n : x^2 \pmod{n} \text{ είναι } B\text{-smooth}\}.$$

Η δεύτερη παρατήρηση είναι ότι δεν χρειάζεται το x να ξεκινάει από το 1, αλλά από το $\lceil \sqrt{n} \rceil$. Επίσης, υποθέτουμε ότι σταματάμε στο $\lceil \sqrt{n} \rceil + n^\varepsilon$ για κάποιο μικρό $\varepsilon > 0$. Τότε, η νέα ευρετική γίνεται:

$$Pr(\lceil \sqrt{n} \rceil \leq x \leq \lceil \sqrt{n} \rceil + n^\varepsilon : x^2 - n \text{ είναι } B\text{-smooth}) \approx \psi(n, B) \sim u^{-u} \quad \left(\text{όπου } u = \frac{\ln n^{1/2}}{\ln B} \right). \quad (10.1.6)$$

Μένει να χρησιμοποιήσουμε το u^{-u} της ευρετικής, για να εκτιμήσουμε το B .

Έστω $T(B)$ ο συνολικός χρόνος που απαιτείται, για να ολοκληρωθεί το βήμα 2 και $T_2(B)$ ο χρόνος που απαιτείται να ελέγξουμε αν ένας ακέραιος $\leq n$ είναι B -ομαλός. Από την ευρετική (10.1.6), το πλήθος των ακεραίων x που είναι B -ομαλοί είναι περίπου u^u , ενώ το πλήθος των σχέσεων που απαιτείται είναι $t+1 = \pi(B) + 1$ και φαίνεται στο παρακάτω λήμμα 10.1.3. Άρα,

$$T(B) = u^u \times (t+1) \times T_2(B), \quad \left(u = \frac{\ln n^{1/2}}{\ln B} = \frac{\ln n}{2 \ln B} \right).$$

Αποδεικνύεται ότι $T_2(B) = \ln \ln B$, δείτε [3, Ενότητα 3.2.5]. Να σημειώσουμε ότι, αν κάνουμε δοκιμαστική διαίρεση για τον έλεγχο ενός ακεραίου, αν είναι B -ομαλός, το $T_2(B) \sim \pi(B) \sim \frac{B}{\ln B}$. Επομένως, το κόστινο που χρησιμοποιείται στην ενότητα 3.2.5 της προηγούμενης αναφοράς, είναι αρκετά σημαντικό. Τέλος, το ελάχιστο της συνάρτησης $T(B)$ προκύπτει, όταν

$$B \approx L(n)^{1/2}$$

που δίνει

$$T(B) \approx L(n).$$

Επομένως, μέχρι το βήμα 2 απαιτείται $O(L(n))$ ευρετική bit πολυπλοκότητα. Αποδεικνύεται ότι και το βήμα Linear Algebra έχει την ίδια bit πολυπλοκότητα. Οπότε, ο αλγόριθμος αυτός έχει ευρετική υποεκθετική πολυπλοκότητα $L(n) = L[1/2, 1]$. Αν χρησιμοποιήσουμε δοκιμαστική διαίρεση, η πολυπλοκότητα γίνεται $L[1/2, 3/2]$ αντί της καλύτερης $L[1/2, 1]$. Αν χρησιμοποιήσουμε και Gauss στο βήμα 5, τότε η συνολική πολυπλοκότητα είναι $L[1/2, 3/2]$. Η σχέση (10.1.6) είναι αρκετή, για να βελτιώσει τον αλγόριθμο του Dixon που έχει πολυπλοκότητα $L[1/2, 2\sqrt{2}]$.

- Στο σύνολο S δεν βάζουμε πρώτους p που το n δεν είναι τετραγωνικό υπόλοιπο mod p . Αυτό διότι, αν $p|x_i^2 - n$, τότε $x_i^2 \equiv n \pmod{p}$. 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 τετραγωνικό υπόλοιπο mod p δεν τους βάζουμε στην factor base.

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

Λήμμα 10.1.3. Αν $L = \{m_1, m_2, \dots, m_t\}$ 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 $\mathbf{e}(m) = (e_1, \dots, e_K) \in \mathbb{F}_2^K$. Τώρα θεωρούμε ένα σύνολο από B -ομαλούς ακέραιους, έστω $m_1, m_2, \dots, m_r$ για κάποιο r . Το γινόμενο $m_1 m_2 \cdots m_r$ είναι τετράγωνο, αν και μόνο αν

$$\mathbf{e}(m_1) + \mathbf{e}(m_2) + \cdots + \mathbf{e}(m_r) \equiv \mathbf{0} \pmod{2}$$

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

- Μέσα στη B -βάση παραγόντων (B-Factor Base) S υπάρχει και ο αριθμός -1 . Ο λόγος είναι διότι στο βήμα 2 μπορεί κάποιο c_i να επιλεγεί αρνητικό, ώστε να είναι B -ομαλός.

- Για να υπολογίσουμε τη 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 $z^2 \equiv x^2 \pmod{n}$.

Βιβλιογραφικά σχόλια. Υπάρχει ψευδοκώδικας στο βιβλίο των A. Menezes, P. van Orschot και S. Vanstone [6, Αλγόριθμος 3.21, σελ. 96], Mollin [7, Ενότητα 6.4] και ίσως ο πιο αναλυτικός βρίσκεται στην αναφορά [3, Ενότητα 6.1.2]. Επίσης, η αναφορά [9, ενότητα 8.2] περιέχει ένα αναλυτικό παράδειγμα και στον αλγόριθμο [9, αλγόριθμος 8.6] υπάρχει ψευδοκώδικας για το πώς βρίσκουμε B -ομαλούς ακέραιους της μορφής $f(x)$ για x ακέραιοι και $f(x) \in \mathbb{Z}[x]$, χρησιμοποιώντας ένα κόσκινο ανάλογο του Ερατοσθένη.

Ο αλγόριθμος του Dixon στο βήμα 2, χρησιμοποιεί ένα τυχαίο σύνολο θετικών ακεραίων $< N$ και χρησιμοποιεί δοκιμαστική διαίρεση, για να ελέγξει αν ένας ακέραιος είναι B -ομαλός.

10.2 Διακριτός Λογάριθμος (discrete logarithm)

Έστω p πρώτος αριθμός. Έχουμε δει ότι το σύνολο των αντιστρέψιμων στοιχείων $\bmod p$, $\mathbb{Z}_p^*$ 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 $g^m = b$. Επομένως, μπορούμε να διαλέξουμε τον $n : 0 \leq n \leq p-2$. Το θέμα που θα συζητήσουμε αφορά την επίλυση της εξίσωσης $a^x = b$ στην ομάδα $\mathbb{Z}_p^*$. Το $x = \text{dlog}_a(b)$ ονομάζεται διακριτός λογάριθμος $\bmod p$ του b με βάση a . Το πρόβλημα αυτό ονομάζεται *πρόβλημα διακριτού λογαρίθμου $\bmod p$* (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, $\text{dlog}_2 7 = 11$ στην ομάδα $\mathbb{Z}_{13}^*$. Μερικές φορές καλούμε τον διακριτό λογάριθμο $\bmod p$ και δείκτη $\bmod p$ (index και γράφουμε $\text{ind}_g(a) = x$).

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:

Έστω ότι δίνονται τα $a = g^x$, $b = g^y$, και $g \in G$, βρείτε το g^{xy} .

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 (τη 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 τον ακόλουθο ορισμό:

Ορισμός 10.2.1. Το DLP είναι δύσκολο στην ομάδα G (κυκλική ομάδα τάξης q), αν για κάθε αποδοτικό αλγόριθμο A η πιθανότητα

$$\Pr(g \xleftarrow{\$} G, x \xleftarrow{\$} \mathbb{Z}_q : A(G, g, g^x) = x) \text{ είναι αμελητέα ως προς } q. \quad (10.2.1)$$

Μια συνάρτηση $f(n)$ ονομάζεται αμελητέα (negligible) ως προς n αν για κάθε σταθερά $c > 0$, υπάρχει φυσικός N_c τέτοιος ώστε για κάθε $n > N_c$ να έχουμε $f(n) < 1/n^c$. Για παράδειγμα, αν η G είναι η ομάδα μίας ελλειπτικής καμπύλης επί ενός πεπερασμένου σώματος (και υποθέτουμε ότι είναι κυκλική), τότε το DLP πιστεύουμε ότι είναι δύσκολο. Δεν έχει αποδειχτεί ότι είναι δύσκολο, αλλά μέχρι σήμερα δεν έχει βρεθεί κάποιος αποδοτικός αλγόριθμος που να το λύνει. Παρόμοια υπόθεση γίνεται και για το κρυπτοσύστημα RSA (δείτε την ανάλογη σχέση (11.2.1)). Επίσης, επί της ομάδας $\mathbb{Z}_p^*$ για πρώτο $p \geq 2^{1023}$ το πρόβλημα του διακριτού λογαρίθμου θεωρείται δύσκολο. Ειδικότερα, το DLP επί της ομάδας μιας ελλειπτικής $\bmod p$ είναι δυσκολότερο από ό,τι στην $\mathbb{Z}_p^*$, και γι' αυτόν τον λόγο χρησιμοποιούμε μικρότερες παραμέτρους.

Παρατήρηση 10.2.1. 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 έναν πρώτο διαιρέτη q με 160 bits. Κατόπιν, διαλέγουμε ένα στοιχείο h της ομάδας με τάξη q . Π.χ. $h = g^{(p-1)/q} \bmod p$ για κάποιο $h : 2 \leq h \leq p-2$, παράγει μια κυκλική ομάδα με q στοιχεία (εκτός από τη σπάνια περίπτωση που $h = 1$). 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 τη λύση του διακριτού λογαρίθμου στην $\langle h \rangle$.

Παρατήρηση 10.2.2. Ο Peter Shor απέδειξε 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.

Άσκηση 10.17 Οι Mullen και White έχουν αποδείξει το παρακάτω θεώρημα.

Θεώρημα. Αν p πρώτος και $y = g^x$ για κάποιο y, g στο $\mathbb{Z}_p$ και x θετικός ακέραιος, τότε

$$x \equiv -1 + \sum_{i=1}^{p-2} \frac{y^i}{g^{-i} - 1} \pmod{p}.$$

Αν $p = 3571$ και $y = 2763$ και $g = 2$, υπολογίστε τον διακριτό λογάριθμο $\text{dlog}_g(y) = x$ με τον προηγούμενο τύπο.

10.2.1 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 10.2.1. : Ο αλγόριθμος του Shanks

Είσοδος. G η κυκλική ομάδα $\mathbb{Z}_p^*$, g : γεννήτορας της G , $y = g^x$.

Έξοδος. $\text{dlog}_g(y)$

```

1  $L_1 = []; L_2 = []; A = \lfloor \sqrt{p} \rfloor;$ 
2 for  $i = 0$  to  $A + 1$  do
3    $L_1 \leftarrow g^{A-i} \pmod{p}$  // giant steps
4    $L_2 \leftarrow y \cdot g^{-i} \pmod{p}$  // baby steps
end
5 if  $L_1 \cap L_2 \neq \emptyset$  then
6   Let  $B \in L_1 \cap L_2$ 
7    $\text{quotient} \leftarrow L_1.\text{index}(B)$  // returns the position of the number  $B$  in the list  $L_1$ 
8    $\text{remainder} \leftarrow L_2.\text{index}(B)$ 
9    $k \leftarrow A \cdot \text{quotient} + \text{remainder}$ 
return  $k$ 
end
```

Στην περίπτωση μας $|G| = p - 1$. Ας είναι $y \in G$ και $y = g^x$ με $x \in \{0, 1, \dots, p - 2\}$. 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

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

και

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

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 γραφτεί 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 $x \geq p$, που είναι άτοπο.

Η απαιτούμενη μνήμη, όπως φαίνεται στις γραμμές 3,4 του προηγούμενου αλγορίθμου, 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 12.2.2) μπορεί να γίνει σε $O(\sqrt{n} \log_2 n)$. Κατόπιν, εφαρμόζουμε δυαδική

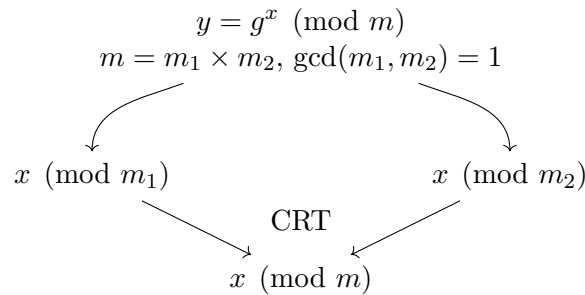
αναζήτηση (χρόνο $O(\log_2 p)$) αν κάποιο στοιχείο $x \in L_2$ ανήκει στο L_1 . Αυτό γίνεται για όλα τα στοιχεία του συνόλου L_2 . Οπότε, συνολικά έχουμε $O(\sqrt{p} \log_2 p)$ πράξεις μέσα στην ομάδα.

Το θετικό με τον αλγόριθμο του Shanks είναι η γενικότητά του (Generic algorithm). Δηλαδή, εφαρμόζεται σε οποιαδήποτε κυκλική ομάδα G . Στην περίπτωση αυτή, στις γραμμές 3,4 οι πράξεις εννοούνται μέσα στην ομάδα (δηλαδή, όχι απαραίτητα $\text{mod } p$). Επίσης, ο αλγόριθμος αυτός θα δούλευε, ακόμη και αν δεν γνωρίζαμε ακριβώς την τάξη της ομάδας G , αλλά μόνο ένα άνω φράγμα αυτής. Το μειονέκτημά του είναι η μεγάλη απαίτησή 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 10.2.3. 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 $m = m_1 m_2$ 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, $w^{m_2} \in G_1$, $w^{m_1} \in G_2$, οπότε έχουν νόημα οι διακριτοί λογάριθμοι. Έστω $x_1 = \log_a(w^{m_2})$. Τότε έχουμε (όλες οι πράξεις είναι στην ομάδα G_1):

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

Παρόμοια (για την ομάδα G_2), $x_2 \equiv x \pmod{m_2}$. Εφαρμόζουμε CRT (δείτε υποενότητα 8.7) 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).



Άσκηση 10.18 Αν $p = 3571$, $y = 2763$, $g = 2$ και υπάρχει x τέτοιο, ώστε $g^x = y$ στο $\mathbb{Z}_p$, εφαρμόστε τον αλγόριθμο του Shanks, για να βρείτε το x . Συγκρίνετε με την άσκηση 10.17.

Υπάρχει μια βελτίωση της μεθόδου του Shanks, αλλά θα χαθεί το ντετερμινιστικό της μεθόδου. Π.χ. στο βιβλίο [4], υπάρχει το ακόλουθο θεώρημα.

Θεώρημα 10.2.1 Ένα κουτί περιέχει n κόκκινες μπάλες και $N - n$ μπλε ($N > n$). Κάνουμε δειγματοληψία m -μπαλών με επανατοποθέτηση.

(i). Έστω $\mathbb{P}$ η πιθανότητα, να διαλέξουμε τουλάχιστον μία κόκκινη μπάλα. Τότε, $\mathbb{P} = 1 - (1 - \frac{n}{N})^m$.

(ii). Αν N μεγάλο, και m, n κοντά στο $\sqrt{N}$ αλλά $< 10\sqrt{N}$, τότε $\mathbb{P} \approx 1 - e^{-mn/N}$.

Απόδειξη. [4, Θεώρημα 4.38]

□

Θα επαναλάβουμε τον αλγόριθμο του Shanks, αλλά τα σύνολα L_1, L_2 θα τα κατασκευάσουμε με τυχαίο τρόπο. Αρχικά, από την εξίσωση $y = g^x$ αν θέσουμε $x = y - z$ γράφεται $yg^z = g^y$. Θα βελτιώσουμε τον χρόνο εκτέλεσης του αλγορίθμου του Shanks. Ας είναι $\mathcal{K}_p = \{1, 2, \dots, p\}$. Έστω ότι υπολογίζουμε τις παρακάτω τιμές, και φτιάχνουμε όπως και προηγουμένως ένα σύνολο (που τώρα γενικά είναι υποσύνολο του L_1),

$$L'_1 = \{g^{y_1}, g^{y_2}, \dots, g^{y_n}\} \text{ με } y_i \xleftarrow{\$} \mathcal{K}_p, i = 1, 2, \dots, n.$$

Το n θα προσδιοριστεί με ακρίβεια παρακάτω. Όλες οι πράξεις είναι mod p . Χρησιμοποιώντας την ορολογία της επίθεσης του Shanks, θέτουμε $M = \{1, g, g^2, \dots, g^{p-1}\}$ και ως χρωματίζουμε τις μπάλες του M κόκκινες αν ανήκουν στο L'_1 και τις υπόλοιπες μπλε. Σχηματίζουμε το σύνολο

$$L'_2 = \{yg^{z_1}, yg^{z_2}, \dots, yg^{z_n}\} \text{ με } z_i \stackrel{\$}{\leftarrow} \mathcal{K}_p, i = 1, 2, \dots, n.$$

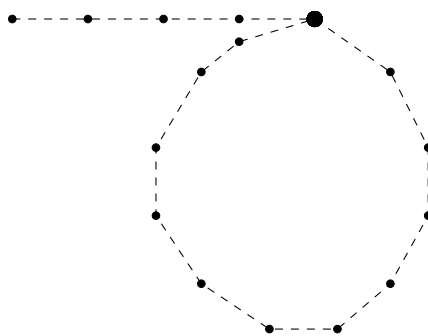
Επειδή το y είναι μια δύναμη του g επίσης το $L'_2 \subset M$.

Διαλέγουμε n μπάλες από το κουτί M (με επανατοποθέτηση) και ζητάμε την πιθανότητα να διαλέξουμε μία κόκκινη, δηλαδή ζητάμε την πιθανότητα να έχουμε μια διένεξη μεταξύ των L'_1, L'_2 . Τότε μπορούμε να χρησιμοποιήσουμε το προηγούμενο θεώρημα (για $m = n, N = p$). Οπότε η ζητούμενη πιθανότητα (για $n, m < 10\sqrt{p}$) είναι $\approx 1 - e^{-n^2/p}$. Αν διαλέξουμε το $n = 3\lfloor\sqrt{p}\rfloor$ έχουμε μια πιθανότητα κοντά στο 0.99. Επομένως έχουμε μια τυχαιοποίηση του αλγορίθμου του Shanks η οποία μας δίνει μια βελτίωση του χρόνου εκτέλεσης από $O(\sqrt{p} \log_2 p)$ σε $O(\sqrt{p})$. Η μνήμη παραμένει $O(\sqrt{p})$. Αυτή η επίθεση ανήκει στη γενικότερη κατηγορία των meet in the middle attacks.

10.2.2 Μέθοδος Pollard- ρ

Το 1978 ο John Pollard ανακάλυψε δύο μεθόδους τύπου Monte Carlo¹⁰⁰, 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 ντετερμινιστικές) και απαιτούν πολύ λιγότερη μνήμη. Ένα άλλο πλεονέκτημά 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 τη 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 $n - \text{bits}(\text{seed})$, τότε αναμένουμε μετά από $2^{n/2}$ βήματα να επαναλαμβάνονται τα προηγούμενα bits.



Σχήμα 10.3: ρ -Pollard.

Αυτό σχηματικά παρουσιάζεται όπως στο σχήμα 10.3. 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 $s + 1$ να είναι το πλήθος των σημείων που βρίσκονται στην ουρά της ακολουθίας

¹⁰⁰Ένας αλγόριθμος τύπου Monte Carlo είναι ένας πιθανοτικός πολυωνυμικός αλγόριθμος που η έξοδός 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.

$(X_i)_i$. Δηλαδή,

$$\text{tail} = \{X_0, \dots, X_s\}, \text{ circle} = \{X_{s+1}, \dots, X_{s+\ell}\}.$$

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 $X_0, X_1, \dots, X_s, \dots, X_{s+\ell} = X_s$ 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 $x < m$ τέτοιο, 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 $b_i \not\equiv b_j \pmod{m}$ τέτοιων, ώστε $g^{a_i} h^{b_i} = g^{a_j} h^{b_j}$. Τότε,

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

Αν η μετάβαση από την τιμή X_i στη 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 (μια άλλη εκδοχή του παράδοξου των γενεθλίων).

Πρόταση 10.2.1 Έστω S ένα σύνολο με N στοιχεία. Αν κάνουμε ομοιόμορφη δειγματοληψία (με επανατοποθέτηση) από το S , ο αναμενόμενος αριθμός δειγμάτων που πρέπει να έχουμε πριν κάποιο στοιχείο εμφανιστεί δύο φορές, είναι $\sqrt{\pi N/2}$.

Απόδειξη. Έστω ότι διαλέγουμε από το S ένα στοιχείο, π.χ. το x . Το ξανατοποθετούμε στο S και διαλέγουμε πάλι ένα τυχαίο στοιχείο. Η πιθανότητα να διαλέξουμε ένα διαφορετικό στοιχείο του x είναι,

$$(N - 1)/N = 1 - 1/N.$$

Το ίδιο πείραμα αν είχαμε διαλέξει ℓ στοιχεία αντί 1 θα μας έδινε πιθανότητα

$$(N - \ell)/N = 1 - \ell/N.$$

Έστω X η τυχαία μεταβλητή που μετράει το πλήθος των στοιχείων που πρέπει να διαλέξουμε από το S τυχαία, ώστε το επόμενο στοιχείο να είναι κάποιο που διαλέξαμε προηγουμένως. Οπότε,

$$\Pr(X > \ell) = (1 - 1/N)(1 - 2/N) \cdots (1 - (\ell - 1)/N).$$

Από τη γνωστή ανισότητα της ανάλυσης,

$$1 - x \leq e^{-x}$$

έχουμε

$$\Pr(X > \ell) \leq e^{-\sum_{j=1}^{\ell-1} \frac{j}{N}} = e^{-\frac{1}{2} \frac{\ell(\ell-1)}{N}} \leq e^{-\frac{(\ell-1)^2}{2N}}.$$

Η αναμενόμενη τιμή είναι

$$\sum_{\ell \geq 1} \ell \Pr(X = \ell) = \sum_{\ell \geq 1} \ell (\Pr(X > \ell - 1) - \Pr(X > \ell)) = \sum_{\ell \geq 0} \Pr(X > \ell) \leq 1 + \sum_{\ell \geq 1} e^{-\frac{(\ell-1)^2}{2N}} \approx$$

$$1 + \int_0^\infty e^{-x^2/2N} dx = 1 + \sqrt{2N} \int_0^\infty e^{-x^2} dx = 1 + \sqrt{2N} \frac{\sqrt{\pi}}{2} \approx \sqrt{\frac{\pi N}{2}}.$$

□

Άσκηση 10.19 Να υπολογίσετε το πλήθος των στοιχείων που πρέπει να ληφθούν από το σύνολο S , ώστε με πιθανότητα 0.9 να έχουμε μία διένεξη.

Στην περίπτωση μας έχουμε μια συνάρτηση $f : G \rightarrow G$, που συμπεριφέρεται τυχαία και $X_i = f(X_{i-1})$, $i \geq 0$ (με X_0 κάποιο συγκεκριμένο σημείο της G). 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 ότι¹⁰¹ το μήκος του κύκλου (δηλαδή, το πλήθος των σημείων επί του κύκλου), αλλά και της ουράς, είναι ασυμπτωτικά περίπου ίσο με $\sqrt{\pi N/8}$.

Ας δούμε αναλυτικά τη 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

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

Θεωρούμε τη 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.

