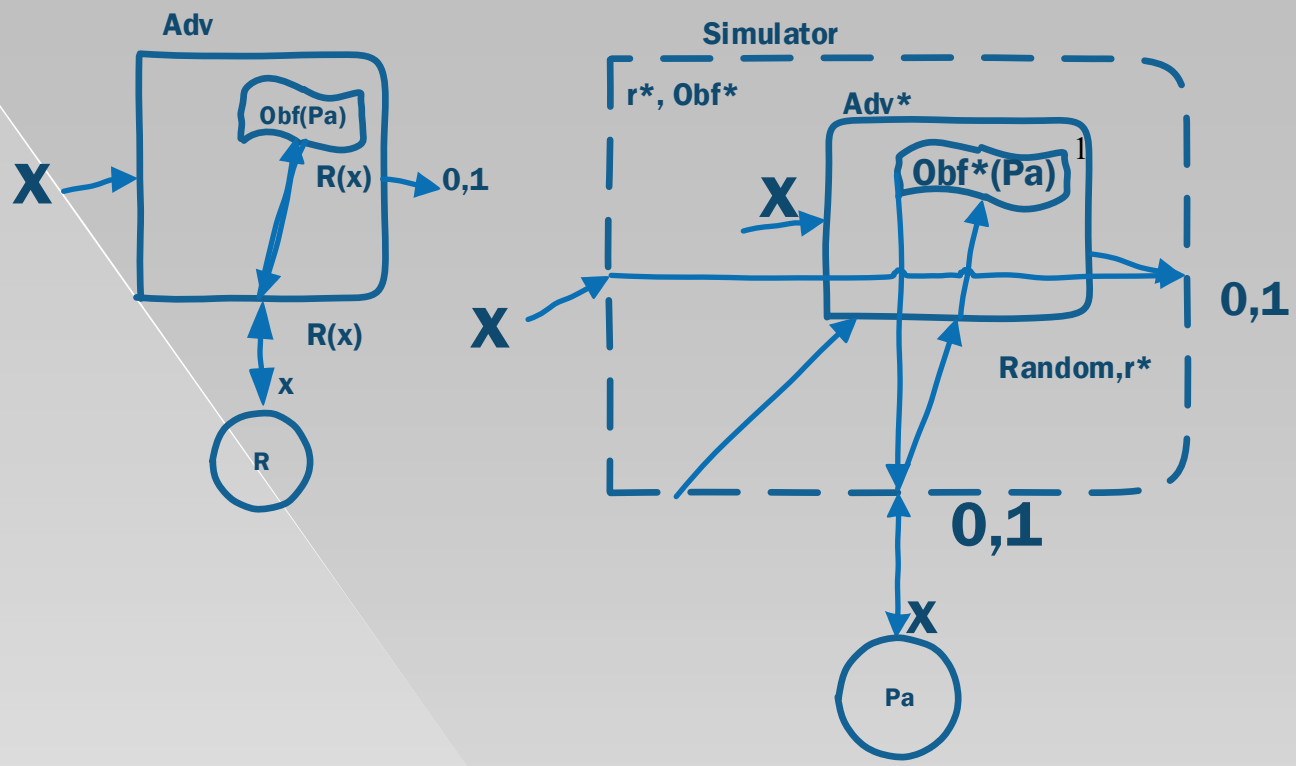


ΥΠΟΛΟΓΙΣΤΙΚΗ ΚΡΥΠΤΟΓΡΑΦΙΑ

ΕΥΣΤΑΘΙΟΣ ΖΑΧΟΣ
ΑΡΙΣΤΕΙΔΗΣ ΠΑΓΟΥΡΤΖΗΣ
ΠΑΝΑΓΙΩΤΗΣ ΓΡΟΝΤΑΣ

Ελληνικά Ακαδημαϊκά Ηλεκτρονικά Συγγράμματα και Βοηθήματα
www.kallipos.gr

HEALLINK
Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

ΕΣΠΑ 2007-2013
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Υπολογιστική Κρυπτογραφία

Συγγραφείς
Ευστάθιος Ζάχος
Αριστείδης Παγουρτζής
Παναγιώτης Γροντάς

Κριτικός αναγνώστης
Δημήτριος Πουλάκης

ΣΥΝΔΕΣΜΟΣ ΕΛΛΗΝΙΚΩΝ ΑΚΑΔΗΜΑΪΚΩΝ ΒΙΒΛΙΟΘΗΚΩΝ
Εθνικό Μετσόβιο Πολυτεχνείο Ηρώων Πολυτεχνείου 9, 15780 Ζωγράφου
www.kallipos.gr
ISBN: 978-960-603-276-9

Πίνακας Περιεχομένων

Πίνακας Συντομογραφιών

AES Advanced Encryption Standard.

AKS Agrawal Kayal Saxena Primality Test.

BDDH Bilinear Decisional Diffie Hellman Problem.

CBC Cipher Block Chaining Mode.

CCA Chosen Ciphertext Attack.

CDH Computational Diffie Hellman Problem.

CFB Cipher BFeedBack Mode.

CO Ciphertext Only Attack.

CPA Chosen Plaintext Attack.

DDH Decisional Diffie Hellman Problem.

DES Data Encryption Standard.

DLOG Discrete Logarithm Problem.

DSS Digital Signature Standard.

ECB Electronic Code Book.

FSR FeedBack Shift Register.

IBE Identity Based Encryption.

IC Index of Coincidence.

IND-CCA Indistinguishability under Chosen Ciphertext Attack.

IND-CPA Indistinguishability under Chosen Plaintext Attack.

KPA Known Plaintext Attack.

LFSR Linear FeedBack Shift Register.

MITM Meet in The Middle.

MPC Secure Multi Party Computation.

NTM Non-Deterministic Turing Machine.

OAEP Optimal Asymmetric Encryption Padding.

OFB Output FeedBack Mode.

OT Oblivious Transfer.

OTP One-Time Pad.

PPT Probabilistic Polynomial Time.

RSA Κρυπτοσύστημα των Rivest Shamir Adleman.

RSA-FDH RSA-Full Domain Hash.

SFE Secure Function Evaluation.

TM Turing Machine.

Tor The Onion Router.

TTP Trusted Third Party.

UC Universal Composability Framework.

ΜΚΔ Μέγιστος Κοινός Διαιρέτης.

Πρόλογος

Είναι πλέον αναμφισβήτητο γεγονός ότι η κρυπτογραφία και οι κάθε είδους εφαρμογές της παίζουν κομβικό ρόλο στη σύγχρονη τεχνολογία, ειδικά στους τομείς της ασφαλούς επικοινωνίας, της ασφαλούς πρόσβασης σε συστήματα και υπηρεσίες, της ανάκτησης και διαχείρισης ευαίσθητων δεδομένων, των ηλεκτρονικών συναλλαγών, των ηλεκτρονικών ψηφοφοριών, και των στρατιωτικών εφαρμογών. Ιδιαίτερα κατά τον 21ο αιώνα, η ραγδαία άνθηση των τηλεπικοινωνιών (με κυριότερο εκφραστή αυτής το διαδίκτυο) έχει καταστήσει την κρυπτογραφία αναπόσπαστο κομμάτι των τεχνολογικών εξελίξεων και αντικείμενο έντονης ερευνητικής δραστηριότητας, η οποία την έχει μετατρέψει από μορφή τέχνης σε επιστήμη, με αυστηρούς ορισμούς και αποδείξεις.

Στο σύγγραμμα αυτό επιχειρούμε να δώσουμε στον αναγνώστη τις απαραίτητες γνώσεις ώστε να κατανοήσει τις βασικές αρχές της σύγχρονης κρυπτογραφίας: τι είναι και πώς ορίζονται τα κρυπτοσυστήματα που χρησιμοποιούνται ευρύτατα, πώς επιχειρηματολογούμε για την ασφάλειά τους, ποια μαθηματικά και υπολογιστικά προβλήματα κρύβονται πίσω τους, και ποιες είναι οι προκλήσεις που αντιμετωπίζει σήμερα η κρυπτογραφική κοινότητα. Επίσης, εστιάζουμε στη σχέση κρυπτογραφίας και υπολογιστικής πολυπλοκότητας, καθώς είναι κοινός τόπος ότι τα σημερινά κρυπτοσυστήματα δεν μπορεί να είναι απόλυτα ασφαλή, αλλά θέλουμε να είναι *υπολογιστικά ασφαλή*, δηλαδή να έχουμε αποδείξεις ή έστω ισχυρές ενδείξεις ότι κανείς δεν μπορεί να τα σπάσει αρκετά γρήγορα ώστε αυτό να έχει κάποια πρακτική επίπτωση.

Στο βιβλίο αυτό παρουσιάζονται οι βασικές έννοιες, μέθοδοι και εφαρμογές της κρυπτογραφίας: κλασική και σύγχρονη συμμετρική κρυπτογραφία, κρυπτογραφία δημοσίου κλειδιού, ψηφιακές υπογραφές, συναρτήσεις σύνοψης, διανομή κλειδιών, θεωρία αριθμών, θεωρία υπολογιστικής πολυπλοκότητας, αλγόριθμοι, αποδείξεις ασφάλειας, κρυπτογραφικά πρωτόκολλα, αποδείξεις μηδενικής γνώσης. Γίνεται ακόμη εκτεταμένη αναφορά σε σύγχρονα θέματα και εφαρμογές: ηλεκτρονικές ψηφοφορίες, ψηφιακό χρήμα, κρυπτονομίσματα, ελλειπτικές καμπύλες, ανωνυμία, συσκότιση κώδικα, κβαντική κρυπτογραφία, ομομορφική κρυπτογραφία.

Το περιεχόμενο του συγγράμματος στηρίχθηκε σε διδακτικές σημειώσεις και υλικό που χρησιμοποιείται στο μάθημα "Κρυπτογραφία" (προγενέστερος τίτλος: "Στοιχεία Θεωρίας Αριθμών και Εφαρμογές στην Κρυπτογραφία") που διδάσκεται στους φοιτητές του ενάτου εξαμήνου της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Ηλεκτρονικών Υπολογιστών του Εθνικού Μετσοβίου Πολυτεχνείου. Το μάθημα διδάσκεται επίσης στη Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών του Ε.Μ.Π. και ως μεταπτυχιακό μάθημα στο διαπανεπιστημιακό Μεταπτυχιακό Πρόγραμμα Λογικής και Αλγορίθμων (Μ.Π.Λ.Α.) με τον τίτλο "Κρυπτογραφία και Πολυπλοκότητα". Γι'αυτό το λόγο πιστεύουμε ότι το βιβλίο αυτό μπορεί να αξιοποιηθεί σε ένα προχωρημένο προπτυχιακό ή μεταπτυχιακό μάθημα, ενώ έχει δομηθεί έτσι ώστε ένα μέρος του, με κατάλληλη επιλογή, να μπορεί να χρησιμοποιηθεί σε εισαγωγικά μαθήματα κρυπτογραφίας και ασφάλειας.

Πολλοί συνάδελφοι και σπουδαστές συνέβαλαν στη συγγραφή και επιμέλεια των αρχικών σημειώσεων που οδήγησαν στο βιβλίο που κρατάτε στα χέρια σας. Θέλουμε να ευχαριστήσουμε ιδιαίτερα τους: Άγγελο Κιαγιά, Τάσο Βίγλα, Κώστα Λημνιώτη, Χρήστο Καπούτση, Βασίλη Ζήκα, Κωνσταντίνο Γεωργίου, Γιάννη Βέτσικα, Γιώργο Αμανατίδη, Δώρα Κάραλη, Παναγιώτη Καρρά, Πέτρο Πετρόπουλο, Αντώνη Καβαρνό, Άρη Τέντε, Όλγα Τσιαντούλη, Ελένη Μπακάλη, και Γιώργο Ζηρδέλη. Θερμές ευχαριστίες ανήκουν στα μέλη των οικογενειών μας, για την αγάπη τους, την υπομονή τους και τη στήριξη που μας παρείχαν σε όλο το διάστημα της συγγραφής του βιβλίου, αλλά και πριν και μετά από αυτό.

Στάθης Ζάχος – Άρης Παγουρτζής – Παναγιώτης Γροντάς, 2015

Κεφάλαιο 1

Εισαγωγή

1.1 Εισαγωγή

Η κρυπτολογία, ως ο κλάδος που ασχολείται με ζητήματα ασφάλειας των επικοινωνιών, έχει μία πλούσια ιστορία χιλιάδων ετών, όσων δηλαδή και οι διάφοροι τρόποι επικοινωνίας: ξεκινάει με την εμφάνιση πρωτόλειων μορφών κρυπτογράφησης, που βασίζονται σε απλές αντικαταστάσεις των συμβόλων του μεταδιδόμενου μηνύματος, και συνεχίζεται μέχρι σήμερα όπου έχει αναπτυχθεί μια πληθώρα πολύπλοκων αλγορίθμων κρυπτογράφησης αλλά και σύνθετων πρωτοκόλλων που στηρίζονται στην απόκρυψη πληροφορίας. Ιδιαίτερα στις τελευταίες δεκαετίες, η ραγδαία άνθηση των τηλεπικοινωνιών (με κυριότερο εκφραστή αυτής το διαδίκτυο) έχει καταστήσει τη διασφάλιση του απορρήτου των επικοινωνιών απόλυτη ανάγκη, φέρνοντας την κρυπτολογία στο επίκεντρο των τεχνολογικών εξελίξεων. Η κρυπτολογία αποτελεί πλέον αντικείμενο έντονης ερευνητικής δραστηριότητας, η οποία την έχει μετατρέψει από μορφή τέχνης σε επιστήμη, με αυστηρούς ορισμούς και αποδείξεις.

Τυπικά με τον όρο ‘κρυπτολογία’ αναφερόμαστε τόσο στην κρυπτογραφία όσο και στην κρυπτανάλυση: η κρυπτογραφία ασχολείται με τον σχεδιασμό κρυπτοσυστημάτων, ενώ η κρυπτανάλυση μελετά το σπάσιμό τους. Καταχρηστικά, ο όρος ‘κρυπτολογία’ έχει “απορροφηθεί” από τον όρο ‘κρυπτογραφία’.

1.1.1 Ιστορική Αναδρομή

Από την Αρχαιότητα έως την Αναγέννηση

Στην αρχαία Ελλάδα, οι έφοροι της Σπάρτης επικοινωνούσαν με τους στρατηγούς χρησιμοποιώντας μακριές και στενές κορδέλες τις οποίες τύλιγαν γύρω από μια

σκυτάλη (κύλινδρο) και μετά έγραφαν το μήνυμα κατά μήκος της σκυτάλης. Για να διαβάσει κάποιος το μήνυμα, έπρεπε να έχει μια παρόμοια σκυτάλη με αυτή που είχε χρησιμοποιηθεί από τον δημιουργό του και να τυλίξει την κορδέλα γύρω από τη σκυτάλη με τον ίδιο τρόπο. Αυτό το σύστημα (διπλής κατεύθυνσης) κρυπτογραφίας είναι ένα κλασικό σύστημα με ένα κλειδί (τη σκυτάλη).

Ο Ιούλιος Καίσαρας επικοινωνούσε με τους συνεργάτες του αντικαθιστώντας κάθε γράμμα με ένα άλλο, το οποίο προέκυπτε με ολίσθηση κατά k βήματα στο αλφάβητο. Αυτό είναι ένα από τα πιο απλά, εύκολα αλλά και ανασφαλή κρυπτοσυστήματα που έχουν προταθεί.

Οι Βενετοί ήταν οι πρώτοι που χρησιμοποίησαν κρυπτογραφία συστηματικά, από το 13ο αιώνα, για διπλωματική αλληλογραφία. Οι πρώτες δημοσιεύσεις (στα λατινικά) περί κρυπτογραφίας φάνηκαν το 1500 (”Στεγανογραφία”) και το 1518 από τον αββά Ιωάννη Τριθέμιο και αργότερα δημοσιεύτηκε το ”Περί κρυπτικών συμβόλων και γραμμάτων” από τον J. B. Porta (1538 - 1615, Ιταλό φυσικό και μαθηματικό). Από τότε η κρυπτογραφία έγινε αντικείμενο ιδιαίτερου ενδιαφέροντος και απέκτησε εφαρμογές. Οι αλχημιστές χρησιμοποιούσαν σύμβολα για να κρυπτογραφήσουν τους τύπους τους, αλλά και πολλοί φιλόσοφοι ενδιαφέρθηκαν για την κρυπτογραφία: Ο Sir Francis Bacon (1561 - 1626) επινόησε ένα σύστημα κρυπτογράφησης όπου κάθε γράμμα αντικαθίσταται με μια λέξη πέντε γραμμάτων και ο Leonardo Da Vinci (1452 - 1519) χρησιμοποιούσε μια μέθοδο κρυπτογράφησης με καθρέπτη.

19ος και 20ος αιώνας

Ο Edgar Allan Poe (1809-1849) στο κλασικό διήγημα ”Το χρυσό έντομο” (”The Gold Bug”) που δημοσίευσε το 1843, εξηγεί τις βασικές αρχές παραβίασης των κωδίκων και υποστηρίζει την άποψη ότι ο ανθρώπινος νους μπορεί να σπάσει οποιοδήποτε κρυπτογραφημένο κείμενο που η ανθρώπινη ευρηματικότητα μπορεί να επινοήσει. Ακόμη περιγράφει ένα σύστημα με το οποίο κάθε κρυπτογραφημένο κείμενο που προέρχεται από μια ευρωπαϊκή γλώσσα μπορεί να αποκρυπτογραφηθεί, αν έχει κρυπτογραφηθεί με αντικατάσταση, μετρώντας τη συχνότητα των γραμμάτων της γλώσσας, τεχνική που πρώτοι συνέλαβαν οι Άραβες.

Ίσως από τα διασημότερα κρυπτογραφήματα, το *σημείωμα του Zimmerman* (*the Zimmerman Note*) ώθησε τις ΗΠΑ στον πρώτο παγκόσμιο πόλεμο. Όταν το κρυπτογράφημα αποκρυπτογραφήθηκε το 1917, οι Αμερικανοί έμαθαν ότι η Γερμανία είχε προσπαθήσει να πείσει το Μεξικό να μπει στον πόλεμο με το μέρος της, υποσχόμενη παραχωρήσεις εδαφών των ΗΠΑ στο Μεξικό.

Τον ίδιο περίπου καιρό, ο Gilbert S. Vernam της AT&T ανέπτυξε τον πρώτο πραγματικά άθραυστο κώδικα που ονομάστηκε βέβαια *κρυπτόγραμμα Vernam* (*The Vernam Cipher)*. Μια ξεχωριστή ιδιότητα αυτού του κώδικα είναι η απαίτηση για

ένα κλειδί με μήκος όσο και το μήνυμα που πρέπει να μεταδοθεί και το οποίο δεν επαναχρησιμοποιείται για την αποστολή άλλου μηνύματος (η κρυπτογράφηση Vernam είναι γνωστή επίσης και ως *κρυπτογράφηση με μπλοκάκι μιας χρήσης* (*one-time-pad*) από την πρακτική της προμήθειας κατασκόπων με το κείμενο-κλειδί γραμμένο σε ένα μπλοκάκι του οποίου κάθε κομμάτι χρησιμοποιείται μια φορά και μετά καταστρέφεται). Η ανακάλυψη του συστήματος αυτού δεν εκτιμήθηκε ιδιαίτερα εκείνη την εποχή, πιο πολύ επειδή δεν είχε αποδειχτεί ακόμη ότι είναι άθραυστος κώδικας και επειδή η απαίτηση για πολλά και μεγάλα κλειδιά την έκαναν μη πρακτική για γενική χρήση.

Α0ξίζει να αναφερθεί ότι το 1967, όταν ο στρατός της Βολιβίας συνέλαβε και εκτέλεσε τον επαναστάτη Che Guevara, βρήκαν στην κατοχή του ένα χαρτί που έδειχνε πως προετοίμαζε ένα μήνυμα για αποστολή στον Κουβανό πρόεδρο Fidel Castro. Ο Che Guevara χρησιμοποιούσε τον άσπαστο κώδικα του Vernam.

Εξαιτίας των μη πρακτικών απαιτήσεων της κρυπτογράφησης Vernam, άλλες (πιο αδύναμες) μέθοδοι συνέχισαν να χρησιμοποιούνται ευρέως. Έτσι, κατά το δεύτερο παγκόσμιο πόλεμο, οι Σύμμαχοι ήταν σε θέση να αποκρυπτογραφούν τα περισσότερα από τα μυστικά μηνύματα που στέλνονταν από τους Γερμανούς. Η εγγενής δυσκολία του σπασίματος των ολοένα και πιο περίπλοκων κρυπτογραφικών μεθόδων ήταν μάλιστα ένας από τους παράγοντες που προώθησε την ανάπτυξη των ηλεκτρονικών υπολογιστών.

Μηχανή Enigma και Alan Turing

Η περίφημη *Μηχανή-Αίνιγμα* (*Enigma*) που χρησιμοποιήθηκε από τους Γερμανούς κατά το δεύτερο παγκόσμιο πόλεμο για κρυπτογράφηση ραδιοτηλεπικοινωνιών ήταν ίσως το πλέον εξελιγμένο κρυπτοσύστημα της εποχής και πυροδότησε μια από τις πιο έντονες προσπάθειες αποκρυπτογράφησης στην ιστορία. Ο κώδικας Αίνιγμα θυμίζει έναν παλιότερο κώδικα (τύπου Vigenére) αλλά είναι πολύ πιο πολύπλοκος. Μια βασική ιδιότητα της μηχανής αυτής ήταν η αυτο-αντιστροφή: εάν το κωδικοποιημένο κείμενο δινόταν ως είσοδος στη μηχανή, τότε η έξοδος θα ήταν το αρχικό μήνυμα (αν φυσικά η μηχανή είχε την ίδια αρχική κατάσταση με τη μηχανή που είχε κάνει την κωδικοποίηση). Παρόλο που αυτό αποτελούσε τρομερή ευκολία για τους χειριστές της μηχανής, αποδείχτηκε ότι ήταν και μεγάλη αδυναμία του κώδικα Αίνιγμα. Πριν τον πόλεμο, η γαλλική αντικατασκοπεία είχε αποκτήσει αντίγραφα των εντολών της μηχανής-Αίνιγμα και έδωσε την πληροφορία αυτή στους Πολωνούς που υπέκλεπταν και ανέλυαν τις γερμανικές ράδιο-επικοινωνίες. Με τη βοήθεια των εντολών αυτών, οι Πολωνοί κρυπταναλυτές μπόρεσαν να συμπεράνουν τη συνδεσμολογία-καλωδίωση της μηχανής, οπότε έγινε δυνατό να διαβάζονται τα κρυπτογραφημένα κείμενα, αρκεί να είναι γνωστή η αρχική κατάσταση της μηχανής. Παρόλο που οι Βρετανοί τα έμαθαν όλα αυτά

από τους Πολωνούς, είχαν μικρή αξία γι' αυτούς επειδή οι Γερμανοί έκαναν κάποιες τροποποιήσεις στη μηχανή πριν τον πόλεμο.

Οι Βρετανοί συγκέντρωσαν μια ομάδα κρυπταναλυτών και μαθηματικών, με επικεφαλής τον Alan Turing, σε μια βικτωριανή έπαυλη στο Buckinghamshire που ονομαζόταν Bletchley Park. Χρησιμοποιώντας τις πληροφορίες των Πολωνών, η ομάδα βάσισε τις προσπάθειές της στη λεγόμενη μέθοδο πιθανής λέξης. Η μέθοδος αυτή βασίζεται στο γεγονός ότι σε κάποιες περιπτώσεις μια συγκεκριμένη ακολουθία συμβόλων σχεδόν σίγουρα αντιπροσωπεύει μια γνωστή λέξη. Μαντεύοντας σωστά μερικές από τις κρυπτογραφημένες λέξεις του κρυπτοκειμένου, μπορούσαν να καθορίζουν τη συνδεσμολογία της μηχανής, δοκιμάζοντας όλες τις πιθανές συνδεσμολογίες και προσδιορίζοντας ποια είχε ως αποτέλεσμα τα υποτιθέμενα ζευγάρια κρυπτογραφημένων-αποκρυπτογραφημένων λέξεων. Ο Turing αντιλήφθηκε ότι μόνο μια αυτόματη και σχετικά γρήγορη μηχανή θα μπορούσε να τα βγάλει πέρα με τις δοκιμές, όποτε και οδηγήθηκε στην κατασκευή ενός εξομοιωτή της μηχανής-Αίνιγμα με το όνομα Bombe. [1]

Σύγχρονη κρυπτογραφία

Η σημερινή μορφή των κρυπτογραφικών συστημάτων έχει καθοριστεί σε πολύ μεγάλο βαθμό από δύο, κεφαλαιώδους σημασίας για την κρυπτογραφία και τις επικοινωνίες γενικότερα, επιστημονικές εργασίες Kerchoffs, Shannon, που δημοσιεύτηκαν στα 1883 και 1949 αντίστοιχα. Στην πρώτη, ο Kerchoffs έθεσε τη βασική σχεδιαστική αρχή που έκτοτε διέπει κάθε κρυπτογραφικό σύστημα, σύμφωνα με την οποία η ασφάλεια ενός συστήματος πρέπει να έγκειται *μόνο* στη μυστικότητα του κλειδιού και να μην εξαρτάται από τη μυστικότητα του αλγορίθμου κρυπτογράφησης. Η δεύτερη εργασία ανήκει στο θεμελιωτή της Θεωρίας Πληροφορίας Claude Shannon. Στην εργασία αυτή η κρυπτογραφία μετατρέπεται σε αυστηρό επιστημονικό πεδίο, όπου ορίζεται η έννοια του κρυπτοσυστήματος και η απόλυτη ασφάλεια. Η εργασία αυτή του Shannon αποτέλεσε ισχυρή κινητήρια δύναμη για την ταχεία εξέλιξη της έρευνας στο χώρο της κρυπτογραφίας, η οποία έλαβε χώρα στο δεύτερο μισό του εικοστού αιώνα και συνεχίζεται μέχρι σήμερα. Όλοι οι σύγχρονοι κρυπτογραφικοί αλγόριθμοι σχεδιάζονται υπό το πρίσμα των εννοιών που εισήγαγε ο Shannon.

Σημαντική τομή επίσης στο χώρο της κρυπτογραφίας αποτέλεσε η εργασία των Diffie-Hellman το 1976 [2], όπου προτάθηκε μία επαναστατική τεχνική η οποία επιλύει το πρόβλημα της ανταλλαγής κλειδιού από απόσταση, χωρίς να απαιτείται άμεση επαφή, θέτοντας έτσι τις βάσεις για την *κρυπτογραφία δημοσίου κλειδιού*. Πράγματι, το 1977 οι Ronald L. Rivest, Adi Shamir και Leonard M. Adleman

[1] Η εκπληκτική αυτή επιτυχία του Turing και της ομάδας του είναι το θέμα της πρόσφατης αξιόλογης ταινίας "The Imitation Game" (2014).

(τότε στο MIT) πρότειναν ένα ιδιαίτερα επιτυχημένο (έως και σήμερα) κρυπτοσύστημα δημοσίου κλειδιού, γνωστό ως RSA [6]. Οι παραπάνω εργασίες καθώς και η δουλειά ερευνητών στην δεκαετία του 1980, όπως η Shafi Goldwasser, ο Silvio Micali κ.ά., οδήγησε στην Σύγχρονη Κρυπτογραφία, στην οποία κεντρικό ρόλο διαδραματίζει η έννοια της αποδείξιμης ασφάλειας και ο κλάδος της υπολογιστικής πολυπλοκότητας με τα οποία θα ασχοληθούμε εκτενώς στο τρέχον σύγγραμμα.

Εκτός από την παραπάνω αλλαγή, η σύγχρονη κρυπτογραφία έχει επεκτείνει σημαντικά το πεδίο εφαρμογής της πέρα από την ιδιωτική επικοινωνία. Οι μέθοδοί της επιτρέπουν τον έλεγχο της ακεραιότητας μηνυμάτων, την μη αποποίηση αποστολής ή λήψης τους, την χρονική τους σήμανση, και πλήθος άλλων ιδιοτήτων. Επιπλέον μπορεί να χρησιμοποιηθεί για την απόδειξη της ταυτότητας κάποιου χρήστη (αυθεντικοποίηση) και την παροχή εξουσιοδότησης για την πραγματοποίηση συγκεκριμένων λειτουργιών. Από την άλλη μπορεί να παρέχει ανωνυμία και δυνατότητα άρνησης σε περίπτωση εκβιασμού. Σε ακόμα πιο υψηλό επίπεδο η κρυπτογραφία δρα ως καταλύτης για την οικοδόμηση εμπιστοσύνης μεταξύ οντοτήτων με διαφορετικά και συχνά αντικρουόμενα συμφέροντα, τα οποία αλληλεπιδρούν από απόσταση, μια πολύ σημαντική λειτουργία στο σημερινό πολλαπλά συνδεδεμένο κόσμο.

21ος αιώνας

Κατά τον 21ο αιώνα, οι κρυπτογραφικές εφαρμογές, πρωτόκολλα και τεχνικές παίζουν κομβικό ρόλο στη σύγχρονη τεχνολογία, ειδικά στους τομείς της ασφαλούς επικοινωνίας, της ασφαλούς πρόσβασης, των ηλεκτρονικών ψηφοφοριών, της ανάκτησης και διαχείρισης ευαίσθητων δεδομένων, και των ηλεκτρονικών συναλλαγών, με πρόσφατη σημαντικότατη εξέλιξη την ανάπτυξη του κρυπτονομίσματος Bitcoin, και αρκετών ήδη διαδόχων του, με κυρίαρχο χαρακτηριστικό την απουσία κεντρικού ελέγχου.

Οι παραπάνω εφαρμογές, και πολλές άλλες που δεν αναφέρθηκαν, μπόρεσαν να πραγματοποιηθούν χάρη στην αλματώδη ανάπτυξη επαναστατικών ιδεών και αλγορίθμων, όπως η τέλεια μυστικότητα, η κρυπτογραφία δημοσίου κλειδιού, η ασφαλής ανταλλαγή κλειδιού από απόσταση, οι ψηφιακές υπογραφές, τα διαλογικά συστήματα αποδείξεων και οι αποδείξεις μηδενικής γνώσης, οι γεννήτριες ψευδοτυχαιότητας, η σύνθεση πρωτοκόλλων, οι συναρτήσεις σύνοψης χωρίς συγκρούσεις, η υπολογιστική πολυπλοκότητα και πολλά άλλα. Κοινό χαρακτηριστικό των παραπάνω μεθόδων είναι ότι η ασφάλειά τους εδράζεται, όλο και περισσότερο, σε αυστηρές μαθηματικές αποδείξεις. Για παράδειγμα, είμαστε πλέον σε θέση να διενεργούμε ηλεκτρονικές ψηφοφορίες που παρέχουν αποδείξεις ορθότητας για διάφορες φάσεις της λειτουργίας τους. Ή, να διενεργούμε συναλλαγές χωρίς κεντρική

αρχή, μέσω του Bitcoin ή άλλων κρυπτονομισμάτων, με απόδειξη εγκυρότητας για κάθε συναλλαγή, που επικυρώνεται συλλογικά! Τα κρυπτονομίσματα είναι μια επανάσταση σε εξέλιξη, ανοίγοντας δρόμους για αποκεντρωμένη και αποδεδειγμένα ασφαλή ψηφιακή υλοποίηση λειτουργιών που μέχρι σήμερα απαιτούσαν την ύπαρξη κάποιας αρχής, όπως για παράδειγμα τη σύναψη συμβολαίων, αλλά και ηλεκτρονικών ψηφοφοριών.

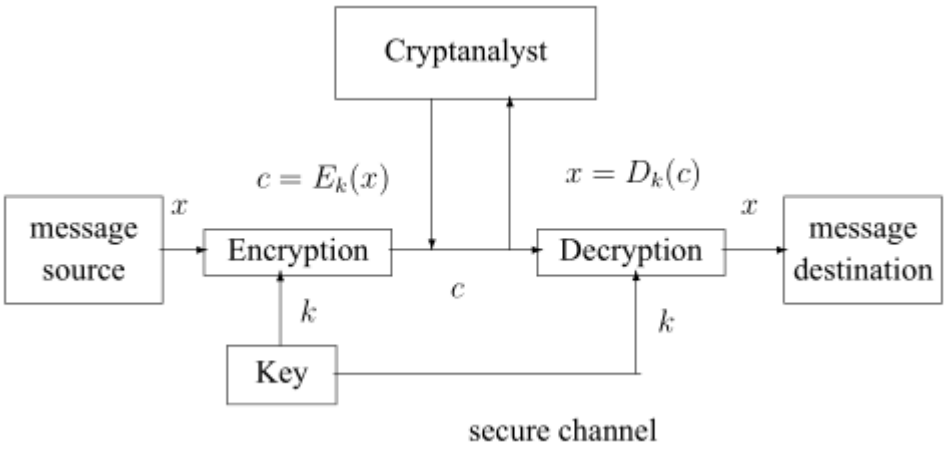
Τα μαθηματικά γίνονται για άλλη μια φορά επίκαιρα, βοηθώντας στην εμπέδωση εμπιστοσύνης σε κρίσιμες λειτουργίες, και μέσω αυτής στο άνοιγμα της κρυπτογραφίας στο πλατύ κοινό. Τα περισσότερα κρυπτοσυστήματα είναι πλέον εντελώς ανοιχτά (ο τρόπος λειτουργίας είναι γνωστός), και η ασφάλειά τους βασίζεται αποκλειστικά σε αριθμητικούς αλγορίθμους που μας επιτρέπουν να εκτελούμε αποδοτικά πράξεις με αριθμούς χιλιάδων ψηφίων, ώστε η υπολογιστική δυσκολία (πολυπλοκότητα) των αντίστροφων πράξεων να είναι τεράστια. Η κρυπτογραφία έχει φύγει οριστικά από τα στεγανά των μυστικών υπηρεσιών και τη στρατιωτική χρήση και είναι έτοιμη να προσφέρει ακόμη περισσότερο τις υπηρεσίες της στο σύνολο της ανθρωπότητας πλέον, προάγοντας τη δημοκρατία, τον σεβασμό της ιδιωτικής ζωής, και τελικά την ενεργό και ισότιμη συμμετοχή όλων στο οικονομικό, πολιτικό, και κοινωνικό γίγνεσθαι.

1.2 Κλασικά Συστήματα

Στην κλασική κρυπτογραφία, οι αλγόριθμοι κρυπτογράφησης και αποκρυπτογράφησης είναι γνωστοί σε όλους, και το ίδιο κλειδί χρησιμοποιείται και για τις δύο κατευθύνσεις (κρυπτογράφηση-αποκρυπτογράφηση). Με άλλα λόγια, στα κλασικά συστήματα, η αποκρυπτογράφηση είναι εύκολη αν το κλειδί κρυπτογράφησης είναι γνωστό. Αντίθετα, στην κρυπτογραφία δημοσίου κλειδιού το κλειδί κρυπτογράφησης k μπορεί με ασφάλεια να δημοσιοποιηθεί χωρίς να αποκαλυφθεί το κλειδί αποκρυπτογράφησης k'. Για αυτό το λόγο τα κλασικά συστήματα αναφέρονται επίσης και ως *συμμετρικά* ή *διπλής κατεύθυνσης συστήματα*, και τα συστήματα δημοσίου κλειδιού ως *μη-συμμετρικά* ή *μονής κατεύθυνσης* συστήματα (αυτό σημαίνει ότι η διαδικασία κρυπτογράφησης είναι μονής κατεύθυνσης - δεν μπορεί εύκολα να αντιστραφεί). Μία κεντρική ιδέα που διέπει τα συμμετρικά συστήματα είναι ότι οι συμμετέχοντες μπορούν να ανταλλάξουν το κλειδί με ασφάλεια (μέσω κάποιου διαφορετικού δίαυλου επικοινωνίας) κάτι που φυσικά δεν ισχύει στα ασύμμετρα.

Μια (πολύ παλιά) κατηγοριοποίηση των κλασικών κρυπτοσυστημάτων είναι σε συστήματα *αντικατάστασης (substitution)* και *μετάθεσης (permutation)* (ή *αναδιάταξης (transposition)*).

Στα συστήματα αντικατάστασης (substitution ciphers), τα γράμματα του αρχι-



Σχήμα 1.1: Συμβατικά (κλασικά) κρυπτοσυστήματα διπλής κατεύθυνσης.

κού κειμένου αντικαθίστανται από άλλα τα οποία διατηρούνται στην ίδια διάταξη όπως και τα πρωτότυπα τους στο αρχικό κείμενο. Αν οι αντικαταστάτες παραμένουν οι ίδιοι σε όλο το κείμενο (κάθε γράμμα του αρχικού κειμένου αναπαρίσταται πάντοτε από το ίδιο σύμβολο-αντικαταστάτη) τότε το σύστημα ονομάζεται *μονοαλφαβητικά*. Αν το αρχικό κείμενο είναι σε κάποια φυσική γλώσσα, η κρυπτανάλυση είναι πάντοτε εφικτή βασιζόμενη στη στατιστική κατανομή των γραμμάτων. Στα *πολυαλφαβητικά* συστήματα αντικατάστασης κάθε γράμμα του αρχικού κειμένου μπορεί να έχει πολλούς αντικαταστάτες και κάθε φορά χρησιμοποιείται διαφορετικός.

Στα συστήματα μετάθεσης (ή αναδιάταξης) τα γράμματα του αρχικού κειμένου αναδιατάσσονται. Αυτή η μέθοδος είναι υπερβολικά απλή, οπότε θα πρέπει να συνδυαστεί με κάποια άλλη ιδέα (μέθοδο παρεμβολής σκουπιδιών, …).

Μια άλλη κατηγοριοποίηση των κρυπτοσυστημάτων θα μπορούσε να είναι σε συστήματα αντικατάστασης *χωρίς συμφραζόμενα (context-free)* και σε συστήματα αντικατάστασης *με συμφραζόμενα (context-sensitive)* : Στα συστήματα χωρίς συμφραζόμενα κάθε γράμμα κωδικοποιείται ξεχωριστά ενώ σε εκείνα με συμφραζόμενα η κωδικοποίηση γίνεται ανά ομάδες (blocks).

1.2.1 Μονοαλφαβητικά Συστήματα Αντικατάστασης

Ένα κρυπτοσύστημα ονομάζεται μονοαλφαβητικά αν κάθε γράμμα του αρχικού κειμένου αναπαρίσταται πάντοτε από το ίδιο σύμβολο-αντικαταστάτη (κάθε εμφάνιση ενός συμβόλου του αρχικού κειμένου κρυπτογραφείται με τον ίδιο πάντα

αντικαταστάτη).

Το Κρυπτοσύστημα του ΚΑΙΣΑΡΑ

Το κρυπτοσύστημα του Καίσαρα είναι από τα πρώτα κρυπτογραφικά σχήματα που χρησιμοποιήθηκαν. Είναι επίσης πολύ απλό, και μπορεί κανείς να το σπάσει πολύ εύκολα. Το σύστημα του Καίσαρα βασίζεται σε αντικαταστάσεις με ολίσθηση κατά k θέσεις, δηλαδή κάθε γράμμα αντικαθίσταται με άλλο, προχωρώντας k θέσεις στο αλφάβητο modulo το μέγεθος του αλφαβήτου. $(k = 1, \ldots, 25)$.
Δηλαδή για $k = 3$ έχουμε:

Τα γράμματα του αρχικού κειμένου:	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Οι αντικαταστάτες του κρυπτοκειμένου:	D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

Έτσι το αρχικό κείμενο I LOVE MATH κρυπτογραφείται ως LORYH PDWK. Η μέθοδος κρυπτογράφησης E_k είναι ολίσθηση μπροστά κατά k βήματα στο αλφάβητο, και η μέθοδος αποκρυπτογράφησης είναι ολίσθηση πίσω κατά k βήματα στο αλφάβητο. Παρακάτω δίνονται κάποιες προφανείς ιδιότητες των E_k και D_k:

$$E_i D_j = D_j E_i \text{ (αντιμεταθετική ιδιότητα)}$$
$$D_k = E_{26-k}\,,$$
$$D_k E_k = D_0 = E_0 = D_{26} = E_{26}.$$

Αν το αρχικό κείμενο ανήκει σε μια γνωστή φυσική γλώσσα, η κρυπτανάλυση αυτού του συστήματος είναι πολύ εύκολη: από τη στιγμή που ο συνολικός αριθμός των δυνατών κλειδιών είναι αρκετά μικρός (25) μπορεί κανείς απλά να τα δοκιμάσει όλα σε ένα μικρό μέρος του κρυπτοκειμένου και να δει ποιο από όλα οδηγεί σε αρχικό κείμενο που έχει νόημα. Επίσης, όπως και όλα τα μονοαλφαβητικά συστήματα αντικατάστασης, το σύστημα του Καίσαρα μπορεί να το σπάσει κανείς υπολογίζοντας τις συχνότητες εμφάνισης των γραμμάτων.

1.2.2 Πολυαλφαβητικά Συστήματα Αντικατάστασης

Στα πολυαλφαβητικά κρυπτοσυστήματα αντικατάστασης, ένα γράμμα δεν αντικαθίσταται από το ίδιο σύμβολο παντού στο κείμενο: η χρήση των αντικαταστατών ποικίλει στα διάφορα μέρη του απλού κειμένου. Για παράδειγμα στη γερμανική μηχανή Αίνιγμα μετά από κάθε γράμμα του κειμένου, τα γρανάζια γυρίζουν, δίνοντας ένα νέο πρότυπο κρυπτογράφησης.

Ο κώδικας αντικατάστασης 2-γραμμάτων PLAYFAIR

Το κρυπτοσύστημα Playfair βασίζεται στο Αγγλικό αλφάβητο. Λειτουργεί ως εξής: Διατάσσουμε τα γράμματα του Αγγλικού αλφαβήτου, παραλείποντας το J (το J και το I θεωρούνται σαν το ίδιο γράμμα) σε έναν πίνακα 5×5 (που λέγεται τετράγωνο Playfair), με κάποιο τυχαίο τρόπο. Συνήθως το τετράγωνο Playfair δημιουργείται βασισμένο σε μια λέξη-κλειδί (ή φράση): τα γράμματα της λέξης κλειδί (οι επαναλήψεις διαγράφονται) τοποθετούνται κατά σειρές, ακολουθούμενα από τα υπόλοιπα γράμματα σε αλφαβητική σειρά. (Οι λέξεις-κλειδιά χρησιμοποιούνται προκειμένου να γίνει η διαχείριση του κλειδιού ευκολότερη).

Για παράδειγμα, η λέξη-κλειδί PLAYFAIR δίνει τον παρακάτω πίνακα:

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

Χωρίζουμε το αρχικό κείμενο σε ομάδες που αποτελούνται από δύο γράμματα το καθένα (2-γράμματα). Δεν πρέπει κανένα block να περιέχει δύο φορές το ίδιο γράμμα: οποτεδήποτε συμβαίνει αυτό, παρεμβάλλουμε (μεταξύ των δύο γραμμάτων στο αρχικό κείμενο) ένα *μηδενικό χαρακτήρα* (συνήθως το Q). Επίσης το απλό κείμενο πρέπει να περιέχει άρτιο αριθμό γραμμάτων (διαφορετικά βάζουμε ένα μηδενικό χαρακτήρα στο τέλος του). Οι διαδικασίες κρυπτογράφησης και αποκρυπτογράφησης λειτουργούν στα 2-γράμματα όπως φαίνεται παρακάτω:

Κρυπτογράφηση:

Αν τα δύο γράμματα του block δεν είναι στην ίδια γραμμή ή στήλη του τετραγώνου Playfair, τότε κωδικοποιούμε χρησιμοποιώντας τις άλλες δύο γωνίες του ορθογωνίου που ορίζεται από τα δύο γράμματα του block. Για παράδειγμα, χρησιμοποιώντας το παραπάνω τετράγωνο Playfair, το 2-γραμμα AD ορίζει το ορθογώνιο AFBD και κρυπτογραφείται ως FB.

Αν τα δύο γράμματα του block είναι στην ίδια γραμμή (στήλη), επιλέγουμε (κυκλικά) τα γειτονικά γράμματα δεξιά (κάτω) για κάθε γράμμα του block. Έτσι, το RC κωδικοποιείται με το BD, το AW με το BA, το UV με το VW.

Με αυτή τη μέθοδο το αρχικό κείμενο MATHEMATICS χωρίζεται σε διγράμματα ως MA TH EM AT IC SQ και κρυπτογραφείται: HF QM GE FQ RD TS.

Αποκρυπτογράφηση:

Η μέθοδος αποκρυπτογράφησης είναι προφανής: αφού το PLAYFAIR είναι ένα κλασικό διπλής-κατεύθυνσης κρυπτοσύστημα η αποκρυπτογράφηση προκύπτει (εύκολα) από τη μέθοδο κρυπτογράφησης.

Η κρυπτανάλυση αυτού του συστήματος μπορεί επίσης να βασιστεί στη μέθοδο της μέτρησης συχνοτήτων (όπως και για όλα τα μονοαλφαβητικά συστήματα αντικατάστασης), αλλά η ανάλυση είναι πιο πολύπλοκη καθώς η κρυπτογράφηση βασίζεται σε 2-γράμματα, και όχι σε απλά γράμματα.

Τα πολυαλφαβητικά συστήματα παρέχουν μια πολύ καλή άμυνα εναντίον της μέτρησης συχνοτήτων διότι κάθε γράμμα δεν αναπαρίσταται με το ίδιο σύμβολο παντού στο απλό κείμενο.

Κρυπτοσύστημα VIGENÉRE

Το σύστημα του Vigenére (Blaise de Vigenére 1523-1596)) είναι από τα πιο παλιά και τα πιο γνωστά πολυαλφαβητικά κρυπτοσυστήματα (στην πραγματικότητα ο κώδικας Αίνιγμα και ο κώδικας Vernam είναι Vigenére συστήματα). Αρχικά αντιστοιχίζουμε σε κάθε γράμμα του αλφαβήτου έναν αριθμό $(A-0, B-1, \ldots, Z-25)$. Το VIGENÉRE μπορεί να θεωρηθεί ένα σύστημα του Καίσαρα στο οποίο το κλειδί αλλάζει από βήμα σε βήμα.

Το κλειδί στο σύστημα VIGENÉRE είναι ένα διάνυσμα rx χαρακτήρων $k = (k_0, k_1, \ldots, k_{r-1})$. Αυτό το διάνυσμα έχει μορφή μιας λέξη-κλειδί, και μπορεί να είναι οποιαδήποτε λέξη ή φράση (επαναλήψεις γραμμάτων επιτρέπονται). Ο αριθμός r λέγεται περίοδος του συστήματος.

Το αρχικό κείμενο διαιρείται σε blocks μεγέθους r και κάθε block κρυπτογραφείται χρησιμοποιώντας τη λέξη-κλειδί ως εξής: γράφουμε τη λέξη-κλειδί κάτω από το block του αρχικού κειμένου και κρυπτογραφούμε κάθε γράμμα του κειμένου χρησιμοποιώντας ένα σύστημα του Καίσαρα με το k να ισούται με τον αριθμό που αντιστοιχεί στο γράμμα της λέξης-κλειδί που είναι γραμμένη από κάτω. Με μαθηματικό συμβολισμό η αντικατάσταση VIGENÉRE για κάθε σύμβολο του αρχικού κειμένου ορίζεται ως:

$$c_i = E_K(x_i) = (x_i + k_{i \bmod r}) \bmod 26, 0 \leq i \leq n-1$$

Για την αποκρυπτογράφηση φυσικά έχουμε:

$$x_i = D_K(c_i) = (c_i - k_{i \bmod r}) \bmod 26, 0 \leq i \leq n-1$$

Οι διαδικασίες κρυπτογράφησης και αποκρυπτογράφησης μπορούν να εκτελεστούν και χωρίς υπολογιστή χρησιμοποιώντας έναν πίνακα παρόμοιο με τον παρακάτω, στον οποίο κάθε στήλη μπορεί να θεωρηθεί ως ένα σύστημα του Καίσαρα. Για να χρησιμοποιήσουμε αυτόν τον πίνακα διαβάζουμε το κείμενο από την πρώτη στήλη, το κλειδί από την πρώτη γραμμή και το κρυπτοκείμενο είναι το γράμμα που βρίσκεται στην τομή τους. Για την αποκρυπτογράφηση, βρίσκουμε

το γράμμα του κρυπτοκειμένου στη στήλη που υποδεικνύεται από το κλειδί και διαβάζουμε το αντίστοιχο γράμμα του αρχικού κειμένου από την πρώτη στήλη της γραμμής στην οποία βρίσκεται το γράμμα του κρυπτοκειμένου. (Σημειώνεται ότι ο ρόλος των γραμμών και των στηλών μπορεί να αλλαχθεί)

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

Η κρυπτογράφηση του αρχικού κειμένου χρειάζεται μια λέξη-κλειδί μήκους r η οποία επαναλαμβάνεται για να καλύψει όλο το αρχικό κείμενο. Τέτοια πολυαλφαβητικά συστήματα, όπου τα αλφάβητα των αντικαταστατών επαναλαμβάνονται περιοδικά συνήθως ονομάζονται *περιοδικά*.

Αν γνωρίζουμε την περίοδο κάποιου περιοδικού πολυαλφαβητικού συστήματος, τότε η κρυπτανάλυση του μπορεί να αναχθεί στην κρυπτανάλυση ενός μονοαλφαβητικά συστήματος: Θεωρούμε ότι η περίοδος είναι r. Διατάσσουμε τα γράμματα του κρυπτοκειμένου σε γραμμές με r στήλες σε κάθε γραμμή (γράφουμε r γράμματα σε κάθε γραμμή). Δύο εμφανίσεις του ίδιου γράμματος στην ίδια στήλη αντιπροσωπεύουν το ίδιο αρχικό γράμμα. Οπότε μπορούμε να αποκρυπτογραφήσουμε κάθε στήλη με μέτρηση συχνοτήτων.