¹⁰¹Fajole, Odlyzko, Random mapping Statistics

Αλγόριθμος 10.2.2. : Ο κυκλικός αλγόριθμος του Floyd

Είσοδος. Συνάρτηση F , αρχική τιμή X_0 , πλήθος επαναλήψεων M

Έξοδος. Μια διένεξη μεταξύ του i και $2i$ ή Fail

```

1  $X \leftarrow X_0$ 
2  $Y \leftarrow X_0$ 
3 for  $i = 1$  to  $M$  do
4    $X \leftarrow F(X)$ 
5    $Y \leftarrow F(F(Y))$ 
6   if  $X = Y$  then
7     return διένεξη μεταξύ  $i$  και  $2i$ 
   end
end
8 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) επομένως, έχει ασήμαντη χωρική πολυπλοκότητα και (χρονική) πολυπλοκότητα $O(t)$. Ειδικότερα, μπορούμε να δείξουμε ότι η (χρονική) πολυπλοκότητα είναι

$$O(\sqrt{\pi|G|/2}).$$

Πράγματι, αρχικά παρατηρούμε ότι $X_t = X_{2t}$, δηλαδή, το t είναι μία περίοδος της ακολουθίας $(X_i)_i$, επομένως, $\ell|t$, όπου ℓ η περίοδος της $(X_i)_i$. Δηλαδή, το t είναι ένα πολλαπλάσιο της περιόδου ℓ . Επειδή το X_t είναι επί του κύκλου της ακολουθίας $(X_i)_i$, προκύπτει ότι $t \geq s$, όπου s το μήκος της ουράς της ακολουθίας $(X_i)_i$. Αλλά ο t είναι ο μικρότερος θετικός ακέραιος με την ιδιότητα $X_t = X_{2t}$, επομένως

$$t = \left\lceil \frac{s}{\ell} \right\rceil \ell. \quad (10.2.2)$$

Επειδή $s + \ell = O(\sqrt{\pi m/2})$ προκύπτει το ζητούμενο, δηλαδή, η πολυπλοκότητα είναι $O(\sqrt{\pi|G|/2})$.

Επιπλέον, αν έχουμε το t είναι εύκολο να βρούμε το ℓ ως εξής: Υπολογίζουμε τις τιμές της ακολουθίας $\{X_{t+i}\}_i$ για $i = 1, 2, \dots$ έως ότου βρούμε πάλι το X_t . Αυτό θα συμβεί ακριβώς μετά από ℓ βήματα. Για να βρούμε το s είναι περισσότερο πολύπλοκο (δείτε [5, Κεφ. 7.1]). Το μόνο που μας δίνει η εξίσωση (10.2.2) είναι ότι

$$t - \ell < s \leq t.$$

Άσκηση 10.20 Αν $p = 3571$, $y = 2763$ και $g = 2$, υπολογίστε τον διακριτό λογάριθμο $dlog_g(y) = x$ με τον αλγόριθμο ρ του Pollard.

Άσκηση 10.21 Σε αυτήν την άσκηση θα δούμε πώς μπορούμε να χρησιμοποιήσουμε τον αλγόριθμο του Pollard για παραγοντοποίηση¹⁰². Θεωρούμε έναν φυσικό αριθμό N και ζητάμε έναν διαιρέτη του. Έστω $F(X) = (X^2 + 1) \bmod N$ και ας είναι X_0 μια τυχαία τιμή από το $\{2, 3, \dots, N-1\}$. Η γραμμή 6 στον αλγόριθμο 10.2.2 τώρα θα είναι

$$\text{If } 1 < \gcd(|X - Y|, N) < N.$$

Αν δεν βρεθεί κάποιος μη τετριμμένος διαιρέτης, διαλέγουμε άλλη τιμή στο X_0 . Επίσης, θα μπορούσατε να κάνετε ένα loop με while, αντί της for, που να βγαίνει από το loop, όταν έχει βρεθεί ένας διαιρέτης. Υλοποιήστε τον αλγόριθμο και βρείτε έναν διαιρέτη του αριθμού $N = 2^{257} - 1$.

¹⁰²Η ευρετική πολυπλοκότητα αυτού του αλγορίθμου είναι $O(N^{1/4})$.

Βιβλιογραφία

- [1] Boneh, Dan; Shoup, V. *A Graduate Course in Applied Cryptography*, Version 0.5, Jan. 2020. <http://toc.cryptobook.us/>.
- [2] Galbraith, Steven. *Mathematics of Public Key Cryptography (version 1.1)*. <http://www.math.auckland.ac.nz/~sgal018/crypto-book/crypto-book.html>, 2011.
- [3] Grandall, R.; Pommerance, C. *Prime Numbers. A Computational Perspective*. 2nd edition, pp. 596, Springer, 2005.
- [4] Hoffstein, J.; Pipher, J.; Silverman, J. H. *An Introduction to Mathematical Cryptography*. Springer, 2008.
- [5] Joux, A. *Algorithmic Cryptanalysis*. Chapman & Hall/CRC, 2009.
- [6] Menezes, A.; van Oorschot, P.; Vanstone, S. *Handbook of Applied Cryptography*, CRC Press. <https://cacr.uwaterloo.ca/hac/>, 1996.
- [7] Mollin, R. *An Introduction to Cryptography*. 2nd edition, Series : Discrete mathematics and its applications. Taylor and Francis Group, 2007.
- [8] Shoup, Victor. *A Computational Introduction to Number Theory and Algebra (2nd edition)*, Cambridge University Press. <https://shoup.net/ntb> , 2008.
- [9] Wagstaff, Samuel. *The joy of factoring*. Student Mathematical Library, vol. 68, 2013. AMS.
- [10] Πουλάκης, Δ. *Θεωρία Αριθμών*, Εκδ. Ζήτη, 2009.
- [11] Πουλάκης, Δ. *Υπολογιστική Θεωρία Αριθμών*, Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. <http://dx.doi.org/10.57713/kallipos-323>, 2015.
- [12] Τζανάκης, Νίκος. *Θεμελιώδης Θεωρία Αριθμών*, Σημειώσεις, Πανεπιστήμιο Κρήτης, 2012.

Κεφάλαιο 11

Trapdoor Functions (TDF)

Σύνοψη - Περίληψη. Σε αυτό το κεφάλαιο ξεκινάμε με τον ορισμό του κρυπτοσυστήματος δημόσιου κλειδιού και παρουσιάζουμε πώς μπορούν δύο οντότητες να το χρησιμοποιήσουν, για να υλοποιήσουν ένα πρωτόκολλο ανταλλαγής κλειδιού. Συνεχίζουμε με κάποιους ορισμούς ασφάλειας. Κατόπιν, παρουσιάζουμε τον ορισμό της Trapdoor function και πώς μπορούμε να χρησιμοποιήσουμε αυτές τις συναρτήσεις για την κατασκευή κρυπτοσυστημάτων δημόσιου κλειδιού. Το πρώτο παράδειγμα μιας TDF είναι η συνάρτηση RSA. Παρουσιάζουμε τη συγκεκριμένη TDF αρκετά αναλυτικά και τελειώνουμε με την παρουσίαση της συνάρτησης του Rabin και με το κρυπτόςύστημα του Rabin.

Προαπαιτούμενη γνώση. Γι' αυτό το κεφάλαιο όλες οι γνώσεις που χρειαζόμαστε έχουν δοθεί στα προηγούμενα κεφάλαια. Πιο αναλυτικά, για την παρουσίαση της RSA-TDF είναι απαραίτητες κάποιες βασικές γνώσεις πιθανοτήτων (π.χ. [12]), ισοτιμίες, παραγοντοποίηση ακεραίων, πρώτοι αριθμοί και τετραγωνικές ρίζες $(\text{mod } N)$. Για το RSA μπορείτε να δείτε τα βιβλία [2, 4, 5, 6, 8] και από την ελληνόγλωσση βιβλιογραφία [3, 11, 13, 14].

11.1 Εισαγωγή

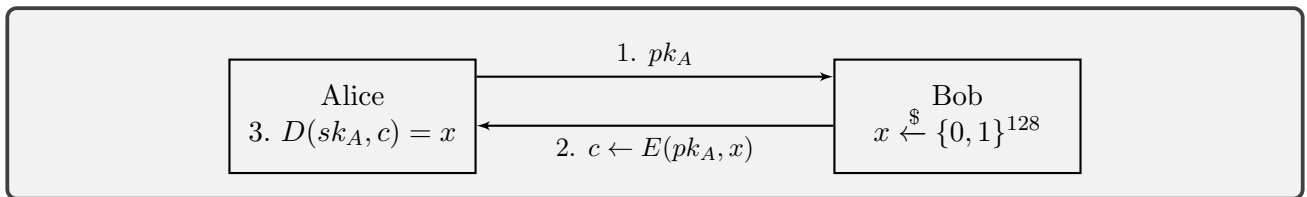
Αρχικά δίνουμε τον ορισμό ενός κρυπτοσυστήματος δημόσιου κλειδιού.

Ορισμός 11.1.1. Μια τριάδα αλγορίθμων (G, E, D) ονομάζεται κρυπτοσύστημα δημόσιου κλειδιού, αν

- (i). G : πιθανοτικός αλγόριθμος για τη 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 (eavesdropper). 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 $\text{EXP}(0)$, διαφορετικά λέμε ότι εκτέλεσε το πείραμα $\text{EXP}(1)$.

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

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

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

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

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

Ορισμός 11.1.2. Το Δ λέμε ότι είναι **σημασιολογικά ασφαλές** (SS), αν

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

Παράδειγμα 11.1.1. Ένα ντετερμινιστικό σύστημα κρυπτογράφησης δεν είναι semantically secure. Πράγματι, η Εύα μπορεί να ξεχωρίσει τα κρυπτογραφημένα μηνύματα που προέρχονται από ίδια μηνύματα, από αυτά που προέρχονται από διαφορετικά μηνύματα. Η $Pr(W_0) = 0$, $Pr(W_1) = 1$ άρα, $Adv_{SS}(\Delta) = 1$.

Παρατήρηση 11.1.1. 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 αποκρυπτογραφήσεις τους να μπορούν να βρεθούν.

Αν δώσουμε στην Εύα πιο ενεργό ρόλο, τότε η Alice μπορεί να δεχτεί επιθέσεις επιλεγμένου κρυπτογραφημένου κειμένου (CCA-Chosen Ciphertext Attack). Όπως παραπάνω, πρέπει να ορίσουμε και μια νέα έννοια της ασφάλειας για τις CCA επιθέσεις. Σε αυτήν την περίπτωση θα ορίσουμε ένα διαφορετικό παίγνιο και θα απαιτήσουμε πάλι η συνάρτηση $Adv_{CCA}(\Delta) \rightarrow 0$. Το παίγνιο ορίζεται ως εξής:

1. Η Εύα ζητάει από την Alice να αποκρυπτογραφήσει τα μηνύματα $c_i, i = 1, 2, \dots, n$.
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 μηνύματα $c'_i \neq c, i = 1, 2, \dots, n$.

Αν στο τέλος η Εύα δεν μπορεί να προσδιορίσει ποιο πείραμα εκτελέστηκε, δηλαδή $Adv_{CCA}(\Delta) \rightarrow 0$, λέμε τότε ότι το σύστημά μας είναι ασφαλές στην επίθεση επιλεγμένων κρυπτογραφημένων κειμένων (CCA-secure).

Ορισμός 11.1.3. Μια τριάδα αλγορίθμων (G, F, F^{-1}) ονομάζεται TDF, όταν

- (i). G : πιθανοτικός αλγόριθμος για τη δημιουργία δημόσιου κλειδιού (pk : public key) και ιδιωτικού κλειδιού (sk : secret key). Επίσης, το pk ορίζει μια συνάρτηση $F(pk, *) : X \rightarrow Y$.
- (ii). $F(pk, *)$: αποδοτικός ντετερμινιστικός αλγόριθμος που ορίζει μια συνάρτηση $F(pk, *) : X \rightarrow Y$
- (iii). υπάρχει $F^{-1}(sk, *)$, που ορίζει μια συνάρτηση από $Y \rightarrow X$ 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 αυτήν την περίπτωση να ορίσουμε τι σημαίνει ασφαλής TDF.

Ορισμός 11.1.4. Μια συνάρτηση TDF, (G, F, F^{-1}) ονομάζεται ασφαλής, αν η F είναι συνάρτηση μίας φοράς.

Μπορεί, δηλαδή, 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 : Trapdoor 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-Adleman¹⁰³ 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 πώς μπορούμε να κατασκευάσουμε ένα ασφαλές κρυπτοσύστημα. Μια 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

¹⁰³και 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. Ενώ το RSA παρουσιάστηκε το 1976, ήδη από το 1973 ο Clifford Cocks είχε αναπτύξει ένα ισοδύναμο σύστημα με το RSA.

$D(sk, c) = D(sk, E(pk, m)) = m$. Αυτό το κρυπτοσύστημα έχει πολλά προβλήματα! Για παράδειγμα, δεν είναι σημασιολογικά ασφαλές, διότι έχουμε μια ντετερμινιστική κρυπτογράφηση (παράδειγμα 11.1.1). 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 padding. Για το RSA το πιο γνωστό είναι το PKCS#1¹⁰⁴. Ο σκοπός είναι αυτές οι πρακτικές να μετατρέψουν το σύστημά μας σε SS-CCA, σημασιολογικά ασφαλές σε επιθέσεις κρυπτογραφημένου κειμένου.

Δεν πρέπει ποτέ να υλοποιούμε ένα κρυπτοσύστημα δημόσιου κλειδιού εφαρμόζοντας απευθείας την TDF (δείτε την ενότητα 12.2.1).

11.1.1 Υβριδικό κρυπτοσύστημα και Key Encapsulation Mechanism

Παρακάτω θα ορίσουμε ένα υβριδικό κρυπτοσύστημα (το οποίο το ονομάζουμε κρυπτοσύστημα ISO) που αποδεικνύεται ότι είναι CCA ασφαλές¹⁰⁵.

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

1. $x \xleftarrow{\$} X$
2. $y \leftarrow F(pk, x)$
3. $k \leftarrow H(x)$
4. $c \leftarrow E_s(k, m)$

Έξοδος: (y, c)

Η αποκρυπτογράφηση $D(sk, (y, c))$ γίνεται ως εξής:

Είσοδος: $sk, C = (y, c)$

1. $x \leftarrow F^{-1}(sk, y)$
2. $k \leftarrow H(x)$
3. $m \leftarrow D_s(k, c)$

Έξοδος: m

Το προηγούμενο σύστημα μπορεί να χρησιμοποιηθεί χωρίς την (E_s, D_s) . Έχουμε ένα κρυπτοσύστημα δημόσιου κλειδιού με δημόσιο κλειδί pk και ιδιωτικό κλειδί sk . Ο Key Encapsulation Mechanism (KEM) έχει δύο συναρτήσεις, ENCAP, DECAP. Αν παραλείψουμε το βήμα 4 της συνάρτησης κρυπτογράφησης $E(pk, \cdot)$, τότε έχουμε ένα encapsulation scheme : ENCAP. Δέχεται ως είσοδο ένα δημόσιο κλειδί, pk , και εξάγει ένα session key και ένα κρυπτομήνυμα, στην περίπτωσή μας το (k, y) . Ενώ η συνάρτηση αποκρυπτογράφησης, τώρα ονομάζεται decapsulation scheme : DECAP, και δέχεται ως είσοδο, το ιδιωτικό κλειδί sk και το κρυπτομήνυμα y , και εξάγει το κλειδί k , δηλαδή, παραλείπουμε το τρίτο βήμα από τη συνάρτηση $D(sk, \cdot)$. Το πλεονέκτημα των συστημάτων ENCAP/DECAP είναι ότι δεν χρειάζεται το σύστημα δημόσιου κλειδιού κάποιο σύστημα padding. Τα τελευταία χρόνια, η τάση είναι να χρησιμοποιούμε τα συστήματα δημόσιου κλειδιού για να κατασκευάσουμε KEM, π.χ. για το RSA δείτε τη δημοσίευση του NIST: SP 800-56B Rev.2¹⁰⁶. Στην υποενότητα 11.2.1 θα παρουσιάσουμε το σχήμα RSA-KEM.

¹⁰⁴Public Key Cryptographic Standards

¹⁰⁵υβριδικό λέμε ένα κρυπτοσύστημα που χρησιμοποιεί κρυπτογραφία ιδιωτικού και δημόσιου κλειδιού.

¹⁰⁶<https://csrc.nist.gov/publications/detail/sp/800-56b/rev-2/final>

11.2 RSA TDF

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

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

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

με

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

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

- 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 $ed \equiv 1 \pmod{\phi(N)}$. Τότε το δημόσιο κλειδί είναι $pk = (e, N)$ και το ιδιωτικό $sk = (d, N)$.
- $y = F(pk, x)$, ορίζεται $y = RSA_e(x)$.
- η $F^{-1}(sk, y) = y^d \pmod{N}$. Πράγματι, από το θεώρημα του Euler (Θεώρημα 9.1.1)

$$y^d = [(RSA_e(x))^d] \equiv x^{ed} \equiv x^{1+k\phi(N)} \equiv x \cdot (x^{\phi(N)})^k \equiv x \cdot 1^k \equiv x \pmod{N}.$$

Παρατήρηση 11.2.1. Ορίσαμε τη συνάρτηση $RSA_e(x)$ να έχει πεδίο ορισμού το $\mathbb{Z}_N^*$. Σε πολλά βιβλία (αλλά όχι σε όλα!) που παρουσιάζουν τον RSA θα δείτε το πεδίο ορισμού να είναι το $\mathbb{Z}_N$. Αυτό διότι ισχύει $x^{ed} \equiv x \pmod{N}$ 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 $c = p^e \pmod{N}$, τότε $\gcd(c, N) = p$. Για να αποφύγουμε αυτήν την προβληματική κατάσταση περιορίσαμε το πεδίο ορισμού της $RSA_e(x)$ να είναι το $\mathbb{Z}_N^*$.

Άσκηση 11.1 Έστω p, q δύο (διαφορετικοί) πρώτοι και $N = pq$. Επίσης e, d δύο ακέραιοι με $ed \equiv 1 \pmod{N}$. Ν.α.ό. για κάθε ακέραιο x ισχύει $x^{ed} \equiv x \pmod{N}$.

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

$$Pr[y \xleftarrow{\$} \mathbb{Z}_N^* : \mathcal{A}(N, e, y) = y^{1/e} \pmod{N}] \ll 1, \quad (11.2.1)$$

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 στις οποίες βασίζεται η σύγχρονη κρυπτογραφία δίνονται από τις σχέσεις 10.2.1, 11.2.1.

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 αποδειχτεί ότι είναι ισοδύναμο με το πρόβλημα της παραγοντοποίησης. Δηλαδή, αν η Εύα μπορεί και αποκρυπτογραφεί μηνύματα που έχουν κρυπτογραφηθεί με το RSA, αυτό δεν σημαίνει ότι μπορεί να παραγοντοποιήσει το N (φυσικά, το αντίστροφο ισχύει). Πρέπει να αναφέρουμε εδώ ότι υπάρχουν συστήματα που είναι ισοδύναμα με το πρόβλημα της παραγοντοποίησης, αλλά αυτά δεν είναι πρακτικά.

Παρατήρηση 11.2.2. (i). Αποδείξαμε ότι $y^d \equiv x \pmod{N}$. Αυτό μπορεί να ισχύει, ακόμη και αν $ed \not\equiv 1 \pmod{\phi(N)}$. 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 τα 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 $\lambda(N) = \text{lcm}(p-1, q-1)$ η συνάρτηση του Carmichael 9.2.2 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 στη δυσκολία εύρεσης e -οστών ριζών $\text{mod } N$, 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 $\text{NP} \cap \text{co-NP}$. Είναι απίθανο να είναι NP-complete.

(iv). 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 EC2¹⁰⁷.

(v). Ποτέ δεν χρησιμοποιούμε την TDF-RSA απευθείας για κρυπτογράφηση. Δηλαδή, αν m ένα μήνυμα, ποτέ δεν στέλνουμε $c = \text{RSA}_e(m)$. Στην ενότητα 12.2.3 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. Ένας τρόπος χρήσης του RSA χωρίς padding είναι να το συνδυάσουμε με ένα σύστημα KEM : Key Encapsulation Mechanism.

11.2.1 RSA-KEM

Για παράδειγμα, αν κάποιος έχει το δημόσιο κλειδί μας, (N, e) , εκτελεί τα εξής βήματα για το encapsulation, που δέχεται εισόδους το δημόσιο κλειδί (N, e) :

1. $x \xleftarrow{\$} \mathbb{Z}_N^*$.
2. $c \leftarrow x^e \pmod{N}$
3. $k \leftarrow \text{Hash}(x)$
4. return c, k

Εμείς, εκτελούμε την εξής διαδικασία (decapsulation) που δέχεται εισόδους το ιδιωτικό κλειδί, (N, d) , και το κρυπτοκείμενο c ,

1. $x \leftarrow c^d \pmod{N}$
2. $k \leftarrow \text{Hash}(x)$
3. return k

Με αυτόν τον τρόπο καταλήγουμε στο ίδιο κλειδί k ¹⁰⁸. Το RSA-KEM είναι ασφαλές εφόσον η συνάρτηση κατακερματισμού συμπεριφέρεται ως ένα τυχαίο μαντείο. Έχει το πλεονέκτημα έναντι του RSA-OAEP ότι έχει λίγο πιο ισχυρή απόδειξη ασφάλειας¹⁰⁹. Χρησιμοποιούμε σήμερα την TDF-RSA για την ψηφιακή

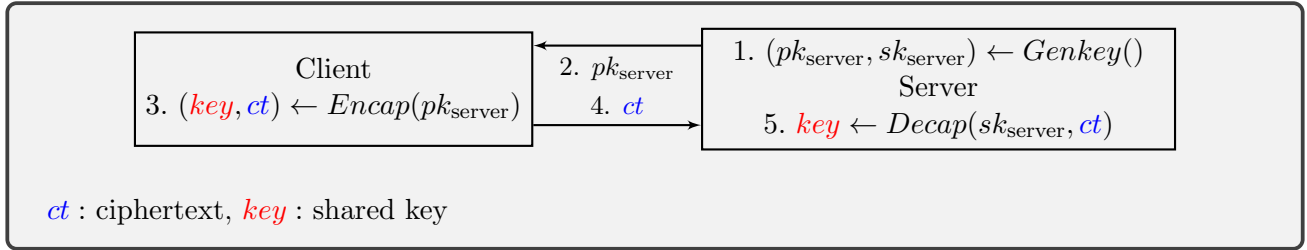
¹⁰⁷<https://github.com/eniac/faas>

¹⁰⁸ Δείτε το RFC: <https://datatracker.ietf.org/doc/html/rfc5990#page-12>

¹⁰⁹<https://eprint.iacr.org/2001/112>

υπογραφή μηνυμάτων και ανταλλαγή κλειδιών (είτε με OAEP είτε με KEM χωρίς padding). Όταν η κρυπτογράφηση γίνεται απευθείας, λέμε ότι χρησιμοποιούμε το κρυπτοσύστημα textbook RSA (ή plain RSA). Τον θετικό αθέρατο N τον ονομάζουμε *RSA-modulus*, το e δημόσιο εκθέτη και το d ιδιωτικό (ή μυστικό) εκθέτη. Ενώ, το (N, e) το ονομάζουμε δημόσιο κλειδί (public key) και το ζευγάρι ακεραίων (N, d) ιδιωτικό κλειδί (secret key).

Παρακάτω δίνουμε ένα σχήμα ανταλλαγής κλειδιού που χρησιμοποιεί KEM.



11.2.2 Η γνώση του d αποκαλύπτει τα p, q

Αν γνωρίζουμε το ιδιωτικό κλειδί d , μπορούμε να βρούμε την παραγοντοποίηση του N (το αντίστροφο ισχύει, διότι, αν γνωρίζουμε την παραγοντοποίηση του N , έχουμε το $\phi(N)$. Άρα, από τη σχέση $ed \equiv 1 \pmod{\phi(N)}$ βρίσκουμε εύκολα το d). Θα δούμε αρχικά έναν 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:

$$p + q = N - x + 1, \quad pq = N,$$

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 στη 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 στη 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, δοθέντος (e, d, N) με $ed \leq N^2$ και οι πρώτοι p, q έχουν ίδιο μήκος, τότε υπάρχει ντετερμινιστικός αλγόριθμος πολυωνυμικού χρόνου που μας δίνει την παραγοντοποίηση του N ¹¹⁰.

Στην περίπτωση που ο θετικός αθέρατος e είναι αρκετά μεγάλος, υπάρχει και πιθανοστικός αλγόριθμος που βρίσκει τα p, q . Ιστορικά, πρώτος ο Victor Miller έδωσε έναν αλγόριθμο που παραγοντοποιεί το N , αν γνωρίζουμε ένα πολλαπλάσιο του $\phi(N)$. Ο αλγόριθμός του χρησιμοποιούσε ως υπόθεση ότι ισχύει η

¹¹⁰Alexander May. Computing the RSA secret key is deterministic polynomial time equivalent to factoring : <https://www.iacr.org/archive/crypto2004/31520213/det.pdf>.

επεκτεταμένη υπόθεση του Riemann.

Η περίπτωση¹¹¹ για οποιοδήποτε e

Έστω g τυχαίο στοιχείο του συνόλου $\mathbb{Z}_N^*$. Από το θεώρημα του Euler έχουμε $g^{\phi(N)} \equiv 1 \pmod{N}$. Θέτουμε $k = ed - 1$. Τότε το k είναι πολλαπλάσιο του $\phi(N)$ οπότε $g^k \equiv 1 \pmod{N}$ και γράφεται $k = 2^\ell r$ για κάποιο περιττό r . Το ℓ είναι ο μεγαλύτερος εκθέτης τέτοιος ώστε $2^\ell | k$. Άρα, το $x = g^{k/2}$ είναι μία τετραγωνική ρίζα της μονάδας $\pmod{N}$ (αφού $x^2 \equiv 1 \pmod{N}$).

Υποθέτουμε ότι το $x = g^{k/2} \not\equiv \pm 1 \pmod{N}$. Θα δείξουμε ότι $\gcd(x - 1, N) = p$ ή q .

Απόδειξη ότι $\gcd(x - 1, N) = p$ ή q : Γενικά από CRT (δείτε ενότητα 8.7) η ισοτιμία $x^2 \equiv 1 \pmod{N}$ είναι ισοδύναμη με το σύστημα $\{x^2 \equiv 1 \pmod{p}, x^2 \equiv 1 \pmod{q}\}$. Καθεμία από τις προηγούμενες τετραγωνικές εξισώσεις έχει δύο λύσεις π.χ. η πρώτη έχει τις λύσεις $x \equiv \pm 1 \pmod{p}$. Επομένως η αρχική εξίσωση $x^2 \equiv 1 \pmod{N}$ έχει τέσσερις τετραγωνικές ρίζες $\pmod{N}$. Οι δύο από αυτές ικανοποιούν την $x \equiv \pm 1 \pmod{N}$ (τετριμμένες) και οι άλλες δύο $x \not\equiv \pm 1 \pmod{N}$. Τα συστήματα ισοτιμιών που προκύπτουν είναι τέσσερα:

$$\{x \equiv \pm 1 \pmod{p}, x \equiv \pm 1 \pmod{q}\}.$$

Κάθε σύστημα μπορεί να λυθεί με CRT, και προκύπτουν τέσσερις λύσεις,

$$x = \pm qv_1 \pm pv_2 \pmod{N}$$

όπου $v_1 = q^{-1} \pmod{p}$, $v_2 = p^{-1} \pmod{q}$.

Από την Πρόταση 8.4.1(vi) παρατηρούμε ότι από τα συστήματα $\{x \equiv 1 \pmod{p}, x \equiv 1 \pmod{q}\}$ και $\{x \equiv -1 \pmod{p}, x \equiv -1 \pmod{q}\}$ προκύπτουν οι λύσεις $x \equiv 1 \pmod{N}$ και $x \equiv -1 \pmod{N}$, αντίστοιχα (οπότε σε αυτές τις δύο περιπτώσεις δεν χρειαζόμαστε CRT). Οι άλλες δύο περιπτώσεις θα δώσουν άλλες δύο λύσεις. Το σύστημα,

$$\{x \equiv 1 \pmod{p}, x \equiv -1 \pmod{q}\}$$

θα δώσει τη λύση $x = qv_1 - pv_2 \pmod{N}$, ενώ το σύστημα,

$$\{x \equiv -1 \pmod{p}, x \equiv 1 \pmod{q}\},$$

θα δώσει τη λύση $x = -qv_1 + pv_2 \pmod{N}$. Οι λύσεις αυτές είναι $\not\equiv \pm 1 \pmod{N}$.

Στην πρώτη περίπτωση $p|x - 1$, $q|x + 1$. Επίσης $q \nmid x - 1$. Αν το $q|x - 1$ τότε $q|2$, άτοπο. Άρα, $\gcd(x - 1, N) = p$. Στην άλλη περίπτωση (δηλ. στο δεύτερο σύστημα), $\gcd(x - 1, N) = q$. Επομένως, $\gcd(x - 1, N) = p$ ή q . Οπότε αποδείχτηκε το ζητούμενο.

Τώρα, αν συμβεί $x = g^{k/2} \equiv \pm 1 \pmod{N}$, τότε ο αριθμός $x = g^{k/4}$ είναι μία τετραγωνική ρίζα της μονάδας $\pmod{N}$. Αν, επίσης, $g^{k/4} \not\equiv \pm 1 \pmod{N}$, τότε εφαρμόζουμε την προηγούμενη διαδικασία, αν όχι, τότε το $x = g^{k/8} \pmod{N}$ είναι μία τετραγωνική ρίζα της μονάδας $\pmod{N}$. Αργά ή γρήγορα θα σταματήσουμε στη χειρότερη περίπτωση στο $g^{k/2^\ell}$, $\ell = O(\ln N)$. Η πιθανότητα να διαλέξω ένα στοιχείο g , ώστε ένα από τα στοιχεία $g^{k/2^i} \not\equiv \pm 1 \pmod{N}$ για κάποιο $i \in \{1, 2, \dots, \ell\}$, είναι $\geq 1/2$. Άρα, έχουμε έναν αλγόριθμο Monte Carlo με πιθανότητα επιτυχίας τουλάχιστον $1/2$ και χρονική πολυπλοκότητα (σε bit) $O((\ln N)^3)$.

Άσκηση 11.2 Έστω ότι στο textbook RSA (δείτε την παρατήρηση 11.2.2 (iv)) έχουμε ένα modulus $N = pq$ και δημόσιο κλειδί e . Έστω το μήνυμα $m = ap$ για κάποιο a . Είναι εύκολο να παραβιάσουμε την ασφάλεια του συστήματος αυτού με τις δοθείσες παραμέτρους;

¹¹¹Proof of Fact1, p.205, <http://www.ams.org/notices/199902/boneh.pdf>

Άσκηση 11.3 Δίνεται το δημόσιο κλειδί $(N, e) = (11413, 19)$. Βρείτε το ιδιωτικό κλειδί και κατόπιν αποκρυπτογραφήστε το μήνυμα

$$C = (3203, 909, 3143, 5255, 5343, 3203, 909, 9958, 5278, 5343, 9958,$$

$$5278, 4674, 909, 9958, 792, 909, 4132, 3143, 9958, 3203, 5343, 792, 3143, 4443)$$

Υποθέστε ότι τα γράμματα στο αρχικό μήνυμα m , αναπαρίστανται από τις ASCII τιμές τους (δουλέψτε block by block το C).

Υπόδ. Παραγοντοποιήστε το N , κατόπιν υπολογίστε το $\phi(N)$.

Άσκηση 11.4 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. Υλοποιήστε μια (πιθανοτική) επίθεση, που να σας επιστρέφει το N . Μπορείτε να μετρήσετε εμπειρικά την πιθανότητα επιτυχίας; (Υπόδ. θεωρήστε και τα $(m_i^2, c_i') \in \mathbb{Z}_N^2$, τότε τα $c_i' - c_i$ διαιρούνται από το N .)

11.2.3 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 11.2.1. CRT-RSA (αποκρυπτογράφηση)

Είσοδος. Τυχαίο ακέραιο c και (p, q, d_p, d_q, i_q) .

Έξοδος. $c^d \in \mathbb{Z}_N$

```

1  $S_p = c^{d_p} \bmod p$ 
2  $S_q = c^{d_q} \bmod q$ 
3  $S = S_q + q(i_q(S_p - S_q) \bmod p)$ 
4 return  $S$ 
```

Παρατηρούμε ότι το κλειδί στο CRT-RSA έχει $\approx (\frac{1}{2} \log_2 N)$ bits, δηλαδή, περίπου τα μισά bits απ' ότι στο κλασικό. Γι' αυτόν τον λόγο χρησιμοποιείται σε smart cards. Επίσης, είναι περίπου τέσσερις φορές πιο γρήγορο από το κλασικό. Μένει να αποδείξουμε ότι η έξοδος S είναι τελικά η αποκρυπτογράφηση του c δηλ. $S = c^d \pmod{N}$.

Αρχικά παρατηρούμε ότι το $c^d \pmod{pq}$ μπορεί να υπολογιστεί αν γνωρίζουμε τα

$$c^d \equiv c^{d \bmod (p-1)} \equiv c^{e^{-1} \bmod (p-1)} \equiv c^{d_p} \pmod{p}, \quad c^d \equiv c^{d \bmod (q-1)} \equiv c^{e^{-1} \bmod (q-1)} \equiv c^{d_q} \pmod{q}.$$

Έχοντας τα S_p, S_q με χρήση του CRT υπολογίζουμε το $c^d \pmod{N}$.

Μια δεύτερη παρατήρηση αφορά την επίλυση του συστήματος $\{x \equiv a \pmod{p}, x \equiv b \pmod{q}\}$ που μπορεί να λυθεί ως εξής. Το $x = b + qu$ για κάποιο u . Τότε, $b + qu \equiv a \pmod{p}$, επομένως $u \equiv i_q(a - b) \pmod{p}$. Άρα, $x = b + q(i_q(a - b) \pmod{p})$. Αν θέσουμε $a = S_p$, $b = S_q$ προκύπτει το ζητούμενο.

Η τεχνική αυτή έχει διάφορες εφαρμογές. Για παράδειγμα ας θεωρήσουμε ένα N με 2048 bits και διαλέγουμε το d έτσι ώστε d_p, d_q να είναι μικρά, π.χ. 256 bits αλλά το $d \pmod{N}$ να είναι μεγάλο. Τότε, αντί να αποθηκεύσουμε το d που είναι περίπου 2048 bits αποθηκεύουμε το p (ή το q) που είναι μήκους 1024 bits. Έχοντας το p υπολογίζουμε το $q = N/p$ και κατόπιν τα d_p, d_q, i_q . Επομένως θα μπορούσαμε να το χρησιμοποιήσουμε σε smart cards.

11.3 Rabin TDF

Η TDF-Rabin(1979) βασίζει την ασφάλειά της στη δυσκολία της παραγοντοποίησης ενός ακεραίου 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

$$y = \text{Rabin}(x) = x^2 \pmod{N}$$

- η $F^{-1}(sk, y)$: η αντίστροφη συνάρτηση υπολογίζεται, εφόσον γνωρίζουμε τους πρώτους παράγοντες p, q του N , με χρήση του κινεζικού Θεωρήματος υπολοίπων (CRT). Η Διαδικασία είναι η εξής:

Από την ισοτιμία

$$x^2 \equiv y \pmod{N}$$

έχουμε

$$x \equiv \pm y_p \pmod{p}, x \equiv \pm y_q \pmod{q}$$

τα οποία μπορούμε εύκολα να λύσουμε με το CRT και δίνουν τέσσερις λύσεις mod N , έστω $\{x_1, x_2, x_3, x_4\}$.



Σχήμα 11.1: Michael O. Rabin (1931-) Turing Medal 1979. Ήταν μαθητής του Alonzo Church.
CC BY-SA 2.0 de licence, https://en.wikipedia.org/wiki/Michael_O._Rabin#/media/File:M_O_Rabin.jpg.

Μία από αυτές είναι η ζητούμενη. Σε αυτό το σημείο υπάρχει μια προσυμφωνημένη 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 τη λύση.

Παράδειγμα 11.3.1. Έστω $p = 7, q = 11, N = 77, x \in \{0, 1, \dots, 76\}$. 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 8.7.1 προκύπτει

$$\begin{cases} x^2 \equiv 15 \pmod{7} = 1 \\ x^2 \equiv 15 \pmod{11} = 4 \end{cases}$$

Μια τετραγωνική ρίζα του $1 \pmod{7}$ είναι το 1, ενώ μία τετραγωνική ρίζα του $4 \pmod{11}$ είναι το 2. Λύνω το σύστημα

$$\begin{cases} x \equiv 1 \pmod{7} \\ x \equiv 2 \pmod{11} \end{cases}.$$

Κατόπιν, εφαρμόζουμε το CRT και σύμφωνα με την ορολογία της ενότητας 8.7, έχουμε

$$m = 77, m_1 = p = 7, m_2 = q = 11, M_1 = m/m_1 = 11, M_2 = m/m_2 = 7,$$

$$v_1 \equiv M_1^{-1} \pmod{m_1} = 2 \pmod{7}, v_2 \equiv M_2^{-1} \pmod{m_2} = 8 \pmod{11}.$$

Το $x \equiv (1 \cdot M_1 \cdot v_1 + 2 \cdot M_2 \cdot v_2) \equiv 57 \pmod{77}$. Οι υπόλοιπες λύσεις προκύπτουν παίρνοντας όλα τα δυνατά πρόσημα στην παράσταση

$$(\pm 1 \cdot M_1 \cdot v_1 \pm 2 \cdot M_2 \cdot v_2) \pmod{77} = 64, 13, \mathbf{20}.$$

11.3.1 Ασφάλεια

Ο Rabin παρατήρησε ότι, αν μπορούμε να υπολογίζουμε τετραγωνικές ρίζες mod N , τότε μπορούμε να βρούμε την παραγοντοποίηση του N σε πολυωνυμικό χρόνο. Διαλέγουμε τυχαία ένα ακέραιο m από το σύνολο $\mathbb{Z}_N^*$ και υπολογίζουμε το τετράγωνο του mod N . Έστω, $y = m^2 \pmod{N}$. Εφόσον μπορούμε να υπολογίζουμε τετραγωνικές ρίζες, λύνουμε την εξίσωση, $x^2 \equiv y \pmod{N}$. Από CRT η ισοδυναμία $x^2 \equiv y \pmod{N}$ 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 αυτήν την περίπτωση

$$a^2 \equiv y \equiv m^2 \pmod{N}.$$

Οπότε,

$$N \mid a^2 - m^2 = (a - m)(a + m).$$

Αν $\gcd(N, a - m) = 1$, τότε $N \mid a + m$ 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, την προηγούμενη απόδειξη μπορούμε να τη 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 secure¹¹².

Αλγόριθμος 11.3.1. : 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  $m \xleftarrow{\$} \mathbb{Z}_N^*$ 
2  $y \leftarrow m^2 \pmod{N}$ 
3  $x \leftarrow \sqrt{y} \pmod{N}$ 
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.

Άσκηση 11.5 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 $m < 20$ (να μην χρησιμοποιηθεί υπολογιστής).

11.3.2 Coin flipping over the phone

Έστω ότι η Alice και ο Bob είναι σε δύο απομακρυσμένα γεωγραφικά σημεία. Ένας από τους δύο θα πάει στο μέρος που ζει ο άλλος για διακοπές. Καθώς κανένας δεν θέλει να μετακινηθεί, προτείνουν να ρίξουν ένα κέρμα. Δυστυχώς, ο μοναδικός τρόπος επικοινωνίας είναι το τηλέφωνο. Πώς θα καταφέρουν να ρίξουν ένα

¹¹²Dan Boneh, Simplified OAEP for the RSA and Rabin Functions

κέρμα (κάποιος από τους δύο) και να συμφωνήσουν με το αποτέλεσμα; Εκτελούν το παρακάτω πρωτόκολλο.

- [1] Alice : Διαλέγει δύο πρώτους αριθμούς p, q ισοδύναμους $\equiv 3 \pmod{4}$. Έστω $n = pq$. Στέλνει το n .
- [2] Bob : Διαλέγει ένα τυχαίο x ($1 < x < n$) και υπολογίζει $y = x^2 \pmod{n}$. Στέλνει το y στην Alice.
- [3] Alice : Η Alice έχοντας το y μπορεί να υπολογίσει τις τέσσερις τετραγωνικές ρίζες $\pmod{n}$, έστω $\pm a, \pm b$. Η Alice ρίχνει ένα κέρμα και διαλέγει το a ή b . Έστω ότι στέλνει το b .
- [4] Bob : Αν ο Bob είχε διαλέξει το a , τότε μπορεί να υπολογίσει τα p, q τα οποία στέλνει στην Alice. Έτσι η Alice μπορεί να επιβεβαιώσει ότι πράγματι ο Bob λέει αλήθεια και σε αυτή την περίπτωση κερδίζει ο Bob.
- [5] Alice : Αν ο Bob στείλει `null` στην Alice, η Alice στέλνει τους δύο πρώτους p, q ώστε να επαληθεύσει ο Bob ότι πράγματι διάλεξε δύο πρώτους αριθμούς, και σε αυτή την περίπτωση κερδίζει η Alice.

Άσκηση 11.6 Υλοποιήστε το προηγούμενο πρωτόκολλο σε μία γλώσσα προγραμματισμού.

11.4 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 .

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 μιας ελλειπτικής καμπύλης επί ενός πεπερασμένου σώματος, είναι η βάση της κρυπτογραφίας με ελλειπτικές καμπύλες.

Άσκηση 11.7 Περιγράψτε, πώς παράγονται τα κλειδιά, πώς γίνεται η κρυπτογράφηση και πώς η αποκρυπτογράφηση στο σύστημα ElGamal. Κατόπιν ν.α.ό. δεν είναι σημασιολογικά ασφαλές υπό επιθέσεις κρυπτογραφημένου επιλεγμένου κειμένου (CCA). Με άλλα λόγια δεν είναι IND-CCA secure.

11.4.1 Κρυπτογραφία με Ελλειπτικές καμπύλες

Χωρίς να επεκταθούμε σε τεχνικές λεπτομέρειες θα αναφερθούμε στην πολύ σημαντική εφαρμογή των ελλειπτικών καμπυλών στην κρυπτογραφία, δείτε [6, 7, 9, 10]. Στην ελληνόγλωσση βιβλιογραφία δείτε [1, Κεφάλαιο 7]. 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 καμπυλών (ECC : Elliptic Curve Cryptography) είναι σήμερα αρκετά διαδεδομένη, αλλά όχι στον 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 καμπυλών 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 (δηλαδή, σύστημα που οι εξισώσεις του είναι δευτέρου βαθμού).

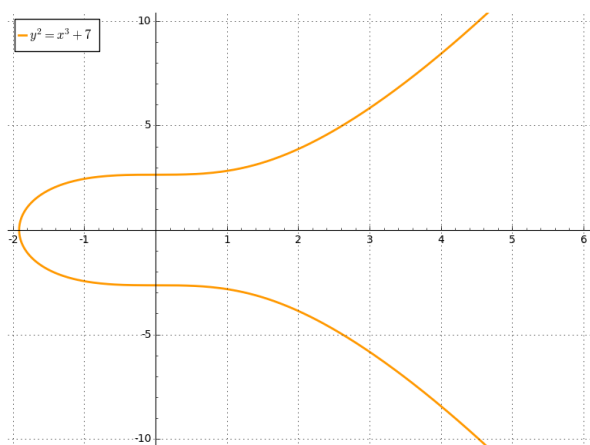
Φυσικά, όπως και στο RSA, χρειάζεται ιδιαίτερη προσοχή στο πώς θα υλοποιήσουμε αυτά τα συστήματα. Προτείνονται συγκεκριμένες ελλειπτικές καμπύλες π.χ. από το NIST, για υλοποίηση αυτών των πρωτοκόλλων. Η δική μας γνώμη είναι να χρησιμοποιείτε τις οικογένειες ελλειπτικών καμπυλών που δόθηκαν από τον Bernstein¹¹³ ή τις ελλειπτικές καμπύλες του Edwards.

Τα συστήματα sagemath και pari gp έχουν πολλές καλές υλοποιήσεις για ελλειπτικές καμπύλες. Ενδεικτικά, παραθέτουμε κάποιον κώδικα στο σύστημα sagemath.