Αν η περίοδος που χρησιμοποιείται σε ένα περιοδικό σύστημα τύπου VIGENÉRE είναι άγνωστη, μπορεί (ίσως) να βρεθεί με την *μέθοδο του Kasiski* (F.W. Kasiski, 1860): Αυτή η μέθοδος υπολογίζει την περίοδο ψάχνοντας τις εμφανίσεις της ίδιας λέξης στο κρυπτοκείμενο. Υποθέτουμε ότι μια συγκεκριμένη λέξη εμφανίζεται δύο φορές στο κρυπτοκείμενο, και ότι μεσολαβούν m γράμματα μεταξύ των εμφανίσεων αυτών (δηλαδή από την αρχή της πρώτης μέχρι την αρχή της δεύτερης).

Αυτό μπορεί να οφείλεται στο ότι οι δύο εμφανίσεις αντιστοιχούν στο ίδιο κομμάτι του αρχικού κειμένου, που έχει κρυπτογραφηθεί ξεκινώντας από την ίδια θέση του κλειδιού. Σε αυτήν την περίπτωση, η απόσταση m μεταξύ των δύο εμφανίσεων στο κρυπτοκείμενο θα πρέπει να είναι πολλαπλάσιο του μήκους του κλειδιού. Αν βρεθούν αρκετές τέτοιες διπλές εμφανίσεις στο κρυπτοκείμενο, μπορούμε να κάνουμε μια καλή εκτίμηση για το μήκος του κλειδιού.

Στην διαδικασία κρυπτανάλυσης σημαντικό ρόλο μπορεί να παίξει και ο δείκτης σύμπτωσης (Index of Coincidence (IC)) ο οποίος εκφράζει την πιθανότητα δύο τυχαίοι χαρακτήρες ενός κειμένου να ταυτίζονται. Έρευνες από τη γλωσσολογία έχουν δείξει ότι η τιμή του σε κείμενο φυσικής γλώσσας διαφέρει σημαντικά από την τιμή του σε τυχαίο κείμενο.

Συγκεκριμένα ο δείκτης σύμπτωσης σε κείμενο X, όπου f_i είναι το πλήθος εμφανίσεων του γράμματος i, ορίζεται από την σχέση:

$$IC(X) = \frac{\sum_{i=0}^{25} \binom{f_i}{2}}{\binom{n}{2}} = \sum_{i=0}^{25} \frac{f_i(f_i-1)}{n(n-1)}$$

όπου C_k^n οι συνδυασμοί n ανά k. Είναι προφανές ότι ο δείκτης σύμπτωσης παραμένει αναλλοίωτος σε ολίσθηση του κειμένου.

Έχει υπολογιστεί ότι σε άγνωστο κείμενο αγγλικής X: $E[IC(X)] \simeq \sum_{i=0}^{25} p_i^2 \simeq 0.065$ όπου p_i: η στατιστική συχνότητα του γράμματος i, ενώ σε εντελώς τυχαίο κείμενο με αγγλικούς χαρακτήρες: $E[IC(X)] \simeq \sum_{i=0}^{25} (\frac{1}{26})^2 = \frac{1}{26} \simeq 0.03$

Έτσι με βάση τα παραπάνω μπορούμε με μεγάλη πιθανότητα να ξεχωρίσουμε ένα τυχαίο κείμενο με αγγλικούς χαρακτήρες από ένα κανονικό αγγλικό κείμενο.

Ο δείκτης σύμπτωσης μπορεί να χρησιμοποιηθεί για να βρεθεί η περίοδος r:

- Δοκιμάζουμε για διαδοχικές τιμές του r να χωρίσουμε το κρυπτοκείμενο σε r στήλες
- Προκύπτουν οι εξής στήλες: $C_i = \{c_{i+jr} \mid 0 \leq j \leq \lceil \frac{n}{r} \rceil - 1\}$
- Στη συνέχεια υπολογίζουμε το $IC(C_i)$ για κάθε στήλη. Αν έχουμε βρει το σωστό μήκος τότε θα λαμβάνει τιμές φυσικής γλώσσας, ενώ σε διαφορετική περίπτωση θα λαμβάνει τίμιες τυχαίου κειμένου.

Στην συνέχεια θα υπολογιστεί το κλειδί: Για να το πετύχουμε αυτό δοκιμάζουμε ολισθήσεις της πρώτης στήλης κατά $j = 1, 2, \ldots, 25$. Μεταξύ της ολισθημένης πρώτης στήλης και της m-στης στήλης (για $2 \leq m \leq r$) υπολογίζουμε τον δείκτη αμοιβαίας σύμπτωσης ως:

$$IMC(C_{1 \gg j}, C_m) = \sum_{i=0}^{25} \frac{f_{(1 \gg j)}(i) f_{(m)}(i)}{|C_1||C_m|}$$

όπου

- $f_{(1)}(i)$ το πλήθος εμφανίσεων χαρακτήρα i στην στήλη 1.
- $f_{(1 \gg j)}(i) = f_{(1)}((i-j) \bmod 26)$ το πλήθος εμφανίσεων χαρακτήρα i στην στήλη 1, μετά από ολίσθηση της στήλης κατά j.

Ο δείκτης αμοιβαίας σύμπτωσης αντιστοιχεί στην πιθανότητα δύο τυχαίοι χαρακτήρες από δύο κείμενα να ταυτίζονται. Έχει παρόμοιες ιδιότητες με το Δείκτη Σύμπτωσης, δηλαδή η τιμή του διαφέρει σημαντικά μεταξύ αγγλικών κειμένων (ή προερχόμενων από αγγλικά κείμενα, με την ίδια ολίσθηση) και τυχαίων κειμένων (ή προερχόμενων από αγγλικό κείμενο, με διαφορετική ολίσθηση).

Ο κώδικας Vernam (one-time pad) (1917)

Μία ακραία περίπτωση ενός πολυαλφαβητικού συστήματος είναι ένα σύστημα όπου το κλειδί έχει ίσο μήκος με το αρχικό κείμενο. Ο κώδικας Vernam, ή αλλιώς "μπλοκάκι μιας χρήσης" (One-Time Pad (OTP)) είναι ένα κρυπτοσύστημα μυστικού κλειδιού όπου το κλειδί έχει το ίδιο μήκος με το κείμενο προς κρυπτογράφηση. Επιπλέον, το κλειδί το χρησιμοποιούμε μόνο μια φορά και μετά το "πετούμε", δεν το ξαναχρησιμοποιούμε.

Το αρχικό κείμενο M αναπαρίσταται ως δυαδική ακολουθία, όπως επίσης και το κλειδί K. Το κρυπτοκείμενο C προκύπτει από τη αποκλειστική διάζευξη ανά bit (XOR) (ή πρόσθεση modulo 2) του αρχικού κειμένου με το κλειδί:

$$C = M \oplus K.$$

Η αποκρυπτογράφηση δίνεται από την ίδια πράξη:

$$M = C \oplus K.$$

Αποδεικνύεται ότι είναι αδύνατο για τον κρυπταναλυτή να σπάσει τον κώδικα που χρησιμοποιεί μπλοκάκι μιας χρήσης: Οποιοδήποτε κρυπτοκείμενο C δεν αποκαλύπτει καμία πληροφορία για το αρχικό κείμενο P αφού κάθε μήνυμα M θα μπορούσε να παραγάγει το C, αν το κλειδί K ήταν ίσο με $K = C \oplus M$.

Η κρυπτογράφηση με μπλοκάκι μιας χρήσης είναι αποδεδειγμένα ασφαλής με πληροφοριοθεωρητική έννοια, αφού ο υποκλοπέας δεν έχει ποτέ αρκετή πληροφορία για να αποκρυπτογραφήσει το κρυπτοκείμενο και κανένα μέγεθος υπολογιστικής δύναμης δεν μπορεί να τον βοηθήσει.

Από την άλλη πλευρά, το μπλοκάκι μιας χρήσης δεν είναι πρακτικό αφού ένα μεγάλο κλειδί πρέπει να δημιουργηθεί, να διανεμηθεί και να αποθηκευτεί.

1.3 Ορισμός κρυπτοσυστήματος

Κάθε κρυπτοσύστημα αποτελείται από 3 αλγόριθμους:

- **Αλγόριθμος Δημιουργίας Κλειδιών** (`KeyGen`): Λαμβάνει ως είσοδο μία παράμετρο ασφαλείας λ και δημιουργεί τα κλειδιά κρυπτογράφησης key_{enc}, key_{dec} και αποκρυπτογράφησης. Στην περίπτωση των συμμετρικών κρυπτοσυστημάτων τα παραπάνω φυσικά ταυτίζονται (δηλ. $key_{enc} = key_{dec}$). Συνήθως η παράμετρος ασφαλείας συμβολίζεται στο μοναδιαίο σύστημα αρίθμησης οπότε συμβολικά γράφουμε:

  $$KeyGen(1^{\lambda}) \rightarrow (key_{enc}, key_{dec})$$

- **Αλγόριθμος Κρυπτογράφησης** (`Encrypt`): Λαμβάνει ως είσοδο ένα κλειδί κρυπτογράφησης και ένα μήνυμα m και δίνει ως έξοδο ένα κρυπτογράφημα c. Δηλαδή:

  $$\texttt{Encrypt}(key_{enc}, m) \rightarrow c$$

  Ο `Encrypt` μπορεί να είναι είτε ντετερμινιστικός, δηλαδή σε κάθε μήνυμα να αντιστοιχεί το ίδιο κρυπτοκείμενο σε κάθε κρυπτογράφηση ή πιθανοτικός, δηλαδή σε κάθε μήνυμα να αντιστοιχεί διαφορετικό κρυπτοκείμενο σε κάθε κρυπτογράφηση.

- **Αλγόριθμος Αποκρυπτογράφησης** (`Decrypt`): Λαμβάνει ως είσοδο ένα κλειδί αποκρυπτογράφησης και ένα μήνυμα c και δίνει ως έξοδο ένα μήνυμα m. Δηλαδή:

  $$\texttt{Decrypt}(key_{dec}, c) \rightarrow m$$

  Ο `Decrypt` είναι συνήθως ντετερμινιστικός.

Από τους παραπάνω ορισμούς, προκύπτουν τρία σύνολα:

- Το σύνολο όλων των δυνατών κλειδιών `K`
- Το σύνολο όλων των δυνατών μηνυμάτων `M`
- Το σύνολο όλων των δυνατών κρυπτοκειμένων `C`

Συνήθως ο αλγόριθμος `KeyGen` λαμβάνει ένα κλειδί από τον `K` χρησιμοποιώντας την ομοιόμορφη κατανομή. Επιπλέον είναι εύκολο να συμπεράνει κανείς από τους παραπάνω ορισμούς ότι το μόνο σύνολο που επιλέγεται 'ελεύθερα' είναι το `M`. Τα υπόλοιπα προκύπτουν ως σύνολο τιμών των `KeyGen` και `Decrypt`.

Με βάση τα παραπάνω κρυπτοσύστημα είναι η εξάδα

$$\mathcal{CS} = (\mathtt{K}, \mathtt{M}, \mathtt{C}, KeyGen, Enc, Dec).$$

Η βασική ιδιότητα που πρέπει να έχει κάθε κρυπτοσύστημα είναι αυτή της ορθότητας, δηλαδή κάθε κρυπτογράφημα πρέπει να αποκρυπτογραφείται σωστά, να δίνει δηλαδή το αρχικό μήνυμα. Δηλαδή:

$$Dec(key_{dec}, Enc(key_{enc}, m)) = m$$

Στους παραπάνω ορισμούς εμπίπτουν τόσο τα συμμετρικά κρυπτοσυστήματα, όπου $key_{dec} = key_{enc}$, όσο και τα κρυπτοσυστήματα δημοσίου κλειδιού, όπου το key_{enc} είναι δημόσιο ενώ το key_{dec} ιδιωτικό.

1.4 Μοντέλα ασφάλειας

1.4.1 Οι αρχές του Kerckhoffs

Από τους πρώτους πειραματισμούς με τα πρώτα κρυπτογραφικά συστήματα προέκυψαν κάποιες εμπειρικές παρατηρήσεις, οι οποίες διατυπώθηκαν από τον Kerckhoffs ως ένα σύνολο από 6 αρχές. Οι πιο σημαντικές από αυτές είναι η 1η και η 2η τις οποίες αναφέρουμε παρακάτω:

Ορισμός 1.1 (1η αρχή του Kerckhoffs)**.** Η μέθοδος κρυπτογράφησης δεν πρέπει να απαιτείται να είναι μυστική. Πρέπει να μπορεί να πέσει στα χέρια του εχθρού χωρίς να δημιουργήσει κανένα πρόβλημα.

Αυτό σημαίνει πως η μόνη μυστική πληροφορία που πρέπει να προφυλάσσεται κατά τη διάρκεια μιας επικοινωνίας είναι το κλειδί. Οι κυριότεροι λόγοι για τον σκοπό αυτό συνοψίζονται παρακάτω:

1. Κατ'αρχήν, το κλειδί έχει μικρότερο μέγεθος από τον αλγόριθμο κρυπτογράφησης και κατά συνέπεια διανέμεται πολύ πιο εύκολα με ασφάλεια.
2. Επιπλέον ένα κρυπτοσύστημα δεν είναι ένα ενιαίο στοιχείο όπως το κλειδί. Αποτελείται από πολλά επιμέρους στοιχεία κάποιο από τα οποία μπορεί να διαρρεύσει διακυβεύοντας την συνολική ασφάλεια του συστήματος.

3. Σε περίπτωση διαρροής το κλειδί είναι πολύ πιο εύκολο να αλλαχθεί από τον αλγόριθμο κρυπτογράφησης.

4. Στην περίπτωση επικοινωνίας περισσότερων από δύο οντοτήτων είναι πολύ πιο εύκολο να χρησιμοποιείται ένα κρυπτοσύστημα με πολλά κλειδιά παρά πολλά κρυπτοσυστήματα.

5. Η δημοσιοποίηση των λεπτομερειών ενός κρυπτοσυστήματος οδηγεί στην εξέταση τους από πολλούς ειδικούς και άρα η απουσία επιτυχημένων επιθέσεων με τον καιρό αποτελεί ισχυρή ένδειξη ασφάλειας.

6. Τέλος σε ένα ανοικτό κρυπτοσύστημα υπάρχει περίπτωση τα διάφορα προβλήματα να γίνουν αντιληπτά από τους 'καλούς' και να διορθωθούν. Αντίθετα σε ένα κλειστό σύστημα τα ελαττώματά μπορεί να είναι γνωστά σε ένα μικρό κύκλο χρηστών μόνο, οι οποίοι ίσως και να τα χρησιμοποιούν προς όφελος τους, αφήνοντας στους υπόλοιπους χρήστες την ψευδαίσθηση της ασφάλειας.

Για τους παραπάνω λόγους η αρχή του Kerckhoffs έχει ισχυρή αποδοχή στη σύγχρονη κρυπτογραφία. Παρά το γεγονός ότι έχει ηλικία 150 χρόνων, υπάρχουν ακόμα και σήμερα 'κλειστά' κρυπτοσυστήματα, τα οποία μάλιστα είναι δημιουργήματα μεγάλων εταιρειών που ισχυρίζονται ότι παρέχουν απόλυτη ασφάλεια, χωρίς όμως να περέχουν οποιαδήποτε απόδειξη ή δυνατότητα εξέτασης. Ένας κατάλληλος όρος για αυτά είναι ο χαρακτηρισμός του Bruce Schneier *snake oil*[2], όπως φαίνεται από τα άρθρα που παραθέτουμε στο τέλος του κεφαλαίου.

Ορισμός 1.2 (2η αρχή του Kerckhoffs)**.** Το κρυπτοσύστημα θα πρέπει να είναι πρακτικά απρόσβλητο, αν δεν γίνεται θεωρητικά.

Η αρχή αυτή εκφράζει το γεγονός ότι ένα κρυπτοκείμενο δεν χρειάζεται να είναι για πάντα απόρρητο: υπάρχει κάποια χρονικό διάστημα μετά από το οποίο δεν πειράζει να αποκαλυφθεί. Αρκεί λοιπόν η κρυπτανάλυση του να απαιτεί περισσότερο χρόνο από το διάστημα αυτό. Επιπλέον κατά το διάστημα που απαιτείται η μυστικότητα δεν χρειάζεται να μηδενιστεί η πιθανότητα κρυπτανάλυσης, αρκεί να είναι πολύ μικρή (αμελητέα). Η αρχή αυτή έχει επηρεάσει πάρα πολύ τη σύγχρονη κρυπτογραφία, θέτοντας τις βάσεις για την λεγόμενη *υπολογιστική ασφάλεια*, με την οποία θα ασχοληθούμε εκτενέστερα στη συνέχεια.

[2]Στα ελληνικά θα μπορούσε να αποδοθεί ως 'νερό Καματερού', ένα υποτιθέμενο "θαυματουργό" νερό που κυκλοφορούσε στην Ελλάδα στα μέσα της δεκαετίας του '70.

1.4.2 Τύποι επιθέσεων

Η ιστορική αναδρομή στα διάφορα κρυπτοσυστήματα αποκαλύπτει μια πληθώρα επιθέσεων που αυτά έχουν υποστεί ανά τους αιώνες (κυρίως βέβαια κατά τους 2 τελευταίους). Οι επιθέσεις αυτές μπορούν να χωριστούν στις παρακάτω κατηγορίες, τις οποίες παρουσιάζουμε σε αύξουσα σειρά ισχύος του αντιπάλου. Φυσικά υποθέτουμε ότι ο αντίπαλος μιας κατηγορίας διαθέτει τις δυνατότητες των αντιπάλων των προηγουμένων κατηγοριών. Στόχος του αντιπάλου σε κάθε περίπτωση είναι η *μη εξουσιοδοτημένη* απόκτηση του μηνύματος που αντιστοιχεί σε κάποιο κρυπτοκείμενο.

Επίθεση Μόνο Κρυπτοκειμένου - Ciphertext Only Attack (CO) Ο αντίπαλος παρακολουθεί παθητικά το κανάλι επικοινωνίας, συλλέγοντας κρυπτοκείμενα. Η επίθεση αυτή ισχύει για οποιοδήποτε κανάλι επικοινωνίας είναι δημόσιο.

Επίθεση Γνωστού Μηνύματος - Known Plaintext Attack (KPA) Ο αντίπαλος όπως και πριν είναι απλά ωτακουστής. Γνωρίζει όμως και κάποια ζεύγη μηνυμάτων και αντίστοιχων κρυπτοκειμένων. Ιστορικά αυτή η επίθεση έχει χρησιμοποιηθεί πάρα πολλές φορές. Για παράδειγμα κάθε έναρξη επικοινωνίας περιέχει μηνύματα χειραψίας τα οποία προέρχονται από ένα συγκεκριμένο σύνολο λέξεων. Στην περίπτωση της μηχανής Αίνιγμα για παράδειγμα ήταν γνωστά κρυπτοκείμενα που αντιστοιχούσαν σε μετεωρολογικές προγνώσεις.

Επίθεση Επιλεγμένου Μηνύματος - Chosen Plaintext Attack (CPA) Στην κατηγορία αυτή των επιθέσεων ο αντίπαλος δεν είναι απλά παθητικός χρήστης. Μπορεί να ενεργήσει επιθετικά έχοντας την επιπλέον δυνατότητα να αποκτήσει κρυπτογραφήσεις για μηνύματα της επιλογής του. Αν και σε πρώτη όψη αυτό δεν φαίνεται ένα ρεαλιστικό σενάριο, υπάρχουν ιστορικά παραδείγματα διεξαγωγής αυτής της επίθεσης, με πιο χαρακτηριστικό αυτό της ναυμαχίας του Midway (1942), όπου το αμερικανικό ναυτικό επιβεβαίωσε τις υποψίες του για επικείμενη επίθεση των Ιαπώνων στην ατόλη Midway, στέλνοντας παραπλανητικά ακρυπτογράφητα μηνύματα που περιείχαν την λέξη Midway και παρατηρώντας τις ιαπωνικές επικοινωνίες για πιθανή κρυπτογραφημένη αναμετάδοση (που πράγματι έγινε, κωδικοποιώντας το ‘Midway’ σε ‘AF’ – το οποίο ‘AF’ είχε ήδη εντοπιστεί από προηγούμενες αποκρυπτογραφήσεις ως σημείο επίθεσης των Ιαπώνων).

Επίθεση Επιλεγμένου Κρυπτοκειμένου - Chosen Ciphertext Attack (CCA) Ο πλέον ισχυρός αντίπαλος μπορεί επιπλέον να αποκρυπτογραφήσει επιλεγμένα κρυπτοκείμενα. Και αυτή η επίθεση έχει παρατηρηθεί πρακτικά σε πολλές περι-

πτώσεις. Ένα τέτοιο παράδειγμα αφορά την παρεμβολή τροποποιημένων κρυπτοκειμένων στην επικοινωνία μεταξύ δύο οντοτήτων. Το κρυπτοκείμενα αυτά θα αποκρυπτογραφηθούν σε μη έγκυρα μηνύματα. Η απρόσεκτη απόρριψη αυτών των μη έγκυρων μηνυμάτων μπορεί να οδηγήσει τον επιτιθέμενο να αποκτήσει πολύτιμες αποκρυπτογραφήσεις των τροποποιημένων κρυπτοκειμένων που εισήγαγε στην επικοινωνία.

1.4.3 Τέλεια Μυστικότητα (κατά Shannon)

Ο Shannon, αναφερόμενος σε επιθέσεις τύπου CO μόνο, διατύπωσε αυστηρά την ιδιότητα το κρυπτοκείμενο να μην παρέχει καμμία πληροφορία για το αρχικό κείμενο και την ονόμασε *τέλεια μυστικότητα* (*perfect secrecy* [7].

Ας θεωρήσουμε το αρχικό κείμενο M, το κλειδί K και το κρυπτοκείμενο C σαν τυχαίες μεταβλητές που παίρνουν τιμές αντίστοιχα από τα σύνολα $\mathtt{M}, \mathtt{K}, \mathtt{C}$. Οι M και K είναι ανεξάρτητες, ενώ η C εξαρτάται από τις άλλες δύο. Ένα κρυπτοσύστημα έχει τέλεια μυστικότητα σύμφωνα με τον Shannon αν:

$$\forall x \in \mathtt{M}, y \in \mathtt{C}: \quad \Pr_{M\in\mathtt{M},K\in\mathtt{K}}[M = x \mid C = y] = \Pr_{M\in\mathtt{M}}[M = x]$$

Η τέλεια μυστικότητα ονομάζεται και πληροφοριοθεωρητική (information theoretic). Σημαίνει, όπως είπαμε, ότι το κρυπτοκείμενο δεν παρέχει καμία νέα πληροφορία για το αρχικό κείμενο, δηλαδή η (*a posteriori* πληροφορία ίδια με την *a priori*). Ισοδύναμες συνθήκες για τα παραπάνω αποτελούν:

1. $\forall x \in \mathtt{M}, y \in \mathtt{C}:\ \Pr[C = y] = \Pr[C = y \mid M = x]$, δηλαδή, η πιθανότητα εμφάνισης ενός κρυπτοκειμένου είναι ανεξάρτητη από το αρχικό κείμενο.
2. $\forall x_1, x_2 \in \mathtt{M}, y \in \mathtt{C}:\ \Pr[C = y \mid M = x_1] = \Pr[C = y \mid M = x_2]$, η οποία συνθήκη είναι χρήσιμη για ανταπόδειξη

Παράδειγμα 1. *Random Shift Cipher Έστω το παρακάτω κρυπτοσύστημα:*

- $\mathtt{M} = \mathtt{K} = \mathtt{C} = \{0, \ldots, 25\}$
- `Encrypt` : $C = Enc(K, M) = M + K \bmod 26$
- `Decrypt` : $C = Dec(K, C) = C - K \bmod 26$

Υποθέτουμε ότι η κατανομή των κλειδιών είναι ομοιόμορφη δηλαδή $\forall K \in \mathtt{K}$*:* $\Pr[K = i] = \frac{1}{26},\ 0 \le i \le 25$*. Με βάση το παραπάνω θα αποδείξουμε ότι έχει τέλεια μυστικότητα:*

1. $\forall y \in \mathtt{C} : \Pr[C = y] = \sum_{x \in \mathtt{M}} \Pr[M = x] \cdot \Pr[K = y - x \bmod 26] = \frac{1}{26} \sum_{x \in \mathtt{M}} \Pr[M = x] = \frac{1}{26}$
2. $\Pr[M = x \mid C = y] = \frac{\Pr[C=y|M=x] \Pr[M=x]}{\Pr[C=y]}$
3. *Από (1) και (2):* $\forall x \in \mathtt{M}, y \in \mathtt{C} : \Pr[M = x \mid C = y] = \frac{\frac{1}{26} \Pr[M=x]}{\frac{1}{26}} = \Pr[M = x]$

Γενικεύοντας για να έχουμε τέλεια μυστικότητα πρέπει το μήκος κλειδιού να είναι μεγαλύτερο ή ίσο του μήκους του μηνύματος, δηλαδή:

$$|\mathtt{M}| \leq |\mathtt{C}| \leq |\mathtt{K}|$$

Πράγματι τότε θα ισχύει

- $|\mathtt{M}| \leq |\mathtt{C}|$, από την απαίτηση για κρυπτογράφηση '1-1'.
- $|\mathtt{C}| \leq |\mathtt{K}|$: Αν $|\mathtt{C}| > |\mathtt{K}|$,
  $\forall x \in \mathtt{M}, \exists y \in \mathtt{C}, \Pr[C = y \mid M = x] = 0 \neq Pr[C = y]$.

Θεώρημα 1.3. *Θεώρημα τέλειας μυστικότητας*
Έστω κρυπτοσύστημα με $|\mathtt{M}| = |\mathtt{C}| = |\mathtt{K}|$. *Το σύστημα έχει τέλεια μυστικότητα ανν ισχύουν τα εξής:*
(1) για κάθε $x \in \mathtt{M}, y \in \mathtt{C}$*, υπάρχει μοναδικό* $k \in \mathtt{K}$*, ώστε* $enc_k(x) = y$
(2) κάθε κλειδί επιλέγεται με την ίδια πιθανότητα, συγκεκριμένα $1/|\mathtt{K}|$

Απόδειξη.
- Ευθύ: Παραβίαση της (1) οδηγεί σε μηδενική δεσμευμένη πιθανότητα κάποιου y με δοσμένο x. Από την (1) και αρχή Περιστερώνα και ιδιότητα '1-1' της enc_{K_i}:

  $$\forall y \in \mathtt{C}, k_1, k_2 \in \mathtt{K}, \exists x_1, x_2 \in \mathtt{M} : enc_{k_1}(x_1) = y, enc_{k_2}(x_2) = y$$

  Με χρήση της δεύτερης Ισοδύναμης Συνθήκης προκύπτει ότι τα k_1, k_2 είναι ισοπίθανα.
- Αντίστροφο: Άμεσο, με χρήση δεύτερης Ισοδύναμης Συνθήκης

□

Αν και η τέλεια μυστικότητα είναι θεωρητικά εφικτή, η παραγωγή και η ανταλλαγή του κλειδιού είναι πρακτικά ασύμφορες λόγω των απαιτήσεων για το μήκος, την τυχαιότητα αλλά και την μοναδική χρήση.

Έτσι έγιναν προσπάθειες διατύπωσης νέων μοντέλων ασφάλειας, που να αντιστοιχούν σε αυτό που συνήθως επιδιώκουμε στην πράξη. Επικράτησε η έννοια της *υπολογιστικής ασφάλειας*, η οποία αντιστοιχεί στην δεύτερη αρχή του Kerchoffs: δεν χρειάζεται να είναι αδύνατη η αποκρυπτογράφηση, αρκεί να είναι υπολογιστικά ανέφικτη.

1.4.4 Υπολογιστική ασφάλεια: Ορισμοί και Αποδείξεις

Η απαίτηση στο μοντέλο αυτό είναι: οποιοσδήποτε αντίπαλος διαθέτει 'λογική' υπολογιστική ισχύ έχει αμελητέα πιθανότητα να σπάσει το κρυπτοσύστημα. Φυσικά, το μοντέλο θα πρέπει να συνοδεύεται από έναν συγκεκριμένο ορισμό ασφάλειας, ο οποίος θα καθορίζει τι ακριβώς σημαίνει το 'σπάσιμο'. Η 'λογική' υπολογιστική ισχύς, σύμφωνα με την Θεωρία Υπολογιστικής Πολυπλοκότητας (που συμφωνεί σε μεγάλο βαθμό με τις εμπειρικές παρατηρήσεις) αντιστοιχεί σε έναν αντίπαλο που διαθέτει πολυωνυμικό χρόνο (ως προς την παράμετρο ασφαλείας του `KeyGen`) και μπορεί να αποτύχει με μικρή πιθανότητα. Ένας τέτοιος αντίπαλος λέγεται Probabilistic Polynomial Time (PPT) (probabilistic polynomial time).

Το αντίστοιχο της τέλειας μυστικότητας για αντίπαλο PPT ονομάζεται **σημασιολογική ασφάλεια (semantic security)**. Εκφράζει την διαίσθηση ότι ένας υπολογιστικά περιορισμένος αντίπαλος δεν μπορεί αποδοτικά να μάθει τίποτε χρήσιμο από το κρυπτοκείμενο παρά μόνο με αμελητέα πιθανότητα. Η έννοια αυτή προτάθηκε από τους Goldwasser και Micali στο [3] και ουσιαστικά θεμελίωσε τη σύγχρονη κρυπτογραφία. Στην πράξη όμως ο ορισμός αυτός αποδείχθηκε δύσχρηστος. Ευτυχώς, οι ίδιοι απέδειξαν ότι μπορεί να χρησιμοποιηθεί ένας εναλλακτικός ορισμός, συγκεκριμένα η έννοια της **μη διακρισιμότητας** (indistinguishability). Η έννοια αυτή περιγράφεται με τη βοήθεια *παιχνιδιών ασφάλειας*.

Παιχνίδια Ασφάλειας

Ένα *παιχνίδι ασφάλειας* είναι ένα παιχνίδι μεταξύ δύο οντοτήτων: του *προκαλούντα (challenger)* $\mathcal{C}$ και του *αντιπάλου (adversary)* $\mathcal{A}$, που αναπαριστά οποιεσδήποτε επιθέσεις μπορούν να εκδηλωθούν στο κρυπτοσύστημα. Ο προκαλών είναι μία φανταστική οντότητα που δημιουργεί την *πρόκληση (challenge)* στην οποία πρέπει να *αποκριθεί (response)* ο $\mathcal{A}$. Και οι δύο οντότητες είναι PPT. Επίσης, σε κάποιους ορισμούς, ο προκαλών παίζει και το ρόλο του "μαντείου" κρυπτογράφησης ή αποκρυπτογράφησης όταν προβλέπεται ο αντίπαλος να έχει πρόσβαση σε τέτοιο μαντείο (βλ. Ενότητα 1.4.2, επιθέσεις CPA και CCA). Ένα παιχνίδι μη διακρισιμότητας έχει συνήθως τα εξής βήματα:

- Οι $\mathcal{A}$ και $\mathcal{C}$ επικοινωνούν ανταλλάσσοντας μηνύματα.

- Κάποια στιγμή ο $\mathcal{A}$ παράγει δύο μηνύματα m_0, m_1.
- Τότε ο $\mathcal{C}$ εσωτερικά διαλέγει ένα τυχαίο bit b και παράγει και στέλνει στον $\mathcal{A}$ το $c_b \leftarrow \texttt{Encrypt}(m_b)$ (challenge).
- $\mathcal{A}$: Με βάση το c_b και όποιες άλλες πληροφορίες συγκέντρωσε, υπολογίζει ένα bit b' (response), προσπαθώντας να "μαντέψει" το b.
- Ο αντίπαλος κερδίζει το παιχνίδι αν $b' = b$.

Συνοπτικά λοιπόν το παιχνίδι και η συνθήκη νίκης μπορούν να γραφούν ως εξής:

$$\text{IND} - \text{Game}(\mathcal{A}) = \begin{cases} 1, & b' = b \\ 0, & b' \neq b \end{cases}$$

Καθώς και οι δύο οντότητες είναι περιορισμένες σε πολυωνυμικό χρόνο, το πλήθος των μηνυμάτων πρέπει να είναι πολυωνυμικό.

Ορίζουμε το πλεονέκτημα του $\mathcal{A}$ ως:

$$Adv_{IND}(\mathcal{A}) = |Pr[\text{IND} - \text{Game}(\mathcal{A}) = 1] - \tfrac{1}{2}|$$

Ορισμός 1.4. Ως αμελητέα συνάρτηση ορίζουμε οποιαδήποτε συνάρτηση $negl$ για την οποία για κάθε πολυώνυμο p υπάρχει n_0 ώστε $\forall n \geq n_0 : negl(n) < \frac{1}{p(n)}$

Για παράδειγμα, η συνάρτηση $f(n) = 1/2^n$ είναι αμελητέα.

Ορισμός 1.5 (Μη διακρισιμότητα)**.** Ένα κρυπτοσύστημα διαθέτει την ιδιότητα της μη διακρισιμότητας όταν υπάρχει αμελητέα συνάρτηση $negl$, ώστε $\forall$ PPT $\mathcal{A}$:

$$Adv_{IND}(\mathcal{A}) \leq negl(\lambda)$$

Σε πρακτικό επίπεδο λοιπόν εάν ο $\mathcal{A}$ διαθέτει δύο αρχικά κείμενα, και του δώσουν το κρυπτοκείμενο ενός από αυτά, δεν μπορεί αποδοτικά να βρει σε ποιο αντιστοιχεί με πιθανότητα σημαντικά μεγαλύτερη του 1/2. Όπως αναφέραμε και παραπάνω, έχει αποδειχθεί ([3],[5]) ότι η σημασιολογική ασφάλεια είναι ισοδύναμη με την μη διακρισιμότητα. Η ιδιότητα αυτή εξειδικεύεται περαιτέρω ανάλογα με την ισχύ του αντιπάλου, όπως περιγράφουμε παρακάτω.

Ας ξεκινήσουμε με την περίπτωση μη φραγμένου χρονικά αντιπάλου, πρόκειται δηλαδή για ουσιαστικά για μεταφορά της έννοιας της τέλειας μυστικότητας κατά Shannon στο πλαίσιο της μη διακρισιμότητας:

Παιχνίδι IND-PS

- Ο $\mathcal{A}$ παράγει δύο μηνύματα m_0, m_1 τα οποία στέλνει στον $\mathcal{C}$.
- Ο $\mathcal{C}$ εκτελεί τον αλγόριθμο $\texttt{KeyGen}$ παράγοντας ένα τυχαίο κλειδί k. Επίσης διαλέγει ένα τυχαίο bit $b \in \{0, 1\}$.
- Στην συνέχεια κρυπτογραφεί το μήνυμα m_b και στέλνει στον $\mathcal{A}$ το κρυπτοκείμενο $c \leftarrow \texttt{Encrypt}(k, m_b)$.
- Ο $\mathcal{A}$ επεξεργάζεται το c και παράγει ένα bit b'.
- Αν $b' = b$ τότε το αποτέλεσμα του παιχνιδιού ορίζεται 1 και ο $\mathcal{A}$ κερδίζει. Σε διαφορετική περίπτωση το αποτέλεσμα ορίζεται 0.

Ορισμός 1.6. Ένα κρυπτοσύστημα διαθέτει την ιδιότητα IND − PS αν στο παιχνίδι IND − PS κάθε $\mathcal{A}$ έχει αμελητέο πλεονέκτημα στον υπολογισμό του b από το να μαντέψει τυχαία.

Στο παραπάνω παιχνίδι παρατηρούμε ότι ο $\mathcal{A}$ μπορεί εύκολα να έχει πιθανότητα επιτυχίας $1/2$, αν μαντέψει τυχαία. Από την άλλη, αποδεικνύεται ότι αν το κρυπτοσύστημα έχει την ιδιότητα της τέλειας μυστικότητας, τότε κανείς αντίπαλος δεν μπορεί να κάνει κάτι καλύτερο από το να μαντέψει τυχαία, δηλαδή το πλεονέκτημα είναι μηδενικό.

Σε περίπτωση που ο αντίπαλος δεν είναι υπολογιστικά πανίσχυρος, αλλά είναι PPT, δηλαδή είναι πιθανοτικός και ο χρόνος του είναι πολυωνυμικά φραγμένος ως προς την παράμετρο ασφάλειας λ, τότε το παιχνίδι έχει ως εξής:

Παιχνίδι IND-EAV

- Ο $\mathcal{A}$ λαμβάνει την παράμετρο ασφαλείας 1^λ.
- Ο $\mathcal{A}$ παράγει δύο μηνύματα m_0, m_1 με ίδιο μήκος τα οποία στέλνει στον $\mathcal{C}$.
- Ο $\mathcal{C}$ εκτελεί τον αλγόριθμο $\texttt{KeyGen}(1^\lambda)$. παράγοντας ένα τυχαίο κλειδί k. Επίσης διαλέγει ένα τυχαίο bit $b \in \{0, 1\}$.
- Στην συνέχεια κρυπτογραφεί το μήνυμα m_b παράγοντας το $c \leftarrow \texttt{Encrypt}(k, m_b)$ και το στέλνει στον $\mathcal{A}$.

- Ο $\mathcal{A}$ επεξεργάζεται το c και παράγει ένα bit b'.
- Αν $b' = b$ τότε το αποτέλεσμα του παιχνιδιού ορίζεται 1 και ο $\mathcal{A}$ κερδίζει. Σε διαφορετική περίπτωση το αποτέλεσμα ορίζεται 0.

Ορισμός 1.7. Ένα κρυπτοσύστημα είναι διαθέτει την ιδιότητα IND − EAV αν στο παιχνίδι IND − EAV κάθε PPT $\mathcal{A}$ έχει αμελητέο πλεονέκτημα στον υπολογισμό του b σε σχέση με το να μαντέψει τυχαία.

Με τον ίδιο τρόπο μπορούμε να ορίσουμε την ασφάλεια μη διακρισιμότητας ως προς τις επιθέσεις CPA και CCA της Ενότητας 1.4.2.

Παιχνίδι Indistinguishability under Chosen Plaintext Attack (IND-CPA)

- Ο $\mathcal{A}$ λαμβάνει την παράμετρο ασφαλείας 1^λ.
- Ο $\mathcal{C}$ λαμβάνει την παράμετρο ασφαλείας 1^λ και εκτελεί τον αλγόριθμο `KeyGen` παράγοντας το κλειδί k.
- Ο $\mathcal{A}$ κρυπτογραφεί πολυωνυμικό πλήθος μηνυμάτων χρησιμοποιώντας τον $\mathcal{C}$.
- Τελικά ο $\mathcal{A}$ παράγει δύο μηνύματα m_0, m_1 τα οποία στέλνει στον $\mathcal{C}$.
- Ο $\mathcal{C}$ διαλέγει ένα τυχαίο bit $b \in \{0, 1\}$.
- Στην ο $\mathcal{C}$ συνέχεια κρυπτογραφεί το μήνυμα m_b παράγοντας το $c \leftarrow \texttt{Encrypt}(k, m_b)$ και το στέλνει στον $\mathcal{A}$.
- Ο $\mathcal{A}$ κρυπτογραφεί πολυωνυμικό πλήθος μηνυμάτων χρησιμοποιώντας τον $\mathcal{C}$.
- Ο $\mathcal{A}$ επεξεργάζεται το c και τις πληροφορίες που συγκέντρωσε και παράγει ένα bit b'.
- Αν $b' = b$ τότε το αποτέλεσμα του παιχνιδιού ορίζεται 1 και ο $\mathcal{A}$ κερδίζει. Σε διαφορετική περίπτωση το αποτέλεσμα ορίζεται 0.

Ορισμός 1.8 (Ασφάλεια IND-CPA)**.** Το κρυπτοσύστημα έχει την ιδιότητα IND-CPA αν στο παιχνίδι IND-CPA κάθε PPT $\mathcal{A}$ έχει αμελητέο πλεονέκτημα στον υπολογισμό του b σε σχέση με το να μαντέψει τυχαία.

Για την ιδιότητα της μη διακρισιμότητας ως προς επίθεση CCA έχουμε το παρακάτω παιχνίδι.

Παιχνίδι Indistinguishability under Chosen Ciphertext Attack (IND-CCA)

- Ο $\mathcal{A}$ λαμβάνει την παράμετρο ασφαλείας 1^λ.
- Ο $\mathcal{C}$ λαμβάνει την παράμετρο ασφαλείας 1^λ και εκτελεί τον αλγόριθμο `KeyGen` παράγοντας το κλειδί k.
- Ο $\mathcal{A}$ κρυπτογραφεί πολυωνυμικό πλήθος μηνυμάτων χρησιμοποιώντας τον $\mathcal{C}$.
- Ο $\mathcal{A}$ χρησιμοποιεί τον $\mathcal{C}$ ως *decryption oracle* και μπορεί να αποκρυπτογραφήσει πολυωνυμικό πλήθος μηνυμάτων
- Τελικά παράγει δύο μηνύματα m_0, m_1, διαφορετικά από αυτά που έχει αποκρυπτογραφήσει, τα οποία στέλνει στον $\mathcal{C}$,
- Ο $\mathcal{C}$ διαλέγει ένα τυχαίο bit $b \in \{0, 1\}$.
- Στην συνέχεια κρυπτογραφεί το μήνυμα m_b παράγοντας το $c \leftarrow \texttt{Encrypt}(k, m_b)$ και το στέλνει στον $\mathcal{A}$.
- Ο $\mathcal{A}$ συνεχίζει να κρυπτογραφεί πολυωνυμικό πλήθος μηνυμάτων και να κάνει οποιονδήποτε άλλο υπολογισμό μπορεί
- *Προαιρετικά* ο $\mathcal{A}$ μπορεί να συνεχίσει να χρησιμοποιεί το *decryption oracle*, αλλά *δεν μπορεί να ζητήσει αποκρυπτογράφηση του* c.
- Ο $\mathcal{A}$ επεξεργάζεται το c και τις πληροφορίες που συγκέντρωσε και παράγει ένα bit b'.
- Αν $b' = b$ τότε το αποτέλεσμα του παιχνιδιού ορίζεται 1 και ο $\mathcal{A}$ κερδίζει. Σε διαφορετική περίπτωση το αποτέλεσμα ορίζεται 0.

Ορισμός 1.9 (Ασφάλεια IND-CCA)**.** Το κρυπτοσύστημα έχει την ιδιότητα IND-CCA1 αν στο παιχνίδι IND-CCA κάθε PPT $\mathcal{A}$ έχει αμελητέο πλεονέκτημα στον υπολογισμό του b από το να μαντέψει τυχαία.

Αν εκτελείται και το προαιρετικό βήμα χωρίς ο αντίπαλος να αποκτήσει σημαντικό πλεονέκτημα, τότε το κρυπτοσύστημα έχει την ιδιότητα IND-CCA2 (ή adaptive IND-CCA).

Η σημασία του προαιρετικού βήματος είναι ότι επιτρέπει ένα άλλο είδος επίθεσης σε κάποια κρυπτοσυστήματα. Συγκεκριμένα, στο παραπάνω παίγνιο ο $\mathcal{A}$ δεν μπορεί να ρωτήσει τον C για την αποκρυπτογραφήση του c. Μπορεί όμως να μετατρέψει το c σε $\hat{c}$, να ζητήσει την αποκρυπτογράφησή του $\hat{c}$ σε $\hat{m}$ και να μετατρέψει 'αντίστροφα' το $\hat{m}$ σε m, κερδίζοντας με πιθανότητα 1. Αυτό είναι δυνατό στα κρυπτοσυστήματα που διαθέτουν μία άλλη ιδιότητα την *ευπλαστότητα* (malleability).

Ορισμός 1.10 (Εύπλαστο (malleable) Κρυπτοσύστημα)**.** Επιτρέπει στον $\mathcal{A}$ να φτιάξει, γνωρίζοντας μόνο το κρυπτοκείμενο $c = \texttt{Encrypt}(m)$, ένα *έγκυρο* κρυπτοκείμενο $c' = \texttt{Encrypt}(h(m))$, για κάποια, συνήθως πολυωνυμικά αντιστρέψιμη, συνάρτηση h γνωστή σε αυτόν.

Για παράδειγμα, το κλασικό σύστημα του Καίσαρα είναι εύπλαστο (γιατί;).

Αποδεικνύεται ότι (η απόδειξη παραλείπεται):

Θεώρημα 1.11. *Non-malleability* $\Leftrightarrow$ *IND-CCA2*

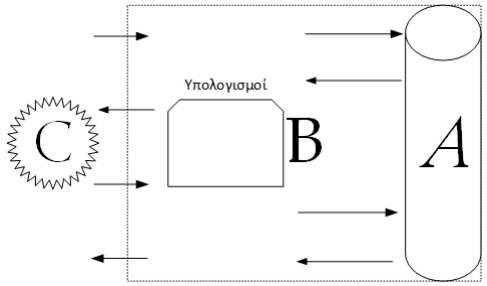
Κρυπτογραφικές υποθέσεις και αποδείξεις ασφάλειας

Για να εδραιώσουμε το υπολογιστικό ανάλογο της τέλειας μυστικότητας χρησιμοποιώντας τους παραπάνω ορισμούς, χρειαζόμαστε και κάποιο είδος απόδειξης. Οι αποδείξεις στην σύγχρονη κρυπτογραφία έχουν πολλά κοινά με τις αποδείξεις αναγωγής μεταξύ προβλημάτων στην θεωρία υπολογιστικής πολυπλοκότητας. Συγκεκριμένα, μετατρέπουν έναν (υπολογιστικά) αποδοτικό αντίπαλο σε έναν αλγόριθμο επίλυσης ενός θεωρούμενου δύσκολου υπολογιστικού προβλήματος. Το αντίστοιχο θεώρημα έχει την γενική μορφή:

Θεώρημα 1.12. *Αν ισχύει η υπόθεση $\mathcal{Y}$, τότε το κρυπτοσύστημα $\mathcal{CS}$ είναι ασφαλές (υπό συγκεκριμένο ορισμό).*

Για την απόδειξη χρησιμοποιούμε αντιθετοαντιστροφή, δηλαδή αποδεικνύουμε ότι αν το $\mathcal{CS}$ **δεν** είναι ασφαλές (υπό συγκεκριμένο ορισμό), τότε δεν ισχύει η $\mathcal{Y}$. Συνήθως δείχνουμε ότι αν υπάρχει PPT $\mathcal{A}$ ο οποίος "σπάει" το $\mathcal{CS}$ (παραβιάζει τον ορισμό ασφαλείας του), τότε μπορούμε να βρούμε PPT αλγόριθμο $\mathcal{B}$ που να επιλύει αποδοτικά το πρόβλημα $\Pi_{\mathcal{Y}}$ (αυτό είναι το πρόβλημα που η $\mathcal{Y}$ υποθέτει ότι είναι δύσκολο), διαψεύδοντας έτσι την $\mathcal{Y}$. [3] Συνήθως μιλάμε για κατασκευαστική

[3] Με όρους αναγωγών από την Θεωρία Υπολογιστικής Πολυπλοκότητας, αυτή είναι μια (ενδεχομένως πιθανοτική) πολυωνυμικού χρόνου αναγωγή από το πρόβλημα $\Pi_{\mathcal{Y}}$ στο πρόβλημα σπασίματος του $\mathcal{CS}$. Πράγματι, μια τέτοια αναγωγή αποδεικνύει ακριβώς ότι αν το $\mathcal{CS}$ σπάει με αποδοτικό πιθανοτικό αλγόριθμο, τότε και το $\Pi_{\mathcal{Y}}$ λύνεται αποδοτικά με πιθανοτικό αλγόριθμο, άρα η $\mathcal{Y}$ δεν ισχύει.



Σχήμα 1.2: Κατασκευή αντιπάλου σε κρυπτογραφική αναγωγή

απόδειξη, αν και μία μη-κατασκευαστική απόδειξη ύπαρξης του $\mathcal{B}$ (δεδομένου του $\mathcal{A}$) μπορεί επίσης να θεωρηθεί ισχυρή ένδειξη ασφάλειας.

Για να πετύχει το στόχο του, ο $\mathcal{B}$ μπορεί να χρησιμοποιεί εσωτερικά τον $\mathcal{A}$, σαν υπορουτίνα, χωρίς όμως να έχει πρόσβαση στο εσωτερικό του (black box access). Μία συνήθης τεχνική είναι ο $\mathcal{B}$ να παίζει τον ρόλο του challenger στο παιχνίδι ασφαλείας του $\mathcal{CS}$, αλληλεπιδρώντας με τον αντίπαλο $\mathcal{A}$. Φυσικά και ο $\mathcal{B}$ θα αλληλεπιδρά με τον δικό του προκαλούντα C, σε ένα πρωτόκολλο διάψευσης της $\mathcal{Y}$.

Η όλη διαδικασία περιγράφεται Σχήμα 1.4.4.

Με απλούστερα λόγια, αποδεικνύουμε πως αν το κρυπτοσύστημα δεν είναι ασφαλές, τότε μπορούμε να διαψεύσουμε μία γενικώς αποδεκτή υπόθεση, όπως για παράδειγμα τη δυσκολία παραγοντοποίησης ενός σύνθετου αριθμού ή τον υπολογισμό του διακριτού λογάριθμου. Εφόσον όμως η υπόθεση (είναι γενικά αποδεκτό ότι) ισχύει, τότε και το κρυπτοσύστημα θα (είναι γενικά αποδεκτό ότι) είναι ασφαλές.