```
1 p = 331
2 E = EllipticCurve(GF(p), [0,0,0,0,7])
3 print(E)
4 print(E.cardinality())
```

Ο προηγούμενος κώδικας υπολογίζει το πλήθος των σημείων της ελλειπτικής καμπύλης $E : y^2 = x^3 + 7$ modulo 331. Το αποτέλεσμα είναι 364. Ο παρακάτω κώδικας μας δίνει το γράφημα στο $\mathbb{R}$.

```
1 pl = plot(EllipticCurve([0,0,0,0,7]), thickness=2, gridlines='true', color=hue(2.1),
2         xmin=-2, xmax=6, ymin=-10, ymax=10, legend_label='$y^2=x^3+7$')
3 pl
```



Σχήμα 11.2: Το γράφημα της ελλειπτικής καμπύλης $y^2 = x^3 + 7$ στο $\mathbb{R}$.

11.4.2 Post Quantum ECC

Ο Couveignes και οι Rostovtsev, Stolbunov το 2006 πρότειναν ένα πρωτόκολλο τύπου Diffie-Hellman βασιζόμενοι στο πρόβλημα εύρεσης ισογένειας μεταξύ δύο ελλειπτικών κανονικών καμπυλών (isogeny key exchange protocols over ordinary curves). Είναι σημαντικό να παρατηρήσουμε ότι πιστεύουμε ότι αυτό το πρόβλημα είναι χβαντικά ασφαλές, με την έννοια ότι δεν ανάγεται σε ένα πρόβλημα παραγοντοποίησης ή διακριτού λογαρίθμου. Επομένως, είναι υποψήφιο για την εποχή μετά τους χβαντικούς υπολογιστές (Post Quantum era). Έχει προταθεί βελτιωμένη έκδοση του προηγούμενου πρωτοκόλλου, SIDH: Supersingular isogeny Diffie - Hellman key exchange, στον διαγωνισμό της NIST για Post Quantum Cryptosystems.

Update: Δυστυχώς το σύστημα αυτό έσπασε από τους Castryck και Decru τον Ιούλιο του 2022¹¹⁴.

¹¹³<https://safecurves.cr.yp.to/>

¹¹⁴<https://eprint.iacr.org/2022/975>

Βιβλιογραφία

- [1] Αντωνιάδης, Ι.; Κοντογεώργης, Α. *Πεπερασμένα Σώματα και Κρυπτογραφία*, Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. <http://dx.doi.org/10.57713/kallipos-878>, 2015.
- [2] Boneh, Dan; Shoup, V. *A Graduate Course in Applied Cryptography*, Version 0.5, Jan. 2020. <http://toc.cryptobook.us/>.
- [3] Burmester, M.; Γκρίτζαλης, Σ.; Κάτσικας, Σ.; Χρυσικόπουλος, Β. *Σύγχρονη Κρυπτογραφία. Θεωρία και Εφαρμογές*. Εκδ. Παπασωτηρίου, 2011.
- [4] Cozzens, M.; Miller, S. J. *The Mathematics of Encryption-An Elementary Introduction*. Mathematical World, 1st edition, vol. 29, 2013. AMS.
- [5] Galbraith, Steven. *Mathematics of Public Key Cryptography (version 1.1)*. <http://www.math.auckland.ac.nz/~sgal018/crypto-book/crypto-book.html>, 2011.
- [6] Hoffstein, J.; Pipher, J.; Silverman, J. H. *An Introduction to Mathematical Cryptography*. Springer, 2008.
- [7] Husemoller, D. *Elliptic Curves*. Springer, 2004.
- [8] Menezes, A.; van Oorschot, P.; Vanstone, S. *Handbook of Applied Cryptography*, CRC Press. <https://cacr.uwaterloo.ca/hac/>, 1996.
- [9] Silverman, J. H. *Advanced Topics in the Arithmetic of Elliptic Curves*. Springer, 1994.
- [10] Silverman, J. H. *The Arithmetic of Elliptic Curves*. Springer, 2009.
- [11] Κάτος, Β.; Στεφανίδης, Γ. *Τεχνικές Κρυπτογραφίας και Κρυπτανάλυσης*, Εκδ. Ζυγός, 2003.
- [12] Μπερτσεκάς, Π. Δ.; Τσιτσικλής, Ν. Γ. *Εισαγωγή στις πιθανότητες*. Εκδ. Τζιόλας, 2010.
- [13] Πουλάκης, Δ. *Κρυπτογραφία*. Εκδ. Ζήτη, 2009.
- [14] Ζάχος, Ε.; Παγουρτζής, Α.; Γροντάς, Π. *Υπολογιστική Κρυπτογραφία*, Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. <http://dx.doi.org/10.57713/kallipos-492>, 2015.

Κεφάλαιο 12

Επιθέσεις στο Κρυπτοσύστημα RSA

Σύνοψη - Περίληψη. Σε αυτό το κεφάλαιο θα παρουσιάσουμε μερικές επιθέσεις στο σύστημα RSA. Αρχικά, παρουσιάζουμε την επίθεση του Wiener η οποία βασίζεται στο θεώρημα του Legendre. Αυτή η επίθεση πετυχαίνει, όταν το μυστικό κλειδί είναι αρκετά μικρό σε σχέση με το RSA-modulus N . Κατόπιν, παρουσιάζουμε τη γενίκευση των Boneh-Durfee. Τέλος, παρουσιάζουμε συνοπτικά την επίθεση του Coppersmith που βασίζεται σε πλέγματα. Καθώς δεν έχουμε μιλήσει για πλέγματα, η παρουσίαση είναι αρκετά συνοπτική. Ο αναγνώστης μπορεί να ενημερωθεί για τις βασικές έννοιες της αναγωγής πλεγμάτων σε παρακάτω κεφάλαιο του βιβλίου.

Προαπαιτούμενη γνώση. Σε αυτό το κεφάλαιο χρειάζεται να μπορούμε να υπολογίσουμε το συνεχές κλάσμα ενός ρητού αριθμού. Η διαδικασία εύρεσης συνεχούς κλάσματος δίνεται αναλυτικά. Βασικές γνώσεις στοιχειώδους θεωρίας αριθμών απαιτούνται. Μπορείτε να δείτε το κεφ. 8 αυτού του βιβλίου. Επίσης, χρειάζονται κάποιες βασικές γνώσεις για τη διαδικασία της ταξινόμησης (sorting). Π.χ. δείτε [1], [5, Κεφ. 8] . Κάποια βασικά στοιχεία για πλέγματα (lattices) θα ήταν χρήσιμα για την καλύτερη κατανόηση των επιθέσεων τύπου Coppersmith. Μπορείτε να δείτε τις αναφορές [2, 3] ή να δείτε παρακάτω το ανάλογο κεφάλαιο σε αυτό το βιβλίο.

12.1 Η επίθεση του Wiener

Ένα απλό συνεχές κλάσμα είναι μια έκφραση της μορφής

$$x = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \frac{1}{a_4 + \dots}}}}$$

το οποίο το συμβολίζουμε και $x = [a_0; a_1, a_2, \dots]$. Οι όροι $a_0, a_2, \dots$ είναι ακέραιοι και ονομάζονται *μερικά πηλικά*. Τα μερικά πηλικά μπορεί να είναι είτε πεπερασμένα είτε άπειρα ως προς 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 12.1.1. : Συνεχές κλάσμα πραγματικού αριθμού

Είσοδος. $x \in \mathbb{R}$

Έξοδος. $[a_0; a_1, a_2, \dots] = x$

```

1   $i = 0, a_0 = \lfloor x \rfloor, x = x - a_0$ 
2  while  $x > 0$  do
     $i = i + 1$ 
     $a_i = \lfloor 1/x \rfloor$ 
     $x = 1/x - a_i$ 
end
```

Αποδεικνύεται ότι το συνεχές κλάσμα είναι πεπερασμένου μήκους αν και μόνο αν ο $x \in \mathbb{Q}$ [4, 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 *συγκλίνοντα κλάσματα* του x .

Παράδειγμα 12.1.1. 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

$$[0; 1] = 1, [0; 1, 2] = \frac{1}{1 + \frac{1}{2}} = \frac{2}{3}, [0; 1, 2, 1] = \frac{3}{4}, \dots, x.$$

Άσκηση 12.1 Βρείτε τα συνεχή κλάσματα των ρητών

$$\frac{12345}{6789}, \frac{54321}{9876}.$$

Πολυπλοκότητα του αλγορίθμου 12.1.1 όταν $x \in \mathbb{Q}$. Έστω $x = m/n$ ανάγωγο κλάσμα με $m, n > 0$. Ο αλγόριθμος εύρεσης του συνεχούς κλάσματος του m/n είναι στην ουσία η εύρεση του μέγιστου κοινού διαιρέτη των m, n . Ας θεωρήσουμε το $x = 13/5$. Αν εφαρμόσουμε τον αλγόριθμο θα έχουμε,

$$1. a_0 = \lfloor 13/5 \rfloor = 2 \text{ και } x \leftarrow x - a_0 = 3/5.$$

$$2. a_1 = \lfloor 5/3 \rfloor = 1 \text{ και } x \leftarrow 1/x - a_1 = 2/3.$$

$$3. a_2 = \lfloor 3/2 \rfloor = 1 \text{ και } x \leftarrow 1/x - a_2 = 1/2.$$

$$4. a_3 = \lfloor 2/1 \rfloor = 2 \text{ και } x \leftarrow 1/x - a_3 = 0.$$

Άρα, $x = [2, 1, 1, 2]$. Αν εφαρμόσουμε τον Ευκλείδειο αλγόριθμο έχουμε,

$$13 = 2 \cdot 5 + 3, \quad 5 = 1 \cdot 3 + 2, \quad 3 = 1 \cdot 2 + 1, \quad 2 = 2 \cdot 1 + 0.$$

Επομένως, η πολυπλοκότητα του αλγορίθμου 12.1.1 για $x \in \mathbb{Q}$ είναι $O(\ln m \ln n)$. Δηλαδή, πολυωνυμικού χρόνου.

Πριν αποδείξουμε το βασικό θεώρημα της ενότητας, χρειαζόμαστε το παρακάτω.

Θεώρημα 12.1.1 (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 $p = N_k$, $q = D_k$.

Στο επόμενο θεώρημα χρησιμοποιούμε την ορολογία της ενότητας 11.2.

Θεώρημα 12.1.2 (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 τη σχέση $ed \equiv 1 \pmod{\phi(N)}$ 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.

Η πολυπλοκότητα του αλγορίθμου ισούται με αυτήν του υπολογισμού του συνεχούς κλάσματος του ρητού e/N , που είναι $O(\ln e \ln N)$. □

Ο Wiener πρότεινε να χρησιμοποιούμε $e > N^{1.5}$, διότι τότε ο αλγόριθμός του δεν δίνει καμία πληροφορία για το d (μπορείτε να το εξηγήσετε αυτό;). Επίσης, αν χρησιμοποιήσουμε το CRT-RSA (υποενότητα 11.2.3) η επίθεση του Wiener δεν δουλεύει.

Παράδειγμα 12.1.2. 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 ακόλουθη:

```

Αλγόριθμος 12.1.2. : Επίθεση του Wiener
Είσοδος.  $N, e$  ( $e < N$ )
Έξοδος. Ο μυστικός εκθέτης  $d$  ή FAIL

1  $\frac{e}{N} = [0; a_1, \dots, a_n]$ 
2 for  $i = 1$  to  $n$  do
3    $\frac{N_i}{D_i} = [0; a_1, \dots, a_i]$ 
   end
4 for  $i = 1$  to  $n$  do
5   if  $(2^e)^{D_i} \equiv 2 \pmod{N}$  then
6     return  $D_i$ 
7     EXIT
   end
end
8 FAIL

```

Μια άλλη υλοποίηση που δίνει τους πρώτους παράγοντες είναι η εξής:

```

Αλγόριθμος 12.1.3. : Επίθεση του Wiener
Είσοδος.  $N, e$  ( $e < N$ )
Έξοδος. Οι πρώτοι αριθμοί  $p, q$  ή FAIL

1  $\frac{e}{N} = [0; a_1, \dots, a_n]$ 
2 for  $i = 1$  to  $n$  do
3    $\frac{N_i}{D_i} = [0; a_1, \dots, a_i]$ 
4    $\phi_i = \frac{eD_i - 1}{N_i}$ 
   end
5 for  $i = 1$  to  $n$  do
6   if  $\phi_i \in \mathbb{Z}$  then
7     Βρείτε τις ρίζες  $x_1, x_2$  της εξίσωσης:  $x^2 - (N - \phi_i + 1)x + N = 0$ 
8     if  $x_1, x_2 \in \mathbb{Z}$  then
9       return  $x_1, x_2$ 
10      EXIT
    end
  end
end
11 FAIL

```

Στη γραμμή 7 γίνεται έλεγχος αν το D_i που διαλέξαμε είναι το σωστό, δηλαδή, το d . Ο έλεγχος γίνεται έμμεσα. Η εξίσωση της γραμμής 7 έχει ως ρίζες τους 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 12.1.3. Έστω ότι $(N, e) = (5697733, 3105251)$. Τότε

$$\frac{e}{N} = [0; 1, 1, 5, 17, 1, 9, 1, 1, 1, 1, 1, 1, 1, 1, 3, 6, 2]$$

και

$$\frac{N_i}{D_i} \in \{1, 1/2, 6/11, 103/189, 109/200, \dots\},$$

$$\phi_i \in \{310250, 6210501, 5692960, \dots\}.$$

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 12.1.1 (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 12.1.1. Γενικά, οι επιθέσεις στο RSA, ανήκουν σε μια από τις παρακάτω κατηγορίες: ευρετική, πιθανοτική ή ντετερμινιστική. Στις ευρετικές επιθέσεις κάνουμε μια υπόθεση την οποία δεν μπορούμε να αποδείξουμε, αλλά ούτε και να βρούμε την πιθανότητα επιτυχίας της επίθεσης. Στις πιθανοτικές κάνουμε μια υπόθεση, όπου μπορούμε να αποδείξουμε ότι ισχύει με μια μεγάλη πιθανότητα. Στις ντετερμινιστικές, δεν χρειαζόμαστε κάποια επιπλέον υπόθεση.

Άσκηση 12.2 Ας θεωρήσουμε $(N, e) = (194749497518847283, 50736902528669041)$ και το κρυπτογραφημένο κείμενο που δίνεται στο: [github.com/drazioti/...](https://github.com/drazioti/textbook-RSA), έχει προκύψει από το textbook-RSA (block by block) και κατόπιν κωδικοποιήθηκε. Εφαρμόστε την επίθεση του Wiener, για να βρείτε το κλειδί d . Υποθέτουμε ότι στο αρχικό κείμενο m κάθε χαρακτήρας έχει αντικατασταθεί από την ASCII τιμή του. Τέλος, βρείτε το αρχικό μήνυμα m .

Επομένως, αν δούμε το προηγούμενο παράδειγμα 12.1.2 με τη νέα επίθεση των Boneh-Durfee, η απαίτηση είναι να έχει ο d τουλάχιστον 300 bits (για την περίπτωση που ο N έχει 1024-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, στα οποία ακόμη και ο LLL (δείτε υποενότητα 14.5) 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 12.1.2 (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. Όμως, 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 $C_i \equiv M_i^e \pmod{N_i}$, με $e_i = 3$. Από CRT βρίσκει $C \equiv C_i \pmod{N_i}$. Επομένως, $C \equiv M^3 \pmod{N_1 N_2 N_3}$. 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 12.1.3 Έστω $N_1, \dots, N_k$ θετικοί ακέραιοι ανά δύο πρώτοι και $N_{\min} = \min_i \{N_i\}$ ($i = 1, 2, \dots, k$). Αν $g_i(x) \in \mathbb{Z}_{N_i}[x]$ βαθμού $d < k$ και M ακέραιος με $M < N_{\min}$ με $g_i(M) \equiv 0 \pmod{N_i}$, για κάθε $i = 1, 2, \dots, k$, τότε υπάρχει αποδοτικός αλγόριθμος που υπολογίζει το M .

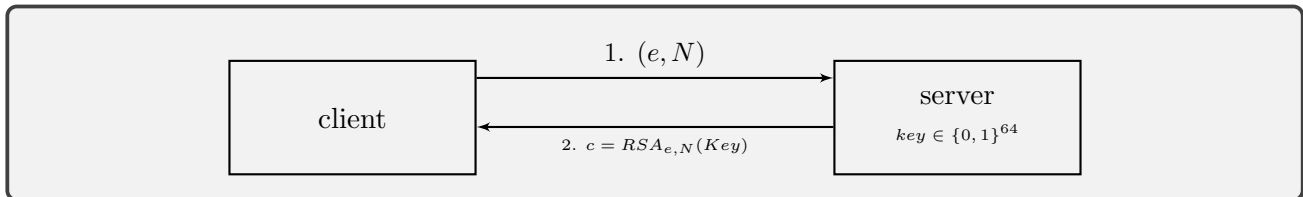
Για να αποφύγουμε την προηγούμενη επίθεση, προτείνεται η εφαρμογή τυχαίου padding πριν την κρυπτογράφηση αντί σταθερού padding. Για παράδειγμα, ο Bob στέλνει στον i -παραλήπτη αντί του μηνύματος M το μήνυμα $i \cdot 2^m + M$ (linear padding), όπου m το πλήθος των bits του M . Αλλά και σε αυτήν την περίπτωση το προηγούμενο θεώρημα σε συνδυασμό με την επίθεση του Coppersmith μπορεί επίσης να αποκαλύψει το μήνυμα M .

12.2 Επιθέσεις στο RSA βασισμένες σε πλέγματα

Σε αυτήν την ενότητα θα αναφέρουμε επιθέσεις στο RSA που βασίζονται σε πλέγματα (μια εισαγωγή δίνεται στην ενότητα 14). Στόχος είναι να καταλάβουμε πότε υλοποιούνται αυτές οι επιθέσεις και όχι να δώσουμε μία μαθηματική ανάλυση αυτών των επιθέσεων.

12.2.1 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 12.3) 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 $\mathbb{Z}_N$):

$$L_1 = \{c/1^e, c/2^e, c/3^e, \dots, c/2^{32e}\}, \quad L_2 = \{1^e, 2^e, 3^e, \dots, 2^{32e}\}.$$

Επομένως, σε χρόνο $O(2^{32})$ (για την ακρίβεια χρειαζόμαστε 2^{33} υψώσεις $\pmod{N}$) 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 τη 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

$$(RSA_{e,N}(key + 2^{1024} + 2^{128} + 1), 2^{1024} + 2^{128} + 1).$$

Άσκηση 12.3 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 $\{\lfloor n/2 \rfloor, \lfloor n/2 \rfloor \pm 1\}$. Χρειαζόμαστε την παρακάτω ρουτίνα, η οποία θα εκτελεστεί K φορές, και για κάθε εκτέλεση θα διαλέγουμε ένα τυχαίο m με n -bits, ώστε να υπολογίζουμε τη ζητούμενη πιθανότητα (αυτή θα είναι πλήθος άσων διά του K).

```

input :  $n, m$ 
output: 1, αν υπάρχει ακέραιος  $d$  διαιρέτης του  $m$  δυαδικού μήκους περίπου  $n/2$  bits
1:  $L = \text{divisors}(m)$ 
2: For  $i$  from 1 to  $\text{len}(L)$ 
3:   If  $\text{len}(d) \in \{\lfloor n/2 \rfloor, \lfloor n/2 \rfloor \pm 1\}$  then return 1
  
```

Στην επίθεση που παρουσιάσαμε στην ενότητα 12.2.1, 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 στην ενότητα 12.2.1 έχει πιθανότητα επιτυχίας $1/4$ και χρονική πολυπλοκότητα $O(2^{32})$. Να το επαληθεύσετε πειραματικά.

12.2.2 Merge Sort

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 αυτήν την πολυπλοκότητα. Ανακαλύφθηκε από τον John von Newman το 1945. Ξεκινάμε, διαιρώντας στη μέση τη λίστα, σε δύο ίδιου μήκους υπολίστες (ας υποθέσουμε για απλότητα ότι ο n είναι άρτιος), έστω L_{right}, L_{left} . Ταξινομούμε τις δύο υπολίστες με τον ίδιο τρόπο, δηλαδή τις χωρίζουμε στη μέση, έτσι θα προκύψουν συνολικά τέσσερις υπολίστες, κ.ο.κ., θα καταλήξουμε σε λίστες με δύο στοιχεία. Συνεπώς, 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 τη 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. Ξαναεφαρμόζω τη συνάρτηση merge (θα χρειαστούμε το πολύ 4 συγκρίσεις για κάθε ζευγάρι), αργά ή γρήγορα θα καταλήξω στη ζητούμενη ταξινομημένη λίστα. Ο αλγόριθμος αυτός (όπως και του Karatsuba) ανήκει στην κατηγορία Divide and Conquer.

Αλγόριθμος 12.2.1. : MergeSort

Είσοδος. Μία λίστα L με n στοιχεία

Έξοδος. Η Ταξινομημένη λίστα

```

1 def Mergesort(L)
2   if  $|L| \leq 1$  then
3     return L
4   end
5   Take  $L_{left}$ 
6   Take  $L_{right}$ 
7    $L_{right} = \text{Mergesort}(L_{right})$ 
8    $L_{left} = \text{Mergesort}(L_{left})$ 
9    $L \leftarrow \text{merge}(left, right)$ 
10  return L
```

Ο προηγούμενος αλγόριθμος καλεί ως υπορουτίνα τον παρακάτω αλγόριθμο.

Αλγόριθμος 12.2.2. 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  $R.remove(R[0])$ 
5     else
6        $C.appendright(L[0])$  and  $L.remove(L[0])$ 
7     end
8   end
9   while  $|L| > 0$  do
10     $C.appendright(L[0])$  and  $L.remove(L[0])$ 
11  end
12  while  $|R| > 0$  do
13     $C.appendright(R[0])$  and  $R.remove(R[0])$ 
14  end
15  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 $T(n) = n \log_2(n)$. Επομένως, 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.

Άσκηση 12.4 Να υλοποιήσετε τον αλγόριθμο Mergesort.

12.2.3 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 LLL¹¹⁵.
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, στην 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 αυτήν την περίπτωση το x έχει 200-bits και γνωρίζει τα 824 πιο σημαντικά bits του M).

Θεώρημα 12.2.1 (Coppersmith) Έστω $0 < \varepsilon < 1/d$. Ας είναι $F(x)$ ένα κανονικό πολυώνυμο (δηλαδή, ο συντελεστής του μεγιστοβάθμιου όρου είναι μονάδα) βαθμού d με μία τουλάχιστον 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 $\text{poly}(d, 1/\varepsilon, \ln N)$.

Αλγόριθμος 12.2.3. : 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 ισοτιμίας $F(x) \equiv 0 \pmod{N}$ 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 $\Phi_X : \mathbb{Z}[x] \rightarrow \mathbb{R}^n$ με $\Phi_X(f_0 + f_1x + \dots + f_nx^n) = (f_0, f_1X, \dots, f_nX^n)$
- 5 Κατασκεύασε τα πολυώνυμα $G_{i,j}(x) = N^{h-1-j}F(x)^jx^i$, $0 \leq j < h$, $0 \leq i < d$
- 6 Κατασκεύασε τα διάνυσμα $\Phi_X(G_{i,j})$ και σχημάτισε έναν 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 καθένα το σύνολο στο οποίο ορίζονται, ώστε ο πίνακας M να προκύψει τριγωνικός άνω¹¹⁶. Ας δούμε ένα παράδειγμα.

Παράδειγμα 12.2.1. 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 $N = 1797693134 \dots 9211$ (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.

¹¹⁵Δείτε υποενότητα 14.5

¹¹⁶Μία υλοποίηση του προηγούμενου αλγορίθμου στο σύστημα Sagemath μπορείτε να βρείτε στην https://github.com/AristotleUniversity/sage/blob/master/our-projects/univariate_coppersmith

Επομένως, η Alice στέλνει το παρακάτω μήνυμα

$$M = 111 \dots 1111 || \text{bin}(x).$$

Η Εύα γνωρίζει το pad και φάχνει τα υπόλοιπα 290 bits του M . Επίσης, η Εύα γνωρίζει την κρυπτογράφηση του M που είναι $C = M^e \pmod{N}$. Η Εύα σχηματίζει το πολυώνυμο $F(x) = (x + 2^{1024} - 2^{290})^3 - C$ και εκτελεί τον αλγόριθμο 12.2.3. Οπότε υπολογίζει διαδοχικά

1. $h = 7$

2. $X = 109095309735365398756518898013497620784775746772155687711811$

6947308772559604191797968896

3. $n = 21$

4. Έναν πίνακα 21×21 στον οποίο εκτελούμε τον LLL. Το πρώτο διάνυσμα την ανηγμένης βάσης δίνει το πολυώνυμο

$$g(x) = 2977907 \dots 116524x^{19} - 109432 \dots 6372752x^{18} + \dots + 3272 \dots 616608$$

το οποίο έχει ρίζα στο $\mathbb{Z}$ τον ακέραιο

$$x_0 = 8594215741625 \dots 76551218155386392$$

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.

12.3 Συμπεράσματα

Σήμερα χρησιμοποιούμε το σχήμα pad OAEP και για υπογραφές RSA το σύστημα PSS. Το σχήμα pad OAEP¹¹⁷ ανακαλύφθηκε από τους Rogaway και Bellare. Το πρότυπο που υλοποιεί το σχήμα OAEP είναι το PKCS#1 ver. 2¹¹⁸. 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 2²⁰ 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 βρει καμιά πληροφορία (για το καθαρό κείμενο) 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 ντετερμινιστική κρυπτογράφηση, επομένως δεν είναι SS. Δηλαδή, 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 12.3.1. 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 ότι οι παράμετροι του RSA είναι (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).

¹¹⁷Optimal Asymmetric Encryption Padding

¹¹⁸όπως είδαμε στην παρατήρηση 11.2.2 iv, μπορούμε να αποφύγουμε το padding

Διαλέγουμε ένα τυχαίο ακέραιο s από το σύνολο

$$\{1, 2, \dots, N-1\} - \{e\}.$$

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 $(m's^{-1})^e \equiv s^e m (s^e)^{-1} \equiv m \bmod N$. 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 αποδειχτεί το εξής αποτέλεσμα όσον αφορά το textbook-RSA:

Θεώρημα 12.3.1 Έστω ότι έχουμε ένα μαντείο που δέχεται ως είσοδο κρυπτογραφημένα μηνύματα και επιστρέφει το LSB^{119} του μηνύματος. Δηλαδή, με είσοδο c επιστρέφει $\text{LSB}(m)$, όπου $m = \text{RSA}_d(c)$. 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 ολόκληρο το μήνυμα. Ο Daniel Bleichenbacher χρησιμοποίησε παρόμοιο RSA-μαντείο, για να σπάσει το RSA-PKCS#1 ver.1.5 που χρησιμοποιούνταν στο SSL v.3.¹²⁰ Συμπεραίνουμε ότι παρόλο που το textbook-RSA βασίζει την ασφάλειά του στη δυσκολία της παραγοντοποίησης μεγάλων ακεραίων, δεν δημιουργεί ένα ασφαλές κρυπτοσύστημα. Έχουμε μέχρι στιγμής τα παρακάτω,

- 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 ^a(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 12.2.1.
- 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 ($m = \text{RSA}_e(C)$), 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).

^a Η Εύα δεν χρησιμοποιεί κάποιο μαντείο, όπως στις επιθέσεις CPA και CCA. Απλά παρακολουθεί την επικοινωνία.

Όσον αφορά την επιλογή $e = 3$, μπορούμε 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 κρυπτογράφησης 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).

¹¹⁹ LSB : Least Significant Bit

¹²⁰ Daniel Bleichenbacher, Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS #1, 1998

Άσκηση 12.5 Να περιγράψετε αναλυτικά το σύστημα ΟΑΕΡ.

Βιβλιογραφία

- [1] Cormen, T. H; Leiserson C. E.; Rivest E. L.; Stein C. *Introduction To Algorithms*, (Third edition), MIT Press, 2009.
- [2] Galbraith, Steven. *Mathematics of Public Key Cryptography (version 1.1)*. <http://www.math.auckland.ac.nz/~sgal018/crypto-book/crypto-book.html>, 2011.
- [3] Miccianchio, D; Goldwasser, S. *Complexity of Lattice problems. A Cryptographic Perspective*. Springer, 2002.
- [4] Wagstaff, Samuel. *The joy of factoring*. Student Mathematical Library, vol. 68, 2013. AMS.
- [5] Τσίχλας, Κ.; Μανωλόπουλος, Ι.; Γούναρης, Α. *Σχεδίαση και Ανάλυση Αλγορίθμων*, Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. <http://dx.doi.org/10.57713/kallipos-613>, 2015.

Κεφάλαιο 13

Ψηφιακές Υπογραφές

Σύνοψη - Περίληψη. Σε αυτήν την ενότητα παρουσιάζουμε βασικές έννοιες των ψηφιακών υπογραφών (digital signatures). Ξεκινάμε με τον ορισμό και κατόπιν δίνουμε τους ορισμούς ασφάλειας. Επίσης, παρουσιάζουμε ψηφιακές υπογραφές που βασίζονται σε συναρτήσεις, TDF όπως είναι η υπογραφή RSA. Κατόπιν, παρουσιάζουμε και μια ψηφιακή υπογραφή, την DSA, που δεν βασίζεται σε TDF. Σήμερα χρησιμοποιούμε την εκδοχή με ελλειπτικές καμπύλες της DSA, τη γράφουμε ECDSA. Για παράδειγμα, στο πρωτόκολλο του Bitcoin για την υπογραφή των συναλλαγών από έναν χρήστη χρησιμοποιούμε ECDSA. Στο δεύτερο μέρος αυτού του κεφαλαίου παρουσιάζουμε και τα συστήματα ταυτοποίησης (identity schemes). Υπάρχει μια βασική αναγωγή που μετατρέπει αυτά τα συστήματα σε ψηφιακές υπογραφές, και αυτός είναι ο λόγος που τα παρουσιάζουμε. Έχουν και ανεξάρτητο ενδιαφέρον, καθώς σχετίζονται με τα συστήματα μηδενικής γνώσης (Zero-knowledge Systems). Θα παρουσιάσουμε το σύστημα ταυτοποίησης του Schnorr το οποίο κατασκευάζει την ψηφιακή υπογραφή του Schnorr (μια παραλλαγή του DSA).

Προαπαιτούμενη γνώση. Για αυτό το κεφάλαιο χρειάζεται να γνωρίζουμε τον ορισμό των TDF: Trapdoor Functions, το σύστημα RSA, καθώς και βασική θεωρία αριθμών. Μπορείτε να ανατρέξετε στα προηγούμενα κεφάλαια αυτού του βιβλίου. Επίσης, μπορείτε να δείτε για θεωρία αριθμών τη βιβλιογραφία [1, 3, 4, 7, 6, 8] ή στην ελληνόγλωσση βιβλιογραφία [9, 10, 11].

13.1 Ορισμός ψηφιακής υπογραφής

Οι ψηφιακές υπογραφές χρησιμοποιούνται για την απόδειξη της γνησιότητας ενός ψηφιακού εγγράφου. Είναι συστήματα δημόσιου κλειδιού, αλλά μοιάζουν λίγο με τις συναρτήσεις MAC. Στις MAC οι διαδικασίες υπογραφής και επαλήθευσης γίνονται με ένα κλειδί. Στις ψηφιακές υπογραφές αυτές οι διαδικασίες γίνονται με διαφορετικά κλειδιά. Επίσης, αν θέλουμε να υπογράψουμε ένα έγγραφο με μία MAC και να το στείλουμε σε δύο διαφορετικές οντότητες, τότε πρέπει να φτιάξουμε δύο διαφορετικά ιδιωτικά κλειδιά. Ωστόσο, με την ψηφιακή υπογραφή μπορούμε να υπογράψουμε για διαφορετικούς χρήστες με το ίδιο κλειδί. Ας υποθέσουμε ότι η Alice θέλει να υπογράψει ένα μήνυμα και να το στείλει στον Bob.

Ειδικότερα, η Alice υπογράφει με το ιδιωτικό κλειδί και ο Bob επαληθεύει με το δημόσιο κλειδί του αποστολέα, δηλαδή της Alice. Επομένως, 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 ενεργειών υπογραφής του εγγράφου (non-repudiation). Επίσης, κατά αναλογία με την ιδιόχειρη υπογραφή, όπου τοποθετούμε το υπογεγραμμένο μήνυμα σε έναν φάκελο και το σφραγίζουμε, μπορούμε το ψηφιακά υπογεγραμμένο μήνυμα να το κρυπτογραφήσουμε με το δημόσιο κλειδί του παραλήπτη. Εκτός από την υπογραφή μηνυμάτων οι ψηφιακές υπογραφές χρησιμοποιούνται στις παιχνιδομηχανές (Xbox, PS3). Αυτή η παράδοση ξεκίνησε από τη Nintendo το 1985 και συνεχίστηκε με την Atari. Η Nintendo χρησιμοποίησε MAC συναρτήσεις για την αυθεντικοποίηση των παιχνιδιών. Μια άλλη εφαρμογή των ψηφιακών υπογραφών είναι η αναβάθμιση (update or patch) κάποιου προγράμματος του υπολογιστή μας. Για παράδειγμα, πώς διασφαλίζεται η ακεραιότητα¹²¹ ενός kernel security patch στο linux; Μέσα από κάποιο κανάλι, που είναι αυθεντικοποιημένο, προμηθευόμαστε το δημόσιο κλειδί του προγραμματιστή (π.χ. του Linus Torvalds) ή της εταιρείας (kernel.org) που προμηθεύει τη βελτίωση ασφάλειας του πυρήνα. Στο πρόγραμμα που κατεβάζουμε υπάρχει και μια ψηφιακή υπογραφή η οποία δημιουργήθηκε από τον προγραμματιστή της βελτίωσης. Επομένως, μπορούμε να επαληθεύσουμε αν η βελτίωση είναι έγκυρη, διότι έχουμε το δημόσιο κλειδί του.

Ορισμός 13.1.1. 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) και δεν δέχεται καμία είσοδο. Το κλειδί pk ονομάζεται δημόσιο κλειδί ή κλειδί επαλήθευσης, ενώ το sk ονομάζεται ιδιωτικό κλειδί ή κλειδί υπογραφής.

- S : αλγόριθμος υπογραφής (Sign).

Είναι ένας πιθανοτικός πολυωνυμικός αλγόριθμος που δέχεται δύο παραμέτρους ως είσοδο sk, m όπου m ένα μήνυμα, $S(sk, m) = s$. Η έξοδος ονομάζεται υπογραφή του μηνύματος.

- V : αλγόριθμος επαλήθευσης (Verify).

Είναι ντετερμινιστικός αλγόριθμος πολυωνυμικού χρόνου ο οποίος δέχεται ως είσοδο την υπογραφή s και το μήνυμα m και επιστρέφει *True* ή *accept*, αν η υπογραφή είναι σωστή, και *False* ή *reject* διαφορετικά.

13.2 Ασφάλεια μιας ψηφιακής υπογραφής

13.2.1 Επιθέσεις σε ψηφιακές υπογραφές

Υπάρχουν οι εξής επιθέσεις:

Παθητική επίθεση. Σε αυτήν την επίθεση απλά ο επιτιθέμενος παρακολουθεί την επικοινωνία χωρίς να μπορεί να παρέμβει.

Επίθεση γνωστού κειμένου¹²². Σε αυτήν την επίθεση απλά ο επιτιθέμενος παρακολουθεί την

¹²¹<https://www.kernel.org/doc/html/latest/process/maintainer-pgp-guide.html>

¹²²Known message attack

επικοινωνία, αλλά του έχουν δοθεί κάποιες επιπλέον υπογραφές μηνυμάτων.

Επίθεση επιλεγμένου κειμένου¹²³. Σε αυτήν την επίθεση απλά ο επιτιθέμενος παρακολουθεί την επικοινωνία αλλά μπορεί και να παρέμβει στην επικοινωνία με τον εξής τρόπο: μπορεί να επιλέξει συγκεκριμένα μηνύματα και να ζητήσει την υπογραφή τους. Δηλαδή, εκτός του δημόσιου κλειδιού και των υπογραφών που ανταλλάσσονται στον δίαυλο επικοινωνίας, μπορεί να έχει στην κατοχή του ζευγάρια (m_i, s_i) , όπου s_i είναι η ψηφιακή υπογραφή του μηνύματος m_i .

Προσαρμοσμένη επίθεση επιλεγμένου κειμένου¹²⁴. Σε αυτήν την επίθεση απλά ο επιτιθέμενος παρακολουθεί την επικοινωνία αλλά μπορεί και να παρέμβει στην επικοινωνία με τον εξής τρόπο: μπορεί να επιλέξει συγκεκριμένα μηνύματα και να ζητήσει την υπογραφή τους. Μετά την υπογραφή τους μπορεί να ξαναστείλει μηνύματα προς υπογραφή, βασιζόμενος σε αυτά που έχει ήδη πάρει.

13.2.2 Ευπάθειες ψηφιακής υπογραφής

Έχουμε τις εξής ευπάθειες σε μια ψηφιακή υπογραφή:

Υπαρκτή Πλαστογράφηση. Λέμε ότι έχουμε υπαρκτή πλαστογράφηση αν κάποιος τρίτος καταφέρνει να υπογράψει ένα μήνυμα. Το μήνυμα δεν έχει απαραίτητα κάποιο νόημα. Αν αυτή η αδυναμία υπάρχει σε μια ψηφιακή υπογραφή, δεν συνεπάγεται ότι το σχήμα ψηφιακής υπογραφής δεν είναι ασφαλές.

Επιλεκτική Πλαστογράφηση. Σε αυτήν την περίπτωση κάποια μηνύματα τα οποία διάλεξε κάποιος τρίτος μπορεί να τα υπογράψει.

Ολική Πλαστογράφηση. Σε αυτήν την περίπτωση κάποιος τρίτος μπορεί και υπογράφει όποιο μήνυμα θέλει. Το σχήμα της ψηφιακής υπογραφής με αυτήν την αδυναμία είναι μη ασφαλές. Δεν είναι απαραίτητο κάποιος να έχει το μυστικό κλειδί, για να πετύχει ολική πλαστογράφηση.

Ολικό σπάσιμο (total crack). Σε αυτήν την περίπτωση το μυστικό κλειδί μπορεί να υπολογιστεί από κάποια τρίτη οντότητα.

13.2.3 Ορισμός Ασφάλειας ψηφιακής υπογραφής

Ορισμός 13.2.1. Λέμε ότι μια ψηφιακή υπογραφή είναι ασφαλής, αν με επίθεση Προσαρμοσμένου Επιλεγμένου Κειμένου δεν πετυχαίνει η υπαρκτή πλαστογράφηση.

Δηλαδή, η πιο ισχυρή επίθεση δεν μπορεί να παράγει ούτε ένα ζευγάρι (m, s) που να περνάει με επιτυχία τον αλγόριθμο επαλήθευσης.

13.3 Ψηφιακές Υπογραφές από TDF

Οι Diffie και Hellman παρατήρησαν ότι ένα κρυπτοσύστημα δημόσιου κλειδιού, που προκύπτει από μια TrapDoor Function, έστω f , μπορεί να χρησιμοποιηθεί και για να υπογράψουμε μηνύματα. Αυτό μπορεί να γίνει απλά επεκτείνοντας το μήνυμα m με την υπογραφή $s = f^{-1}(sk, m)$, όπου sk το ιδιωτικό κλειδί της TrapDoor function f και στέλνει στον παραλήπτη (m, s) . Ο παραλήπτης επαληθεύει 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 υπογράψει.
- (ii). Οι χρήστες, που γνωρίζουν το δημόσιο κλειδί, μπορούν να ελέγξουν ή να επαληθεύσουν την εγκυρότητα της υπογραφής ενός μηνύματος (verification).
- (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) (non-repudiation).

¹²³Chosen message attack

¹²⁴Adaptive chosen message attack

13.3.1 Υπογραφή RSA

Όπως αναφέραμε στην προηγούμενη ενότητα 13.3, εφόσον το RSA είναι μια TDF, μπορεί να χρησιμοποιηθεί για την κατασκευή ψηφιακής υπογραφής. Έστω m το μήνυμα που θέλουμε να υπογράψουμε. Ας είναι (e, N) το δημόσιο και (d, N) το ιδιωτικό κλειδί του συστήματος RSA. Η υπογραφή γίνεται με χρήση του ιδιωτικού κλειδιού :

Υπογραφή του m : $s = m^d \bmod N$. Η επαλήθευση γίνεται από όλους όσοι έχουν το δημόσιο κλειδί μας.

Επαλήθευση της υπογραφής s .

i. Υπολογίζουμε την $\alpha = s^e \bmod N$.

ii. Ελέγχουμε αν $\alpha = m$

Αν ο έλεγχος είναι επιτυχής, τότε αποδεχόμαστε την ψηφιακή υπογραφή, διαφορετικά όχι. Παρόλα αυτά, η συγκεκριμένη θέλει τροποποίηση, για να γίνει ασφαλής.

Πρόταση 13.3.1 Η ψηφιακή υπογραφή που βασίζεται στο RSA είναι ολικά παραποιήσιμη με επίθεση επιλεγμένου κειμένου.

Απόδειξη. Έστω m το μήνυμα που θέλουμε να υπογράψουμε. Υλοποιούμε την επίθεση επιλεγμένου κειμένου και σχηματίζουμε τα εξής μηνύματα:

1ο μήνυμα. $m_1 = mr \bmod N$, όπου r τυχαίος αριθμός από το $\mathbb{Z}_N^* - \{1\}$.

2ο μήνυμα. $m_2 = r^{-1} \bmod N$.

Ζητάμε τις υπογραφές των m_1, m_2 , έστω s_1, s_2 , αντίστοιχα. Τότε, η υπογραφή του μηνύματος m είναι απλά το γινόμενο των s_1 και s_2 . Πράγματι,

$$s_1 s_2 = (mr)^d r^{-d} \bmod N = m^d \bmod N$$

που είναι η ψηφιακή υπογραφή του μηνύματος m . □

Ένας τρόπος να αποφύγουμε την προηγούμενη επίθεση είναι, πριν υπογράψουμε το μήνυμα, να εφαρμόσουμε μια ασφαλή συνάρτηση¹²⁵ κατακερματισμού H .

Υπογραφή του m : $s = H(m)^d \bmod N$. Η επαλήθευση γίνεται από όλους όσοι έχουν το δημόσιο κλειδί μας.

Επαλήθευση της υπογραφής s .

i. Υπολογίζουμε την $\alpha = s^e \bmod N$.

ii. Υπολογίζουμε την $H(m)$.

iii. Ελέγχουμε αν $\alpha \equiv H(m) \bmod N$.

Παράδειγμα 13.3.1. Έστω $p = 31, q = 23$ και $N = p \times q = 713$. Επίσης, $e = 283, d = 667$. Θεωρούμε το μήνυμα $m = 4 \in \mathbb{Z}_N^*$. Θα χρησιμοποιήσουμε τη naïve RSA υπογραφή (δηλαδή, χωρίς την εφαρμογή της hash). Η υπογραφή του μηνύματος είναι

$$s = m^d \bmod N = 4^{667} \bmod 713.$$

Με χρήση του αλγορίθμου της ενότητας 7.2.3, μπορούμε να υπολογίσουμε

$$4^{667} \bmod 713 = 698.$$

Άρα στέλνουμε το ζευγάρι $(m = 4, s = 698)$, που είναι η ψηφιακή υπογραφή του m , καθώς και το μήνυμα m . Για να επαληθεύσει κάποιος την ψηφιακή υπογραφή, κάνει τις εξής πράξεις:

$$s^e = 698^{283} \bmod 713 = 4.$$

Οπότε δέχονται την ψηφιακή υπογραφή ως σωστή.

¹²⁵ Δηλαδή, να είναι collision free.

Άσκηση 13.1 Αν $p = 17, q = 19$ και $N = 323, e = 17, d = 305$, να υπολογίσετε την RSA-ψηφιακή υπογραφή του μηνύματος $m = 2$.

Άσκηση 13.2 Δίνεται το δημόσιο κλειδί μιας (naive) ψηφιακής υπογραφής RSA, $(N, e) = (899, 839)$. Αν η ψηφιακή υπογραφή του μηνύματος $m = 3$ είναι $s = 301$, μπορείτε να επαληθεύσετε ότι είναι σωστή η ψηφιακή υπογραφή;

13.3.2 DSA

DSA : Digital Signature Algorithm είναι μια ψηφιακή υπογραφή που χρησιμοποιείται επίσημα από την αμερικάνικη κυβέρνηση από το 1991. Επίσης, στο SSL/TLS χρησιμοποιείται για την αυθεντικοποίηση της επικοινωνίας. Σήμερα χρησιμοποιείται η εκδοχή με τις ελλειπτικές καμπύλες¹²⁶. Αυτή η ψηφιακή υπογραφή βασίζει την ασφάλειά της στον διακριτό λογάριθμο και δεν προέρχεται από κάποια TDF, όπως η υπογραφή RSA που παρουσιάσαμε προηγουμένως.

Έστω ότι η Alice θέλει να υπογράψει ένα μήνυμα και να το στείλει στον Bob. Αρχικά, η Alice παράγει δύο πρώτους αριθμούς p, q με $q|p - 1$. Τα μήκη των (q, p) είναι $(160, 1024)$ ή $(224, 2048)$ ή $(256, 2048)$ bits. Ας είναι g ένα στοιχείο του $\mathbb{Z}_p^*$ τάξης q . Δηλαδή, το g παράγει μια υποομάδα τάξης q . Επίσης, διαλέγει τυχαία έναν αριθμό a (ιδιωτικό κλειδί) από το σύνολο $\{2, 3, \dots, q - 2\}$ και υπολογίζει το $R = g^a \pmod{p}$. Το δημόσιο κλειδί είναι η τετράδα (p, q, g, R) , καθώς και μια κρυπτογραφικά ασφαλής συνάρτηση κατακερματισμού έστω

$$h : \{0, 1\}^* \rightarrow \{1, 2, \dots, q - 1\}$$

και το ιδιωτικό κλειδί είναι ο ακέραιος x . Έστω $m < q$ το μήνυμα που θέλει η Alice να υπογράψει. Διαλέγει ένα εφήμερο κλειδί (ephemeral key ή nonce) k , τυχαία από το σύνολο $\{1, 2, \dots, q - 1\}$ και υπολογίζει τους αριθμούς

$$r = (g^k \bmod p) \bmod q, \quad s = k^{-1}(h(m) + ar) \bmod q.$$

Το (r, s) είναι η υπογραφή του μηνύματος m . Ο Bob επαληθεύει την υπογραφή υπολογίζοντας τον αριθμό

$$(g^{s^{-1}h(m) \bmod q} R^{s^{-1}r \bmod q} \bmod p) \bmod q$$

και ελέγχει αν ισούται με r .

Παρατήρηση 13.3.1. Η ψηφιακή υπογραφή DSA βασίζεται στη δυσκολία επίλυσης του διακριτού λογάριθμου. Για παράδειγμα η Εύα μπορεί να επιτεθεί στο σύστημα αν λύσει το πρόβλημα διακριτού λογάριθμου στην ομάδα $\mathbb{Z}_p^*$ χρησιμοποιώντας τον index calculus method, που είναι υποκθετικής πολυπλοκότητας. Επίσης, επειδή υπάρχει υποομάδα G τάξης q της ομάδας $\mathbb{Z}_p^*$ η Εύα μπορεί να χρησιμοποιήσει έναν γενικό αλγόριθμο, όπως του Pollard για να επιτεθεί στον διακριτό λογάριθμο στην ομάδα G τάξης q . Αυτό θα κόστιζε $O(\sqrt{q})$ bit-complexity.

Άσκηση 13.3 Να αποδείξετε ότι, αν στην ψηφιακή υπογραφή DSA κάποιος χρησιμοποιήσει δύο φορές το ίδιο εφήμερο κλειδί (για δύο διαφορετικά μηνύματα), τότε μπορεί να υπολογίσει το ιδιωτικό κλειδί a .

13.3.3 Επίλυση διαφωνιών

Μια ασφαλής ψηφιακή υπογραφή δεν μπορεί να πλαστογραφηθεί. Ας υποθέσουμε ότι η Εύα υποστηρίζει ότι η Alice υπέγραψε ένα κείμενο, πράγμα που δεν παραδέχεται η Alice. Σε αυτήν την περίπτωση η Alice πρέπει να αποδείξει ότι το δημόσιο κλειδί που χρησιμοποίησε η Εύα δεν είναι αυτό που έχει η Alice. Γι' αυτόν τον λόγο υπάρχουν οι αρχές πιστοποίησης¹²⁷. Η Alice, πριν αρχίσει να χρησιμοποιεί την υπογραφή της, πηγαίνει σε μια αρχή πιστοποίησης που υπογράφει το δημόσιο κλειδί της. Η αρχή πιστοποίησης ελέγχει

¹²⁶το 1998 προτάθηκε η εκδοχή του DSA με ελλειπτικές καμπύλες. Για παράδειγμα, το bitcoin χρησιμοποιεί αυτήν την εκδοχή του DSA.

¹²⁷certification authorities

σχολαστικά την ταυτότητα της Alice και κατόπιν συνδέει το δημόσιο κλειδί της με την Alice, υπογράφοντάς το ψηφιακά. Έτσι, η Εύα που υποστηρίζει ότι η Alice υπέγραψε ένα κείμενο, πρέπει να δείξει το δημόσιο κλειδί που χρησιμοποίησε μαζί με την υπογραφή της αρχής πιστοποίησης. Είναι πρακτικά αδύνατο να έχει η Εύα ένα δημόσιο κλειδί διαφορετικό της Alice που να πλαστογραφεί υπογραφές της Alice. Δείτε [5, ενότητα 1.8.3].

Οι αρχές πιστοποίησης, εκτός από την ψηφιακή υπογραφή των δημόσιων κλειδιών των χρηστών, υπογράφουν και SSL πιστοποιητικά που χρησιμοποιούνται κατά τη διαδικασία αυθεντικοποίησης ενός διακομιστή (server) από έναν πελάτη (client) στο πρωτόκολλο SSL/TLS (δείτε το κεφάλαιο 15). Επίσης, οι αρχές πιστοποίησης υπογράφουν δημόσια κλειδιά ψηφιακών καρτών (etoken ή smart cards) πολιτών. Στην τελευταία περίπτωση η αρχή πιστοποίησης συνήθως είναι μια κυβερνητική υπηρεσία. Σε κάθε άλλη περίπτωση η αρχή πιστοποίησης είναι συνήθως ένας πολύ μεγάλος ιδιωτικός (ή μπορεί και μη κερδοσκοπικός) οργανισμός που συνήθως χρεώνει αυτήν την υπηρεσία. Αυτό συμβαίνει, γιατί η συγκεκριμένη αρχή πιστοποίησης γίνεται αποδεκτή από μεγάλες εταιρείες, όπως είναι οι google, microsoft, apple, και από μη κερδοσκοπικούς οργανισμούς, όπως είναι η mozilla foundation.

Υπάρχει ακόμη ένας τρόπος να επιλύσουμε διαφωνίες χωρίς τη χρήση των αρχών πιστοποίησης. Αυτή η μέθοδος χρησιμοποιείται στο πρωτόκολλο κρυπτογράφησης/ψηφιακών υπογραφών για ηλεκτρονικά μηνύματα, PGP, δείτε το κεφάλαιο 15.

13.4 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 στη μηχανή, με 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 στη 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 βρει 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 να τρέχουν αυτά τα συστήματα 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 στη δυσκολία υπολογισμού των τετραγωνικών ριζών mod n , όταν δεν γνωρίζουμε την παραγοντοποίηση του n . Επίσης, πιο νέα συστήματα ταυτοποίησης βασίζουν την ασφάλειά 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 σε τσιπάκια όπως τα rfid-tags. Στα τελευταία, όμως έχουν βρεθεί λύσεις όσον αφορά τη συμμετρική κρυπτογραφία, π.χ. ο αλγόριθμος hummingbird και SEA. Στην κρυπτογραφία δημόσιου κλειδιού δεν υπάρχουν ανάλογοι (lightweight) αλγόριθμοι.

13.4.1 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 $r \in \{1, 2, \dots, q-1\}$ και στέλνει στον Bob $x = g^r \pmod{p}$.

2ο βήμα. Ο Bob διαλέγει έναν τυχαίο αριθμό $e \in \{1, 2\}$ και τον στέλνει στην Alice.

3ο βήμα. Η Alice υπολογίζει τον αριθμό $y = r + es \pmod{q}$ και τον 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 13.4.1. Λέμε ότι ένα σύστημα ταυτοποίησης τριών κινήσεων είναι σημασιολογικά ασφαλές (sound), αν και μόνο 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 $\{1, 2, \dots, q-1\}$ 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 $r \in \{1, 2, \dots, q-1\}$ και στέλνει στον Bob $x = g^r \pmod{p}$.

2ο βήμα. Ο Bob διαλέγει έναν τυχαίο αριθμό $e \in \{1, 2, \dots, 2^t\}$ και τον στέλνει στην Alice

3ο βήμα. Η Alice υπολογίζει τον αριθμό $y = r + es \pmod{q}$ και τον 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 13.4.1 (soundness). Έστω ότι υπάρχει πιθανοτικός αλγόριθμος $\mathcal{A}$ τέτοιος, ώστε με είσοδο (pk, e) , όπου το e είναι τυχαίο από το σύνολο $\{1, \dots, 2^t\}$, επιστρέφει με πιθανότητα $\varepsilon > 2^{-t+1}$ ένα ζευγάρι (x, y) το οποίο περνάει το τεστ αυθεντικοποίησης και αυτό γίνεται σε χρόνο $|A|$. Τότε, με θετική σταθερή πιθανότητα και σε χρόνο $O(|A|/\varepsilon)$, μπορεί να υπολογιστεί ο διακριτός λογάριθμος $\log_g(v)$ στο $\mathbb{Z}_p^*$.

Σύμφωνα με αυτό το θεώρημα, 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 $\{1, \dots, 2^t\}$, εκτός και αν μπορεί να υπολογίσει τον διακριτό λογάριθμο του v . Με άλλα λόγια αν ο Bob συμπεριφέρεται έντιμα (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 13.4.2. 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 $e = 1, 2, \dots, 2^t$ και σχηματίζουμε τη γραμμή που αποτελείται από τα $S_{\mathcal{A}}(IS_{\mathcal{A}}, pk, e)_{e=1,2,\dots}$. 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

$$M_{pk} = \begin{matrix} & \begin{matrix} 1 & 2 & \dots & 2^t \end{matrix} \\ \begin{matrix} IS_1 \\ IS_2 \\ \vdots \\ IS_N \end{matrix} & \begin{pmatrix} 0 & 1 & \dots & 1 \\ 1 & 0 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{pmatrix} \end{matrix}$$

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 στη 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 άσους, ώστε με σταθερή πιθανότητα οι δύο στήλες που διαλέγουμε τυχαία να μας προμηθεύουν με δύο επιτυχημένες τριάδες $(x, e_1, y_1), (x, e_2, y_2)$. Δηλαδή,

$$M_{pk}(IS_i, e_1) = 1, M_{pk}(IS_i, e_2) = 1$$

με $e_1 \neq e_2$. Τότε, εφαρμόζουμε το παρακάτω λήμμα. Η παρακάτω ιδιότητα που θα αποδείξουμε στο λήμμα ονομάζεται και special soundness.

Λήμμα 13.4.1. (*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 13.4.1. 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. Επομένως, 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 13.4.3. Μία γραμμή του πίνακα M_{pk} την ονομάζουμε βαριά (*heavy*), αν η σχετική συχνότητα των 1 στη γραμμή είναι $\geq \epsilon/2$.

Λήμμα 13.4.2. Έστω g το πλήθος των άσων στις βαριές γραμμές του πίνακα M_{pk} . Επίσης, $h = 2^t \times N$ είναι το πλήθος των στοιχείων του M_{pk} . Τότε, $g/h \geq \epsilon/2$.

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 $h\epsilon$, ενώ στον M' είναι $< h'\epsilon/2$. Επομένως,

$$g \geq h\epsilon - h'\epsilon/2 \geq h\epsilon - h\epsilon/2 = h\epsilon/2.$$

□

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 13.4.3. *Αν διαλέξουμε τυχαία τουλάχιστον $\frac{1}{\epsilon}$ 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 $\frac{1}{\epsilon}$ 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 τη στήλη), οπότε υπολογίζουμε το y . Αν η γραμμή είναι βαριά, τότε με πιθανότητα $1/2$, η τριάδα (x, e, y) είναι πετυχημένη (δηλαδή, περνάει το τεστ αυθεντικοποίησης). Αλλά μια γραμμή είναι βαριά με πιθανότητα $\epsilon/2$. Επομένως, μετά από $\frac{1}{\epsilon}$ τυχαίες επιλογές του $IS_{\mathcal{A}}$ θα πετύχουμε μια βαριά γραμμή με πιθανότητα $\frac{1}{2}$. $\square$

Απόδειξη του Θεωρήματος. Καλούμε τον αλγόριθμο $\mathcal{A}$, έως ότου βρούμε τον πρώτο άσο. Αυτό μπορεί να επιτευχθεί ύστερα από το πολύ $1/\epsilon$ 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

$$\frac{h\epsilon/2}{h} \cdot \# \text{κλήσεις του } \mathcal{A} = \frac{\epsilon}{2} \cdot \frac{1}{\epsilon} = \frac{1}{2}.$$

Καλούμε $1/\epsilon$ φορές τον αλγόριθμο $\mathcal{A}$. Μετά από $1/\epsilon$ προσπάθειες θα βρούμε το πρώτο 1, αν ψάξουμε στον H τυχαία (δηλαδή, αν διαλέγουμε μια τυχαία εσωτερική κατάσταση και ένα τυχαίο διάνυσμα 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

$$\frac{\frac{\epsilon}{2} 2^t - 1}{2^t}.$$

Και γι' αυτό χρειαζόμαστε

$$\frac{2^t}{\frac{\epsilon}{2} 2^t - 1}$$

προσπάθειες για επιτυχία. Αφού

$$\frac{2^t}{\frac{\epsilon}{2} 2^t - 1} < \frac{2^t}{\frac{\epsilon}{2} 2^{t-1}} = 4/\epsilon,$$

αυτό σημαίνει ότι με λιγότερες από $\frac{4}{\epsilon}$ 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 δύο 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 έναν τυχαίο αριθμό μεταξύ των $1, 2, \dots, d$ (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 $O(|\mathcal{A}|/\epsilon)$. 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 $(1 - \varepsilon/d)^{k-1} \leq 1$ 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 πιθανότητα επιτυχίας για το βήμα βήμα 2 να γίνει $1/2$. Επιπλέον, αν επιλέξουμε $d = 16$, τότε το βήμα 2 τερματίζει μετά 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 $\frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$. Οπότε, αν συμβεί αυτό, τότε με πιθανότητα τουλάχιστον $1/8$ παίρνουμε ένα δεύτερο 1.

$$\mathcal{A} \xrightarrow[\text{Pr} > 1/2]{O(|A|/\varepsilon)} 1\text{η δοκιμή} \longrightarrow \mathcal{A}_1 \xrightarrow[\text{Pr} > 1/8]{O(|A|/\varepsilon)} 2\text{η δοκιμή}$$

Έτσι, 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 $O(|A|/\varepsilon)$ και πετυχαίνει να βρει δυο 1 στην ίδια γραμμή με σταθερή πιθανότητα $> 1/16$. Και 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 13.4.1, το θεώρημα έπεται. $\square$

Αφού υποθέσαμε ότι το να βρούμε στοιχεία από το Σ_b είναι δύσκολο, δεν μπορούμε να βελτιώσουμε την πιθανότητα επιτυχίας, ώστε $> 2^{-t+2}$. Δείτε την αναφορά [2] για περισσότερες λεπτομέρειες.

Το σύστημα ταυτοποίησης του Schnorr μπορεί να χρησιμοποιηθεί για να κατασκευάσουμε μία ψηφιακή υπογραφή εφαρμόζοντας τον μετασχηματισμό των Fiat-Shamir. Γενικά, αυτός ο μετασχηματισμός μπορεί να χρησιμοποιηθεί σε οποιοδήποτε σύστημα ταυτοποίησης τριών βημάτων. Μπορεί να αποδειχθεί ότι ο μετασχηματισμός αυτός στην περίπτωση του Schnorr, δίνει μια ασφαλή ψηφιακή υπογραφή έναντι επιθέσεων επιλεγμένου χειμένου.

Βιβλιογραφία

- [1] Boneh, Dan; Shoup, V. *A Graduate Course in Applied Cryptography*, Version 0.5, Jan. 2020. <http://toc.cryptobook.us/>.
- [2] Damgard, I. *On Σ protocols*. <https://cs.au.dk/~ivan/Sigma.pdf>, 2010.
- [3] Galbraith, Steven. *Mathematics of Public Key Cryptography (version 1.1)*. <http://www.math.auckland.ac.nz/~sgal018/crypto-book/crypto-book.html>, 2011.
- [4] Grandall, R.; Pommerance, C. *Prime Numbers. A Computational Perspective*. 2nd edition, pp. 596, Springer, 2005.
- [5] Menezes, A.; van Oorschot, P.; Vanstone, S. *Handbook of Applied Cryptography*, CRC Press. <https://cacr.uwaterloo.ca/hac/>, 1996.
- [6] Mollin, R. *An Introduction to Cryptography*. 2nd edition, Series : Discrete mathematics and its applications. Taylor and Francis Group, 2007.
- [7] Shoup, Victor. *A Computational Introduction to Number Theory and Algebra (2nd edition)*, Cambridge University Press. <https://shoup.net/ntb> , 2008.
- [8] Stein, W. *Elementary Number Theory : Primes, Congruences, and Secrets*. <http://wstein.org/ent/>, 2011.
- [9] Πουλλάκης, Δ. *Θεωρία Αριθμών*, Εκδ. Ζήτη, 2009.
- [10] Τζανάκης, Νίκος. *Θεμελιώδης Θεωρία Αριθμών*, Σημειώσεις, Πανεπιστήμιο Κρήτης, 2012.
- [11] Ζάχος, Ε.; Παγουρτζής, Α.; Γροντάς, Π. *Υπολογιστική Κρυπτογραφία*, Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. <http://dx.doi.org/10.57713/kallipos-492>, 2015.

Κεφάλαιο 14

Πλέγματα

Σύνοψη - Περίληψη. Σε αυτήν την ενότητα παρουσιάζουμε βασικές έννοιες των πλεγμάτων (lattices). Τα πλέγματα χρησιμοποιούνται στην κρυπτογραφία είτε για κρυπτανάλυση (π.χ. η επίθεση του Coppersmith στο RSA) είτε για την κατασκευή κρυπτοσυστημάτων ανθεκτικών σε κβαντικές επιθέσεις. Το βασικό πρόβλημα με το οποίο θα ασχοληθούμε είναι το πρόβλημα του μικρότερου διανύσματος (shortest vector problem).

Προαπαιτούμενη γνώση. Για αυτό το κεφάλαιο χρειάζονται πολύ καλές γνώσεις βασικής γραμμικής άλγεβρας. Για παράδειγμα: η διαδικασία Gram-Schmidt για εύρεση μιας ορθογώνιας βάσης, βασική άλγεβρα πινάκων, εσωτερικό γινόμενο, μέτρο διανύσματος, και γενικά εξοικείωση με διανυσματικούς χώρους. Για μια εισαγωγή στη γραμμική άλγεβρα δείτε, [3, 6, 7]. Επίσης, χρειάζονται βασικές γνώσεις γραφημάτων και αλγορίθμων, όπως ο DFS. Για τα γραφήματα μπορείτε να δείτε [10, Κεφ. 6], [11, Κεφ. 5] και [8, Κεφ. 11]. Για τα πλέγματα μπορείτε να συμβουλευτείτε τις αναφορές [1, 4]. Τέλος, χρειάζονται βασικές γνώσεις θεωρίας πολυπλοκότητας και εξοικείωση με τα σύμβολα του Landau.

14.1 Εισαγωγή στα Πλέγματα

To not know math is a severe limitation to understanding the world.

– Richard P. Feynman

Ορισμός 14.1.1. Ένα υποσύνολο $L \subset \mathbb{R}^n$ καλείται πλέγμα (lattice), αν υπάρχουν γραμμικώς ανεξάρτητα διανύσματα $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k$ του $\mathbb{R}^n$ ($n \geq k$) τέτοια, ώστε

$$L = L(\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k) = \left\{ \sum_{j=1}^k \alpha_j \mathbf{b}_j : \alpha_j \in \mathbb{Z}, 1 \leq j \leq k \right\} = \{ \mathbf{x}B : \mathbf{x} \in \mathbb{Z}^k \},$$

όπου B έχει ως γραμμές τα $\mathbf{b}_1, \dots, \mathbf{b}_k$. Τα διανύσματα $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k$ καλούνται βάση του πλέγματος L .

Τα βασικά στοιχεία των πλεγμάτων (όπως και η προχωρημένη θεωρία) μπορούν να βρεθούν στις αναφορές [1, Κεφ.17], [9]. 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 και αυτός ο κοινός αριθμός k , ονομάζεται τάξη (rank) του πλέγματος L . Ο πίνακας M_L διαστάσεων $k \times n$ που σχηματίζεται από τα διανύσματα $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k$ ονομάζεται πίνακας βάσης του πλέγματος (basis matrix). Παρατηρήστε ότι η τάξη του πίνακα βάσης ισούται με την τάξη του πλέγματος L δηλαδή k . Κάθε στοιχείο του L , π.χ. το διάνυσμα του πλέγματος

$$a_1 \mathbf{b}_1 + \dots + a_k \mathbf{b}_k,$$

ισούται με $\mathbf{a}M_L$, όπου $\mathbf{a} = (a_1, \dots, a_k)$. Πράγματι, (αν θεωρήσουμε το $\mathbf{a}$ ως πίνακα γραμμή και τα $\mathbf{b}_i$ ως πίνακες γραμμή),

$$\mathbf{a}M_L = a_1 \mathbf{b}_1^T + a_2 \mathbf{b}_2^T + \dots + a_k \mathbf{b}_k^T.$$

Δηλαδή, $L = \mathbb{Z}^k M_L$. 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 14.1.1. Πολλές φορές στη βιβλιογραφία συναντάμε τον εξής ορισμό του πλέγματος.

Ορισμός 14.1.2. Μια προσθετική διακριτή υποομάδα $\mathcal{L}$ του $\mathbb{R}^n$ ονομάζεται πλέγμα του $\mathbb{R}^n$.

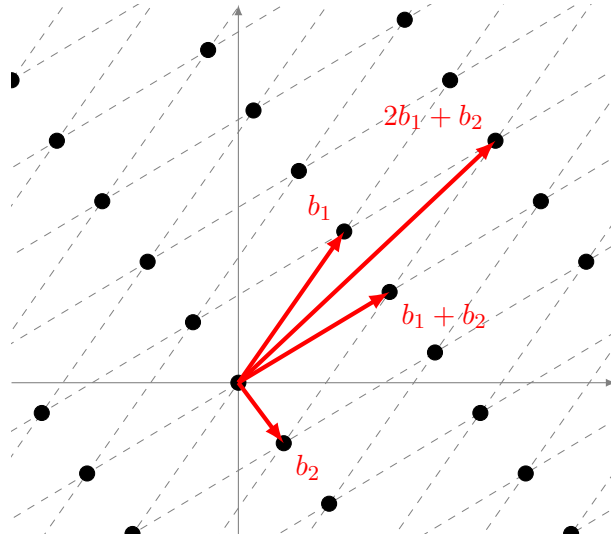
Ένα υποσύνολο $\mathcal{L}$ του $\mathbb{R}^n$ ονομάζεται διακριτό, αν και μόνο αν για κάθε συμπαγές υποσύνολο K του $\mathbb{R}^n$ έχουμε $|\mathcal{L} \cap K| < \infty$. Μπορούμε να αποδείξουμε ότι οι δύο ορισμοί είναι ισοδύναμοι, δηλαδή ο ορισμός (14.1.1) και (14.1.2) είναι ισοδύναμοι. Αρκεί να αποδείξουμε τη συνεπαγωγή (14.1.2) $\Rightarrow$ (14.1.1). Για την απόδειξη μπορείτε να δείτε [5, Ενότητα 4.1. Theorem I]. Το αντίστροφο, (14.1.1) $\Rightarrow$ (14.1.2), προκύπτει διότι τα συμπαγή υποσύνολα του $\mathbb{R}^n$ είναι μια ένωση πεπερασμένου πλήθους κλειστών μπαλών¹²⁸ και γνωρίζουμε ότι μια μπάλα περιέχει πεπερασμένου πλήθους σημεία του $\mathcal{L}$.

Παρατήρηση 14.1.2. Θεωρούμε τώρα τη γραμμική απεικόνιση $T : \mathbb{R}^k \rightarrow \mathbb{R}^n$ με

$$T(x_1, \dots, x_k) = x_1 \mathbf{b}_1 + \dots + x_k \mathbf{b}_k.$$

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, $T(\mathbf{e}_1) = T((1, 0, \dots, 0)) = \mathbf{b}_1$ κ.ο.κ. $T(\mathbf{e}_k) = \mathbf{b}_k$. Κατόπιν σχηματίζουμε τον πίνακα που έχει ως στήλες τα στοιχεία $T(\mathbf{e}_1), \dots, T(\mathbf{e}_k)$.

¹²⁸μια κλειστή μπάλα με κέντρο ένα σημείο $\mathbf{x}_0$ του $\mathbb{R}^n$ και ακτίνα $R > 0$ είναι ένα σύνολο της μορφής $\overline{B}_n(R, \mathbf{x}_0) = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x} - \mathbf{x}_0\| \leq R\}$.

Σχήμα 14.1: Ένα πλέγμα που παράγεται από τα $\mathbf{b}_1, \mathbf{b}_2$.

Παράδειγμα 14.1.1. (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 τη λύση $(0, 0)$, έτσι το πλέγμα είναι το $L_1 = \mathbb{Z} \cdot \mathbf{0}$ που είναι μηδενικής τάξης.

(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 ακέραιων λύσεων της γραμμικής εξίσωσης

$$a_1x_1 + \cdots + a_nx_n = a_0,$$

με $a_0 \neq 0$ και $\gcd(a_1, \dots, a_n) = 1$ δεν είναι πλέγμα, διότι $\mathbf{0} \notin L$.

(iv). Τα σύνολα

$$A_n = \{(x_0, \dots, x_n) \in \mathbb{Z}^{n+1} : \sum_{i=0}^n x_i = 0\} \quad (n \geq 1) \quad (14.1.1)$$

και

$$D_n = \{(x_1, \dots, x_n) \in \mathbb{Z}^n : \sum_{i=1}^n x_i \equiv 0 \pmod{2}\} \quad (n \geq 3) \quad (14.1.2)$$

είναι πλέγματα¹²⁹.

Ορισμός 14.1.3. Ονομάζουμε στοιχειώδες παραλληλόγραμμο ενός πλέγματος $L(B)$, όπου B πίνακας διάστασης $k \times n$, 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 ημι-ανοιχτό παραλληλεπίπεδο με κορυφές $\mathbf{O}, \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_1 + \mathbf{b}_2$.

Ορισμός 14.1.4. Ονομάζουμε όγκο (volume) ενός πλέγματος $L(B)$ τον όγκο του στοιχειώδους παραλληλόγραμμου του πλέγματος. Τον συμβολίζουμε $\det(L)$ ή $\text{vol}(L)$.

¹²⁹Τα πλέγματα αυτά ονομάζονται και root lattices, διότι προέρχονται από σύνολα διανυσμάτων που είναι root systems.

Άσκηση 14.1 Να αποδείξετε ότι ο όγκος του πλέγματος (14.1.1), για $n = 2$, ισούται με $\sqrt{3}$. Δηλαδή, $\text{vol}(A_2) = \sqrt{3}$. Επίσης, να αποδείξετε ότι $\text{vol}(D_3) = 2$.

Άσκηση 14.2 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 $V(\mathbb{Z}^2)$.

Άσκηση 14.3 Περιγράψτε γεωμετρικά το $V(\mathcal{L})$, όπου

$$\mathcal{L} = \text{span}_{\mathbb{Z}}(\{\mathbf{b}_1 = (1, 1), \mathbf{b}_2 = (-1, 3)\}).$$

Παρατήρηση 14.1.3. Μερικές φορές ονομάζουμε τον όγκο ενός πλέγματος και co-volume και ισούται με τον όγκο του τόρου (torus) $\text{span}(L)/L$.

Λήμμα 14.1.1. 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 η B' είναι μια 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, $\mathbf{y}B' = \mathbf{z}B'$ οπότε $(\mathbf{y} - \mathbf{z})B' = \mathbf{0}$, επομένως $\mathbf{y} - \mathbf{z} \in \text{Ker}(B'^T)$, όπου $\text{Ker}(B'^T)$ είναι ο αριστερός μηδενοχώρος (cokernel) του B' . Ισχύει

$$\dim[\text{Ker}(B'^T)] = k - \text{rank}(B') = 0.$$

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

$$\text{span}(\mathbf{b}_1, \dots, \mathbf{b}_k) = \text{span}(\mathbf{b}'_1, \dots, \mathbf{b}'_k),$$

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, $\mathbf{x}' = \mathbf{0}$, συνεπώς $\mathbf{x} = \lfloor \mathbf{y} \rfloor B' \in L(B')$. Δηλαδή, $L(B) \subseteq L(B')$. Τέλος, επειδή οι πίνακες B, B' έχουν ίδια τάξη, προκύπτει ότι $L(B) = L(B')$. Δηλαδή, B' είναι μια βάση του $L(B)$. □

Παράδειγμα 14.1.2. 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 14.1.1 Ας είναι B, B' δύο $n \times n$ πίνακες βάσης ενός μέγιστης τάξης πλέγματος L . Τότε υπάρχει πίνακας $U \in GL_n(\mathbb{Z})$ (ο U καλείται unimodular) με $B = UB'$, και αντίστροφα.

Απόδειξη. ($\Rightarrow$) Ας είναι $B = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n]^T$, $B' = [\mathbf{b}'_1, \mathbf{b}'_2, \dots, \mathbf{b}'_n]^T$. Τότε,

$$\mathbf{b}'_i = \sum_{j=1}^n a_{ij} \mathbf{b}_j.$$

Άρα, υπάρχει πίνακας U με ακέραια στοιχεία, $U = [a_{ij}]_{1 \leq i, j \leq n}$, τέτοιο ώστε $B' = UB$. Παρόμοια, υπάρχει πίνακας U' τέτοιο ώστε $B = U'B'$. Άρα, $B' = UU'B'$. Επειδή τα διανύσματα $\mathbf{b}'_1, \mathbf{b}'_2, \dots, \mathbf{b}'_n$ 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, $U \in GL_n(\mathbb{Z})$.

($\Leftarrow$) Έστω $B' = UB$, τότε $\mathbf{x}B \in L$. Αλλά, $\mathbf{x}B = \mathbf{x}U'B' = \mathbf{y}B' \in L'$, όπου L' το πλέγμα που παράγεται από τον πίνακα B' . 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, $L = L'$. $\square$

Η προηγούμενη Πρόταση ισχύει και γενικότερα.

Πρόταση 14.1.2 Ας είναι B, B' δύο $k \times n$ πίνακες βάσης ενός πλέγματος L . Τότε υπάρχει πίνακας $U \in GL_k(\mathbb{Z})$ με $B = UB'$, και αντίστροφα.

Απόδειξη. [1, Κεφ. 17, λήμμα 16.1.6] $\square$

Γενικά αν έχουμε μια βάση ενός πλέγματος, για να κατασκευάσουμε μια άλλη, εφαρμόζουμε στοιχειώδεις πράξεις στις γραμμές του πίνακα βάσης. Π.χ. $\mathbf{R}_i \leftarrow \mathbf{R}_i + \alpha \mathbf{R}_j$, όπου $\mathbf{R}_i$ είναι η i γραμμή.

Άσκηση 14.4 Υπολογίστε τον όγκο του πλέγματος που έχει βάση $\mathbf{b}_1 = (1, 1, 1)$, $\mathbf{b}_2 = (0, -1, 2)$.

Άσκηση 14.5 Αποδείξτε ότι $\dim(\text{rowspace}(B)) = \text{rank}(L(B))$, όταν το $L(B)$ είναι μέγιστης τάξης.

Άσκηση 14.6 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 τον δείκτη ορθογωνιότητας $\delta(L) = \frac{\|\mathbf{b}_1\| \cdot \|\mathbf{b}_2\|}{\text{vol}(L)}$. Κατόπιν, αποδείξτε ότι $\delta(L) = 1$, αν και μόνο αν τα $\mathbf{b}_1, \mathbf{b}_2$ είναι ορθογώνια.

Άσκηση 14.7 (*) Να αποδείξετε ότι $\text{vol}(V(\mathcal{L})) = \text{vol}(\mathcal{L})$ (όπου $V(\mathcal{L})$ το Voronoi cell του $\mathcal{L}$, δείτε την άσκηση 14.2).

Άσκηση 14.8 Δίνονται οι δύο πίνακες,

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ -1 & 1 & 2 & 3 \\ 4 & 3 & 2 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 5 & 8 & 11 & 13 \\ 4 & 6 & 8 & 9 \\ 8 & 9 & 10 & 10 \\ 4 & 6 & 9 & 10 \end{pmatrix}$$

- (i). Ν.α.ο. οι γραμμές τους παράγουν το ίδιο πλέγμα.
- (ii). Ν.α.ο. το διάνυσμα $(43, 69, 95, 110)$ ανήκει στο πλέγμα.
- (iii). Να υπολογίσετε τον όγκο του.
- (iv). Πόσα σημεία του πλέγματος υπάρχουν μέσα στον κύκλο με κέντρο $\mathbf{O}$ και ακτίνα 10; Το ίδιο, αλλά για ένα τετράγωνο με πλευρά 10 μονάδες και κέντρο στην αρχή των αξόνων.

Άσκηση 14.9 Έστω E_8 το πλέγμα με βάση που δίνεται από τις γραμμές του πίνακα:

$$A = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 1/2 & 1/2 & 1/2 & 1/2 & 1/2 & 1/2 & 1/2 & 1/2 \end{pmatrix}$$

(i). Να υπολογίσετε τον όγκο του πλέγματος E_8 .

(ii). Ν.α.ο. $E_8 = \{\mathbf{x} \in \mathbb{Z}^8 \cup (\mathbb{Z} + \frac{1}{2})^8 : \sum_i x_i \equiv 2x_1 \equiv \dots \equiv 2x_8 \pmod{2}\}$.

(iii). Πόσα σημεία του πλέγματος υπάρχουν μέσα στον κύκλο με κέντρο $\mathbf{O}$ και ακτίνα $\sqrt{2}$;

14.2 Διαδικασία Gram-Schmidt

Την προβολή του διανύσματος $\mathbf{u} \in \mathbb{R}^n$ επί του $\mathbf{v} \neq \mathbf{0}$ τη 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 $\lambda \neq 0$. Αλλά, 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 $\{\mathbf{b}_1, \dots, \mathbf{b}_k\} \subset \mathbb{R}^n$ μια διατεταγμένη βάση ενός διανυσματικού υποχώρου του $\mathbb{R}^n$ ($n \geq k$). Με την παρακάτω διαδικασία προκύπτει μια ορθογώνια βάση.

$$\mathbf{b}_1^* = \mathbf{b}_1, \quad \mathbf{b}_2^* = \mathbf{b}_2 - \text{proj}_{\mathbf{b}_1^*} \mathbf{b}_2 = \mathbf{b}_2 - \frac{\mathbf{b}_1 \cdot \mathbf{b}_2}{\mathbf{b}_1 \cdot \mathbf{b}_1} \mathbf{b}_1,$$

$$\mathbf{b}_i^* = \mathbf{b}_i - \sum_{j=1}^{i-1} \text{proj}_{\mathbf{b}_j^*} \mathbf{b}_i, \dots$$

$$\mathbf{b}_k^* = \mathbf{b}_k - \sum_{j=1}^{k-1} \text{proj}_{\mathbf{b}_j^*} \mathbf{b}_k.$$

Θέτουμε

$$\mu_{ij} = \frac{\mathbf{b}_i \cdot \mathbf{b}_j^*}{\mathbf{b}_j^* \cdot \mathbf{b}_j^*} \quad (i > j).$$

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-Schmidt*. 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 14.2.1. : 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  $(\mu_{ij})$ .

 $\mathbf{R}_1 \leftarrow \mathbf{b}_1$ 
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  $\mathcal{B}' = (\mathbf{R}_1, \dots, \mathbf{R}_n), \mu = (\mu_{i,j})_{i,j}$ 