Για να είναι ορθή η απόδειξη πρέπει να ακολουθούνται οι εξής κανόνες:

- Προσομοίωση: Ο $\mathcal{A}$ δεν θα πρέπει να μπορεί να διακρίνει τον $\mathcal{B}$ από οποιονδήποτε άλλο προκαλούντα.
- Πιθανότητα επιτυχίας: Αν ο $\mathcal{A}$ έχει μη αμελητέα πιθανότητα επιτυχίας τότε και ο $\mathcal{B}$ θα πρέπει να έχει μη αμελητέα πιθανότητα επιτυχίας.
- Πολυπλοκότητα: Ο $\mathcal{B}$ θα πρέπει να είναι PPT. Αυτό πρακτικά σημαίνει ότι οποιαδήποτε εσωτερική επεξεργασία πρέπει να είναι πολυωνυμικού χρόνου.
- Πρέπει να είναι όσο πιο αυστηρή (tight) γίνεται, δηλαδή ο χρόνος που καταναλώνουν ο $\mathcal{A}$ και ο $\mathcal{B}$ θα πρέπει να είναι "κοντά", το ίδιο και η πιθανότητα επιτυχίας τους (χρησιμοποιώντας συμβολισμό θα θέλαμε: $t_{\mathcal{B}} \approx t_{\mathcal{A}}$ και $\epsilon_{\mathcal{B}} \approx \epsilon_{\mathcal{A}}$).

Παρατήρηση: συνήθως είναι πιο εύκολο να αποδειχθεί το αντίστροφο του παραπάνω θεωρήματος, δηλαδή ότι "αν δεν ισχύει η υπόθεση τότε το κρυποσύστημα δεν είναι ασφαλές". Στο σημείο αυτό χρειάζεται προσοχή, καθώς η μία κατεύθυνση δεν συνεπάγεται την άλλη. Για παράδειγμα, το γεγονός ότι αν μπορούμε να παραγοντοποίησουμε γρήγορα έναν πολύ μεγάλο σύνθετο μας επιτρέπει να σπάσουμε το κρυπτοσύστημα RSA, δεν σημαίνει απαραίτητα ότι αν δεν μπορούμε να παραγοντοποιήσουμε γρήγορα δεν μπορούμε να σπάσουμε το RSA (μπορεί να υπάρχει κάποιος άλλος τρόπος). Για να δείξουμε το τελευταίο θα πρέπει να αποδείξουμε ότι ένας αλγόριθμος που αποκρυπτογραφεί μηνύματα κρυπτογραφημένα με RSA θα μπορούσε να οδηγήσει σε έναν αλγόριθμο για παραγοντοποίηση.

Αν και οι κρυπτογραφικές αναγωγές είναι γενικά αποδεκτό εργαλείο, δεν έχουν λείψει και επικρίσεις και μάλιστα από ιδιαίτερα σημαντικές φωνές ([4]). Σε κάθε περίπτωση λοιπόν δεν πρέπει να τις αντιμετωπίζουμε ως πανάκεια [1] καθώς εξ' ορισμού παρέχουν *σχετικές* εγγυήσεις ως προς ένα θεωρούμενο δύσκολο πρόβλημα και κάτω από έναν συγκεκριμένο ορισμό ασφάλειας.

Η πραγματική τους χρησιμότητα είναι ότι δίνουν την ευκαιρία να ορίσουμε σαφέστερα το κρυπτοσύστημα/πρωτόκολλο και να σκεφτούμε καλύτερα τις λεπτομέρειες και τις πιθανές αδυναμίες του. Επιπλέον επιτρέπουν την συγκέντρωση των κρυπταναλυτικών προσπαθειών στο πρόβλημα στο οποίο αντιστοιχεί ένα κρυπτοσύστημα, επιτρέποντας τη 'μαζική' ανάλυση ομοειδών πρωτοκόλλων, αντί για κάθε ένα χωριστά. Επίσης, δίνουν τη δυνατότητα ακριβέστερης ρύθμισης της παραμέτρου ασφάλειας σε σχέση με την επιθυμητή δυσκολία του υπολογιστικού προβλήματος.

Φυσικά η ύπαρξη κρυπτογραφικής αναγωγής δεν αρκεί από μόνη της, καθώς υπάρχουν συνήθως πολλές λεπτομέρειες, θεωρητικές και πρακτικές, που μπορούν να θέσουν σε κίνδυνο την ασφάλεια ενός κρυπτοσυστήματος. Είναι όμως μια καλή βάση εκκίνησης για τη δημιουργία αποδεδειγμένα αξιόπιστων κρυπτοσυστημάτων.

1.5 Ασκήσεις

1. Δίνεται το παρακάτω κρυπτοκείμενο, το οποίο γνωρίζουμε ότι έχει κρυπτογραφηθεί με κρυπτοσύστημα Vigenére. Αποκρυπτογραφήστε το (ξεκινήστε βρίσκοντας το μήκος του κλειδιού). Εξηγήστε σύντομα τη μέθοδο που ακολουθήσατε.

ZHRULIXEFHCMTDRDKTESBFPIRSVQZXUL
VWPKYWVWOWATCUPVIICOLEKAYWOEOTURBB
COENJWSMRUJMCIGKVCZMBUHTOTLSSMGSHU
LEOTURBBIOAVJQKNPHLLACNWPWTWVWO
WATPKHZGCGHYAIIRQJMCIGKVCZHHPPTOTLZ
VZYHVSDQZHBXAAGCELMQIE

2. Σε ένα κρυπτοσύστημα που διαθέτει τέλεια μυστικότητα, είναι αναγκαίο κάθε κλειδί να επιλέγεται με την ίδια πιθανότητα; Αποδείξτε τον ισχυρισμό σας.

3. Χρησιμοποιήστε τον ορισμό της τέλειας μυστικότητας κατά Shannon για να δείξετε ότι το παρακάτω κρυπτοσύστημα XOR δεν έχει τέλεια μυστικότητα: το 1ο κλειδί (για τον πρώτο χαρακτήρα) επιλέγεται με ομοιόμορφη πιθανότητα από το σύνολο των κλειδιών, το 2ο κλειδί επιλέγεται ομοιόμορφα από το σύνολο των υπολοίπων κλειδιών, και η διαδικασία επαναλαμβάνεται.

4. Δίνεται ένα κρυπτοσύστημα που διαθέτει τέλεια μυστικότητα για την ομοιόμορφη κατανομή πιθανότητας πάνω στα αρχικά κείμενα. Εξετάστε αν ισχύει στο κρυπτοσύστημα αυτό η ιδιότητα της τέλειας μυστικότητας για την εξής κατανομή πιθανότητας: ένα από τα αρχικά κείμενα εμφανίζεται με πιθανότητα $1/2$, ενώ τα υπόλοιπα με πιθανότητα $1/2(|\mathtt{M}| - 1)$, όπου $|\mathtt{M}|$ ο πληθάριθμος του χώρου μηνυμάτων.

5. Κατασκευάστε κρυπτοσύστημα που να έχει την ιδιότητα της τέλειας μυστικότητας, με $\mathtt{M} = A, B, C, D$ και $|\mathtt{M}| < |\mathtt{C}| < |\mathtt{K}|$: ορίστε κατάλληλα κλειδιά και κρυπτοκείμενα, τις πιθανότητες εμφάνισης των απλών κειμένων, και την πιθανότητα επιλογής κάθε κλειδιού – η τελευταία μπορεί να διαφέρει ανάλογα με το απλό κείμενο. Αποδείξτε ότι πράγματι το σύστημα διαθέτει τέλεια μυστικότητα.

6. Θεωρήστε το κρυπτοσύστημα `XOR` όπου δίνεται ένα κλειδί $K = K_0K_1 \cdots K_m-1$ όπου κάθε K_i είναι ένα byte. Ένα μήνυμα μήκους r bytes, έστω $K = M_0M_1 \cdots M_r - 1$ κρυπτογραφείται ως εξής: $C_i = M_i \oplus K_{i \bmod m}$. Δηλαδή κάθε byte του αρχικού κειμένου κρυπτογραφείται με κάποιο byte του κλειδιού με την πράξη XOR (bitwise) και το κλειδί επαναχρησιμοποιείται σε κάθε block από m bytes. Δείξτε πώς μπορείτε να αποκρυπτογραφήσετε το μήνυμα χρησιμοποιώντας ιδέες από τη μέθοδο του Δείκτη σύμπτωσης και τις ιδιότητες της πράξης XOR.

7. Μπορεί ένα κρυπτοσύστημα δημοσίου κλειδιού να έχει την ιδιότητα μη διακρισιμότητας IND-CCA;

8. Έχει το κρυπτοσύστημα Vigenére την ιδιότητα IND-CPA;
9. Είναι το κρυπτοσύστημα του Καίσαρα εύπλαστο (malleable); Εξηγήστε την απάντησή σας. Τι ισχύει για το Vigenére;
10. Αποδείξτε την αντίστροφη κατεύθυνση του Θεωρήματος 1.11, δηλαδή ότι ένα malleable κρυπτοσύστημα δεν έχει την ιδιότητα IND-CCA2.

1.6 Ηλεκτρονικό Υλικό

Στο τέλος κάθε κεφαλαίου θα παρουσιάζεται υλικό σε ηλεκτρονική μορφή που σκοπό έχει να βοηθήσει την καλύτερη κατανόηση των αντικειμένων του κεφαλαίου με την παρουσίαση τους σε διαδραστική μορφή.

- Χρήσιμα άρθρα:
  - Bruce Schneier's Blog
  - Memo to the Amateur Cipher Designer (`https://goo.gl/92TW36`)
  - Crypto Snake Oil (`https://goo.gl/FaFoSK`)
- Διαδραστικές Παρουσιάσεις - Video
  - Το πανεπιστήμιο του Rhode Island έχει συγκεντρώσει παρουσιάσεις κλασικών κρυπτοσυστημάτων στους παρακάτω συνδέσμους :
    * Κρυπτοσυστήματα Ολίσθησης
    * Κρυπτοσυστήματα Affine
    * Κρυπτοσυστήματα Αντικατάστασης
    * Vigenére
  - The BLACK Chamber, Διαδραστικές παρουσιάσεις κλασικών συστημάτων από τον Simon Singh, συγγραφέα ενός από τα πιο διάσημα βιβλία κρυπτογραφίας για το ευρύ κοινό
  - Προσομοιωτής Μηχανής Enigma
  - Αποσυναρμολόγηση Μηχανής Enigma
  - Ανακατασκευή μηχανής Bombe του Turing
  - Επίδειξη τέλειας μυστικότητας Shannon
  - Διαλέξεις για αποδείξεις μέσω κρυπτογραφικών αναγωγών
- Διαδραστικές Υλοποιήσεις

  - Sharky's Vigenére Cipher
  - Vigenére Cipher Codebreaker

- Κώδικας
  - Βιβλιοθήκη κλασικών κρυπτοσυστημάτων στο Sage
  - Μηχανή Enigma σε Javascript
  - Κρυπτογράφηση και κρυπτανάλυση κλασικών κρυπτοσυστημάτων σε Python

Βιβλιογραφία

[1] Ivan Damgard. A 'proof-reading' of some issues in cryptography. In *Automata, Languages and Programming*, pages 2–11. Springer, 2007.

[2] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Trans. Inf. Theor.*, 22(6):644–654, November 1976.

[3] Shafi Goldwasser and Silvio Micali. Probabilistic encryption & how to play mental poker keeping secret all partial information. In *STOC '82: Proceedings of the fourteenth annual ACM symposium on Theory of computing*, pages 365–377, New York, NY, USA, 1982. ACM Press.

[4] Neal Koblitz and Alfred J Menezes. Another look at "provable security". *Journal of Cryptology*, 20(1):3–37, 2007.

[5] Silvio Micali, Charles Rackoff, and Bob Sloan. The notion of security for probabilistic cryptosystems. *SIAM Journal on Computing*, 17(2):412–426, 1988.

[6] R.L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21:120–126, 1978.

[7] Claude E. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28(4):656–715, 1949.

Κεφάλαιο 2

Μαθηματικό Υπόβαθρο

Σε αυτό το κεφάλαιο Θα παρουσιάσουμε ορισμένα στοιχεία από την Θεωρία Αριθμών, την Θεωρία Ομάδων και την Θεωρία Πιθανοτήτων. Θα περιοριστούμε στις ελάχιστες γνώσεις που μας χρειάζονται για την κατανόηση της σύγχρονης κρυπτογραφίας. Οι έννοιες στις οποιες θα αναφερθούμε παίζουν σημαντικό ρόλο στον ορισμό, την υλοποίηση και την ασφάλεια των σύγχρονων κρυπτοσυστημάτων. Η ύλη του παρόντος κεφαλαίου στηρίχθηκε σε μεγάλο βαθμό στα [1], [5], [4], [6], [2], [3] και [7].

2.1 Θεωρία Αριθμών

Η Θεωρία Αριθμών ασχολείται κυρίως με τις ιδιότητες των φυσικών αριθμών (ή αλλιώς θετικών ακεραίων): $1, 2, 3, \ldots$. Θα αρχίσουμε παρουσιάζοντας τα βασικά (εισαγωγικά) στοιχεία της θεωρίας Αριθμών, δηλαδή την έννοια της διαιρετότητας, των πρώτων αριθμών, του μεγίστου κοινού διαιρέτη (Μέγιστος Κοινός Διαιρέτης (ΜΚΔ)) και της συνάρτησης Euler (ϕ).

2.1.1 Διαιρετότητα

Ορισμός 2.1. Αν $a, b \in \mathbb{Z}$ (ακέραιοι) λέμε ο "a διαιρεί τον b" αν $\exists c \in \mathbb{Z} : b = ca$ και συμβολίζουμε με $a \mid b$.

Η άρνηση του $a \mid b$ δηλαδή το γεγονός ότι το a δεν διαιρεί το b, συμβολίζεται με $a \nmid b$. Ο a λέγεται **διαιρέτης** του b και ο b πολλαπλάσιο του a. Ο a είναι **γνήσιος διαιρέτης** του b αν: $a \mid b \ \wedge \ 0 < a < b$. Ο a είναι **μη τετριμμένος διαιρέτης** του b: $a \mid b \ \wedge \ a \neq 1 \ \wedge \ a \neq b$.

Ο παραπάνω ορισμός της διαιρετότητας έχει τις παραπάνω συνέπειες ως ιδιότητες:

Πρόταση 2.2. *Έστω* $a, b, c, x, y \in \mathbb{Z}$

1. $a \mid 0$.
2. *Κάθε αριθμός μεγαλύτερος του 1 έχει τουλάχιστον δύο διαιρέτες : το 1 και τον εαυτό του.*
3. $a \mid b \;\Rightarrow\; a \mid (bc)$.
4. $a \mid b \;\wedge\; b \mid c \;\Rightarrow\; a \mid c$.
5. $a \mid b \wedge b \mid a \;\Rightarrow\; |a| = |b|$.
6. $a \mid b \wedge a \mid c \;\Rightarrow\; a \mid (b + c)$.
7. $a \mid b \wedge a \mid c \;\Rightarrow\; a \mid (bx + cy)$
8. $a \mid b \wedge b > 0 \;\Rightarrow\; a \leq b$

Παρακάτω ορίζουμε την ακέραια διαίρεση.

Πρόταση 2.3. *Διαίρεση: Για κάθε* $a, b \in \mathbb{Z}, b \neq 0$ *υπάρχουν* $q, r \in \mathbb{Z}$ *έτσι ώστε* $a = qb + r$ *και* $0 \leq r < |b|$. *Τα* q, r *είναι μοναδικά (για* $r = 0$ *έχουμε* $b \mid a$*) και λέγονται* πηλίκο *και* υπόλοιπο *αντίστοιχ.*

Απόδειξη. Ορίζουμε r τον μικρότερο μη αρνητικό όρο της εξής αριθμητικής προόδου: $\ldots, a - 3b, a - 2b, a - b, a, a + b, a + 2b, a + 3b, \ldots$

- Το r ικανοποιεί τις συνθήκες της πρότασης (r είναι της μορφής $a - qb$).
- Μοναδικότητα: Έστω $r, r'(q, q')$.
  Τότε: $0 = (q - q')b + (r - r') \;\Rightarrow\; r' - r = (q - q')b$
  αλλά: $|r' - r| < b$, άρα $r = r'$ και $q = q'$.

□

Παρατήρηση 1. Η πράξη 'mod' δίνει το υπόλοιπο της ακέραιας διαίρεσης δύο αριθμών, το r του παραπάνω ορισμού. Ισοδύναμα:

$$a \bmod b = a - \lfloor \frac{a}{b} \rfloor b$$

2.1.2 Πρώτοι Αριθμοί

Ορισμός 2.4. Πρώτος αριθμός λέγεται ένας ακέραιος μεγαλύτερος του 1 που δεν έχει άλλους διαιρέτες εκτός από το 1 και τον εαυτό του.

Ένας αριθμός που δεν είναι πρώτος λέγεται σύνθετος.
Για παράδειγμα:

- $2, 3, 5, \ldots, 1997, \ldots, 6469, \ldots$
- $(333 + 10^{793})10^{791} + 1$ (με 1585 ψηφία, παλίνδρομος βρέθηκε το 1987 από τον H. Dubner)
- $2^{1257787} - 1$ (με 378632 ψηφία βρέθηκε το 1996, από τους David Slowinski και Paul Gage)
- $2^{13466917} - 1$ (με 4053946 ψηφία βρέθηκε το 2001, από τον Michael Cameron, μέσω του GIMPS[1])
- $2^{43112609} - 1$ (με 12978189 ψηφία βρέθηκε το 2008, από τον Hans-Michael Elvenich, μέσω του GIMPS)
- $2^{57885161} - 1$ (με 17425170 ψηφία βρέθηκε το 2013, από τον Curtis Cooper, μέσω του GIMPS)
- $2^{74207281} - 1$ (με 22338618 ψηφία βρέθηκε το 2015, από τον Curtis Cooper, μέσω του GIMPS)

Πρόταση 2.5. *Κάθε ακέραιος μεγαλύτερος του 1 είναι είτε πρώτος είτε γινόμενο πρώτων αριθμών.*

Θεώρημα 2.6 (Ευκλείδη)**.** *Οι πρώτοι είναι άπειροι σε πλήθος.*

Απόδειξη. Έστω ότι οι πρώτοι είναι πεπερασμένοι σε πλήθος, δηλ. $p_1, p_2, \ldots, p_n$ τότε ο αριθμός $p_1 p_2 \ldots p_n + 1$ δε διαιρείται από κανένα πρώτο παρά μόνο από το 1 και τον εαυτό του, άρα είναι πρώτος, κάτι που είναι άτοπο. □

Θεώρημα 2.7 (Θεμελιώδες Θεώρημα Αριθμητικής)**.** *Κάθε αριθμός μπορεί να γραφεί με μοναδικό τρόπο σε γινόμενο πρώτων αριθμών (όχι απαραίτητα διαφορετικών ανά δύο).*

[1] Great Internet Mersenne Prime Search, διαδικτυακό κατανεμημένο πρόγραμμα αναζήτησης πρώτων αριθμών Mersenne, δηλ. της μορφής $2^p - 1$ όπου p πρώτος.

Η απόδειξη του Θεμελιώδους Θεωρήματος στηρίζεται στην μαθηματική επαγωγή και στην παρακάτω πρόταση. Η πλήρης απόδειξη μπορεί να βρεθεί στα βασικά συγγράμματα θεωρίας αριθμών, αλλά συνιστάται και ως άσκηση για τον ενδιαφερόμενο αναγνώστη.

Πρόταση 2.8. *p πρώτος και $p \mid ab \Rightarrow p \mid a \vee p \mid b$*

2.1.3 Μέγιστος Κοινός Διαιρέτης

Ορισμός 2.9. Ο μέγιστος κοινός διαιρέτης των a, b είναι ο μεγαλύτερος ακέραιος που διαιρεί τον a και τον b. συμβολίζουμε με (a, b), ή με ΜΚΔ(a, b), ή με gcd(a, b).

Ορισμός 2.10. *Σχετικά πρώτοι (coprime)* λέγονται οι a, b αν ισχύει $(a, b) = 1$.

Παρακάτω φαίνονται κάποιες ιδιότητες του ΜΚΔ:

Παρατήρηση 2.

- $(ma, mb) = m(a, b)$
- $(a, m) = (b, m) = 1 \Rightarrow (ab, m) = 1$
- $(a, b) = (a, b + ax)$
- p πρώτος $\wedge\, p \mid ab \Rightarrow p \mid a \vee p \mid b$
- $m \mid a \wedge n \mid a \wedge (m, n) = 1 \Rightarrow mn \mid a$

Πολλές από τις παραπάνω ιδιότητες, και αρκετές άλλες, μπορούν να αποδειχθούν με τη χρήση του παρακάτω ισχυρού θεωρήματος, που μεταξύ άλλων εξηγεί την ορθότητα του Ευκλείδειου αλγορίθμου για εύρεση ΜΚΔ αλλά και τον τρόπο υπολογισμού αντιστρόφου σε αριθμητική modulo, μέσω του Επεκτεταμένου Ευκλείδειου Αλγόριθμου, όπως θα δούμε αργότερα. Με απλά λόγια το θεώρημα αυτό λέει ότι ο ΜΚΔ δύο αριθμών είναι ο *μικρότερος θετικός ακέραιος που γράφεται σαν γραμμικός συνδυασμός τους.*

Θεώρημα 2.11 (ΜΚΔ). *Για κάθε $a, b \in \mathbb{Z}$ ισχύει $(a, b) = \min\{xa + yb \mid x, y \in \mathbb{Z}, xa + yb \geq 0\}$*

Απόδειξη. Έστω $d = \min\{xa + yb \mid x, y \in \mathbb{Z}, xa + yb \geq 0\}$. Θα δείξουμε ότι $d \mid a$ και $d \mid b$.

Έστω $\kappa, \lambda \in \mathbb{Z}$ τ.ώ. $d = \kappa a + \lambda b$. Θα δείξουμε ότι $d \mid a$.

Έστω $d \nmid a$. Τότε υπάρχουν $q, r \in \mathbb{Z}$ τέτοια ώστε $a = qd + r, \quad 0 < r < d,$.

Επομένως $r = a - qd = a - q(\kappa a + \lambda b) = (1 - q\kappa)a + (-\lambda q)b$, οπότε $r \in \{xa + yb \mid x, y \in \mathbb{Z}, xa + yb \geq 0\}$ και $r < d$, άτοπο.

Όμοια δείχνουμε $d \mid b$. Επομένως ο d είναι κοινός διαιρέτης των a, b.

Έστω τώρα ένας άλλος κοινός διαιρέτης d' των a, b. Τότε $a = c_1 d', b = c_2 d'$, οπότε:

$$d = \kappa c_1 d' + \lambda c_2 d' \Rightarrow d' \mid d \Rightarrow d' \leq d$$

Επομένως ο d είναι ο μέγιστος κοινός διαιρέτης των a, b. □

Ένα χρήσιμο πόρισμα είναι το εξής:

Πόρισμα 2.12. *Αν ένας κοινός διαιρέτης των a, b γράφεται σαν γραμμικός συνδυασμός τους τότε είναι ο μέγιστος κοινός διαιρέτης τους.*

2.1.4 Η συνάρτηση Euler

Ορισμός 2.13. *Η συνάρτηση ϕ του Euler.* Ορίζουμε $\phi(n)$ να είναι το πλήθος των αριθμών από το 1 μέχρι και το n που είναι σχετικά πρώτοι με τον n.

Πρόταση 2.14. *Ιδιότητες της συνάρτησης ϕ:*

1. $\phi(p) = p - 1$ *όπου p πρώτος.*
2. $\phi(p^a) = p^a(1 - \frac{1}{p})$ *όπου p πρώτος.*
3. $\phi(mn) = \phi(m)\phi(n)$ *για m,n σχετικά πρώτους.*

Παρατήρηση: Από τα (2) (3) είναι φανερό ότι αν είναι γνωστή η ανάλυση ενός αριθμού n σε πρώτους παράγοντες τότε είναι εύκολο να υπολογίσουμε το $\phi(n)$.

Πόρισμα 2.15. *Για κάθε $n \in \mathbb{N}$ ισχύει $\phi(n) = n \prod_{p|n}(1 - \frac{1}{p})$.*

Πρόταση 2.16. *Έστω $n = pq$, p, q πρώτοι, $p \neq q$. Γνώση των δύο πρώτων p, q είναι ισοδύναμη με γνώση του $\phi(n)$ (οι αντίστοιχοι υπολογισμοί χρειάζονται $O(\log^3 n)$ bit operations).*

Θεώρημα 2.17 (Euler)**.** $\forall n \geq 1; \sum_{d|n} \phi(d) = n$

2.1.5 Σχέση ισοτιμίας (congruence)

Ορισμός 2.18. Δύο αριθμοί a, b λέγονται *ισότιμοι modulo* m αν $m \mid (a - b)$. Συμβολικά γράφουμε $a \equiv b \pmod{m}$. Ισοδύναμα μπορούμε να πούμε ότι οι a και b αν διαιρεθούν με το m δίνουν το ίδιο υπόλοιπο.