```

Παράδειγμα 14.2.1. 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 $\{\mathbf{b}_1^*, \dots, \mathbf{b}_k^*\}$ είναι οι εξής:

- (i). $\text{span}(\mathbf{b}_1^*, \dots, \mathbf{b}_i^*) = \text{span}(\mathbf{b}_1, \dots, \mathbf{b}_i)$, $1 \leq i \leq k$,
- (ii). $\mathbf{b}_i^* = \mathbf{b}_i - (\alpha_1 \mathbf{b}_1 + \dots + \alpha_{i-1} \mathbf{b}_{i-1})$ ($\alpha_1, \dots, \alpha_{i-1} \in \mathbb{R}$),
- (iii). $\mathbf{b}_i^* \cdot \mathbf{b}_j = 0$, $j < i$,
- (iv). $\|\mathbf{b}_i^*\| = \text{distance}(\mathbf{b}_i, \text{span}(\mathbf{b}_1, \dots, \mathbf{b}_{i-1})^\perp)$, $i \geq 2$.

Η πρώτη προκύπτει, διότι

$$\mathbf{b}_i^* = \mathbf{b}_i - \sum_{j=1}^{i-1} \text{proj}_{\mathbf{b}_j^*} \mathbf{b}_i \quad (14.2.1)$$

επομένως,

$$\mathbf{b}_i = \mathbf{b}_i^* + \sum_{j=1}^{i-1} \text{proj}_{\mathbf{b}_j^*} \mathbf{b}_i$$

άρα, $\mathbf{b}_i \in \text{span}(\mathbf{b}_1^*, \dots, \mathbf{b}_i^*)$, συνεπώς

$$\text{span}(\mathbf{b}_1^*, \dots, \mathbf{b}_i^*) \subset \text{span}(\mathbf{b}_1, \dots, \mathbf{b}_i).$$

Αντίστροφα, εξ ορισμού της κατασκευής Gram-Schmidt έχουμε

$$\text{span}(\mathbf{b}_1^*, \dots, \mathbf{b}_i^*) \subset \text{span}(\mathbf{b}_1, \dots, \mathbf{b}_i).$$

Η δεύτερη επίσης προκύπτει άμεσα, διότι

$$\mathbf{b}_i^* - \mathbf{b}_i = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{ij} \mathbf{b}_j^* - \mathbf{b}_i =$$

$$-\sum_{j=1}^{i-1} \mu_{ij} \mathbf{b}_j^* \in \text{span}(\mathbf{b}_1^*, \dots, \mathbf{b}_{i-1}^*) = \text{span}(\mathbf{b}_1, \dots, \mathbf{b}_{i-1}).$$

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,

$$\mathbf{b}_\rho^* \cdot \mathbf{b}_j = 0, \quad 1 \leq j < \rho \leq i - 1.$$

Θα αποδείξουμε ότι ισχύει για i . Για $j < i$ έχουμε

$$\mathbf{b}_i^* \cdot \mathbf{b}_j = \dots = 0.$$

Αυτή η ιδιότητα ισοδύναμα γράφεται

$$\mathbf{b}_i^* \in \text{span}(\mathbf{b}_1, \dots, \mathbf{b}_{i-1})^\perp.$$

Η τέταρτη ιδιότητα εξ ορισμού ισχύει για $i = 2$. Γενικά, αρκεί να παρατηρήσουμε ότι, αν $\mathcal{A}_i = \text{span}(\mathbf{b}_1, \dots, \mathbf{b}_{i-1})^\perp$, 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 14.2.2. 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 $\{\mathbf{b}_1^*, \dots, \mathbf{b}_k^*\}$. Όμως,

$$\pi_i(\mathbf{x}) = \sum_{j=i}^k \text{proj}_{\mathbf{b}_j^*}(\mathbf{x}),$$

είναι η ανάλυση του $\mathbf{x}$ στις ορθογώνιες συνιστώσες $\{\mathbf{b}_i^*, \dots, \mathbf{b}_k^*\}$. Ισχύει $\pi_i(\mathbf{b}_i) = \mathbf{b}_i^*$. Εξ ορισμού,

$$\pi_i(\mathbf{x}) \in \text{span}(\mathbf{b}_i^*, \dots, \mathbf{b}_k^*).$$

Ειδικότερα, από την ιδιότητα (iii)

$$\mathbf{b}_i^* \cdot \mathbf{b}_j = 0 \quad (j < i),$$

άρα,

$$\mathbf{b}_\ell^* \in \mathcal{A}_i = \text{span}(\mathbf{b}_1, \dots, \mathbf{b}_{i-1})^\perp$$

για $\ell = i, i + 1, \dots, k$. Επίσης, $\dim \mathcal{A}_i = k - i + 1$, επομένως

$$\mathcal{A}_i = \text{span}(\mathbf{b}_i^*, \dots, \mathbf{b}_k^*).$$

Δηλαδή, η προβολή $\pi_i(\mathbf{x}) \in \mathcal{A}_i$.

Ορισμός 14.2.1. Ονομάζουμε ορθογώνια προβολή στον υποχώρο (subspace) $\text{span}(\mathbf{b}_i^*, \dots, \mathbf{b}_k^*) \subset \mathbb{R}^n$, την απεικόνιση

$$\pi_i : \mathbb{R}^n \rightarrow \text{span}(\mathbf{b}_i^*, \dots, \mathbf{b}_k^*) \quad (1 \leq i \leq k),$$

$$\pi_i(\mathbf{x}) = \sum_{j=i}^k \text{proj}_{\mathbf{b}_j^*}(\mathbf{x}) = \sum_{j=i}^k \frac{\mathbf{x} \cdot \mathbf{b}_j^*}{\mathbf{b}_j^* \cdot \mathbf{b}_j^*} \mathbf{b}_j^*$$

και

$$\pi(\mathbf{x}) = \pi_1(\mathbf{x}).$$

Παρατήρηση 14.2.1. 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 14.2.3. 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 υπολογίσουμε μια βάση του $\pi(\mathcal{L})$. Μετά 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

$$\pi : \mathcal{L} \rightarrow \text{span}(\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3)^\perp.$$

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 $\pi(\mathcal{L})$ είναι πλέγμα, διότι, 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 $\pi(\mathcal{L})$. Δηλαδή, $\pi(\mathcal{L}) = \text{span}_{\mathbb{Z}}\{\pi(\mathbf{b}_1), \pi(\mathbf{b}_2), \pi(\mathbf{b}_3)\}$. 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, δηλαδή, είναι $\subset \mathbb{Q}^3$ και $\not\subset \mathbb{Z}^3$. Καταλήγουμε ότι

$$\pi(\mathcal{L}) = \text{span}_{\mathbb{Z}}\{(1, 0, 0), (1, 1, 1), (1, 1/2, 5/2)\}.$$

Τέλος, ο όγκος του $\pi(\mathcal{L})$ είναι $\text{vol}(\pi(\mathcal{L})) = 2$.

Λήμμα 14.2.1. 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 $\text{vol}(\pi_i(\mathcal{L})) = \prod_{j=i}^n \|\mathbf{b}_j^*\|$.

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 14.2.1)

$$\mathbf{x} - \mathbf{y} = \pi_i(\mathbf{a}) - \pi_i(\mathbf{b}) = \pi_i(\mathbf{a} - \mathbf{b}) \in \pi_i(\mathcal{L}).$$

Παρατηρούμε ότι τα διανύσματα $\pi_i(\mathbf{b}_j)$ για $j = 1, 2, \dots, n$ 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 $\{\pi_i(\mathbf{b}_i), \pi_i(\mathbf{b}_{i+1}), \dots, \pi_i(\mathbf{b}_n)\}$. Αυτό το σύνολο είναι και γραμμικά ανεξάρτητο. Τέλος, επειδή

$$\text{span}\{\pi_i(\mathbf{b}_i), \pi_i(\mathbf{b}_{i+1}), \dots, \pi_i(\mathbf{b}_n)\} = \text{span}\{\mathbf{b}_i^*, \mathbf{b}_{i+1}^*, \dots, \mathbf{b}_n^*\}$$

και το σύνολο $\{\mathbf{b}_i^*, \mathbf{b}_{i+1}^*, \dots, \mathbf{b}_n^*\}$ είναι γραμμικά ανεξάρτητο, προκύπτει ότι ο όγκος είναι

$$\text{vol}(\pi_i(\mathcal{L})) = \prod_{j=i}^n \|\mathbf{b}_j^*\|.$$

□

Πόρισμα 14.2.1. $\text{vol}(\mathcal{L}) = \text{vol}(\pi(\mathcal{L}))$ ($\pi(\mathcal{L}) = \pi_1(\mathcal{L})$).

Άσκηση 14.10 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 14.11 Ν.α.ό.

$$\|\pi_{i-1}(\mathbf{b}_i)\|^2 = \frac{(\mathbf{b}_i \cdot \mathbf{b}_{i-1}^*)^2}{\|\mathbf{b}_{i-1}^*\|^2} + \|\mathbf{b}_i^*\|^2 \quad (i \geq 2).$$

Άσκηση 14.12 Έστω $\mathbf{x}$ τυχαίο διάνυσμα του $\mathbb{R}^n$ και $\{\mathbf{b}_1, \dots, \mathbf{b}_n\}$, μια βάση του $\mathbb{R}^n$.

(i). Ν.α.ό.

$$\|\pi_{i-1}(\mathbf{x})\|^2 = \|\pi_i(\mathbf{x})\|^2 + \frac{(\mathbf{x} \cdot \mathbf{b}_{i-1}^*)^2}{\|\mathbf{b}_{i-1}^*\|^2} \geq \|\pi_i(\mathbf{x})\|^2 \quad (i = 2, 3, \dots, n).$$

(ii). Τι είδους μονοτονία έχει η ακολουθία $\{\|\pi_i(\mathbf{x})\|\}$; ($i = 1, 2, \dots, n$).

Παρατήρηση 14.2.2. (i). Έστω

$$\hat{B} = \begin{bmatrix} | & | & \dots & | \\ \hat{\mathbf{b}}_1^T & \hat{\mathbf{b}}_2^T & \dots & \hat{\mathbf{b}}_n^T \\ | & | & \dots & | \end{bmatrix},$$

όπου $\hat{\mathbf{b}}_j = \frac{\mathbf{b}_j^*}{\|\mathbf{b}_j^*\|}$. Η διαδικασία κατασκευής των $\hat{\mathbf{b}}_j$ ονομάζεται *ορθοκανονικοποίηση κατά Gram-Schmidt*.

(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 $P: L \rightarrow \mathbb{R}^k$, $P(\mathbf{v}) = \mathbf{v}\hat{B}$ 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,

$$\begin{bmatrix} -\mathbf{b}_1^* \\ -\mathbf{b}_2^* \\ \vdots \\ -\mathbf{b}_k^* \end{bmatrix} \cdot \begin{bmatrix} | & | & \dots & | \\ \frac{\mathbf{b}_1^*}{\|\mathbf{b}_1^*\|} & \frac{\mathbf{b}_2^*}{\|\mathbf{b}_2^*\|} & \dots & \frac{\mathbf{b}_k^*}{\|\mathbf{b}_k^*\|} \\ | & | & \dots & | \end{bmatrix} = \begin{bmatrix} \mathbf{b}_i^* \cdot \mathbf{b}_j^* \\ \|\mathbf{b}_j^*\| \end{bmatrix}_{1 \leq i \leq j \leq k}$$

Από την τελευταία παρατήρηση και το γεγονός ότι $\text{vol}(\mathcal{P}(B)) = \text{vol}(\mathcal{P}(B^*))$, προκύπτει το επόμενο.

Λήμμα 14.2.2. Ο όγκος του πλέγματος L ισούται με τον θετικό πραγματικό αριθμό

$$\det(B^* \hat{B}) = \prod_{i=1}^k \|\mathbf{b}_i^*\|.$$

Παρατήρηση 14.2.3. Ο όγκος του πλέγματος L είναι ανεξάρτητος του πίνακα βάσης B .

Παράδειγμα 14.2.4. 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, $\text{vol}(L) = \det[B^* \hat{B}] = \sqrt{21}$.

Παρατήρηση 14.2.4. Αν $\{\mathbf{b}_1, \dots, \mathbf{b}_n\}$ γραμμικά ανεξάρτητα και $\{\mathbf{b}_1^*, \dots, \mathbf{b}_n^*\}$ η Gram-Schmidt βάση, τότε $\|\mathbf{b}_i^*\| \leq \|\mathbf{b}_i\|$ και $\mathbf{b}_i \cdot \mathbf{b}_i^* = \mathbf{b}_i^* \cdot \mathbf{b}_i^*$ ($i \geq 1$).

Παρατήρηση 14.2.5. Έστω $\mathcal{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_n\}$ με $\mathbf{b}_j \in \mathbb{R}^m$ μία βάση ενός πλέγματος L και B ο πίνακας βάσης. Τότε ισχύει,

$$\text{vol}(L)^2 = \det(BB^T).$$

Ο BB^T είναι ο Gram matrix των διανυσμάτων $\mathcal{B}$.

Άσκηση 14.13 Να αποδείξετε την παρατήρηση 14.2.3.

Άσκηση 14.14 Να αποδείξετε ότι $\text{vol}(A_n) = \sqrt{n+1}$ όπου $A_n \subset \mathbb{Z}^{n+1}$ το οποίο ορίζεται ως

$$A_n = \{(x_0, x_1, \dots, x_n) \in \mathbb{Z}^{n+1} : \sum_{i=0}^n x_i = 0\}.$$

Άσκηση 14.15 Να αποδείξετε ότι $\text{vol}(D_n) = 2$, όπου $D_n \subset \mathbb{Z}^n$ που ορίζεται

$$D_n = \{(x_1, \dots, x_n) \in \mathbb{Z}^n : \sum_{i=1}^n x_i = 0 \pmod{2}\}.$$

Άσκηση 14.16 (*) Έστω πίνακας A διάστασης $m \times n$ και διάνυσμα $\mathbf{d} \in \mathbb{Z}^m$. Δίνεται ο πίνακας,

$$B = \left[\begin{array}{c|c|c} I_n & \mathbf{0}_{n \times 1} & N_2 A^T \\ \hline \mathbf{0}_{1 \times n} & N_1 & -N_2 \mathbf{d} \end{array} \right],$$

όπου $N_1, N_2 \in \mathbb{Z}$. Αν L το πλέγμα που παράγεται από τις γραμμές του B , να υπολογίσετε τον όγκο του L .
Υπόδειξη. Δείτε παρατήρηση 14.2.5. Αν δυσκολεύεστε να βρείτε τον τύπο, μπορείτε να ξεκινήσετε από την περίπτωση $m = n = 2, N_1 = N_2 = 1$,

$$\text{vol}(L)^2 = \text{vol}(L(A))^2 + (d_1 \mathbf{r}_2 - d_2 \mathbf{r}_1)^2 + \sum_{i,j} a_{ij}^2 + \|\mathbf{d}\|^2 + 1.$$

14.3 Θεωρήματα του Minkowski

Η εύρεση ενός διανύσματος με το μικρότερο μήκος σε ένα πλέγμα L ονομάζεται *πρόβλημα του μικρότερου διανύσματος* ή **SVP** : **Shortest Vector Problem** και είναι δύσκολο πρόβλημα. Ειδικότερα, ο Ajtai απέδειξε ότι είναι NP-hard με εφαρμογή τυχαίων αναγωγών. Επίσης, ο Miccianchio απέδειξε ότι το SVP_γ είναι NP-hard για $\gamma = \sqrt{2}$ (με εφαρμογή τυχαίων αναγωγών). 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 14.3.1 Υπάρχει διάνυσμα $\mathbf{z} \in L$ τέτοιο, ώστε $\|\mathbf{z}\| = \lambda_1(L)$.

Απόδειξη. Από τη χαρακτηριστική ιδιότητα του $\inf$ υπάρχει ακολουθία $(\mathbf{z}_n)_n \in L$ τέτοια, ώστε $\|\mathbf{z}_n\| \rightarrow \lambda_1$. 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 υποακολουθία $\mathbf{z}_{i_j}$ της ακολουθίας $\mathbf{z}_n$ που συγκλίνει, έστω στο διάνυσμα $\mathbf{z}$. Τότε, για κάθε $\varepsilon > 0$ υπάρχει $j_0 > 0$ τέτοιο, ώστε $\|\mathbf{z}_{i_j} - \mathbf{z}\| < \varepsilon/\delta$ για όλα τα $j > j_0$. Επομένως, για $j, v > j_0$ έχουμε

$$\|\mathbf{z}_{i_j} - \mathbf{z}_{i_v}\| = \|(\mathbf{z}_{i_j} - \mathbf{z}) + (\mathbf{z} - \mathbf{z}_{i_v})\| \leq 2\varepsilon/\delta.$$

Διαλέγοντας, αντί ε το $\varepsilon\delta/2$ προκύπτει $\|\mathbf{z}_{i_j} - \mathbf{z}_{i_v}\| < \varepsilon$. Εφόσον τα πλέγματα είναι διακριτές ομάδες, προκύπτει $\mathbf{z}_{i_j} = \mathbf{z}_{i_v} = \mathbf{z} \in L$. Άρα, το ζητούμενο διάνυσμα είναι το $\mathbf{z}$. $\square$

Παρατήρηση 14.3.1. Ο αριθμός $\lambda_1(L)$ δεν εξαρτάται από την επιλογή βάσης του πλέγματος. Επίσης, ονομάζεται πρώτο διαδοχικό ελάχιστο (first successive minima).

Θεώρημα 14.3.2 (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 $\mathbf{v} \in L - \{\mathbf{0}\}$ με $\|\mathbf{v}\| \leq 2R$.

Απόδειξη. Έστω $f : \mathbb{R}^n \rightarrow \mathbb{R}^n/L$,

$$f(\mathbf{x}) = \mathbf{x} \pmod{L}.$$

Η συνάρτηση αυτή είναι επί. Θεωρούμε τον περιορισμό

$$\text{rest}_{\overline{B}_n(R)} f : \overline{B}_n(R) \rightarrow \mathbb{R}^n/L,$$

όπου $\overline{B}_n(R)$ η σφαίρα ακτίνας R του $\mathbb{R}^n$ με κέντρο την αρχή των αξόνων¹³⁰. Ο περιορισμός δεν είναι 1-1, λόγω της υπόθεσης με τον όγκο. Πράγματι, αν ήταν 1-1, τότε

$$\text{vol}(\overline{B}_n(R)) = V_n R^n \leq \text{vol}(\mathbb{R}^n/L) = \det L,$$

το οποίο αντίκειται στην υπόθεση του θεωρήματος. Επομένως, υπάρχουν διανύσματα $\mathbf{x}, \mathbf{y}$ του $\overline{B}_n(R)$ 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 $\|\mathbf{v}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\| \leq 2R$. $\square$

¹³⁰ταυτίζουμε τους όρους *σφαίρα* και *μπάλα*

Παρατήρηση 14.3.2. Ο όγκος της n -διάστατης μοναδιαίας σφαίρας,

$$\overline{B}_n = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| \leq 1\}$$

είναι

$$V_n = \frac{\pi^{n/2}}{\Gamma(\frac{n}{2} + 1)},$$

όπου

$$\Gamma(z) = \int_0^\infty x^{z-1} e^{-x} dx,$$

για $\operatorname{Re}(z) > 0$. Ισχύει

$$\Gamma(n) = (n-1)! \quad (n \in \mathbb{Z}_{\geq 1})^{131}. \quad (14.3.1)$$

Μια προσέγγιση του V_n είναι:

$$V_n \approx \left(\frac{2\pi e}{n}\right)^{n/2} \frac{1}{\sqrt{\pi n}}.$$

Όμως, της n -διάστατης σφαίρας ακτίνας R ,

$$V_n(R) = \operatorname{vol}(\overline{B}_n(R)) = V_n R^n.$$

Επίσης, η ακτίνα $R_n(V)$ μιας σφαίρας όγκου V , ισούται με

$$R_n(V) = \frac{(\Gamma(\frac{n}{2} + 1)V)^{1/n}}{\sqrt{\pi}}.$$

Το θεώρημα 14.3.2 του Minkowski μας λέει ότι μια σφαίρα με ακτίνα τουλάχιστον

$$R_0 = 2 \left(\frac{\det L}{V_n}\right)^{1/n} = \frac{2(\det L)^{1/n} \Gamma(\frac{n}{2} + 1)^{1/n}}{\sqrt{\pi}} \quad (14.3.2)$$

περιέχει ένα μη μηδενικό σημείο του L .

Δίνουμε τον παρακάτω ορισμό.

Ορισμός 14.3.1. Συμβολίζουμε με GH : **Gaussian Heuristic** τον θετικό πραγματικό αριθμό

$$GH(L) = \left(\frac{\det L}{V_n}\right)^{1/n} = \frac{(\det L)^{1/n} \Gamma(\frac{n}{2} + 1)^{1/n}}{\sqrt{\pi}} \approx \sqrt{\frac{n}{2\pi e}} (\det(L))^{1/n},$$

όπου $n = \operatorname{rank}(L)$.

Πόρισμα 14.3.1. $\lambda_1(L) < 2GH(L)$.

Απόδειξη. Για κάθε ακτίνα R με

$$GH(L) < R < GH(L) + \varepsilon,$$

για κάθε $\varepsilon > 0$, η προϋπόθεση του θεωρήματος 14.3.2 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. $\square$

Δίνουμε τον ορισμό μιας πολύ βασικής ευρετικής υπόθεσης που ισχύει σε τυχαία (ακέραια) πλέγματα.

¹³¹ Αυτή η ιδιότητα μας επιτρέπει να επεκτείνουμε το σύμβολο $!$ σε πραγματικούς αριθμούς. Π.χ. $(\frac{1}{2})! = \Gamma(1/2)$.

Ορισμός 14.3.2. (Ευρετική του Gauss). Σε τυχαίο ακέραιο πλέγμα L τάξης n και διάστασης m (δηλαδή, $L \subset \mathbb{Z}^m$) υποθέτουμε ότι ισχύει

$$\lambda_1(L) \approx GH(L).$$

Αυτή η ευρετική δεν ισχύει για όλα τα ακέραια πλέγματα¹³². 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 (14.3.2).

Στη βιβλιογραφία συνήθως η ευρετική του Gauss δίνεται ως εξής:

Gaussian Heuristic. Αν έχουμε ένα πλέγμα L και ένα σύνολο S , τότε το πλήθος των σημείων $S \cap L$ είναι προσεγγιστικά ίσο με $\text{vol}(S)/\text{vol}(L)$.

Παρατήρηση 14.3.3. Αν ισχύει η ευρετική του Gauss σε ένα πλέγμα L , τότε η ελάχιστη ακτίνα R , ώστε

$$|L \cap \overline{B}_n(R)| = 1 \approx \frac{\text{vol}(\overline{B}_n(R))}{\det L} = \frac{R^n V_n}{\det L}.$$

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 $0 < \alpha < n^{-\varepsilon}$ για κάποιο $\varepsilon > 0$, τότε, καθώς το n αυξάνεται, αποδεικνύεται ότι η ευρετική του Gauss δεν ισχύει.

Άσκηση 14.17 Να αποδείξετε την ισότητα 14.3.1.

Άσκηση 14.18 Να αποδείξετε ότι

$$\left(\frac{1}{2}\right)! = \frac{\sqrt{\pi}}{2}.$$

Άσκηση 14.19 (Complement formula). Να αποδείξετε ότι

$$\Gamma(s)\Gamma(1-s) = \frac{\pi}{\sin(\pi s)}.$$

Υπόδειξη.

$$J = \Gamma(s)\Gamma(1-s) = \int_{y=0}^{\infty} e^{-y} \left[\int_{x=0}^{\infty} e^{-x} \left(\frac{x}{y}\right)^{s-1} \frac{dx}{y} \right] dy.$$

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 τη 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).

$$HF(L, B) = \frac{\|\mathbf{b}_1\|}{(\det L)^{1/n}}.$$

¹³²J. E. Mazo and A. M. Odlyzko, Lattice points in high dimensional spheres (1990), *Monats. Math.* vol. 17

Ο Hermite factor εξαρτάται μόνο από τη βάση Gram-Schmidt, 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 απέδειξε 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 14.3.3. (Σταθερά του Hermite). Ορίζουμε τη σταθερά του Hermite διάστασης n

$$\gamma_n = \sup_{L \subset \mathbb{Z}^n, \text{rank}(L)=n} \left(\frac{\lambda_1(L)}{(\det L)^{1/n}} \right)^2.$$

Λήμμα 14.3.1. Ισχύει,

$$\sqrt{\gamma_n} \leq \frac{2}{V_n^{1/n}}.$$

Απόδειξη. Από το πόρισμα 14.3.1, 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 14.3.4. Μέχρι στιγμής έχουμε τρεις βασικές ποσότητες ανεξάρτητες της βάσης του πλέγματος, την ποσότητα $\lambda_1(L)$, τον όγκο $\det(L)$ και τη σταθερά Hermite $\gamma_n(L)$. Η τελευταία ποσότητα είναι και ανεξάρτητη του πλέγματος, δηλαδή, $\gamma_n(L) = \gamma_n$.

Παρατήρηση 14.3.5. Ο Lagrange απέδειξε ότι $\gamma_2 = \sqrt{\frac{4}{3}}$. Επίσης, οι Korkine-Zolotarev απέδειξαν ότι $\gamma_4 = \sqrt[4]{4}$, $\gamma_5 = \sqrt[5]{8}$. Όμως, ο Hofreiter (καθώς και ο Blichfeld) απέδειξαν ότι $\gamma_6 = \sqrt[6]{\frac{64}{3}}$. Ο Blichfeld απέδειξε $\gamma_7 = \sqrt[7]{64}$, $\gamma_8 = 2$. Η σταθερά του Hermite ορίστηκε αρχικά από τον Hermite για θετικά ορισμένες τετραγωνικές μορφές, δηλαδή, πολυώνυμα της μορφής

$$q(x_1, \dots, x_n) = \sum_{1 \leq i, j \leq n} a_{ij} x_i x_j \quad (a_{ij} = a_{ji} \in \mathbb{Z}),$$

με $q(\mathbf{x}) \geq 0$ για κάθε $\mathbf{x} \in \mathbb{Z}^n$. Θέτουμε $\Delta_q = \det([a_{ij}])$. Ο Hermite απέδειξε 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 τη διάσταση n . Έτσι ο Hermite όρισε

$$\gamma_n = \sup_q \frac{\|q\|}{\Delta_q^{1/n}}.$$

Πρόταση 14.3.1 (Ανισότητα του Hermite).

$$\gamma_n < \gamma_2^{(n-1)/2} = \left(\frac{4}{3}\right)^{(n-1)/2} (n > 2).$$

Απόδειξη. [2, Θεώρημα 7.5] □

Μια αλγοριθμική μορφή της προηγούμενης πρότασης είναι ο αλγόριθμος LLL.

Παρατήρηση 14.3.6. Υπάρχουν και γραμμικά (άνω) φράγματα για τη 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 14.3.2 (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

$$\|\mathbf{b}_1\| = \delta^n (\det L)^{1/n}.$$

Δηλαδή, $\delta^n = HF(L, B)$. 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.

14.3.1 GSA : Geometric Series Assumption

Μια άλλη βασική υπόθεση που κάνουμε πολύ συχνά στα πλέγματα είναι η GSA. Πρώτη φορά διατυπώθηκε από τους Schnor-Euchner.

Ορισμός 14.3.4. Λέμε ότι ισχύει η GSA σε ένα πλέγμα L για τη 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 14.3.7. Δηλαδή, η ακολουθία

$$\left(\|\mathbf{b}_1^*\|, \|\mathbf{b}_2^*\|, \dots, \|\mathbf{b}_n^*\| \right)$$

είναι μια γεωμετρική πρόοδος με λόγο $r < 1$.

Ο συντελεστής GSA και ο RHF συνδέονται μεταξύ τους.

Λήμμα 14.3.2. 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 14.3.2. 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 εφαρμογή του πορίσματος είναι στους αλγορίθμους απαρίθμησης για να κάνουμε πρόβλεψη των $\|\mathbf{b}_i^*\|$ ξεκινώντας από μια βάση με Root Hermite Factor δ .

Άσκηση 14.20 Να σχεδιάσετε ένα γράφημα που στον οριζόντιο άξονα έχει τη διάσταση n της σφαίρας και στον κατακόρυφο τον όγκο της. Σχεδιάστε μέχρι το $n = 10$.

Άσκηση 14.21 Σε τυχαία πλέγματα διάστασης 3 εξετάστε την ευρετική του Gauss, όπως διατυπώνεται στην παρατήρηση 14.3.3.

14.4 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 14.4.1. Μια βάση $\mathbf{b}_1, \mathbf{b}_2$ του L ονομάζεται *ανηγμένη κατά Gauss-Lagrange*, αν και μόνο αν

$$\|\mathbf{b}_1\| \leq \|\mathbf{b}_2\| \leq \|\mathbf{b}_2 + q\mathbf{b}_1\| \quad (14.4.1)$$

για κάθε $q \in \mathbb{Z}$.

Παρατήρηση 14.4.1. Ισοδύναμα ο ορισμός ισχύει, αν στο δεύτερο μέλος της ανισότητας έχουμε

$$\|\mathbf{b}_2 - q\mathbf{b}_1\|.$$

Θεώρημα 14.4.1 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 $\mathbf{b}_1$ τέτοιο, ώστε $\|\mathbf{b}_1\| \leq \|\mathbf{b}'\| < \|\mathbf{b}_2\|$, 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 τέτοιο ώστε $m_2 = m_1a + b$, $0 \leq b < |m_2|$. 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 την τριγωνική ανισότητα,

$$\begin{aligned} \|\mathbf{v}\| &\geq |m_2| \|\mathbf{b}_2 + a\mathbf{b}_1\| - b \|\mathbf{b}_1\| = \\ &-b \|\mathbf{b}_1\| + |m_2| \|\mathbf{b}_2 + a\mathbf{b}_1\| + b \|\mathbf{b}_2 + a\mathbf{b}_1\| - b \|\mathbf{b}_2 + a\mathbf{b}_1\| = \\ &(|m_2| - b) \|\mathbf{b}_2 + a\mathbf{b}_1\| + b(\|\mathbf{b}_2 + a\mathbf{b}_1\| - \|\mathbf{b}_1\|). \end{aligned}$$

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 14.4.1. Αν για τη 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 τη 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 14.4.2. Η γωνία μεταξύ των δύο διανυσμάτων $\mathbf{b}_1$ και $\mathbf{b}_2$ του προηγούμενου λήμματος ανήκει στο διάστημα $(\pi/3, 2\pi/3)$. Δηλαδή, διαφέρουν το πολύ κατά 30° , για να γίνουν κάθετα.

Λήμμα 14.4.2. 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 αυστηρή 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 διότι $\mathbf{b}_1 \neq \mathbf{0}$ ως διάνυσμα βάσης). Άρα, ισχύει για κάθε $q < -1$. Το ζητούμενο έπεται. $\square$

Πόρισμα 14.4.1. 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

$$\|\mathbf{b}_1\| \leq \|\mathbf{b}_2\| \leq \|\mathbf{b}_1 + \mathbf{b}_2\|, \|\mathbf{b}_1 - \mathbf{b}_2\|. \quad (14.4.2)$$

Απόδειξη. ($\Rightarrow$) Για $q = \pm 1$ στον ορισμό 14.4.1, 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. $\square$

Λήμμα 14.4.3. Ας είναι $\{\mathbf{b}_1, \mathbf{b}_2\}$ μια βάση του πλέγματος $L = L(\mathbf{b}_1, \mathbf{b}_2)$. 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 $\mu = \frac{\mathbf{b}_1 \cdot \mathbf{b}_2}{B_1}$. $\square$

Παρατήρηση 14.4.3. (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 τη συνάρτηση που είναι πλησιέστερα σε έναν ακέραιο είναι

$$\lfloor x \rfloor = \lfloor x + \frac{1}{2} \rfloor, \lceil x \rceil = \lceil x - \frac{1}{2} \rceil$$

Έτσι, $\lfloor 0.5 \rfloor = 1$, ενώ $\lceil 0.5 \rceil = 0$.

Λήμμα 14.4.4. 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

$$\mu_0 = \left\lfloor \frac{\mathbf{b}_1 \cdot \mathbf{b}_2}{\|\mathbf{b}_1\|^2} \right\rfloor.$$

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 κοντά 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 14.4.1 (ii) προκύπτει $\mu_0 = 0$. $\square$

Αλγόριθμος 14.4.1. : Αλγόριθμος των Gauss-Lagrange

Είσοδος. $\mathbf{b}_1, \mathbf{b}_2$ γραμμικά ανεξάρτητα

Έξοδος. Μια ανηγμένη κατά Gauss-Lagrange βάση

```

1   $B_1 \leftarrow \|\mathbf{b}_1\|^2$ 
2   $\mu \leftarrow \mathbf{b}_1 \cdot \mathbf{b}_2 / B_1$ 
3   $\mathbf{b}_2 \leftarrow \mathbf{b}_2 - \lceil \mu \rceil \mathbf{b}_1$ 
4   $B_2 \leftarrow \|\mathbf{b}_2\|^2$ 
5  while  $\|\mathbf{b}_2\| < \|\mathbf{b}_1\|$  do
6       $\text{swap}(\mathbf{b}_1, \mathbf{b}_2)$ 
7       $B_1 \leftarrow B_2$ 
8       $\mu \leftarrow \mathbf{b}_1 \cdot \mathbf{b}_2 / B_1$ 
9       $\mathbf{b}_2 \leftarrow \mathbf{b}_2 - \lceil \mu \rceil \mathbf{b}_1$ 
10      $B_2 \leftarrow \|\mathbf{b}_2\|^2$ 
11 end
12 return  $\mathbf{b}_1, \mathbf{b}_2$ 
```

Ορθότητα του αλγορίθμου

Αν το loop δεν εκτελεστεί, τότε καταλήγω σε δύο διανύσματα $\mathbf{b}_1$ και $\mathbf{b}_2 = \mathbf{b}_2 - \lceil \mu \rceil \mathbf{b}_1$. Επομένως, από το προηγούμενο λήμμα 14.4.4, 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 νέα διανύσματα θα είναι αυστηρά μικρότερα. Αλλά μέσα στον

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 14.4.2 Έστω $\|\mathbf{b}_i\| \leq B$ ($i = 1, 2$). Μετά από $O((\log_2(B))^3)$ bit-πράξεις ο αλγόριθμος θα τερματίσει.

Απόδειξη. [1, Θεώρημα 17.1.10] □

14.5 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 Lenstra, Hendrik Lenstra και László Lovász, 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 $n = \text{rank}(L)$.

Δίνουμε τον ορισμό της LLL-ανηγμένης βάσης.

Ορισμός 14.5.1. Έστω $\delta \in (0, 1/4)$. Μια βάση $B = \{\mathbf{b}_1, \dots, \mathbf{b}_n\}$ ονομάζεται δ -LLL-ανηγμένη (δ -LLL-reduced) αν:

- (i). $|\mu_{ij}| \leq 1/2$ για όλα $1 \leq i < j \leq n$.
- (ii). $\|\mathbf{b}_{i+1}^*\|^2 \geq (\delta - \mu_{i+1,i}^2) \|\mathbf{b}_i^*\|^2$, για όλα τα i με $1 \leq i \leq n-1$ (ανισότητα του Lovász).

Η συνθήκη του Lovász, μοιάζει αρκετά πολύπλοκη. Για παράδειγμα, ένα πράγμα που μας λέει είναι ότι

$$h \text{ probol του } \mathbf{b}_{i+1} \text{ επ του } \text{span}(\mathbf{b}_1, \dots, \mathbf{b}_{i-1})^\perp \geq \sqrt{\delta} \times h \text{ probol του } \mathbf{b}_i \text{ επ του } \text{span}(\mathbf{b}_1, \dots, \mathbf{b}_{i-1})^\perp$$

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) ο LLL λύνει το SVP. Ο αλγόριθμος για τη 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 $< \text{Poly}(n)\lambda_1(L)$, όπου $\text{Poly}(n)$ ένα πολυώνυμο του n .

Αλγόριθμος 14.5.1. : LLL Pseudocode

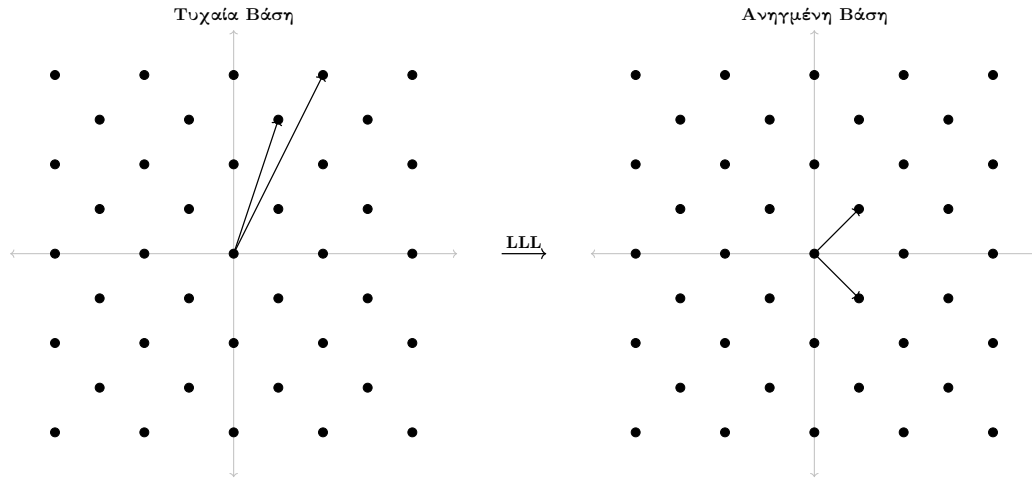
Είσοδος. Μια διατεταγμένη βάση $\mathcal{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_n\} \subset \mathbb{Z}^m$ του πλέγματος $\mathcal{L}(\mathcal{B})$ και έναν πραγματικό αριθμό $\delta \in (1/4, 1)$. Χρησιμοποιούμε τη συνάρτηση $gso = GSO[\mathcal{B}]$ (αλγόριθμος 14.2.1) η οποία επιστρέφει μια ορθογώνια βάση $gso[0]$ και τον Gram-Schmidt matrix $gso[1]$

Έξοδος. Μια LLL-ανηγμένη βάση του $\mathcal{L}$

```

--Initialization:
1   $i = 2$ 
2   $gso \leftarrow GSO(\mathcal{B})$ 
3   $(\mathbf{b}_i)_i \leftarrow gso[1]$ 
4   $(\mu_{i,j})_{i,j} \leftarrow gso[2]$ 
-- Size Reduction Step:
5  while  $i \leq n$  do
6    for  $j = i - 1$  to 1 do
7       $c_{i,j} \leftarrow \lfloor \mu_{i,j} \rfloor$        $\# \lfloor x \rfloor = \lfloor x + 0.5 \rfloor$ 
8       $\mathbf{b}_i \leftarrow \mathbf{b}_i - c_{i,j} \mathbf{b}_j$ 
9      update  $\mathcal{B}$  and  $gso \leftarrow GSO(\mathcal{B})$ 
10      $(\mathbf{b}_i)_i \leftarrow gso[1]$ 
11      $(\mu_{i,j})_{i,j} \leftarrow gso[2]$ 
    end
-- Swap step:
12  if  $\delta \|\mathbf{b}_i^*\|^2 > \|\mu_{i+1,i} \mathbf{b}_i^* + \mathbf{b}_{i+1}^*\|^2$  then
13     $\mathbf{b}_i \leftrightarrow \mathbf{b}_{i+1}$ 
14     $i = \max(2, i - 1)$ 
15    update  $\mathcal{B}$  and  $gso \leftarrow GSO(\mathcal{B})$ 
16     $(\mathbf{b}_i)_i \leftarrow gso[1]$ 
17     $(\mu_{i,j})_{i,j} \leftarrow gso[2]$ 
  else
18     $i \leftarrow i + 1$ 
  end
end
19 return  $\mathcal{B}$ 

```



Το βασικό θεώρημα που αποδείχτηκε από τους Lenstra, Lenstra και Lovász είναι το παρακάτω.

Θεώρημα 14.5.1 Έστω $L(\mathcal{B})$ με $\mathcal{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_n\}$, $\mathbf{b}_i \in \mathbb{Z}^m$, $\Lambda = \max_i (\|\mathbf{b}_i\|^2)$ 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 $O(n^5 m (\log_2 \Lambda)^2)$.

Ένα βασικό αποτέλεσμα του προηγούμενου αλγορίθμου είναι το εξής:

Θεώρημα 14.5.2 Έστω $L(\mathcal{B})$ με $\mathcal{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_n\}$, $\mathbf{b}_i \in \mathbb{Z}^m$ και $\delta = 3/4$. Τότε, ο προηγούμενος

αλγόριθμος υπολογίζει ένα διάνυσμα $\mathbf{v} \in L - \{\mathbf{0}\}$ τέτοιο, ώστε

$$\|\mathbf{v}\| \leq 2^{(n-1)/4} \det(L)^{1/n}.$$

14.5.1 Βοηθητικά λήμματα

Λήμμα 14.5.1. Αν $\{\mathbf{b}_1, \dots, \mathbf{b}_n\}$ είναι μια ανηγμένη LLL βάση, τότε για την GS βάση ισχύει

$$\|\mathbf{b}_{i-1}^*\| < \frac{2}{\sqrt{4\delta - 1}} \|\mathbf{b}_i^*\|, \quad 2 \leq i \leq n.$$

Απόδειξη. Από τη συνθήκη του Lovász έχουμε

$$\delta \|\mathbf{b}_{i-1}^*\|^2 \leq \|\mu_{i,i-1} \mathbf{b}_{i-1}^*\|^2 + \|\mathbf{b}_i^*\|^2, \quad 2 \leq i \leq n.$$

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. □

14.6 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 επίλυσή του σπάει κρυπτοσυστήματα που πιστεύουμε ότι έχουν ασφάλεια από κβαντικούς υπολογιστές. Ασυμπτωτικά οι αλγόριθμοι κοσκινίσματος έχουν καλύτερη πολυπλοκότητα από τους αλγόριθμους απαρίθμησης. Ο καλύτερος αλγόριθμος¹³³ απαιτεί χρόνο $O(2^{2.465n})$ και μνήμη $O(2^{1.23n})$. Το μειονέκτημά τους είναι η εκθετικά μεγάλη απαίτηση σε μνήμη. Στην πράξη χρησιμοποιούμε και τις δύο κατηγορίες αλγορίθμων. Το πανεπιστήμιο του Darmstadt¹³⁴ τρέχει έναν διαγωνισμό για το approximation SVP όπου φαίνεται ότι και οι δύο κατηγορίες αλγορίθμων δουλεύουν αρκετά καλά (οι Sieving algorithms φαίνεται να δουλεύουν αρκετά καλύτερα στην πράξη).