Παρατήρηση 3. $a \bmod m = b \bmod m$ αν και μόνο αν $a \equiv b \pmod{m}$ για $a, b \in \mathbb{Z}$. Σημείωση: Η πράξη $a \bmod m$ (δηλ. το mod χωρίς παρένθεση) συμβολίζει το υπόλοιπο της ευκλείδειας διαίρεσης του ακεραίου a με το m.

Παρατήρηση 4. Στη βιβλιογραφία θα βρείτε και άλλους συμβολισμούς για την ισοτιμία, π.χ. $a = b \pmod{m}$ ή και $a = b\,(m)$.

Κλάσεις ισοτιμίας, ο δακτύλιος $\mathbb{Z}_m$

Η σχέση της ισοτιμίας modulo m είναι σχέση ισοδυναμίας για τους ακεραίους και τους χωρίζει σε m κλάσεις $C_0, C_1, C_2, \ldots, C_{m-1}$.
Κάθε κλάση C_k περιέχει τους ακεραίους που αφήνουν υπόλοιπο k αν διαιρεθούν με το m. π.χ. για $m = 5$ έχουμε τις εξής κλάσεις:

$$\begin{aligned} C_0 &= \{0, 5, 10, 15, 20, 25, ..., 5k, ...\} \\ C_1 &= \{1, 6, 11, 16, 21, 26, ..., 5k+1, ...\} \\ C_2 &= \{2, 7, 12, 17, 22, 27, ..., 5k+2, ...\} \\ C_3 &= \{3, 8, 13, 18, 23, 28, ..., 5k+3, ...\} \\ C_4 &= \{4, 9, 14, 19, 24, 29, ..., 5k+4, ...\} \end{aligned}$$

Το σύνολο των κλάσεων $\{C_0, C_1, C_2, \ldots, C_{m-1}\}$ που για απλότητα θα γράφουμε ως $\{0, 1, 2, \ldots, m-1\}$, θα το συμβολίζουμε με $\mathbb{Z}_m$ και το ονομάζουμε *σύνολο ακεραίων modulo m*.

Ορισμός 2.19. Οι πράξεις που ορίζονται μεταξύ των κλάσεων ορίζονται με τον εξής τρόπο: $C_k + C_j = C_{(k+j) \bmod m}$ και $C_k \cdot C_j = C_{kj \bmod m}$

Παρατήρηση 5. Με τις πράξεις $+, \cdot$ (όπως ορίστηκαν παραπάνω) μπορεί να δειχθεί ότι ο $(\mathbb{Z}_m, +, \cdot)$ είναι αντιμεταθετικός δακτύλιος με μοναδιαίο στοιχείο.

Ένας ισοδύναμος τρόπος να ορίσουμε το $\mathbb{Z}_m$ είναι $\mathbb{Z}_m = \{a \bmod m \mid a \in \mathbb{Z}\}$ και τότε οι πράξεις είναι ακριβώς η πρόσθεση και ο πολλαπλασιασμός modulo m μεταξύ ακεραίων,

$$(a \bmod m + b \bmod m) \bmod m = (a + b) \bmod m \,,$$

$$\big((a \bmod m) \cdot (b \bmod m)\big) \bmod m = (a \cdot b) \bmod m \,,$$

έχουμε δηλαδή στην ισοτιμία εκτός της ισοδυναμίας και ιδιότητες ομομορφισμού.

Παρατήρηση 6. Γενικά μπορούμε να εφαρμόσουμε στους ακέραιους modulo m πρόσθεση, αφαίρεση, πολλαπλασιασμό θεωρώντας τους απλούς ακέραιους αρκεί πάντα στο τέλος να παίρνουμε σαν αποτέλεσμα το υπόλοιπο της διαίρεσης με το m. Βέβαια όπως θα δούμε και αργότερα αυτός ο τρόπος δεν είναι πάντα ο καλύτερος για υλοποίηση σε υπολογιστή: όταν έχουμε πράξεις με πολύ μεγάλους αριθμούς, και θέλουμε το αποτέλεσμα modulo m, με το m σημαντικά μικρότερο από τους αρχικούς αριθμούς ή/και από τα ενδιάμεσα αποτελέσματα, τότε συμφέρει να εφαρμόζουμε στους αρχικούς αριθμούς και στα ενδιάμεσα αποτελέσματα την πράξη mod$m$, να δουλεύουμε δηλαδή διαρκώς στον δακτύλιο $\mathbb{Z}_m$.

Ιδιότητες ισοτιμιών

Πρόταση 2.20. *Ισχύουν:*

1. *Αν* $a \equiv b \pmod{m}$ *τότε* $a \equiv b \pmod{d}$ *για κάθε* $d \mid m$.
2. *Αν* $a \equiv b \pmod{m}$ *και* $a \equiv b \pmod{n}$ *με* $(m, n) = 1$ *τότε* $a \equiv b \pmod{mn}$.

Τα στοιχεία του $\mathbb{Z}_m$ δεν έχουν όλα αντίστροφο ως προς τον πολλαπλασιασμό (modulo m). Μπορεί να δειχθεί ότι ικανή και αναγκαία συνθήκη για να έχει ένα στοιχείο του $\mathbb{Z}_m$ αντίστροφο είναι να είναι σχετικά πρώτος με τον m.

Πρόταση 2.21. $(a, m) = 1$ *αν και μόνο αν* $\exists c \in \mathbb{Z}_m : a \cdot c \equiv 1 \pmod{m}$.

Απόδειξη. Αν ισχύει $(a, m) = 1$ τότε $\exists x, y : 1 = xa + ym$. Παίρνοντας το υπόλοιπο της διαίρεσης με το m των δύο μελών έχουμε ότι

$$1 \bmod m = xa \bmod m + ym \bmod m \Rightarrow$$

$$(x \bmod m)(a \bmod m) = 1 \bmod m \ .$$

Άρα το $a \pmod{m}$ έχει αντίστροφο το $x \pmod{m}$. Όπως θα δούμε αργότερα το x μπορεί να υπολογιστεί σε πολυωνυμικό χρόνο με την επέκταση του αλγορίθμου του Ευκλείδη.

Αντίστροφα αν το $a \pmod{m}$ έχει αντίστροφο το $x \pmod{m}$ τότε ισχύει ότι το $ax \equiv 1 \pmod{m} \Rightarrow m \mid (ax - 1)$. Αν $(a, m) = d > 1$ τότε $d \mid (ax - 1) \Rightarrow d \mid 1$, άτοπο αφού $d > 1$. □

Ορισμός 2.22. Το υποσύνολο του $\mathbb{Z}_m$ που κάθε στοιχείο του είναι σχετικά πρώτο με το m το συμβολίζουμε με $U(\mathbb{Z}_m)$ (units). Δηλαδή $U(\mathbb{Z}_m) = \{a \in \mathbb{Z}_m : (a, m) = 1\}$.

Παρατήρηση 7. Ο πληθικός αριθμός του $U(\mathbb{Z}_m)$ είναι $\phi(m)$.

2.2 Θεωρία Ομάδων

2.2.1 Βασικές Έννοιες

Ορισμός 2.23. *Ομάδα* λέγεται ένα ζεύγος $(G, *)$ όπου G είναι ένα σύνολο και $*$ πράξη: $* : G \times G \to G$ ώστε να ισχύει:

1. $\forall\, a, b \in G : a * b \in G$ (G κλειστότητα)

2. $\forall\, a, b, c \in G : a * (b * c) = (a * b) * c$ (προσεταιριστική)

3. $\exists\, \mathbf{1} \in G, \forall\, a \in G : a * \mathbf{1} = a$ (ουδέτερο στοιχείο)

4. $\forall a\, \exists a^{-1} \in G : a * a^{-1} = e$ (αντίστροφο στοιχείο)

Ορισμός 2.24. *Αντιμεταθετική ομάδα* λέγεται η ομάδα $(G, *)$ όπου για την πράξη $*$ ισχύει επιπλέον:

1. $\forall\, a, b \in G : a * b = b * a$ (αντιμεταθετική ιδιότητα)

Παραδείγματα: $(Q, +)(R, +), (Q - \{0\}, \cdot), (\mathbb{Z}_m, +), (U(\mathbb{Z}_m), \cdot)$.

Ορισμός 2.25. Έστω $(A, \oplus)$ και $(B, \otimes)$ δύο αλγεβρικές δομές. Μια συνάρτηση $f : A \to B$ λέγεται *ομομορφισμός* αν $f(a \oplus b) = f(a) \otimes f(b)$, *μονομορφισμός* εάν επιπλέον είναι *1-1*, *επιμορφισμός* αν είναι *επί*. *Ισομορφισμός* λέγεται ένας ομομορφισμός που είναι 1-1 και επί.

Ορισμός 2.26. Υποομάδα μιας ομάδας $(G, *)$ λέγεται το ζεύγος $(S, *)$ τέτοιο ώστε $S \subseteq G$ και $(S, *)$ είναι ομάδα.

Θεώρημα 2.27 (Κλειστότητα υποομάδας)**.** *Έστω ομάδα $(G, *)$ και $S \subseteq G$, τέτοιο ώστε S κλειστό ως προς $*$. Τότε το $(S, *)$ είναι υποομάδα.*

Η απόδειξη του παραπάνω θεωρήματος αφήνεται ως άσκηση. Έχει εφαρμογές σε αποδείξεις σημαντικών θεωρημάτων της θεωρίας αριθμών καθώς και ιδιοτήτων αλγορίθμων (βλ. και Θ. Lagrange παρακάτω).

Πολλές φορές είναι χρήσιμο να αναπαραστήσουμε την επαναλαμβανόμενη εφαρμογή της πράξης μιας ομάδας σε κάποιο στοιχείο της. Ο συμβολισμός στην περίπτωση αυτή εξαρτάται από το αν η ομάδα είναι πολλαπλασιαστική ή προσθετική.

Ορισμός 2.28 (Δυνάμεις σε Ομάδες)**.** Έστω μια ομάδα G ένα στοιχείο της g και ένας θετικός ακέραιος m. Σε περίπτωση που η ομάδα είναι προσθετική ορίζουμε:

$$mg = g + \cdots + g \quad \textit{(m φορές)}$$

Σε περίπτωση που η ομάδα είναι προσθετική ορίζουμε:

$$g^m = g \cdots g \quad \textit{(m φορές)}$$

Παρατήρηση 8.

- Ο παραπάνω συμβολισμός συμπεριφέρεται και στις δύο περιπτώσεις κατά τα γνωστά μας από τους φυσικούς αριθμούς.
- Το αντίστροφο πρόβλημα από την ύψωση σε δύναμη αφορά την εύρεση του m για το οποίο ισχύει $y = g^m$ ή $y = mg$. Καθότι ο m είναι ακέραιος ονομάζεται το πρόβλημα της *Εύρεσης του Διακριτού Λογαρίθμου*.

Ορισμός 2.29. Μια ομάδα $(G, *)$ λέγεται κυκλική αν υπάρχει στοιχείο $g \in (G, *)$ με την ιδιότητα $\forall x \in G \ \exists y : x = g^y$. Το στοιχείο αυτό το ονομάζουμε και γεννήτορα της $(G, *)$.

Ορισμός 2.30. Τάξη ενός στοιχείου a μιας ομάδας G είναι το μικρότερο y έτσι ώστε $a^y = e$.

Πρόταση 2.31. *Αν ένα στοιχείο είναι γεννήτορας της $(G, *)$ τότε ισχύει ότι η τάξη του είναι $|G|$.*

Ορισμός 2.32. Έστω H υποομάδα της $(G, *)$ και $a \in G$. Το $H * a = \{h * a \mid h \in H\}$ λέγεται δεξί σύμπλοκο (coset) της H στη G.

Πρόταση 2.33. *Το G/H (= σύνολο των δεξιών cosets της H στην G) είναι ομάδα και λέγεται* quotient group *με πράξη $(H * a) \circledast (H * b) = H * (a * b)$.*

Θεώρημα 2.34 (Lagrange)**.** *Αν G είναι πεπερασμένη ομάδα και H είναι υποομάδα της G, τότε $|G| = |G/H| \cdot |H|$, δηλαδή το $|H|$ διαιρεί το $|G|$.*

Η απόδειξη στηρίζεται στο γεγονός ότι δύο σύμπλοκα είτε ταυτίζονται είτε είναι ξένα μεταξύ τους.

Πόρισμα 2.35. *Αν $(S, *)$ υποομάδα της (πεπερασμένης) ομάδας $(G, *)$ και $S \neq G$ τότε $|S| \leq |G|/2$.*

Το πόρισμα αυτό, σε συνδυασμό με το Θεώρημα 2.27 (κλειστότητα υποομάδας) μας λέει επιπλέον ότι *κάθε ιδιότητα στοιχείων μιας ομάδας που είναι κλειστή ως προς την πράξη της ομάδας, είτε ισχύει για όλη την ομάδα, είτε για τα μισά το πολύ στοιχεία της.*

Αυτό έχει άμεση εφαρμογή στην απόδειξη ορθότητας σημαντικών αλγορίθμων, όπως του τεστ Fermat και του τεστ Miller-Rabin για έλεγχο πρώτων αριθμών.

Ορισμός 2.36. Δακτύλιος λέγεται μια τριάδα $(R, +, *)$ όπου το $(R, +)$ είναι αντιμεταθετική ομάδα και ισχύει η προσεταιριστική ιδιότητα για την πράξη $*$ και η επιμεριστική ιδιότητα της $*$ ως προς την $+$:

$$\forall a, b, c \in R \begin{cases} a * (b + c) = a * b + a * c, \\ (b + c) * a = b * a + c * a \end{cases}$$

Ο δακτύλιος λέγεται "αντιμεταθετικός δακτύλιος" αν η πράξη $*$ έχει την αντιμεταθετική ιδιότητα ή "δακτύλιος με μονάδα" αν υπάρχει μοναδιαίο στοιχείο ως προς την $*$.

Ορισμός 2.37. Σώμα λέγεται μια τριάδα $(F, +, *)$ όπου το $(F, +)$ και $(F - \{e_+\}, *)$ είναι αντιμεταθετικές ομάδες και ισχύει η επιμεριστική ιδιότητα της * ως προς την +. Με e_+ συμβολίζουμε το ουδέτερο ως προς την πράξη +.

Το πλήθος των στοιχείων του συνόλου F ονομάζεται *τάξη* του σώματος. Ένα σώμα πεπερασμένης τάξης ονομάζεται *πεπερασμένο σώμα*. Ένα σώμα τάξης m υπάρχει αν και μόνον αν το m είναι δύναμη πρώτου, δηλ. $m = p^n$ με $n \in \mathbb{N}$ και p πρώτο. Το p ονομάζεται χαρακτηριστική του σώματος. (Στο εξής θα συμβολίζουμε με $\oplus$ την πρόσθεση σε ένα σώμα χαρακτηριστικής 2.)

Δύο σώματα της ίδιας τάξης είναι ισομορφικά: αναπαριστούν την ίδια ακριβώς αλγεβρική δομή διαφέροντας μόνο στην ονομασία των στοιχείων. Με άλλα λόγια, για κάθε δύναμη πρώτου υπάρχει ακριβώς ένα πεπερασμένο σώμα, το οποίο συμβολίζουμε $GF(p^n)$.

Τα στοιχεία του $GF(p)$, όπου p πρώτος, μπορούν να παρασταθούν από τους ακεραίους $0, 1, \ldots, p-1$, ενώ οι δύο πράξεις του σώματος είναι τότε η "πρόσθεση ακεραίων modulo p" και ο "πολλαπλασιασμός ακεραίων modulo p". Σε σώματα τάξης που δεν είναι πρώτος, η πρόσθεση και ο πολλαπλασιασμός δεν μπορούν να ορισθούν modulo κάποιου αριθμού και είναι αναγκαία η εισαγωγή πιο περίπλοκων αναπαραστάσεων των στοιχείων του $GF(p^n)$ με $n > 1$. Εδώ αναπαριστούμε τα στοιχεία του $GF(p^n)$ με πολυώνυμα πάνω στο $GF(p)$.

Παρατήρηση 9. Το $(U(\mathbb{Z}_m), \cdot)$ είναι αντιμεταθετική ομάδα. Αν $m = p$ πρώτος τότε $U(\mathbb{Z}_p) = \mathbb{Z}_p - \{0\} = \mathbb{Z}_p^*$ και σε αυτήν την περίπτωση το $\mathbb{Z}_p$ είναι σώμα (τότε συμβολίζεται με $GF(p)$ ή F_p).

2.2.2 Πολυώνυμα

Ένα πολυώνυμο πάνω από ένα σώμα F είναι μία έκφραση της μορφής

$$b(x) = b_{n-1}x^{n-1} + b_{n-2}x^{n-2} + \cdots + b_2x^2 + b_1x + b_0,$$

όπου το x καλείται *απροσδιόριστη* του πολυωνύμου και τα $b_i \in F$ οι *συντελεστές* αυτού. Θα θεωρούμε τα πολυώνυμα ως αφηρημένες οντότητες μονάχα, οι οποίες σε καμία περίπτωση δεν αποτιμώνται.

Ο *βαθμός* ενός πολυωνύμου ισούται με λ αν $b_j = 0, \forall j > \lambda$ και το λ είναι ο μικρότερος αριθμός μ' αυτή την ιδιότητα. Το σύνολο των πολυωνύμων πάνω σ' ένα σώμα F συμβολίζεται με $F[x]$. Το σύνολο των πολυωνύμων πάνω σ' ένα σώμα F, τα οποία έχουν βαθμό μικρότερο από λ, συμβολίζεται με $F[x]|_\lambda$.

Στη μνήμη του υπολογιστή, τα πολυώνυμα του $F[x]|_\lambda$, όπου F ένα πεπερασμένο σώμα, είναι εύκολο να αποθηκευτούν αποθηκεύοντας τους συντελεστές σ' ενα string μήκους λ.

Παράδειγμα 2. *Έστω ότι το σώμα F είναι το $GF(2)$ και έστω $\lambda = 8$. Τα πολυώνυμα είναι βολικό να αποθηκευτούν ως ακολουθίες των 8 bits ή bytes:*

$$b(x) \mapsto b_7b_6b_5b_4b_3b_2b_1b_0$$

bit pattern	character
0000	0
0001	1
0010	2
0011	3

bit pattern	character
0100	4
0101	5
0110	6
0111	7

bit pattern	character
1000	8
1001	9
1010	a
1011	b

bit pattern	character
1100	c
1101	d
1110	e
1111	f

Συχνά για τα bytes θα χρησιμοποιούμε τον δεκαεξαδικό συμβολισμό:

Παράδειγμα 3. *Το πολυώνυμο $x^6 + x^4 + x^2 + x + 1$ στο $GF(2)|_8$ αντιστοιχεί στο string 01010111, ή στον δεκαεξαδικό 57.*

Ορίζουμε τις παρακάτω πράξεις στα πολυώνυμα.

Πρόσθεση. Η πρόσθεση πολυωνύμων γίνεται προσθέτοντας τους συντελεστές των αντίστοιχων δυνάμεων του x. Η πρόσθεση των συντελεστών γίνεται στο σώμα F πάνω από το οποίο είναι ορισμένα τα πολυώνυμα:

$$c(x) = a(x) + b(x) \Leftrightarrow c_i = a_i + b_i, \;\; 0 \leq i \leq n.$$

Είναι εύκολο να διαπιστώσει κανείς ότι η δομή $\langle F[x]|_\lambda, + \rangle$ είναι αβελιανή ομάδα με ουδέτερο στοιχείο το πολυώνυμο που έχει όλους τους συντελεστές ίσους με το μηδενικό στοιχείο του F.

Παράδειγμα 4. *Έστω F το σώμα $GF(2)$. Έχουμε:*

$$\begin{gathered}(x^6 + x^4 + x^2 + x + 1) \oplus (x^7 + x + 1) = \\ x^7 + x^6 + x^4 + x^2 + (1 \oplus 1)x + (1 \oplus 1) = \\ x^7 + x^6 + x^4 + x^2.\end{gathered}$$

Χρησιμοποιώντας bytes θα είχαμε: $01010111 \oplus 10000011 = 11010100$ *ή, με δεκαεξαδικό συμβολισμό,* $57 \oplus 83 = d4$. Είναι φανερό ότι εκτελούμε αποκλειστική διάζευξη κατά ψηφίο (bitwise XOR).

Πολλαπλασιασμός. Ο πολλαπλασιασμός πολυωνύμων είναι πράξη προσεταιριστική, αντιμεταθετική, επιμεριστική ως προς την πρόσθεση και έχει ουδέτερο

στοιχείο (το $u(x) = 1$). Προκειμένου να κάνουμε το $F[x]|_\lambda$ κλειστό ως προς τον πολλαπλασιασμό επιλέγουμε ένα πολυώνυμο $m(x)$ βαθμού λ που το ονομάζουμε *πολυώνυμο αναγωγής*.

Ο πολλαπλασιασμός δύο πολυωνύμων $a(x)$ και $b(x)$ ορίζεται τότε ως το αλγεβρικό τους γινόμενο modulo το πολυώνυμο $m(x)$.

Παράδειγμα 5. *Έστω F το σώμα $GF(2)$. Αν $m(x) = x^8 + x^4 + x^3 + x + 1$ έχουμε:*

$(x^6 + x^4 + x^2 + x + 1) \times (x^7 + x + 1) =$
$(x^{13} + x^{11} + x^9 + x^8 + x^7) \oplus (x^7 + x^5 + x^3 + x^2 + x) \oplus (x^6 + x^4 + x^2 + x + 1) =$
$x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1$

Παρατηρούμε ότι

$x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1$
$= (x^8 + x^4 + x^3 + x + 1) \times (x^5 + x^3) \oplus (x^7 + x^6 + 1)$
$= m(x) \times (x^5 + x^3) \oplus (x^7 + x^6 + 1)$

Επομένως

$x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1$
$\equiv x^7 + x^6 + 1 \pmod{x^8 + x^4 + x^3 + x + 1}$.

Αν επιλέξουμε το $m(x)$ να είναι *ανάγωγο* στο σώμα F — δηλαδή να μην υπάρχουν δυο πολυώνυμα του $F[x]$ με βαθμό θετικό που το γινόμενο τους να ισούται με $m(x)$ — αποδεικνύεται ότι κάθε στοιχείο του $F[x]|_\lambda$ έχει αντίστροφο στο $F[x]|_\lambda$ το οποίο υπολογίζουμε χρησιμοποιώντας τον επεκτεταμένο ευκλείδιο αλγόριθμο. Έστω $a(x)$ ένα πολυώνυμο του οποίου αναζητάμε το αντίστροφο. Ο επεκτεταμένος ευκλείδιος αλγόριθμος μας δίνει δύο πολυώνυμα $b(x)$ και $c(x)$ τέτοια ώστε:

$$a(x) \times b(x) + m(x) \times c(x) = \text{gcd}(a(x), m(x)).$$

Επειδή $\text{gcd}(a(x), m(x)) = 1$ *ανν* το $m(x)$ είναι ανάγωγο, προκύπτει ότι

$$a(x) \times b(x) \equiv 1 \pmod{m(x)}$$

και επομένως το $b(x)$ είναι το αντίστροφο του $a(x)$.
Συνοψίζοντας, αν το F είναι το $GF(p)$ και το $m(x)$ είναι ανάγωγο στο F, τότε η δομή $\langle F[x]|_v, +, \cdot\rangle$ είναι σώμα με p^n στοιχεία, δηλαδή $\langle F[x]|_v, +, \cdot\rangle = GF(p^n)$

Πολλαπλασιασμός με σταθερό πολυώνυμο. Έστω $b(x)$ σταθερό πολυώνυμο βαθμού 3:

$$b(x) = b_0 + b_1 x + b_2 x^2 + b_3 x^3$$

και $c(x), d(x)$ πολυώνυμα με συντελεστές c_i και d_i αντίστοιχα $(0 \leq i < 4)$. Αν $d(x) = b(x) \times c(x)$, εκτελώντας τις πράξεις έχουμε ότι οι συντελεστές του $d(x)$ δίνονται από το γινόμενο πινάκων:

$$\begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} b_0 & b_3 & b_2 & b_1 \\ b_1 & b_0 & b_3 & b_2 \\ b_2 & b_1 & b_0 & b_3 \\ b_3 & b_2 & b_1 & b_0 \end{bmatrix} \times \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

2.3 Σημαντικά Θεωρήματα Θεωρίας Αριθμών

2.3.1 Μικρό Θεώρημα Fermat

Θεώρημα 2.38 (Μικρό Θεώρημα Fermat)**.**

$$\forall a \in \mathbb{Z},\ \forall \text{prime } p \not| a: \quad a^{p-1} \equiv 1 \pmod{p}$$

Απόδειξη. Έστω ένα $a \in \mathbb{Z}$ με $p \not| a$. Τότε το a αντιστοιχεί σε ένα στοιχείο του $\mathbb{Z}_p$ διαφορετικό του 0. Το στοιχείο αυτό το συμβολίζουμε με a. Τότε τα στοιχεία,

$$a \cdot 1, a \cdot 2, \ldots, a \cdot (p-1)$$

είναι διαφορετικά ανά δύο στην ομάδα $\mathbb{Z}_p - \{0\}$. Πράγματι αν $i \cdot a = j \cdot a \pmod{p}$ τότε έχουμε

$$p \mid (a \cdot i - a \cdot j) \Rightarrow p \mid a(i-j) \Rightarrow p \mid a \vee p \mid (i-j)$$

Όμως αφού το a είναι διαφορετικό του 0 στον $\mathbb{Z}_p$ δεν ισχύει ότι $p \mid a$ άρα $p \mid (i-j)$ συνεπώς $i = j \pmod{p}$.