Ο Ajtai το 1996 απέδειξε ότι το SVP_γ είναι NP-hard under randomized reductions για $\gamma = O(1)$. Για $\gamma = O(\sqrt{n})$ είναι co-NP $\cap$ NP και για $\gamma = 2^{\sqrt{n}}$ έχουμε υπο-εκθετικούς αλγόριθμους. Για $\gamma = O(2^n)$ έχουμε πολυωνυμικούς αλγόριθμους (π.χ. αλγόριθμος LLL).

¹³³Xavier Pujol and Damien Stehle, Solving the shortest lattice vector problem in time $2^{2.465n}$

¹³⁴<https://www.latticechallenge.org/svp-challenge/halloffame.php>

14.6.1 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 έναν 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

$$\pi_n(L), \pi_{n-1}(L), \dots, \pi_1(L)$$

με μήκος $\leq R$. Έστω $\{\mathbf{b}_1, \dots, \mathbf{b}_n\}$ μια βάση του πλέγματος L . Ισχύει

$$\mathbf{b}_i = \mathbf{b}_i^* + \sum_{j=1}^{i-1} \mu_{i,j} \mathbf{b}_j^*. \quad (14.6.1)$$

Οπότε, ένα διάνυσμα

$$\mathbf{v} = \sum_{j=1}^n v_j \mathbf{b}_j \in L$$

γράφεται

$$\mathbf{v} = v_1 \mathbf{b}_1^* + v_2 (\mathbf{b}_2^* + \mu_{2,1} \mathbf{b}_1^*) + \dots + v_n \left(\mathbf{b}_n^* + \sum_{j=1}^{n-1} \mu_{n,j} \mathbf{b}_j^* \right) =$$

$$\mathbf{b}_1^* (v_1 + \mu_{2,1} v_2 + \dots + \mu_{n,1} v_n) + \dots + \mathbf{b}_n^* v_n =$$

$$\sum_{j=1}^n (v_j + \sum_{i=j+1}^n \mu_{i,j} v_i) \mathbf{b}_j^*.$$

Επομένως,

$$\|\mathbf{v}\|^2 = \sum_{j=1}^n \left(v_j + \sum_{i=j+1}^n \mu_{i,j} v_i \right)^2 \|\mathbf{b}_j^*\|^2,$$

και

$$\|\pi_r(\mathbf{v})\|^2 = \sum_{j=r}^n (v_j + \sum_{i=j+1}^n \mu_{i,j} v_i)^2 \|\mathbf{b}_j^*\|^2 \quad (1 \leq r \leq n). \quad (14.6.2)$$

Παρατηρούμε ότι

$$\|\pi_n(\mathbf{v})\|^2 \leq \|\pi_{n-1}(\mathbf{v})\|^2 \leq \dots \leq \|\pi_1(\mathbf{v})\|^2 = \|\mathbf{v}\|^2 \leq R^2.$$

Δηλαδή, το άθροισμα (14.6.2) 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 τη 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

$$\sum_{j=n+1-k}^n \left(v_j + \sum_{i=j+1}^n \mu_{i,j} v_i \right)^2 \|\mathbf{b}_j^*\|^2 \leq R^2,$$

ισοδύναμα

$$\left(v_{n+1-k} + \sum_{i=n+2-k}^n \mu_{i,n+1-k} v_i \right)^2 \|\mathbf{b}_{n+1-k}^*\|^2 + \sum_{j=n+2-k}^n \left(v_j + \sum_{i=j+1}^n \mu_{i,j} v_i \right)^2 \|\mathbf{b}_j^*\|^2 \leq R^2,$$

ισοδύναμα

$$\left| v_{n+1-k} + \sum_{i=n+2-k}^n \mu_{i,n+1-k} v_i \right| \leq \frac{\sqrt{R^2 - \sum_{j=n+2-k}^n \left(v_j + \sum_{i=j+1}^n \mu_{i,j} v_i \right)^2 \|\mathbf{b}_j^*\|^2}}{\|\mathbf{b}_{n+1-k}^*\|}.$$

Αν θέσουμε

$$l_j = \left(v_j + \sum_{i=j+1}^n \mu_{i,j} v_i \right)^2 \|\mathbf{b}_j^*\|^2, \quad (14.6.3)$$

τότε

$$\left| v_{n+1-k} + \sum_{i=n+2-k}^n \mu_{i,n+1-k} v_i \right| \leq \frac{\sqrt{R^2 - \sum_{j=n+2-k}^n l_j}}{\|\mathbf{b}_{n+1-k}^*\|}.$$

Για $k = 1$ έχουμε

$$|v_n| \leq \frac{R}{\|\mathbf{b}_n^*\|}.$$

Άρα,

$$v_n \in \left[-\frac{R}{\|\mathbf{b}_n^*\|}, \frac{R}{\|\mathbf{b}_n^*\|} \right].$$

Επειδή, τα $\pm \mathbf{v}$ έχουν το ίδιο μέτρο, θεωρούμε

$$v_n \in I_1 = \left[0, \frac{R}{\|\mathbf{b}_n^*\|} \right]. \quad (14.6.4)$$

Για $k = 2$ προκύπτει

$$|v_{n-1} + \mu_{n,n-1} v_n| \leq \frac{\sqrt{R^2 - v_n^2 \|\mathbf{b}_n^*\|^2}}{\|\mathbf{b}_{n-1}^*\|},$$

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,

$$v_{n-1} \in I_2 = \left[-\mu_{n,n-1} v_n - \frac{\sqrt{R^2 - l_n}}{\|\mathbf{b}_{n-1}^*\|}, -\mu_{n,n-1} v_n + \frac{\sqrt{R^2 - l_n}}{\|\mathbf{b}_{n-1}^*\|} \right] \quad (14.6.5)$$

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 . Ως ρίζα του δένδρου (root) θεωρούμε το μηδενικό διάνυσμα $\mathbf{0} = \pi_{n+1}(L)$. Οι κόμβοι (nodes) του δένδρου σε βάθος (depth) $k = n + 1 - d$ ($d = n, n - 1, \dots, 1$) ¹³⁵ 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 $\leq R$).

¹³⁵ Βάθος ενός κόμβου σ' ένα δένδρο είναι το πλήθος των ακμών που ενώνουν τη ρίζα με τον κόμβο. Όμως 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.

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 ως προς την τάξη του πλέγματος). Ο αλγόριθμος αυτός, για να διασχίσει το δένδρο, χρησιμοποιεί DFS : **D**epth **F**irst **S**earch. Ο λόγος είναι επειδή ο DFS δεν χρησιμοποιεί πολλή μνήμη.

Άσκηση 14.22 Αν ισχύει η GSA σε ένα πλέγμα και ο συντελεστής GSA είναι $r > 1/2$, τότε $|I_1 \cap \mathbb{Z}| < |I_2 \cap \mathbb{Z}|$.

Παράδειγμα 14.6.1. 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 τη σχέση (14.6.4) 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 τη σχέση (14.6.5) 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 τη σχέση (14.6.5) (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 $v_2 \in \{-1, 0, 1\}$. Και το $l_3 = 0$. Δηλαδή,

$$I_3(v_2) = \left[-\frac{\sqrt{R^2 - l_2}}{\|\mathbf{b}_1\|} - \mu_{2,1}v_2, \frac{\sqrt{R^2 - l_2}}{\|\mathbf{b}_1\|} - \mu_{2,1}v_2 \right]$$

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

$$I_3(-1) = [0.33, 1.96].$$

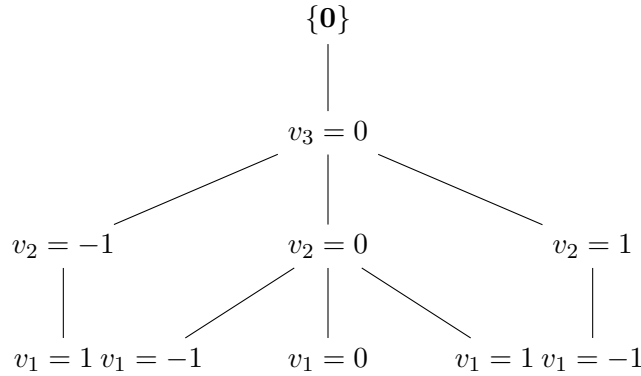
Επομένως, $v_1 = 1$. Για $v_2 = 1$ έχουμε

$$I_3(1) = [-1.96, -0.33],$$

άρα, $v_1 = -1$. Όμως, $I_3(0) = [-1, 1]$, επομένως, $v_1 \in \{-1, 0, 1\}$. 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 $\mathbf{v}_2 = (3, 6, 13)$ που έχουν μέτρο μικρότερο ή ίσο του R .

Παρατήρηση 14.6.1. Αν στο δέντρο απαρίθμησης ανήκει το κλαδί $[v_n, v_{n-1}, \dots, v_1]$, τότε και το κλαδί $[-v_n, -v_{n-1}, \dots, -v_1]$ 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' αυτόν τον λόγο, πριν εφαρμόσουμε τον αλγόριθμο απαρίθμησης, χρησιμοποιούμε κάποιον αλγόριθμο αναγωγής (LLL/BKZ).

Όσον αφορά την πολυπλοκότητα του αλγορίθμου έχουμε το παρακάτω αποτέλεσμα.

Θεώρημα 14.6.1 (Hanrot-Stehlé). Υπάρχει πολύωνμο $p(x, y) \in \mathbb{R}[x, y]$ 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 ντετερμινιστικός αλγόριθμος που βασίζεται στα κελιά Voronoi, με ασυμπτωτική πολυπλοκότητα $2^{2n+o(n)}$ και μνήμη $2^{n+o(n)}$.

14.6.2 Ο ψευδοκώδικας του αλγορίθμου απαρίθμησης

Αλγόριθμος 14.6.1. : Αλγόριθμος Απαρίθμησης (KFP enumeration algorithm)

Είσοδος. Μια διατεταγμένη βάση $B = \{\mathbf{b}_1, \dots, \mathbf{b}_n\} \subset \mathbb{Z}^m$ του πλέγματος $\mathcal{L}(B)$ και έναν 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.  $\mathbf{x} = (x_i) \leftarrow \mathbf{0}_n, \mathbf{c} = (c_i) \leftarrow \mathbf{0}_n, \ell = (\ell_i) \leftarrow \mathbf{0}_n, \text{sumli} \leftarrow 0, S = \emptyset, i \leftarrow 1$ 
03. While  $i \leq n$ 
04.    $c_i \leftarrow -\sum_{j=i+1}^n x_j \mu_{ji}$ 
05.    $\ell_i \leftarrow B_i(x_i - c_i)^2$ 
06.    $\text{sumli} \leftarrow \sum_{j=i}^n \ell_j$ 
07.   If  $\text{sumli} \leq R^2$ 
08.     If  $i = 1$ 
09.        $S \leftarrow S \cup \{\sum_{j=1}^n x_j \mathbf{b}_j\}$ 
10.        $x_1 \leftarrow x_1 + 1$ 
11.     else
12.        $i \leftarrow i - 1$ 
13.        $x_i \leftarrow \text{left part of the interval } I_i$ 
14.     end if
15.   else
16.      $i \leftarrow i + 1$ 
17.      $x_i \leftarrow x_i + 1$ 
18.   end if
19. return  $S$ 

```

Στη γραμμή 13, το x_i παίρνει την τιμή

$$\left\lceil -\sum_{j=i+1}^n \mu_{j,i} x_j - \sqrt{\frac{R^2 - \sum_{j=i+1}^n \ell_j}{B_i}} \right\rceil$$

που είναι το αριστερό άκρο του διαστήματος I_{n+1-i} . Στον ψευδοκώδικα το i είναι το ύψος (height) του δένδρου απαρίθμησης. Ο αλγόριθμος αυτός δεν παίρνει έτοιμα τα διαστήματα I_i , αλλά φτιάχνει αυτά τα διαστήματα με τον έλεγχο που γίνεται στη γραμμή 7, που είναι $l_i \leq R^2$, όπου το l_i δίνεται από την ισότητα (14.6.3). Όταν το $i = 1$ (γραμμή 8), σημαίνει ότι έχουμε βρει ένα διάνυσμα με μέτρο $\leq R$, οπότε το αποθηκεύουμε στη λίστα S . Αν το $i > 1$ (γραμμή 11), τότε μειώνουμε το ύψος (δηλαδή, 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 (γραμμές 15-16) θα αυξήσουμε 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 εξής 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 $O(n)$.

14.6.3 Schnorr-Euchner

Οι Schnorr-Euchner πρότειναν το x_i να ξεκινά από την τιμή

$$\left[- \sum_{j=i+1}^n \mu_{j,i} x_j \right]$$

που είναι πολύ κοντά στο κέντρο του διαστήματος I_i και να αυξομειώνεται το x_i κατά 1. Η παραλλαγή αυτή ονομάζεται αλγόριθμος των Schnorr-Euchner και χρησιμοποιείται ως υπορουτίνα στον αλγόριθμο BKZ. Στον αλγόριθμο των Schnorr-Euchner υπάρχει μια ακόμη παραλλαγή όπου επιτρέπουμε αντί του R , όταν βρισκόμαστε σε βάθος i στη γραμμή 7, να έχουμε $sumli \leq R_i$ όπου $R_i = \alpha_i R$ για κάποιο $\alpha_i \in (0, 1)$. Τότε λέμε ότι έχουμε απαρίθμηση με κλάδεμα (enumeration with pruning) που επίσης πρότειναν οι Schnorr-Euchner-Horner.

Αλγόριθμος 14.6.2. : Αλγόριθμος Απαρίθμησης των Schnorr-Euchner (Schnorr-Euchner enumeration algorithm)¹³⁶

Είσοδος. Μια διατεταγμένη βάση $B = \{\mathbf{b}_1, \dots, \mathbf{b}_n\} \subset \mathbb{Z}^m$ του πλέγματος $\mathcal{L}(B)$ και έναν θετικό αριθμό R .

Έξοδος. Όλα τα διανύσματα $\mathbf{x} \in \mathcal{L}$ ($x_n \geq 0$) $\|\mathbf{x}\| \leq R$.

01. Compute $\{\mu_{ij}\}$ and $B_i = \|\mathbf{b}_i^*\|^2$
02. $\mathbf{x} = (x_i) \leftarrow \mathbf{0}_n, \mathbf{c} = (c_i) \leftarrow \mathbf{0}_n, \ell = (\ell_i) \leftarrow \mathbf{0}_n$
03. $\Delta \mathbf{x} \leftarrow (1, \mathbf{0}_{n-1}), \Delta^2 \mathbf{x} \leftarrow (1, (-1)_{n-1}), sumli \leftarrow 0, S = \emptyset, i \leftarrow 1$
04. **While** $i \leq n$
05. $c_i \leftarrow - \sum_{j=i+1}^n x_j \mu_{ji}$
06. $\ell_i \leftarrow B_i (x_i - c_i)^2$
07. $sumli \leftarrow \sum_{j=i}^n \ell_j$
08. **If** $sumli \leq R^2$ and $i = 1$
09. $S \leftarrow S \cup \{\sum_{j=1}^n x_j \mathbf{b}_j\}$
10. **end if**
11. **If** $sumli \leq R^2$ and $i > 1$
12. $i \leftarrow i - 1$
13. $c_i \leftarrow - \sum_{j=i+1}^n x_j \mu_{ji}$ # center of the interval I_{n+1-i}
14. $x_i \leftarrow \lfloor c_i \rfloor$
15. $\Delta x_i \leftarrow 0$
16. **If** $c_i < x_i$

¹³⁶ Δείτε Fig. 1 του Hanrot, Stehle, Rigorous and Efficient Short Lattice Vectors Enumeration

```

17.       $\Delta^2 x_i \leftarrow 1$ 
18.      else
19.           $\Delta^2 x_i \leftarrow -1$ 
20.      end if
21.      elif  $i = n$  # i.e. if  $sumli > R^2$  and  $i = n$ 
22.          break
23.      else # i.e. if  $sumli > R^2$  or  $i = 1$ 
24.           $i \leftarrow i + 1$ 
25.           $\Delta^2 x_i \leftarrow -\Delta^2 x_i$ 
26.           $\Delta x_i \leftarrow -\Delta x_i + \Delta^2 x_i$ 
27.           $x_i \leftarrow x_i + \Delta x_i$ 
28.      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 ο προηγούμενος αλγόριθμος δεν διαφέρει από τον αλγόριθμο των KFP.

Αν στη γραμμή 08 (και 11) του αλγορίθμου, αντί R^2 θέσουμε $R_{n+1-i}^2 = a_{n+1-i}^2 R^2$, για κάποια $a_i \in (0, 1)$, τότε έχουμε τον αλγόριθμο enumeration with pruning και pruning vector

$$\mathbf{a} = (a_1, a_2, \dots, a_n),$$

με

$$0 < a_1 \leq a_2 \leq \dots \leq a_n = 1.$$

Δηλαδή, η γραμμή 8 θα αντικατασταθεί από την

08. **If** $sumli \leq (a_{n+1-i}R)^2$ and $i = 1$

και η γραμμή 11 από την

11. **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 $\mathbf{a} = (a_1, \dots, a_n)$, τότε για κάθε i στον αλγόριθμο θεωρούμε τη 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 $\mathbf{a} = (1, 1, \dots, 1)$, τότε εκτελείται ο αλγόριθμος χωρίς pruning. Όταν χρησιμοποιούμε pruning, 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 NTL¹³⁷.

Άσκηση 14.23 Έστω $\{\mathbf{b}_i\}_{1 \leq i \leq n}$ μια βάση ενός πλέγματος $\mathcal{L}$. Να υλοποιηθεί ο αλγόριθμος (14.6.2) με pruning vector τέτοιο, ώστε $a_i = \min\{1, \sqrt{1.05i/n}\}$ ($1 \leq i \leq n$).

14.7 CVP

Το πρόβλημα CVP_γ , $\gamma \geq 1$ (approximate Closest Vector Problem) είναι το εξής:

Ορισμός 14.7.1. ($CVP_\gamma(\mathcal{L}, \mathbf{t})$) 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 $\mathbf{y} \in \mathcal{L}$ να ισχύει

$$\|\mathbf{x} - \mathbf{t}\| \leq \gamma \|\mathbf{y} - \mathbf{t}\|,$$

για κάποιο $\gamma \geq 1$.

Το πρόβλημα απόφασης είναι το παρακάτω:

¹³⁷<https://github.com/fplll/fplll> και <https://shoup.net/ntl/>

Ορισμός 14.7.2. (*Decision CVP_γ(L, 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

$$\text{dist}(\mathbf{t}, \mathcal{L}) = \min\{\|\mathbf{x} - \mathbf{t}\| : \mathbf{x} \in \mathcal{L}\},$$

είναι $\leq r$ ή $> \gamma r$.

Με χρήση του αλγορίθμου LLL μπορούμε να λύσουμε το προσεγγιστικό $\text{CVP}_\gamma(\mathcal{L})$ για κάποιο πλέγμα $\mathcal{L} \subset \mathbb{Z}^m$, για $\gamma = 2^{n/2}$ και $n = \text{rank}(\mathcal{L})$ σε πολυωνυμικό χρόνο. Ο αλγόριθμος που προκύπτει ονομάζεται αλγόριθμος του Babai.

Αλγόριθμος 14.7.1. : Babai's nearest plane algorithm

Είσοδος. Μια διατεταγμένη βάση $\mathcal{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_n\} \subset \mathbb{Z}^m$ του πλέγματος $\mathcal{L}(\mathcal{B})$ και ένα διάνυσμα $\mathbf{t} \in \mathbb{R}^m$.

Έξοδος. $\mathbf{x} \in \mathcal{L}$ τέτοιο, ώστε $\|\mathbf{x} - \mathbf{t}\| \leq 2^{n/2} \text{dist}(\mathcal{L}, \mathbf{t})$

```

1  $B \leftarrow \text{LLL}(B)$  ( $\delta = 3/4$ )
2 Compute  $B^* \leftarrow \text{GSO}(B)$ 
3  $\mathbf{b} \leftarrow \mathbf{t}$ 
4 for  $j = n$  to 1 do
5    $c_j \leftarrow \left\lfloor \frac{\mathbf{b} \cdot \mathbf{b}_j^*}{\|\mathbf{b}_j^*\|^2} \right\rfloor$     $\# \lfloor x \rfloor = \lfloor x + 0.5 \rfloor$ 
6    $\mathbf{b} \leftarrow \mathbf{b} - c_j \mathbf{b}_j$ 
7 end
8 return  $\mathbf{t} - \mathbf{b}$ 
```

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 $\mathbf{t} - \sum_{i=1}^n \lambda_i \mathbf{b}_i$ για κάποια $\lambda_i \in \mathbb{Z}$. Άρα, πράγματι το $\mathbf{t} - \mathbf{b} \in \mathcal{L}$.

Η ιδέα του αλγορίθμου.

Περιγράφουμε στη 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 του προηγούμενου αλγορίθμου είναι να βρει ένα σημείο του πλέγματος επί της ευθείας ε_c , ώστε να είναι πιο κοντά στο σημείο $\mathbf{t}$. Να παρατηρήσουμε 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 14.7.1. $c\mathbf{b}_2 \in \varepsilon_c$.

Απόδειξη. $\mathbf{b}_2^* = \mathbf{b}_2 - \mu_{2,1}\mathbf{b}_1$. Επομένως,

$$c\mathbf{b}_2 = c\mathbf{b}_2^* + c\mu_{2,1}\mathbf{b}_1 \in c\mathbf{b}_2^* + \text{span}(\mathbf{b}_1) = \varepsilon_c.$$

□

14.8 Συστήματα που βασίζονται σε πλέγματα

Το πλεονέκτημα αυτών των συστημάτων είναι ότι συνήθως έχουμε αποδείξεις ασφάλειας που βασίζονται σε δύσκολα προβλήματα των πλεγμάτων στη χειρότερη περίπτωση. Ειδικότερα, οι αναγωγές αυτές είναι από τη μέση περίπτωση στη 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 κρυπτούστημα που βασίζεται στο πρόβλημα υποσυνόλου (Subset Sum). Σε κάθε περίπτωση, αυτά τα συστήματα δεν ήταν πετυχημένα. Αυτό συνέβαινε διότι ήταν δύσκολα στη χειρότερη περίπτωση και όχι κατά μέσο όρο. Δηλαδή, 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 κάποιον τρόπο έλυσε το προηγούμενο πρόβλημα. Βρήκε μια κατηγορία προβλημάτων, που είναι κατά μέσο όρο δύσκολα (average case hard problems) και απέδειξε ότι η επίλυσή τους οδηγεί στην επίλυση ενός προβλήματος που είναι δύσκολο στη 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 στη χειρότερη περίπτωση.

Οι περισσότερες εργασίες, που αφορούν συστήματα που βασίζονται σε πλέγματα, είναι αποδείξεις του τύπου average-worst case reductions. Δηλαδή, οι αποδείξεις αυτές μετασχηματίζουν μια επιτυχημένη επίθεση στο κρυπτοσύστημα, σε έναν αποδοτικό αλγόριθμο που λύνει οποιοδήποτε παράδειγμα ενός προβλήματος στα πλέγματα στη 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 στη χειρότερη περίπτωση.

Επίσης, υπάρχουν χβαντικές επιθέσεις σε μερικά προβλήματα που βασίζονται σε πλέγματα. Η χβαντική 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-Halevi¹³⁸. Τα τρία προηγούμενα συστήματα ανήκουν στην κατηγορία των ideal/lattice based cryptosystems.

Εξαιρετικό ενδιαφέρον παρουσιάζει η χβαντική επίθεση των Eldar-Shor¹³⁹. Όταν παρουσιάστηκε αυτή η επίθεση, πολλοί ερευνητές ασχολήθηκαν προσεκτικά με την προηγούμενη εργασία και πολύ σύντομα βρέθηκε ένα κενό στη μέθοδό τους από τον Oded Regev. Φυσικά η εργασία αποσύρθηκε. Μερικοί πιστεύουν¹⁴⁰ ότι αυτό ήταν ικανό να δημιουργήσει αμφιβολίες για τα συστήματα που βασίζονται σε πλέγματα, ώστε η Google να διακόψει το πρόγραμμα CECQP¹⁴¹ το οποίο αφορούσε ένα υβριδικό σύστημα ανταλλαγής κλειδιών που βασίζεται σε πλέγματα και σε διακριτό λογάριθμο (τα συστήματα X25519 και NEW HOPE). Ο NEWHOPE είναι Ring LWE τύπου σύστημα ανταλλαγής κλειδιού και σε περίπτωση που ένας χβαντικός υπολογιστής έσπαζε αυτό το σύστημα ο αλγόριθμος X25519, που βασίζεται σε ελλειπτικές καμπύλες, εγγυάται τουλάχιστον την κλασική ασφάλεια του συστήματος.

14.8.1 Learning With Errors (LWE)

Το πρόβλημα Learning With Errors (LWE) μας δίνει μια σειρά από κρυπτοσυστήματα (ομομορφικά και μη) ανθεκτικά σε χβαντικές επιθέσεις. Πρώτη φορά παρουσιάστηκε από τον Regev το 2005¹⁴². 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 συνόλου $\mathbb{Z}_{11}$. 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 $\mathbb{Z}_q$ ($i = 1, 2, \dots, k$) και εξάγει τα k -ζεύγη

$$(\mathbf{a}_i, \mathbf{a}_i \cdot \mathbf{s} + e_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q.$$

¹³⁸ <http://web.eecs.umich.edu/~cpeikert/soliloquy.html>

¹³⁹ <https://arxiv.org/abs/1611.06999>

¹⁴⁰ <https://groups.google.com/forum/#!topic/cryptanalytic-algorithms/WNMuTfJuSRc>

¹⁴¹ <https://en.wikipedia.org/wiki/CECPQ1>

¹⁴² Oded Regev, On Lattices, Learning with Errors, Random Linear Codes, and Cryptography, STOC 2005

Ορισμός 14.8.1. (*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 $\mathbb{Z}_q$, δηλαδή, $LWE_{n,k,q,\chi}$. Αν $q = 2$ και χ η κατανομή Bernouli, δηλαδή, $\chi(0) = \tau$, $\chi(1) = 1 - \tau$ ($\tau \in (0, 1)$), τότε το πρόβλημα *LWE* ονομάζεται *LPN : Learning Parity with Noise problem*.¹⁴³

Ορίζουμε και το πρόβλημα απόφασης.

Ορισμός 14.8.2. (*DLWE ή Decision-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 δείγμα 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 $q_i = \text{poly}(n)$. Πριν δούμε τη δυσκολία αυτού του προβλήματος, χρειαζόμαστε έναν ορισμό για ένα κλασικό πρόβλημα στα πλέγματα.

Ορισμός 14.8.3. (*SIVP*) Έστω πλέγμα $\mathcal{L}$ διάστασης n . Το πρόβλημα *SIVP: Shortest Independent Vectors Problem*, είναι η εύρεση n γραμμικά ανεξάρτητων διανυσμάτων $\mathbf{x}_i$ ($i = 1, 2, \dots, n$) του $\mathcal{L}$ τέτοιων, ώστε η ποσότητα $M = \max_i \|\mathbf{x}_i\|$ να γίνεται ελάχιστη. Αν αντί του M θεωρήσουμε το $M_\gamma = \gamma M$ ($\gamma \in \mathbb{R}_{>1}$), τότε έχουμε το προσεγγιστικό *SIVP* που συμβολίζεται *SIVP_γ*.

Γενικότερα, ο Regev απέδειξε ότι, αν έχουμε ένα μαντείο *LWE*, τότε υπάρχει κβαντικός αλγόριθμος που λύνει το *SIVP_γ* και ο αλγόριθμος χρησιμοποιεί απαντήσεις από το *LWE*-μαντείο. Δηλαδή, έχουμε μια κβαντική αναγωγή του *SIVP_γ* στο πρόβλημα *LWE*. Επομένως, το πρόβλημα *LWE* είναι τόσο δύσκολο, όσο η χειρότερη περίπτωση του προβλήματος *SIVP_γ*, 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 αναγωγή¹⁴⁴, όταν $q = O(\text{poly}(n))$. Οπότε, η μελέτη της δυσκολίας αυτού του προβλήματος έχει απαντηθεί ικανοποιητικά.

Θεώρημα 14.8.1 Έστω $q = \text{poly}(n)$ και $\alpha q > \sqrt{n}$ ($\alpha \in (0, 1)$). Η λύση του *LWE_q* συνεπάγεται μια λύση σε ένα δύσκολο πρόβλημα των πλεγμάτων στη χειρότερη περίπτωση.

Η κατανομή χ είναι συνήθως η (διακριτή) κατανομή του Gauss επί του $\mathbb{Z}_q$, με τυπική απόκλιση αq και $\alpha > 1/\text{poly}(n)$.

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 τη 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 (14.2).

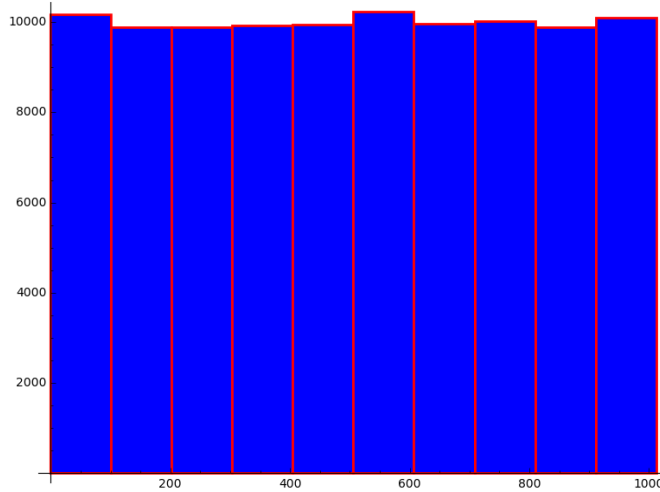
Επιθέσεις. Μια απλή επίθεση στο *LWE* είναι η εξής: Αν έχω ένα *LWE* μαντείο (δηλαδή, μια κατανομή $\mathcal{A}_{\mathbf{s},\chi}$), τότε ζητάω $\text{poly}(n)$ δείγματα. Αν τύχει ένα ζευγάρι $(\mathbf{a}, b)$ με $\mathbf{a} = (1, 0, \dots, 0)$, τότε έχω την εξίσωση $b = \mathbf{a} \cdot \mathbf{s} + e = s_1 + e$. 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 : Bounded Distance Decoding problem*. Χρειαζόμαστε τον παρακάτω ορισμό.

¹⁴³A. Blum, M. L. Furst, M. J. Kearns and R. J. Lipton, Cryptographic Primitives Based on Hard Learning Problems, Crypto 1993

¹⁴⁴Brakerski et al. Classical Hardness of Learning with Errors



Σχήμα 14.2: (Εμπειρική) επαλήθευση της υπόθεσης LWE.

Ορισμός 14.8.4. 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 στην απαρίθμηση των KFP. Τέλος, υπάρχει η συνδυαστική επίθεση (BKW) που ανάγει το DLWE στο SIS : Short Integer Solution problem¹⁴⁵. Εμείς θα δούμε αναλυτικά την επίθεση που ανάγει το πρόβλημά μας στο BDD_α , διότι με βάση αυτήν την επίθεση, 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 (στη χειρότερη περίπτωση). Οπότε και είναι υποψήφια για κρυπτοσυστήματα, με το δεύτερο να έχει το πλεονέκτημα ότι παράγει και ομομορφικά κρυπτοσυστήματα. Τέλος, υπάρχουν συναρτήσεις κατακερματισμού που βασίζονται στο SIS.

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.

14.8.2 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,

$$\mathbf{c} = \text{Enc}_{\mathbf{s}}(m) = (\mathbf{a}, \mathbf{a} \cdot \mathbf{s} + te + m) = (\mathbf{a}, b) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$$

και

$$\text{Dec}_{\mathbf{s}}(\mathbf{c}) = (b - \mathbf{a} \cdot \mathbf{s} \pmod{t}) = (te + m) \pmod{t} = m \pmod{t} = m.$$

Καθώς το σύστημα είναι συμμετρικό η Alice γνωρίζει το κλειδί αποκρυπτογράφησης $\mathbf{s}$. Η κρυπτογράφηση είναι πιθανοτική (και όχι ντετερμινιστική), δηλαδή, 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 $\mathbf{a} \cdot \mathbf{s} + te + m$.

¹⁴⁵Fitzpatrick, Some Algorithms for Learning with Errors, Thesis (2014)

Για κάθε κρυπτογραφημένο μήνυμα ορίζεται η γραμμική συνάρτηση $\mathbb{Z}_q^n \rightarrow \mathbb{Z}_q$

$$f_{\mathbf{a},b}(\mathbf{x}) = b - \mathbf{a} \cdot \mathbf{x} = b - \sum_{i=1}^n a_i x_i.$$

Η αποκρυπτογράφηση τότε γράφεται

$$Dec_{\mathbf{s}}(\mathbf{c}) = f_{\mathbf{a},b}(\mathbf{s}) \pmod{t}.$$

Αλγόριθμος 14.8.1. : 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 $\{0, 1, \dots, q-1\}$.

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  $enc(m) = (\mathbf{a}, \mathbf{a} \cdot \mathbf{s} + te + m \pmod{q})$ 
4 return  $enc(m)$ 
```

Η αποκρυπτογράφηση γίνεται ως εξής:

Αλγόριθμος 14.8.2. : 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 14.8.1. 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

$$\mathbb{Z}_q^4 = \{(x_1, x_2, x_3, x_4) : x_i \in \{-(q-1)/2, \dots, 0, \dots, (q-1)/2\}\},$$

για q περιττό πρώτο. 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 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

$$Dec(c) = 1048571 - t = 1038571 - 2^{20} = -5.$$

14.8.3 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

$$F = (b - (\mathbf{a} \cdot \mathbf{x})) \in \mathbb{Z}_q[x_1, \dots, x_n]$$

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 $F(\mathbf{x}) = f_{\mathbf{a}_1 + \mathbf{a}_2, b_1 + b_2}(\mathbf{x}) = f_{\mathbf{a}_1, b_1}(\mathbf{x}) + f_{\mathbf{a}_2, b_2}(\mathbf{x})$. Δηλαδή, 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 $m_1, \dots, m_k$, τότε αρκεί να γίνει ο προηγούμενος υπολογισμός για το διάνυσμα,

$$\left(\sum_{i=1}^k \mathbf{c}_i, \sum_{i=1}^k b_i \right).$$

Δηλαδή, το σύστημά μας επιτρέπει όσες προσθέσεις θέλουμε και οι υπολογισμοί γίνονται αποδοτικά (αφού δεν αυξάνεται 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

$$f_{\mathbf{a}_1, b_1}(\mathbf{x}) \cdot f_{\mathbf{a}_2, b_2}(\mathbf{x}) =$$

$$h_0 + h_1 x_1 + \dots + h_n x_n + \sum_{i,j} h_{i,j} x_i x_j \in \mathbb{Z}_q[x_1, \dots, x_n].$$

Δηλαδή, στέλνει το διάνυσμα

$$(h_0, h_1, \dots, h_n, h_{11}, \dots, h_{nn}) \in \mathbb{Z}_q^{(n+1)(n+2)/2}.$$

Σε αυτήν την περίπτωση η Alice υπολογίζει το πολυώνυμο G ,

$$G = h_0 + h_1x_1 + \cdots + h_nx_n + \sum_{i,j} h_{i,j}x_ix_j$$

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 $m_1, m_2 \in \mathbb{Z}_{\lceil \sqrt{t} \rceil}$. Δηλαδή, 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 πολυωνύμων, 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 αυτήν 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 $\mathbf{s} = (s_1, \dots, s_n)$, 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

$$\mathbf{a}_i = (a_{ik})_k, b_i = \mathbf{a}_i \cdot \mathbf{r} + t e_i + s_i, \mathbf{a}_{ij} = (a_{ijk})_k, b_{ij} = \mathbf{a}_{ij} \cdot \mathbf{r} + t e_{ij} + s_i s_j. \quad (14.8.1)$$

Υποθέτουμε ότι η Alice δημοσιεύει τις προηγούμενες κρυπτογραφήσεις ($n^2 + n$ το πλήθος).

Λήμμα 14.8.1. 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 (14.8.1). Κατόπιν, ο Bob υπολογίζει το γινόμενο των πολυωνύμων

$$f_{\mathbf{A}_1, B_1}(\mathbf{x}) \cdot f_{\mathbf{A}_2, B_2}(\mathbf{x}) = h_0 + h_1x_1 + \cdots + h_nx_n + \sum_{i,j} h_{i,j}x_ix_j$$

και στέλνει στην Alice το διάνυσμα του $\mathbb{Z}_q^{n+1}$,

$$(H_0, H_1, \dots, H_n) =$$

$$\left(h_0 + \sum_{i=1}^n h_i b_i + \sum_{i=1}^n h_{ij} b_{ij}, \sum_{i=1}^n h_i a_{i,1} + \sum_{i,j} h_{ij} a_{i,j,1}, \dots, \sum_{i=1}^n h_i a_{i,n} + \sum_{i,j} h_{ij} a_{i,j,n} \right).$$

Τότε, η αποκρυπτογράφηση από την Alice με το κλειδί $\mathbf{r}$ θα εμφανίσει το γινόμενο $m_1 m_2$.

Απόδειξη. Έστω, $G(x_1, \dots, x_n) = H_0 + H_1 x_1 + \cdots + H_n x_n$. Τότε, $G(r_1, \dots, r_n) = H_0 + H_1 r_1 + \cdots + H_n r_n$. 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,

$$f_{\mathbf{A}_1, B_1}(\mathbf{s}) f_{\mathbf{A}_2, B_2}(\mathbf{s}) \pmod{t} =$$

$$\begin{aligned}
& h_0 + \sum_{i=1}^n h_i s_i + \sum_{i,j} h_{ij} s_i s_j \pmod{t} = \\
& h_0 + \sum_{i=1}^n h_i (b_i - \mathbf{a}_i \cdot \mathbf{r} - t e_i) + \sum_{i,j} h_{ij} (b_{ij} - \mathbf{a}_{ij} \cdot \mathbf{r} - t e_{ij}) \pmod{t} = \\
& h_0 + \sum_{i=1}^n h_i (b_i - \mathbf{a}_i \cdot \mathbf{r}) + \sum_{i,j} h_{ij} (b_{ij} - \mathbf{a}_{ij} \cdot \mathbf{r}) = \\
& H_0 + H_1 r_1 + \cdots + H_n r_n = G(r_1, \dots, r_n) = G(\mathbf{r}).
\end{aligned}$$

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

$$Dec : \mathbb{Z}_q^{(n+1)(n+2)/2} \rightarrow \mathbb{Z}_q$$

ενώ με Relinearization είναι

$$Dec : \mathbb{Z}_q^{n+1} \rightarrow \mathbb{Z}_q.$$

Το σύστημα, όπως το παρουσιάσαμε, χρησιμοποιεί μόνο μία 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 $\mathbf{r}_1, \dots, \mathbf{r}_L$, 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 Bootstrapping, 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, στη 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 στην 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-Regev¹⁴⁶ και Lindner-Peikert (LP)¹⁴⁷. Η πρώτη επίθεση είναι η κατασκευή ενός διαχωριστή (distinguisher) που χρησιμοποιεί το SIS-problem, για να λύσει το πρόβλημα απόφασης LWE (DLWE), ενώ στη δεύτερη επίθεση γίνεται μια επίθεση επί του BDD-problem σε κατάλληλο q -αδίκιο 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 ο 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 $(M_1, \dots, M_m)$ το μαντέιο απαντάει με m -δείγματα $(\mathbf{a}_i, b_i) \leftarrow \mathcal{A}_{\mathbf{s}, \chi}$, $i = 1, 2, \dots, m$. Ισχύουν οι εξισώσεις, $\mathbf{a}_i \cdot \mathbf{s} \equiv e_i t + M_i \pmod{q}$.

¹⁴⁶Miccianchio-Regev, 2009, <https://eprint.iacr.org/2011/405.pdf>

¹⁴⁷Lindner-Peikert, 2011, <http://web.eecs.umich.edu/~cpeikert/pubs/lwe-analysis.pdf>

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 $\mathbf{b} = (b_i)_{i=1, \dots, m} \in \mathbb{Z}_q^m$ 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 $B = (\mathbf{b}_1, \dots, \mathbf{b}_n)$ 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 $M_1 = M_2 = \dots = M_m = 0$, 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

$$\mathcal{P}_{t/2}(B^*) = \left\{ \sum_{i=1}^n x_i \mathbf{b}_i^* : \frac{-t}{2} \leq x_i < \frac{t}{2} \right\}.$$

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, δηλαδή, έδωσαν μια εκτίμηση για την πιθανότητα $Pr(\mathbf{z} : \mathbf{b} - \mathbf{z} \in \mathcal{P}_{t/2}(B))$. Προσέγγισαν τη διακριτή κατανομή του Gauss με τη συνεχή κατανομή του Gauss.

Λήμμα 14.8.2.

$$Pr(\mathbf{z} : \mathbf{b} - \mathbf{z} \in \mathcal{P}_{t/2}(B)) = \prod_{i=1}^m erf\left(\frac{td_i \|\mathbf{b}_i^*\| \sqrt{\pi}}{2s}\right)$$

Η συνάρτηση

$$erf(x) = \frac{1}{\sqrt{\pi}} \int_{-x}^x e^{-t^2} dt$$

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. Η υπόθεση αυτή ελέγχθηκε σε τυχαία πλέγματα και τουλάχιστον πειραματικά επαληθεύει την υπόθεση GSA και 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

$$m = \sqrt{\frac{n \log_2(q)}{\log_2(\delta)}}.$$

Βιβλιογραφία

- [1] Galbraith, Steven. *Mathematics of Public Key Cryptography (version 1.1)*. <http://www.math.auckland.ac.nz/~sgal018/crypto-book/crypto-book.html>, 2011.
- [2] Gerstein, L. J. *Basic Quadratic Forms*. vol. 90, Graduate studies in mathematics. AMS, 2008.
- [3] Halmos, P. *Finite-Dimensional Vector Spaces*. Springer, 1993.
- [4] Miccianchio, D; Goldwasser, S. *Complexity of Lattice problems. A Cryptographic Perspective*. Springer, 2002.
- [5] Samuel, Pierre. *Algebraic Theory of Numbers*. Hermann, 1970.
- [6] Sheldon, A. *Linear Algebra Done Right*. Springer, 2015.
- [7] Strang, G. *Introduction to Linear Algebra*. Wellesley-Cambridge Press, 2016.
- [8] Τσίχλας, Κ.; Μανωλόπουλος, Ι.; Γούναρης, Α. *Σχεδίαση και Ανάλυση Αλγορίθμων*, Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. <http://dx.doi.org/10.57713/kallipos-613>, 2015.
- [9] Τζανάκη, Ελένη. *Γεωμετρία των Αριθμών*. Διπλωματική Εργασία, Πανεπιστήμιο Κρήτης, 2000.
- [10] Ζάχος, Ε.; Παγουρτζής, Α.; Σούλιου, Θ. *Θεμελίωση επιστήμης υπολογιστών*, Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. <http://dx.doi.org/10.57713/kallipos-491>, 2015.
- [11] Κολουτζάκης, Μ.; Παπαχριστόδουλος, Χ. *Διακριτά μαθηματικά*, Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. <http://dx.doi.org/10.57713/kallipos-517>, 2015.

Κεφάλαιο 15

SSL/TLS & PGP

*A false sense of security is worse
than no security at all.*

Unknown

Σύνοψη - Περίληψη. Σε αυτό το κεφάλαιο παρουσιάζουμε δύο πολύ βασικά πρωτόκολλα, το SSL/TLS και το PGP. Το πρώτο χρησιμοποιείται οπουδήποτε απαιτείται κρυπτογράφηση δύο οντοτήτων στο διαδίκτυο, ενώ το δεύτερο μας επιτρέπει να στέλνουμε κρυπτογραφημένα ηλεκτρονικά μηνύματα.

Προαπαιτούμενη γνώση. Για αυτήν την ενότητα χρειάζονται κάποιες βασικές γνώσεις για δίκτυα υπολογιστών (μπορείτε να δείτε τα εισαγωγικά σ' ένα από τα βιβλία [4, 5]). Επίσης, είναι απαραίτητες κάποιες γνώσεις για ασφάλεια υπολογιστικών συστημάτων, π.χ. [1, 2, 3, 6]. Τέλος, στην ελληνόγλωσση βιβλιογραφία μπορείτε να δείτε [7, 8].

15.1 SSL/TLS

Το πρωτόκολλο SSL (δείτε [8]) αναπτύχθηκε από τη 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) καθώς και η έκδοση 1.3. Χρησιμοποιεί περίπου 200 ρουτίνες (ciphersuites), όπως:

TLS_RSA_WITH_AES_128_CBC_SHA256 (ασφαλής),

TLS_ECDSA_RSA_WITH_AES_128_GCM_SHA256 (ασφαλής),¹⁴⁸

TLS_KRB5_WITH_3DES_EDE_CBC_MD5 (αδύναμη),

TLS_NULL_WITH_NULL_NULL (χωρίς ασφάλεια!).

TLS_AES_128_CCM_8_SHA256 Χρήσιμη για embedded devices που επιθυμούν να χρησιμοποιήσουν ένα κύκλωμα AES τόσο για κρυπτογράφηση όσο και για αυθεντικοποίηση. Υπάρχει μόνο στην έκδοση 1.3.

Η πρώτη ciphersuite, TLS_RSA_WITH_AES_128_CBC_SHA256, χρησιμοποιεί RSA για αυθεντικοποίηση και για ανταλλαγή κλειδιού, ενώ για συμμετρική κρυπτογράφηση χρησιμοποιεί AES_128_CBC. Η hash που χρησιμοποιείται είναι SHA256. Οτιδήποτε πριν από τη λέξη with αφορά την αυθεντικοποίηση (του server στον client) και το σύστημα ανταλλαγής κλειδιού.

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 διαχείριση 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 επιτρέπει με MiMT να χρησιμοποιεί ο server αλγόριθμο RSA με 512-bit modulus. Όπως έχουμε γράψει, τέτοιου μήκους modulus παραγοντοποιούνται εύκολα.

Handshake Protocol	ChangeCipher Protocol	Alert Protocol	Http
Record Protocol			
TCP			

Σχήμα 15.1: Δομή πρωτοκόλλου TLS.

Συνεχίζουμε με την περιγραφή του πρωτοκόλλου. Το SSL/TLS¹⁴⁹ 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

¹⁴⁸<https://datatracker.ietf.org/doc/html/rfc4492#section-2>

¹⁴⁹ver 1.2. : <https://www.rfc-editor.org/rfc/rfc5246>, ver. 1.3 :<https://datatracker.ietf.org/doc/html/rfc8446>

εγκαθίδρυση ενός κοινού κλειδιού που θα χρησιμοποιηθεί από το record layer. Επίσης, κάποιοι συμμετρικοί αλγόριθμοι κρυπτογράφησης χρησιμοποιούν IV, και αυτά θα παραχθούν κατά τη διάρκεια αυτού του πρωτοκόλλου, καθώς και τα κλειδιά που θα χρησιμοποιηθούν στις MAC. Συνήθως το SSL/TLS αυθεντικοποιεί τον εξυπηρετητή (server), σπάνια γίνεται αυθεντικοποίηση του client. Κατά τη διάρκεια αυτού του πρωτοκόλλου γίνονται 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**, έστω **client_nonce**, στον Server.
2. Ο Server θα στείλει ένα **nonce**, έστω **server_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 κάνοντας χρήση του Key Derivation Protocol, θα σχηματίσει το **master_secret_key: ms** από το pms και τα nonces: **client_nonce**, **server_nonce**
8. Ο Client κάνοντας επίσης χρήση του Key Derivation Protocol, θα σχηματίσει το **master_secret_key: ms**¹⁵⁰.
9. Ο Server θα στείλει **ServerFin=PRF(ms, transcript)** (ως PRF χρησιμοποιούμε μια HMAC).
10. Ο Client θα επαληθεύσει το **ServerFin** (αφού έχει υπολογίσει το ms).

Αποφασίζεται από τον client ποια μέθοδος θα χρησιμοποιηθεί, όταν στέλνει ClientHello ενώ ο server επιβεβαιώνει την επιλογή του client και απαντά με ServerHello. Άλλες επιλογές, αντί του RSA-PKCS ver.2, είναι:

- ο 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} .
- ο Ephemeral Diffie-Hellman. Ο server και ο client ανταλλάσσουν τις παραμέτρους (η ομάδα και ο γεννήτορας αποφασίζονται από τον server). Η υπογραφή των παραμέτρων γίνεται αποκλειστικά από τον server. Αυτή η μέθοδος ανταλλαγής κλειδιού προσφέρει forward secrecy και πρέπει να προτιμηθεί από τις υπόλοιπες. Για παράδειγμα, αν χρησιμοποιούσαμε ECDHE_RSA, στο βήμα 2 ο server θα στείλει το εφήμερο κλειδί του υπογεγραμμένο με το ιδιωτικό κλειδί RSA που αντιστοιχεί στο δημόσιο κλειδί του πιστοποιητικού του. Στο πρωτόκολλο SSL/TLS ver. 1.3 χρησιμοποιείται το σύστημα ECDHE by default.
- ο Anonymous Diffie-Hellman. Όπως προηγουμένως, αλλά δεν γίνεται υπογραφή (man in the middle attack).

Το **pre_master_secret** δεν είναι το τελικό κλειδί που θα χρησιμοποιηθεί.

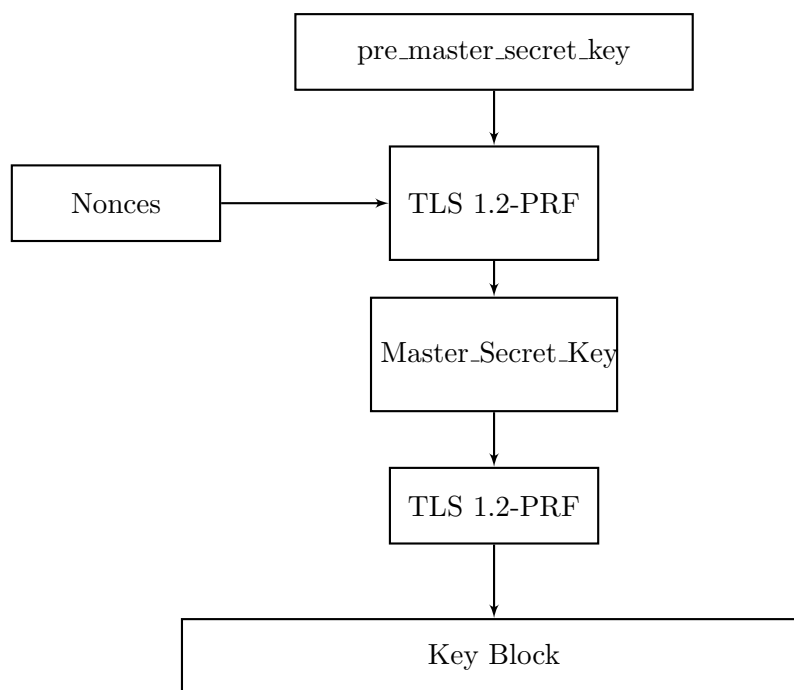
Υπάρχουν και πιστοποιητικά που παράγουν κλειδιά για τον αλγόριθμο υπογραφής ECDSA και έτσι, αντί ο server να υπογράφει με τον RSA η υπογραφή γίνεται με τον ECDSA.

Key derivation Protocol. Το τελικό κλειδί θα προκύψει από το **pre_master_secret_key** με τη βοήθεια μιας PRF που συνήθως είναι η HMAC-SHA256. Τα nonces και η PRF έχουν ανταλλαχθεί κατά τη διάρκεια του πρωτοκόλλου χειραψίας. Από το **master_secret** θα προκύψει το key block από το οποίο το επόμενο πρωτόκολλο record layer θα δανειστεί τα Mac-keys, encryption keys καθώς και τα IV.

- Πρωτόκολλο καταγραφής (record layer).

Παρέχει τις υπηρεσίες, της Εμπιστευτικότητας με χρήση ενός συμμετρικού αλγορίθμου, της Ακεραιότητας δεδομένων με χρήση MAC, της Ακεραιότητας της πηγής των δεδομένων, της προστασίας από replay attacks με χρήση ακολουθίας που προστατεύεται από MAC, και προαιρετικά μας δίνει τη δυνατότητα συμπίεσης των δεδομένων. Στο διάγραμμα 15.3 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

¹⁵⁰ η TLS 1.2 PRF είναι μια HMAC. Ισχύει, $master_secret = HMAC(ms, client_nonce || server_nonce)$



Σχήμα 15.2: TLS- Key Derivation Protocol.

στα δεδομένα (payload). Προστίθεται το Mac-tag και το pad, αν χρειάζεται, προστίθεται. Κατόπιν, γίνεται η κρυπτογράφηση με έναν 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 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 αποσυμπίσει (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.

Παρατήρηση 15.1.1. Στο openssl για να κρυπτογραφήσουμε (αντ. αποκρυπτογραφήσουμε) π.χ. με AES-CBC το αρχείο foo με τον κωδικό password:

```

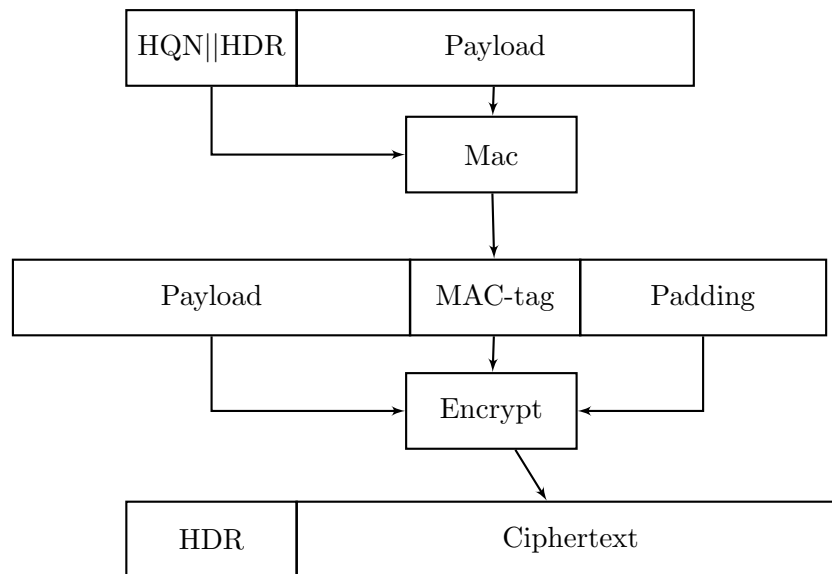
1 $openssl enc -aes-256-cbc -pbkdf2 -iter 20000 -in foo -out foo.enc -k password
2 $openssl enc -d -aes-256-cbc -pbkdf2 -iter 20000 -in foo.enc -out foo.out # decryption

```

Άσκηση 15.1 Εξηγήστε γιατί ένα σύστημα που χρησιμοποιεί RSA για αυθεντικοποίηση και για ανταλλαγή κλειδιού δεν προσφέρει forward secrecy.

Άσκηση 15.2 Περιγράψτε τις βασικές διαφορές των SSL/TLS ver. 1.2 και ver. 1.3.

Άσκηση 15.3 Τι ακριβώς είναι η επέκταση heartbeat και τι σχέση έχει με το heartbleed bug;



Σχήμα 15.3: TLS-record layer protocol.

15.2 GNU PG

*the way to make people
trustworthy is to trust them.*
– Ernest Hemingway
Selected Letters 1917–1961

Το GNU PG ή GPG¹⁵¹ 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 ενδιάμεσου (man in the middle). Στο PGP αυτή η αυθεντικοποίηση επιτυγχάνεται με τον κύκλο εμπιστοσύνης (web of trust). Υποθέτουμε ότι η εμπιστοσύνη έχει τη 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 ζευγάρι (PK_A, SK_A) και (PK_B, SK_B) είναι ένα ζεύγος δημόσιου και ιδιωτικού κλειδιού των 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 από τον αποστολέα.

¹⁵¹GPG: GNU Privacy Guard (<https://gnupg.org/>)

Αλγόριθμος 15.2.1. : PGP (για τον αποστολέα)

Είσοδος. Ένα μήνυμα m

Έξοδος. Αυθεντικοποιημένη κρυπτογράφηση του m από τον αποστολέα

Αυθεντικοποίηση

1 $H \leftarrow \text{hash}(m)$

2 $S \leftarrow \text{RSA}_{SK_A}(H)$

3 $m' \leftarrow [m, S]$

Εμπιστευτικότητα

4 $c' \leftarrow E_s(K_s, m')$

5 $K' \leftarrow \text{RSA}_{PK_B}(K_s)$

6 $c \leftarrow [c', K']$

7 Ο Α στέλνει στον Β το c

Αλγόριθμος 15.2.2. : PGP (για τον παραλήπτη)

Είσοδος. Ένα μήνυμα c

Έξοδος. Αποκρυπτογράφηση του c από τον παραλήπτη

1 $K_s \leftarrow \text{RSA}_{SK_B}(c[2])$ (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 $H' \leftarrow \text{Hash}(m'[1])$

4 **if** $H' = \text{RSA}_{PK_A}(m'[2])$ **then**
 return True, $m'[1]$ (εδώ γίνεται επαλήθευση της ψηφιακής υπογραφής και σε περίπτωση επιτυχίας τυπώνουμε το μήνυμα m)

else
 return FAIL

end

end

Στα βήματα 1-3 του αλγορίθμου 15.2.1, γίνεται η αυθεντικοποίηση του μηνύματος, δηλαδή το μήνυμα υπογράφεται με τον αλγόριθμο RSA , αφού προηγουμένως 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.

Παρατήρηση 15.2.1. Όπως το SSL/TLS, έτσι και το PGP χρησιμοποιεί το σχήμα *first Sign then Encrypt*.

Παρατήρηση 15.2.2. Μπορούμε να χρησιμοποιήσουμε το PGP, για να κρυπτογραφήσουμε ένα κείμενο με έναν συμμετρικό αλγόριθμο. Για να κρυπτογραφήσουμε, για παράδειγμα, το αρχείο `foo.pdf` με τον συμμετρικό αλγόριθμο `blowfish` (με CFB mode of operation¹⁵²):

```
1 $gpg -c --armor --no-symkey-cache --cipher-algo blowfish foo.pdf
```

Κατά τη διαδικασία αυτή θα σας ζητηθεί και κωδικός. Για να αποκρυπτογραφήσουμε:

```
1 $gpg foo.pdf
```

Οι αλγόριθμοι που υποστηρίζονται από το `gpg`:

```
1 $gpg --version
2 ...
3 Cipher: IDEA, 3DES, CAST5, BLOWFISH, AES, AES192, AES256, TWOFISH,
4         CAMELLIA128, CAMELLIA192, CAMELLIA256
```

Επίσης, το `gpg` μπορεί να υπολογίσει hashes, για παράδειγμα ο παρακάτω κώδικας υπολογίζει την hash `ripemd160` του κειμένου `AAA`:

```
1 $echo -n AAA | gpg --print-md ripemd160
2 E4E1 30AC C1D2 A5A6 3C17 EFB1 EEDB D02B E284 43D
```

¹⁵²full-block feedback

Οι συναρτήσεις κατακερματισμού που υποστηρίζονται είναι:

```
1 $gpg --version
2 ...
3 Hash: SHA1, RIPEMD160, SHA256, SHA384, SHA512, SHA224
```

Παρατήρηση 15.2.3. Μια επικίνδυνη επίθεση βρέθηκε στο PGP (καθώς και σε ένα άλλο πρωτόκολλο για emails, το S/MIME¹⁵³), όταν χρησιμοποιούμε email clients, όπως είναι ο Thunderbird, Apple email, Outlook 2017, Win 10 mail κ.ά. Η επίθεση αυτή έχει το όνομα eFail¹⁵⁴. Δεν χτυπάει το πρωτόκολλο PGP, αλλά μια ευπάθεια των email clients. Ο πιο απλός τρόπος να προστατευτούμε από αυτήν την επίθεση είναι να μην χρησιμοποιούμε email clients για την αποκρυπτογράφηση (π.χ. μπορούμε να χρησιμοποιούμε την κονσόλα του λειτουργικού μας συστήματος).

Παρατήρηση 15.2.4. 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 διαχείρισης των πιστοποιητικών βασίζεται στο web of trust. Το πρόγραμμα ανοιχτού κώδικα GNU/GPG υλοποιεί και τα δύο πρωτόκολλα, PGP και S/MIME.

Άσκηση 15.4 Στείλτε ένα κρυπτογραφημένο μήνυμα στον συγγραφέα (το δημόσιο κλειδί του έχει αναγνωριστικό **0xEB1185F82713D6DF**). Από το δημόσιο κλειδί το αναγνωριστικό προκύπτει (σε bash) αν δώσουμε τον παρακάτω κώδικα

```
1 $gpg --list-packets pk.asc | awk '/keyid:/{ print $2 }'
```

¹⁵³Secure/Multipurpose Internet Mail Extensions

¹⁵⁴<https://efail.de>

Βιβλιογραφία

- [1] Du, Wenliang. *Computer Security: A Hands-on Approach*. Independently published, 2019.
- [2] Stallings, W. *Network Security Essentials: Applications and Standards*. 6th Edition, Pearson, 2017.
- [3] Stallings, W. *Cryptography and Network Security: Principles and Practice*. Fifth Edition, Pearson, 2011.
- [4] Stevens, W. Richard. *TCP/IP Illustrated, Vol. 1: The Protocols*. Addison-Wesley Professional, 2011.
- [5] Tittel, E.; Johnson, D. *Guide to Networking Essentials*. 2nd Edition, Certification Insider Press, 1998.
- [6] Weidman, Georgia. *Penetration Testing: A Hands-On Introduction to Hacking*. No starch press, 2014.
- [7] Χειλάς, Κ.; Βακαλούδης, Α.; Πολίτης, Α. *Εργαστηριακές ασκήσεις δικτύων Η/Υ*, Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. <http://dx.doi.org/10.57713/kallipos-826>, 2015.
- [8] Πομπόρτσας, Α.; Παπαδημητρίου, Γ. *Ασφάλεια δικτύων υπολογιστών*. Εκδ. Τζιόλας, 2003.

Το παρόν σύγγραμμα χρηματοδοτήθηκε από το Πρόγραμμα
Δημοσίων Επενδύσεων του Υπουργείου Παιδείας.