Από τα παραπάνω έχουμε ότι αφού τα στοιχεία $a \cdot 1, a \cdot 2, \ldots, a \cdot (p-1)$ είναι διαφορετικά ανά δύο θα αποτελούν μια μετάθεση των στοιχείων της ομάδας $(\mathbb{Z}_p - \{0\}, \cdot)$. Άρα,

$$a1 \cdot a2 \cdot \ldots \cdot a(p-1) = 1 \cdot 2 \cdot \ldots \cdot (p-1) \pmod{p}$$

$$a^{p-1} \cdot 1 \cdot 2 \cdot \ldots \cdot (p-1) = 1 \cdot 2 \cdot \ldots \cdot (p-1) \pmod{p}$$

Τα στοιχεία $1, 2, \ldots, (p-1)$ έχουν αντίστροφο στο $\mathbb{Z}_p$ επομένως έχουμε ότι $a^{p-1} = 1 \pmod{p}$. □

Το μικρό Θεώρημα του Fermat είναι ειδική περίπτωση ενός πολύ πιο γενικού θεωρήματος από τη θεωρία ομάδων, που βασίζεται στο Θεώρημα Lagrange.

Θεώρημα 2.39. *Αν $(G, \cdot)$ πεπερασμένη ομάδα τότε $\forall a \in G$ ισχύει $a^{|G|} = e$ όπου e το ουδέτερο στοιχείο της ομάδας.*

Σαν πόρισμα προκύπτει το μικρό θεώρημα Fermat, εφαρμόζοντας το παραπάνω θεώρημα για την ομάδα $(\mathbb{Z}_p - \{0\}, \cdot)$. Επίσης εφαρμόζοντας για την ομάδα $(U(\mathbb{Z}_m), \cdot)$ έχουμε το εξής:

Πόρισμα 2.40. *(Euler)* $\forall a \in \mathbb{Z}, (a, m) = 1$ *ισχύει* $a^{\phi(m)} \equiv 1 \pmod{m}$.

Πόρισμα 2.41. *Αν ο πρώτος* p *δε διαιρεί τον* a *και* $n = m \pmod{p-1}$ *τότε* $a^n = a^m \pmod{p}$.

Θεώρημα 2.42 (Wilson)**.** p *πρώτος* $\Rightarrow (p-1)! = -1 \pmod{p}$

2.3.2 Κινέζικο Θεώρημα Υπολοίπων

Θεώρημα 2.43 (Κινέζικο Θεώρημα Υπολοίπων, ΚΘΥ ή Chinese Remainder Theorem, CRT)**.** *Έστω ένα σύστημα εξισώσεων*

$$\begin{aligned} x &= a_1 \pmod{m_1} \\ x &= a_2 \pmod{m_2} \\ &\vdots \\ x &= a_k \pmod{m_k} \end{aligned}$$

ώστε $(m_i, m_j) = 1$ *για* $i \neq j$*. Τότε το σύστημα έχει μοναδική λύση στο δακτύλιο* $\mathbb{Z}_M$ *όπου* $M = m_1 m_2 \dots m_k$*. Ισοδύναμα μπορούμε να πούμε ότι το σύστημα έχει άπειρες λύσεις και αν* s_1, s_2 *δύο λύσεις ισχύει* $s_1 = s_2 \pmod{M}$.

Απόδειξη. Έστω $M_i = \frac{M}{m_i}$ τότε έχουμε ότι $(M_i, m_i) = 1$ και αν θεωρήσουμε το M_i στοιχείο του $\mathbb{Z}_{m_i}$ τότε είναι αντιστρέψιμο. Επομένως υπάρχει ένα στοιχείο N_i με $N_i \cdot M_i = 1 \pmod{m_i}$ για κάθε i. Παρατηρούμε επίσης ότι για κάθε $i \neq j$ $M_i \cdot N_i = 0 \pmod{m_j}$. Έστω

$$y = \sum_{i=1}^{k} N_i \cdot M_i \cdot a_i$$

Η y είναι η ζητούμενη λύση. Πράγματι

$$\forall i: \quad y = (M_1 N_1 a_1 + \ldots + M_i N_i a_i + \ldots + M_k N_k a_k) = a_i \pmod{m_i}$$

Αν s_1, s_2 δύο διαφορετικές λύσεις τότε έχουμε ότι για κάθε i,

$$s_1 = a_i \pmod{m_i} \qquad s_2 = a_i \pmod{m_i}$$

Από την πρόταση 2.20 έχουμε ότι $s_1 = s_2 \pmod{M}$. Άρα το σύστημα έχει μοναδική λύση στο $\mathbb{Z}_M$ την $y \pmod{M}$. □

Είναι φανερό από την παραπάνω απόδειξη ότι η λύση ενός τέτοιου συστήματος μπορεί να βρεθεί σε πολυωνυμικό χρόνο.

Παρατήρηση 10. Το Κινέζικο Θεώρημα Υπολοίπων συνεπάγεται δύο ισομορφισμούς:

i. $\mathbb{Z}_{m_1 m_2 \ldots m_k} \cong \mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \ldots \times \mathbb{Z}_{m_k}$

ως προς *πρόσθεση, αφαίρεση και πολλαπλασιασμό* (οι πράξεις στις k-άδες ορίζονται κατά μέλη με τον προφανή τρόπο: τα στοιχεία στη θέση i αθροίζονται / πολλαπλασιάζονται στον δακτύλιο $\mathbb{Z}_{m_i}$.)

και

ii. $U(\mathbb{Z}_{m_1 m_2 \ldots m_k}) \cong U(\mathbb{Z}_{m_1}) \times U(\mathbb{Z}_{m_2}) \times \ldots \times U(\mathbb{Z}_{m_k})$

ως προς *πολλαπλασιασμό και διαίρεση*.

2.4 Τετραγωνικά Υπόλοιπα

Εδώ θα παρουσιάσουμε την έννοια των τετραγωνικών υπολοίπων και το σύμβολο Legendre.

Το κεντρικό ενδιαφέρον της ενότητας αυτής είναι η εξίσωση $x^2 = \pmod p$ που στα πρώτα εδάφια την εξετάζουμε για συγκεκριμένο p. Η μελέτη της συγκεκριμένης εξίσωσης δεν παρουσιάζει ιδιαίτερες δυσκολίες, πάραυτα η μελέτη των λύσεων της εξίσωσης αυτής αν κρατήσουμε σταθερό το a και εξετάσουμε ποιοι πρώτοι την επαληθεύουν παρουσίασε μεγάλο ενδιαφέρον στο παρελθόν. Σε αυτήν την ενότητα θα εξετάσουμε διάφορα "εργαλεία" που έπαιξαν ρόλο στα αποτελέσματα αυτά, τα οποία έχουν σημαντικές εφαρμογές στην μοντέρνα κρυπτογραφία.

Πρόταση 2.44. *Έστω p, q πρώτοι και $a \neq \pmod p$, τότε:*

1. *Η εξίσωση $x^2 = a \pmod p$ έχει είτε 0 είτε 2 λύσεις στο $\mathbb{Z}_p$.*
2. *Η εξίσωση $x^2 = a \pmod{pq}$ έχει είτε 0 είτε 4 λύσεις στο $\mathbb{Z}_{pq}$.*

Απόδειξη. 1. Έστω ότι η x είναι λύση της εξίσωσης, τότε και η $-x$ είναι λύση. Επίσης αν $x = -x \pmod p$ τότε $2x = 0 \pmod p \Rightarrow x = 0 \pmod p$, όποτε για $x \neq 0 \pmod p$ έχουμε ότι $x \neq -x \pmod p$. Αν x, y λύσεις της εξίσωσης τότε $x^2 = y^2 \pmod p$ άρα:

$$p \mid (x^2 - y^2) \Rightarrow p \mid (x - y)(x + y) \Rightarrow$$

$$p \mid (x - y) \vee p \mid (x + y) \Rightarrow x = y \pmod p \vee x = -y \pmod p$$

2. Η λύση της εξίσωσης είναι ισοδύναμη με τη λύση των δύο εξισώσεων $x^2 = a \pmod p$, $x^2 = a \pmod q$. Έστω ότι η πρώτη έχει λύσεις τις $x_1, -x_1$ και η δεύτερη τις $x_2, -x_2$.Για κάθε ένα από τους συνδυασμούς των λύσεων αυτών (που είναι 4) προκύπτει μια διαφορετική λύση για την εξίσωση, από το σύστημα $x = \pm x_1 \pmod p$, $x = \pm x_2 \pmod q$. Η ύπαρξη μοναδικής λύσης στο $\mathbb{Z}_{pq}$ αυτού του συστήματος προκύπτει από το κινέζικο θεώρημα υπολοίπων.

□

Παρατήρηση 11. Η παραπάνω πρόταση μπορεί να γενικευτεί και για τυχαίο $n = p_1^{a_1} p_2^{a_2} \dots p_k^{a_k}$ όπου η αντίστοιχη εξίσωση έχει είτε 0 είτε 2^k λύσεις.

Στην επόμενη πρόταση θα δούμε ένα κριτήριο που θα μας επιτρέπει να ελέγχουμε αν ένας αριθμός στο $\mathbb{Z}_p$ έχει τετραγωνικές ρίζες.

Πρόταση 2.45. *Η εξίσωση $x^2 = a \pmod p$ έχει λύση αν και μόνο αν $a^{\frac{p-1}{2}} = 1 \pmod p$.*

Απόδειξη. Θα δείξουμε ότι και οι δύο συνθήκες ισχύουν αν και μόνο αν το a είναι άρτια δύναμη ενός γεννήτορα.

Έστω ότι $a \equiv g^k \pmod p$ για γεννήτορα g της $\mathbb{Z}_p^*$ (θυμηθείτε ότι η $\mathbb{Z}_p^*$ είναι κυκλική). Τότε:

$\exists x : x^2 \equiv a \pmod p \Leftrightarrow \exists l : g^{2l} \equiv g^k \pmod p \Leftrightarrow 2l \equiv k \pmod{p-1} \Leftrightarrow k \bmod 2 = 0$.

Επίσης, από μικρό Θ. Fermat:

$a^{\frac{p-1}{2}} \equiv g^{\frac{k}{2}(p-1)} \equiv 1 \pmod p \Leftrightarrow p-1 \mid \frac{k}{2}(p-1) \Leftrightarrow k \bmod 2 = 0$ □

Πρόταση 2.46. *Ακριβώς τα μισά στοιχεία του $U(\mathbb{Z}_p)$ είναι τετραγωνικά υπόλοιπα ενώ τα υπόλοιπα είναι τετραγωνικά μη υπόλοιπα.*

Απόδειξη. Όπως είπαμε, οι λύσεις της εξίσωσης $x^2 = a \pmod p$ είναι πάντα στη μορφή $x, -x$. Χωρίζουμε τα στοιχεία του $U(\mathbb{Z}_p)$ σε 'θετικά' $\{1, 2, \dots, \frac{p-1}{2}\}$ και 'αρνητικά' $\{\frac{p-1}{2} + 1, \dots, p-1\}$.

Είναι προφανές ότι κάθε ζευγάρι λύσεων μιας εξίσωσης $x^2 = a \pmod p$ θα έχει το ένα μέλος στο ένα σύνολο των 'θετικών' στοιχείων και το άλλο στο σύνολο των 'αρνητικών'. Έτσι οι αριθμοί που θα προκύψουν αν τετραγωνίσουμε modulo p τα στοιχεία του ενός υποσυνόλου θα είναι ίδιοι με τους αριθμούς που θα προκύψουν αν τετραγωνίσουμε τα στοιχεία του άλλου υποσυνόλου. Επομένως το πλήθος των τετραγωνικών υπολοίπων θα είναι ακριβώς $\frac{p-1}{2}$. □

2.4.1 Σύμβολα Legendre και Jacobi

Ένας ιδιαίτερα βολικός τρόπος να εκφράσουμε το αν ένας αριθμός $a \in \mathbb{Z}_p$ έχει τετραγωνικές ρίζες $\pmod p$ είναι με το *σύμβολο Legendre*.

Ορισμός 2.47. Το σύμβολο Legendre ορίζεται ως εξής,

$$(\tfrac{a}{p}) = \begin{cases} 1, & \exists x : x^2 = a \pmod p \\ -1, & \nexists x : x^2 = a \pmod p \\ 0, & p \mid a \end{cases}$$

Αν $(\frac{a}{p}) = 1$ τότε το a θα το ονομάζουμε και *τετραγωνικό υπόλοιπο* modulo p. Αν $(\frac{a}{p}) = -1$ τότε το a θα το ονομάσουμε *τετραγωνικό μη υπόλοιπο* modulo p.

Τα στοιχεία του $U(\mathbb{Z}_p)$ που δίνουν 1 και -1 στο σύμβολο Legendre είναι ίσα σε πλήθος, όπως φαίνεται από την προηγούμενη πρόταση.

Πρόταση 2.48. *Μερικές ιδιότητες του συμβόλου Legendre,*

1. $m = n \pmod p \Rightarrow (\frac{m}{p}) = (\frac{n}{p})$
2. $(\frac{a}{p}) = a^{\frac{p-1}{2}} \pmod p$
3. $(\frac{ab}{p}) = (\frac{a}{p})(\frac{b}{p})$

Απόδειξη. 1. Προκύπτει άμεσα από τον ορισμό.

2. Έχουμε ότι από την πρόταση 2.45 αν η εξίσωση $x^2 = a \pmod p$ έχει λύσεις τότε $a^{\frac{p-1}{2}} = 1 = (\frac{a}{p}) \pmod p$. Για την περίπτωση που η εξίσωση δεν έχει λύσεις έχουμε γενικά ότι,

$$a^{p-1} = 1 \pmod p \Rightarrow p \mid (a^{p-1} - 1) \Rightarrow$$

$$p \mid (a^{\frac{p-1}{2}} - 1)(a^{\frac{p-1}{2}} + 1) \Rightarrow a^{\frac{p-1}{2}} = \pm 1 \pmod p .$$

Άρα αναγκαστικά έχουμε το 2.

3. Το 3. προκύπτει εύκολα εφαρμόζοντας την ιδιότητα 2.

□

Πρέπει να παρατηρηθεί ότι η ιδιότητα 2. μας δίνει επίσης έναν εύκολο αλγόριθμο υπολογισμού του συμβόλου Legendre, χρησιμοποιώντας τον αλγόριθμο επαναλαμβανόμενου τετραγωνισμού.

Άλλη μία σημαντική ιδιότητα για τον υπολογισμό του συμβόλου Legendre είναι το εξής:

Λήμμα 2.49. *(Gauss) Αν το πλήθος των στοιχείων του συνόλου* $\{a \pmod p, 2a \pmod p, , \ldots, \frac{p-1}{2}a \pmod p, \}$ *που είναι μεγαλύτερα του* $\frac{p}{2}$ *το συμβολίσουμε με* μ *τότε ισχύει ότι* $(\frac{a}{p}) = (-1)^{\mu}$.

Με βάση το λήμμα αυτό μπορούμε να δείξουμε την επόμενη σημαντική πρόταση.

Πρόταση 2.50. *1.* $(\frac{-1}{p}) = (-1)^{\frac{p-1}{2}} = \begin{cases} 1, & p = 1 \pmod 4 \\ -1, & p = 3 \pmod 4 \end{cases}$

2. $(\frac{2}{p}) = \begin{cases} 1, & p = 1 \pmod 8 \vee p = 7 \pmod 8 \\ -1, & p = 3 \pmod 8 \vee p = 5 \pmod 8 \end{cases}$

Απόδειξη. 1. Προκύπτει άμεσα από την ιδιότητα 2 της πρότασης 2.48.

2. Για το σύνολο $2 \cdot 1, 2 \cdot 2, \ldots, 2\frac{p-1}{2}$, έχουμε αν m είναι τέτοιο ώστε, $2m \leq \frac{p-1}{2}$ και $2(m+1) > \frac{p-1}{2}$τότε το πλήθος των στοιχείων του συνόλου αυτού μεγαλύτερων του $\frac{p-1}{2}$ είναι $\frac{p-1}{2} - m$.

   Αν $p = 1 \pmod 8$ τότε από τα παραπάνω έχουμε ότι το μ του λήμματος 2.49 είναι ίσο με $\frac{p-1}{2} - m$ και επειδή $\frac{p-1}{2} = 4k$ έχουμε $m = 2k$. Άρα $\mu = 4k - 2k = 2k$ και συνεπώς $(\frac{2}{p}) = (-1)^{2k} = 1$. Με όμοιο τρόπο αποδεικνύονται και οι υπόλοιπες περιπτώσεις.

□

Το πιο σημαντικό θεώρημα της ενότητας είναι το ακόλουθο που αποδείχθηκε για πρώτη φορά από τον Gauss το 1796. Η απόδειξη παραλείπεται.

Θεώρημα 2.51 (Νόμος Τετραγωνικής Αντιστροφής)**.**
$(\frac{p}{q}) = \begin{cases} -(\frac{q}{p}), & p, q = 3 \pmod 4 \\ (\frac{q}{p}), & p, q \neq 3 \pmod 4 \end{cases}$

Μια χρήσιμη γενίκευση του συμβόλου Legendre είναι το *σύμβολο Jacobi* που ορίζεται πάνω σε όλους τους αριθμούς.

Ορισμός 2.52 (Σύμβολο Jacobi)**.** Αν $n = p_1^{a_1} p_2^{a_2} \ldots p_k^{a_k}$ τότε ορίζουμε το σύμβολο Jacobi

$$(\frac{m}{n}) = \prod_{i=1}^{k} (\frac{m}{p_i})^{a_i}.$$

Μπορεί να επαληθευθεί με βάση τον ορισμό ότι το σύμβολο Jacobi ικανοποιεί τις προτάσεις 2.48, 2.50 και 2.51.

Παρατήρηση 12. Πρέπει να σημειωθεί ότι το σύμβολο Jacobi δεν χαρακτηρίζει απολύτως την ύπαρξη λύσεων της αντίστοιχης εξίσωσης $x^2 = a \pmod{n}$. Πράγματι είναι εύκολο να δούμε ότι αν αυτή η εξίσωση έχει λύσεις τότε $(\frac{a}{n}) = 1$ αλλά δεν ισχύει το αντίστροφο. Π.χ. για $n = pq$, $(\frac{a}{p}) = (\frac{a}{q}) = -1 \Rightarrow (\frac{a}{n}) = 1$, δηλαδή ενώ η ισοτιμία δεν έχει λύσεις το σύμβολο Jacobi έχει την τιμή 1.

Ο έλεγχος πρώτων αριθμών Solovay-Strassen στηρίζεται στον έλεγχο του συμβόλου Jacobi, με επανάληψη δοκιμών για να μειωθεί η πιθανότητα της παραπάνω περίπτωσης.

2.5 Πιθανότητες

2.5.1 Εισαγωγή

Στην ενότητα αυτή θα κάνουμε μία σύντομη εισαγωγή στη Θεωρία Πιθανοτήτων. Θα περιοριστούμε στη μελέτη διακριτών πιθανοτήτων, όπου θεωρούμε ένα πεπερασμένο ή αριθμήσιμο σύνολο πιθανών *γεγονότων (events)*, και κάθε *γεγονός (event)* έχει μία συγκεκριμένη πιθανότητα να συμβεί, η οποία είναι ένας αριθμός στο διάστημα $[0, 1]$. Επιπλέον, το άθροισμα των πιθανοτήτων όλων των πιθανών γεγονότων είναι ίσο με 1.

Αν τα γεγονότα είναι τιμές που μπορεί να πάρει κάποια μεταβλητή (π.χ. το αποτέλεσμα της ρίψης ενός ζαριού), τότε αυτή η μεταβλητή λέγεται *διακριτή τυχαία μεταβλητή*.

Επομένως, μια (διακριτή) τυχαία μεταβλητή X, είναι μια μια μεταβλητή η οποία λαμβάνει τιμές από ένα (πεπερασμένο ή αριθμήσιμο) σύνολο V με συγκεκριμένες πιθανότητες. Αν για παράδειγμα η πιθανότητα να πάρει την τιμή v είναι $1/6$ (π.χ. ζάρι) τότε γράφουμε:

$$Pr[X = v] = \tfrac{1}{6}$$

Η *κατανομή πιθανότητας* μιας (διακριτής) τυχαίας μεταβλητής X που λαμβάνει τιμές από το σύνολο V είναι μία συνάρτηση που αντιστοιχίζει σε κάθε στοιχείο $v \in V$ την πιθανότητα η X να πάρει την τιμή v. Με βάση τα παραπάνω μια κατανομή πιθανότητας έχει τις εξής ιδιότητες:

$$Pr[X = v] \geq 0$$
$$\textstyle\sum_{v \in V} Pr[X = v] = 1$$

Η *αναμενόμενη (expected)*, ή μέση (mean) τιμή μιας τυχαίας μεταβλητής που παίρνει τιμές από το $\mathbb{R}$ δίνεται από τον τύπο:

$$E[X] = \sum_{v \in V} Pr[X = v] \cdot v$$

Για δύο τυχαίες μεταβλητές X, Y ορίζουμε την *από κοινού πιθανότητα* ως την πιθανότητα η X να λάβει την τιμή x και ταυτόχρονα η μεταβλητή Y να λάβει την τιμή y. Συμβολίζουμε $Pr[X = x, Y = y]$. Οι τυχαίες μεταβλητές X, Y είναι ανεξάρτητες αν:

$$Pr[X = x, Y = y] = Pr[X = x] \cdot Pr[Y = y]$$

Ένα παράδειγμα τυχαίων μεταβλητών που είναι ανεξάρτητες είναι οι τιμές που θα λάβει ένα νόμισμα σε δύο διαδοχικές ρίψεις.

Θεώρημα Bayes

Η δεσμευμένη πιθανότητα $Pr[X = x \mid Y = y]$ είναι η πιθανότητα η μεταβλητή X να λάβει την τιμή x, με δεδομένο ότι η μεταβλητή Y έχει λάβει την τιμή y. Για τον υπολογισμό της δεσμευμένης πιθανότητας μπορούμε να χρησιμοποιήσουμε το θεώρημα του Bayes:

Θεώρημα 2.53 (Bayes). $Pr[X = x \mid Y = y] = \frac{Pr[X=x] \cdot Pr[Y=y|X=x]}{Pr[Y=y]} = \frac{Pr[X=x, Y=y]}{Pr[Y=y]}$

2.5.2 Το παράδοξο των γενεθλίων

Το παρακάτω πείραμα από την Θεωρία Πιθανοτήτων έχει πολλές εφαρμογές στην Κρυπτογραφία.

Έστω ένα σάκος με m σφαίρες διαφορετικού χρώματος. Λαμβάνουμε μία σφαίρα τη φορά και σημειώνουμε το χρώμα της. Στη συνέχεια την ξανατοποθετούμε στον σάκο. Επαναλαμβάνουμε τη διαδικασία n φορές ($m > n$). Η πιθανότητα να έχουμε λάβει την ίδια μπάλα δύο φορές (σύγκρουση) είναι:

$$1 - \frac{m \cdot (m-1) \cdots (m-n+1)}{m^n}$$

Κάνοντας πράξεις προκύπτει ότι η πιθανότητα σύγκρουσης ξεπερνάει το $1/2$ όταν γίνουν περίπου $1 + 1.17\sqrt{m}$ λήψεις (για την απόδειξη δείτε την Πρόταση 8.7 στην Ενότητα 8.2.1). Μπορεί επίσης να αποδειχθεί ότι το αναμενόμενο πλήθος λήψεων για να πάρουμε την πρώτη σύγκρουση είναι $\sqrt{\frac{\pi \cdot m}{2}}$.

Η διαδικασία αυτή ονομάστηκε παράδοξο γενεθλίων καθώς ταυτίζεται με την πιθανότητα δύο άνθρωποι να έχουν την ίδια ημερομηνία γενεθλίων. Αντίθετα με την διαίσθηση μας σε ένα σύνολο 23 ατόμων η πιθανότητα να συμβεί αυτό, σύμφωνα με τους παραπάνω τύπους, είναι λίγο παραπάνω από $1/2$ ενώ για να φτάσουμε σε πιθανότητα 99% αρκούν 58 άτομα.

2.5.3 Στατιστική Απόσταση

Έστω δύο μεταβλητές X, Y που λαμβάνουν τιμές από δύο σύνολα S_x, S_y με κατανομές πιθανότητας D_x, D_y ορίζεται ως:

$$\Delta(X,Y) = \frac{1}{2}\sum_{v \in S_x \cup S_y} \mid Prob_{X \sim D_x}[X = v] - Prob_{Y \sim D_y}[Y = v] \mid$$

Ορισμός 2.54. Δύο μεταβλητές είναι στατιστικά αδιαχώριστες όταν η στατιστική τους απόσταση είναι αμελητέα.

Ορισμός 2.55. Μία συνάρτηση f είναι αμελητέα, εάν $\forall c \in \mathbb{R}, \exists n_0 \in \mathbb{N} : f(n) \leq \frac{1}{n^c} \forall n \geq n_0$

2.6 Ασκήσεις

1. Σχεδιάστε σαφή και αποδοτικό αλγόριθμο ακέραιας διαίρεσης, χρησιμοποιώντας μόνο τις πράξεις $(+, -, >>, <<)$, όπου με $<<$ συμβολίζουμε την ολίσθηση κατά 1 bit προς τα αριστερά (και με $>>$ την ολίσθηση προς τα δεξιά). Θεωρήστε ότι δίνονται σαν είσοδος δύο μη αρνητικοί ακέραιοι αριθμοί N, D στην δυαδική τους αναπαράσταση (0-1 arrays) και θα πρέπει να επιστρέφονται δύο ακέραιοι Q, R τ.ώ. $N = Q \cdot D + R, 0 \leq R < D$. Βρείτε και εξηγήστε την χρονική πολυπλοκότητα του αλγορίθμου σας.

2. Υπολογίστε τον ΜΚΔ των αριθμών $5^{56} - 1$ και $5^{72} - 1$

3. Δίνονται p πρώτος και g ένας γεννήτορας της πολλαπλασιαστικής ομάδας $\mathbb{Z}_p^*$. Αν μας δώσουν ένα στοιχείο h της ομάδας, την τάξη του d, και ένα τυχαίο στοιχείο a, πως μπορούμε να διαπιστώσουμε (αποδοτικά) αν το a ανήκει στην υποομάδα που παράγει το h;

4. Έστω $a \in U(Zed_n)$ τάξης k και $b \in U(Zed_n)$ τάξης m. Αποδείξτε ότι αν $gcd(k, m) = 1$, τότε το $ab \in U(Zed_n)$ έχει τάξη km.

5. Αποδείξτε ότι αν $a > 1$ και m, n ακέραιοι, τότε $gcd(a^m - 1, a^n - 1) = a^{gcd(m,n)} - 1$.

6. Αποδείξτε ότι για κάθε $n > 2$ τουλάχιστον ένας από τους $2^n - 1$ και $2^n + 1$ είναι σύνθετος.

7. Να αποδειχθεί ότι $\phi(mn) = \phi(m)\phi(n)$ για m, n σχετικά πρώτους.

8. Να αποδειχθεί ότι $\phi(mn) = gcd(m, n)\phi(lcm(m, n))$ για οποιαδήποτε m, n.

9. Να αποδειχθεί η ιδιότητα της συνάρτησης ϕ του Euler: $\phi(p^a) = p^a(1 - \frac{1}{p})$ όπου p πρώτος.

10. Υπολογίστε τις τετραγωνικές ρίζες του 119 modulo 209. Χρησιμοποιήστε μεθόδους της θεωρίας αριθμών αλλά και εμπειρικές παρατηρήσεις.

11. Υπολογίστε τα $(\frac{19}{61}), (\frac{17}{31}), (\frac{107}{117})$ χρησιμοποιώντας μόνο το θεώρημα τετραγωνικής αντιστροφής καθώς και άλλες ιδιότητες των συμβόλων, χωρίς χρήση παραγοντοποίησης (εκτός με το 2)

12. Από το κινέζικο θεώρημα υπολοίπων γνωρίζουμε, πώς να λύσουμε το σύστημα
$$x \equiv a_1 \pmod{m_1}$$
$$x \equiv a_2 \pmod{m_2},$$
στην περίπτωση που $gcd(m_1, m_2) = 1$. Τι γίνεται όταν $gcd(m_1, m_2) > 1$; Πότε έχει λύση το σύστημα; Πως λύνουμε ένα τέτοιο σύστημα όταν έχουμε περισσότερες από δύο εξισώσεις;

13. Αποδείξτε ότι αν p, q διαφορετικοί πρώτοι, τότε $p^{q-1} + q^{p-1} \equiv \pmod{pq}$.

14. Αποδείξτε ότι N πρώτος αν και μόνο αν $(N-1)! \equiv -1 \pmod{N}$. (Υπόδειξη: για το ευθύ θεωρήστε ζεύγη αντιστρόφων στο $\mathbb{Z}_p^*$, για το αντίστροφο εξετάστε τον $gcd(N, (N-1)!)$ για N σύνθετο.)

15. Αποδείξτε ότι αν p πρώτος και $q = (q-1)/2$, τότε $(q!)^2 + (-1)^q \equiv 0 \pmod{p}$.

16. Αποδείξτε ότι αν p πρώτος, τότε $a^p + b^p \equiv (a+b)^p \pmod{p}$.

17. Αποδείξτε ότι αν p πρώτος και αν $a^p + b^p \equiv 0 \pmod{p}$, τότε $a^p + b^p \equiv 0 \pmod{p^2}$.

18. Ποιο το μέγεθος της ομάδας των τετραγωνικών υπολοίπων modulo n, όπου $n = pq$, με p, q πρώτους;

19. Περιγράψτε δύο αποδοτικούς τρόπους εύρεσης πολλαπλασιαστικού αντιστρόφου του a modulo n, όπου $n = pq$ και p, q γνωστοί πρώτοι αριθμοί που δεν διαιρούν το a. Εφαρμόστε τους για να υπολογίσετε το 28^{-1} mod 51 (δείξτε τις πράξεις αναλυτικά).

20. Φτιάξτε ένα πρόγραμμα που να υλοποιεί το Κινέζικο Θεώρημα Υπολοίπων με χρήση μιας γλώσσας προγραμματισμού της επιλογής σας (π.χ. C/C++, Pascal κτλ..). Το πρόγραμμά σας θα πρέπει να δέχεται σαν είσοδο τα a_i και m_i, $1 \leq i \leq k$, και αν τα m_i είναι πρώτα μεταξύ τους, να δίνει για έξοδο τη μοναδική λύση mod$m_1 \cdots m_k$.

21. Κατασκευάστε πρόγραμμα για εύρεση τετραγωνικών ριζών $\pmod n$, όπου $n = pq$, με p, q πρώτους και $p = q = 3 \pmod 4$.
Υπόδειξη: μπορείτε να χρησιμοποιήσετε το πρόγραμμα της προηγούμενης άσκησης για το Κινέζικο Θεώρημα Υπολοίπων.

2.7 Ηλεκτρονικό Υλικό

- Διαδραστικές Παρουσιάσεις - Video
  - Αναλυτική Εξήγηση του Παράδοξου των Γενεθλίων
  - Παρουσίαση Θεωρήματος Fermat
  - Συνάρτηση του Euler
- Διαδραστικές Υλοποιήσεις
  - Online Υπολογισμός ΜΚΔ
  - Πειραματισμός με το παράδοξο των γενεθλίων
  - Online Υπολογισμός Συνάρτησης του Euler σε Javascript
- Κώδικας
  - Βιβλιοθήκες Θεωρίας Αριθμών σε C++ από τον Victor Shoup
  - Βιβλιοθήκες Θεωρίας Αριθμών σε Python

Βιβλιογραφία

[1] Tom M. Apostol. *Introduction to Analytic Number Theory*. Springer-Verlag, 1976.

[2] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography (Chapman & Hall/Crc Cryptography and Network Security Series)*. Chapman & Hall/CRC, 2007.

[3] Aggelos Kiayias. Cryptography primitives and protocols, 2015. Διαθέσιμο στο http://crypto.di.uoa.gr/class/Kryptographia/Semeioseis_files/Cryptograph_Primitives_and_Protocols.pdf.

[4] Neal Koblitz. *A Course in Number Theory and Cryptography*. Springer-Verlag New York, Inc., New York, NY, USA, 1987.

[5] Victor Shoup. *A computational introduction to number theory and algebra*. Cambridge university press, 2009.

[6] Douglas R Stinson. *Cryptography: theory and practice*. CRC press, 2005.

[7] Stathis Zachos and Aris Pagourtzis. Στοιχεία Θεωρίας Αριθμών και Εφαρμογές στην Κρυπτογραφία. Πανεπιστημιακές Σημειώσεις, 2015.

Κεφάλαιο 3

Στοιχεία Θεωρίας Υπολογισμού

Στο κεφάλαιο αυτό παρουσιάζεται μια εισαγωγή σε βασικές έννοιες της θεωρίας υπολογισμού, με έμφαση στην υπολογιστική πολυπλοκότητα. Η εξοικείωση με τις έννοιες αυτές είναι ιδιαίτερα χρήσιμη, καθώς η δυσκολία πολλών σημαντικών κρυπτοσυστημάτων βασίζεται στη δυσκολία επίλυσης υπολογιστικών προβλημάτων. Θα εξετάσουμε διάφορα υπολογιστικά μοντέλα, αλλά και την επίδραση της τυχαιότητας σε αυτά, θέμα που θα μελετήσουμε και σε επόμενα κεφάλαια. Ο κορμός του παρόντος κεφαλαίου βασίζεται στα [3], [2], [1] και [4].

Στην Θεωρία Υπολογισμού, μας ενδιαφέρει μόνον αν ένα πρόβλημα είναι υπολογίσιμο ή όχι. Αν μάλιστα είναι υπολογίσιμο, τότε είναι αδιάφορο τι ποσότητα πόρων (resources) πρέπει να διατεθεί, για να επιλυθεί το πρόβλημα. Αντίθετα, στην Θεωρία Πολυπλοκότητας, θεωρούμε μόνο υπολογίσιμα προβλήματα και προσπαθούμε να δούμε αν μπορούν να επιλυθούν με περιορισμούς στους διαθέσιμους υπολογιστικούς πόρους, όπως ο χρόνος υπολογισμού, ο επιπλέον χώρος μνήμης που απαιτείται για ενδιάμεσα αποτελέσματα κατά την επίλυση, η τυχαιότητα που χρειάζεται να εφαρμοστεί (καθώς και το είδος της) και πολλοί άλλοι. Αυτοί οι περιορισμοί, καθώς και άλλα χαρακτηριστικά των υπολογισμών, ορίζουν *κλάσεις πολυπλοκότητας* μέσα στις οποίες τοποθετούμε τα διάφορα προβλήματα.

3.1 Βασικοί ορισμοί

Αρχικά θα ασχοληθούμε με κλάσεις πολυπλοκότητας που ορίζονται με βάση περιορισμούς είτε στον χρόνο εκτέλεσης είτε στον επιπλέον χώρο (προκειμένου να αποθηκευτούν ενδιάμεσα αποτελέσματα) που απαιτείται για τον υπολογισμό. Το υπολογιστικό μοντέλο που χρησιμοποιούμε είναι η μηχανή Turing (είτε στην ντετερμινιστική TM είτε στη μη ντετερμινιστική της εκδοχή Non-Deterministic Turing Machine (NTM)):

Ορισμός 3.1. Στην κλάση $\texttt{TIME}(t(n))$ (ή $\texttt{DTIME}(t(n))$) ανήκουν τα προβλήματα που μπορούν να επιλυθούν από *ντετερμινιστική* μηχανή Turing σε χρόνο $t(n)$.

Ορισμός 3.2. Στην κλάση $\texttt{NTIME}(t(n))$ ανήκουν τα προβλήματα που μπορούν να επιλυθούν από *μη ντετερμινιστική* μηχανή Turing σε χρόνο $t(n)$.

Ορισμός 3.3. Στην κλάση $\texttt{SPACE}(s(n))$ (ή $\texttt{DSPACE}(s(n))$) ανήκουν τα προβλήματα που μπορούν να επιλυθούν από *ντετερμινιστική* μηχανή Turing χρησιμοποιώντας επιπλέον χώρο $s(n)$.

Ορισμός 3.4. Στην κλάση $\texttt{NSPACE}(s(n))$ ανήκουν τα προβλήματα που μπορούν να επιλυθούν από *μη ντετερμινιστική* μηχανή Turing χρησιμοποιώντας επιπλέον χώρο $s(n)$.

Με βάση τα παραπάνω, ορίζουμε:

- $\texttt{P} = \texttt{PTIME} = \bigcup_{i\geq 1} \text{DTIME}(n^i)$
- $\texttt{NP} = \texttt{NPTIME} = \bigcup_{i\geq 1} \texttt{NTIME}(n^i)$
- $\texttt{PSPACE} = \bigcup_{i\geq 1} \texttt{DSPACE}(n^i)$
- $\texttt{NPSPACE} = \bigcup_{i\geq 1} \texttt{NSPACE}(n^i)$
- $\texttt{L} = \texttt{DSPACE}(\log n)$
- $\texttt{NL} = \texttt{NSPACE}(\log n)$
- $\texttt{EXP} = \bigcup_{i\geq 1} \texttt{DTIME}(2^{n^i})$
- $\texttt{EXPSPACE} = \bigcup_{i\geq 1} \texttt{DSPACE}(2^{n^i})$

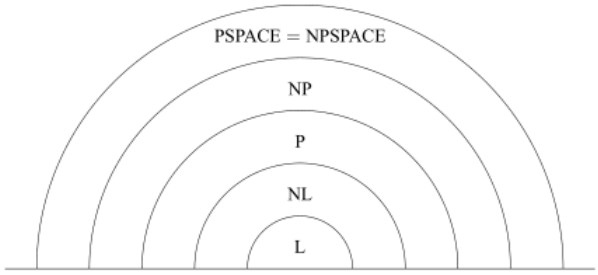
Αν η f είναι μία συνάρτηση πολυπλοκότητας (constructible)[1] τότε ισχύουν:

- $\texttt{DSPACE}(f(n)) \subseteq \texttt{NSPACE}(f(n))$
- $\texttt{DTIME}(f(n)) \subseteq \texttt{NTIME}(f(n))$

διότι κάθε ντετερμινιστική μηχανή Turing μπορεί να θεωρηθεί ως μη ντετερμινιστική με μία μόνο επιλογή σε κάθε βήμα.

- $\texttt{DTIME}(f(n)) \subseteq \texttt{DSPACE}(f(n))$

[1] Η f πρέπει να είναι κατασκευάσιμη, δηλαδή πρέπει να υπάρχει μία TM τέτοια ώστε: $\forall$ input x με $|x| = n$, αποδέχεται την είσοδο σε χρόνο $O(n + f(n))$ (time-constructible) ή working space $O(f(n))$ (space-constructible).



Σχήμα 3.1: Κλάσεις Πολυπλοκότητας

- $\texttt{NTIME}(f(n)) \subseteq \texttt{DSPACE}(f(n))$

διότι σε χρόνο $f(n)$ δεν μπορεί να εξεταστεί χώρος (αριθμός θέσεων στην ταινία της $T.M.$) παραπάνω από $f(n)$.

- $\texttt{NSPACE}(f(n)) \subseteq \texttt{DTIME}(k^{\log n + f(n)})$

Αν $f(n) > \log n$ τότε:

- $\texttt{DSPACE}(f(n)) \subseteq \texttt{DTIME}(c^{f(n)})$
- $\texttt{NTIME}(f(n)) \subseteq \texttt{DTIME}(c^{f(n)})$

Το παρακάτω θεώρημα οφείλεται στον Savitch (1970):

Θεώρημα 3.5. *Αν* $f(n) \geq \log n$ *τότε* $\texttt{NSPACE}(f(n)) \subseteq \texttt{DSPACE}(f^2(n))$.

Άμεσα από το θεώρημα του Savitch προκύπτει ότι $\texttt{PSPACE} = \texttt{NPSPACE}$.

Από τις παραπάνω σχέσεις προκύπτει η εξής ιεραρχία:

$$\texttt{L} \subseteq \texttt{NL} \subseteq \texttt{P} \subseteq \texttt{NP} \subseteq \texttt{PSPACE} = \texttt{NPSPACE}$$

Γνωρίζουμε ότι $\texttt{L} \neq \texttt{PSPACE}$ και $\texttt{NL} \neq \texttt{PSPACE}$ (αυτό προκύπτει από το θεώρημα ιεραρχίας για χωρικές κλάσεις πολυπλοκότητας, που αναφέρεται παρακάτω).
Ανοιχτά παραμένουν τα προβλήματα:

$$\texttt{L} \supseteq \texttt{NL} \supseteq \texttt{P} \supseteq \texttt{NP} \supseteq \texttt{PSPACE}$$

Ο κόσμος μοιάζει, ως τώρα, να είναι όπως στο παρακάτω σχήμα:
Οι παραπάνω κλάσεις πολυπλοκότητας αφορούν προβλήματα *απόφασης*. Μπορούμε επίσης να ορίσουμε κλάσεις πολυπλοκότητας για μηχανές Turing που υπολογίζουν *συναρτήσεις*. Ένα χαρακτηριστικό παράδειγμα είναι η παρακάτω κλάση:

Ορισμός 3.6. FP = το σύνολο των συναρτήσεων που υπολογίζεται από ντετερμινιστική μηχανή Turing σε πολυωνυμικό χρόνο.

Η κλάση FP θα φανεί χρήσιμη παρακάτω στον ορισμό των αναγωγών, αφού περιλαμβάνει τις "εύκολα" υπολογιζόμενες συναρτήσεις.

3.2 Συμπληρώματα κλάσεων πολυπλοκότητας

Έστω γλώσσα L. Ως γνωστόν, το συμπλήρωμα της γλώσσας συμβολίζεται και ορίζεται ως εξής: $\overline{L} = \{x \mid x \notin L\}$. Τώρα, για μία κλάση γλωσσών $\mathcal{C}$, ορίζουμε (με την βοήθεια του συμπληρώματος):

$$\text{co}\mathcal{C} = \{\overline{L} \mid L \in \mathcal{C}\}.$$

Για παράδειγμα η κλάση coNP αποτελείται από τις γλώσσες που είναι συμπληρώματα γλωσσών στο NP. Ένα πρόβλημα που ανήκει στην κλάση coNP είναι το $\overline{\text{SAT}}$ ή το στενά σχετιζόμενο με αυτό πρόβλημα της ταυτολογίας, αν δηλαδή ένας λογικός τύπος που δίνεται είναι ταυτολογία.

Έχει ενδιαφέρον να δούμε ποιες κλάσεις πολυπλοκότητας είναι *κλειστές ως προς συμπλήρωμα* (δηλαδή για ποιες κλάσεις $\mathcal{C}$ ισχύει $\mathcal{C} = co\mathcal{C}$.

Γενικά, οι ντετερμινιστικές κλάσεις πολυπλοκότητας (είτε χρονικές, είτε χωρικές) είναι κλειστές ως προς συμπλήρωμα. Για παράδειγμα, η κλάση P είναι κλειστή ως προς συμπλήρωμα, αφού για να απαντήσουμε για κάθε συμπληρωματικό πρόβλημα σε μία ντετερμινιστική μηχανή αρκεί απλώς να εναλλάξουμε την έξοδο, παραμένοντας στην ίδια πολυπλοκότητα. Δηλαδή, οι $DTIME(t(n))$ και $DSPACE(s(n))$ είναι κλειστές ως προς συμπλήρωμα.

Αν θεωρήσουμε μη ντετερμινισμό, το πρόβλημα είναι ανοιχτό στην περίπτωση της χρονικής πολυπλοκότητας. Για παράδειγμα δεν γνωρίζουμε αν $\texttt{coNP} \neq \texttt{NP}$. Μάλιστα, το τελευταίο συνδέεται και με το πρόβλημα αν $\texttt{P} \neq \texttt{NP}$, αφού προφανώς αν $\texttt{coNP} \neq \texttt{NP}$, τότε $\texttt{P} \neq \texttt{NP}$.

Ενώ η κατάσταση φαινόταν να είναι παρόμοια και για τον χώρο στην μη ντετερμινιστική περίπτωση, στα μέσα της δεκαετίας του 1980 απεδείχθη το παρακάτω:

Θεώρημα 3.7 (Immerman-Szelepcsényi)**.** *Η κλάση* $\texttt{NSPACE}(s(n))$ *είναι κλειστή ως προς συμπλήρωμα.*

3.3 Αναγωγές

Η έννοια της *αναγωγής* σε πολυωνυμικό χρόνο πρέπει να συνδέει μεταξύ τους προβλήματα με υπολογιστικά "εύκολο" τρόπο. Θεωρούμε εύκολες συναρτήσεις

(και προβλήματα) που υπολογίζονται σε πολυωνυμικό χρόνο. Δηλαδή αν η f είναι υπολογίσιμη σε χρόνο $O(n^2)$, τότε θεωρείται εύκολη. Αν οι συναρτήσεις f και g είναι "εύκολες", τότε και η σύνθεσή τους $f \circ g$ θα θέλαμε να είναι "εύκολη", κάτι που ισχύει για τα πολυώνυμα Άρα θεωρούμε εύκολα προβλήματα (και συναρτήσεις) αυτά που υπολογίζονται σε πολυωνυμικό χρόνο (έστω και σε $O(n^{1000})$). Για τους παραπάνω λόγους, ορίζουμε την αναγωγή κατά Karp:

Ορισμός 3.8 (Αναγωγή κατά Karp)**.**

$$A \leq_m^P B: \quad \exists f \in \texttt{FP}, \forall x (x \in A \Leftrightarrow f(x) \in B)$$

Υπάρχουν και άλλες χρήσιμες αναγωγές, όπως η λεγόμενη log-space, που χρησιμοποιεί λογαριθμικό χώρο, και η οποία είναι χρήσιμη για αναγωγές προβλημάτων σε μικρότερες κλάσεις πολυπλοκότητας, όπως η P:

Ορισμός 3.9 (Log-space Αναγωγή)**.**

$$A \leq_m^L B: \quad \exists f \in \texttt{FL}, \forall x (x \in A \Leftrightarrow f(x) \in B)$$

Ισχύει: $A \leq_m^L B \Rightarrow A \leq_m^P B$, αλλά όχι το αντίστροφο.

Μία επιθυμητή ιδιότητα μίας αναγωγής είναι να είναι κλειστή ως προς διάφορες κλάσεις γλωσσών:

Ορισμός 3.10. Λέμε ότι μία κλάση γλωσσών C είναι *κλειστή* ως προς μία αναγωγή $\leq$ αν

$$A \leq B \wedge B \in C \;\Rightarrow\; A \in C.$$

Μερικές από τις κλάσεις πολυπλοκότητας που είναι κλειστές ως προς την αναγωγή κατά Karp ($\leq_m^P$) είναι οι εξής: P, PSPACE, EXP, EXPSPACE (βλέπε παραπάνω για τους ορισμούς τους).

Ορισμός 3.11 (Hardness)**.** Λέμε ότι A είναι C-hard (C-δύσκολο), ως προς την $\leq$, αν:

$$\forall B \in C : B \leq A.$$

Η έννοια της hardness δίνει ένα κάτω όριο για την πολυπλοκότητα ενός προβλήματος, δεδομένου ότι το πρόβλημα A είναι τουλάχιστον τόσο δύσκολο όσο οποιοδήποτε πρόβλημα μίας κλάσης C.

Ορισμός 3.12 (Completeness)**.** Λέμε ότι A είναι $C - complete$ (C-πλήρες), ως προς την $\leq$, αν: A είναι $C-$ hard ως προς $\leq \quad \wedge \quad A \in C$.

Παρακάτω δίνουμε πλήρη προβλήματα για μερικές από τις σημαντικότερες κλάσεις πολυπλοκότητας:

Για την κλάση NL έχουμε το πλήρες πρόβλημα REACHABILITY (log-space αναγωγές). Για την P τα εξής προβλήματα είναι πλήρη: CIRCUIT-VALUE και LINEAR PROGRAMMING (πάλι με log-space αναγωγές). Για την κλάση NP, το 3SAT. Για την κλάση PSPACE, το QBF (Quantified Boolean Formula satisfiability problem). Για την EXP, το $n \times n$ Go. Για την EXPSPACE, το RegExp$(\cup, \cdot, *, ^2)$, που είναι το πρόβλημα ελέγχου ισοδυναμίας regular expressions, που χρησιμοποιούν τους τελεστές $\cup$ (ένωση), $\cdot$ (παράθεση), $*$ (άστρο του Kleene) και ^{2}, όπου $\alpha^2 = \alpha \cdot \alpha$.

3.4 Μοντέλο δένδρων υπολογισμού για TM

Για να μελετήσουμε την συμπεριφορά μη ντετερμινιστικών μηχανών Turing, θα κωδικοποιήσουμε τους υπολογισμούς μίας NTM με ένα δέντρο υπολογισμού. Ο υπολογισμός ξεκινά στην ρίζα του δένδρου. Θεωρούμε ότι αν σε κάποιο σημείο του υπολογισμού έχουμε μία μη ντετερμινιστική επιλογή τότε έχουμε μία διακλάδωση στο δένδρο. Στα φύλλα της μηχανής TM έχουμε τις απαντήσεις της μηχανής Turing. Κάθε μονοπάτι από την ρίζα του δένδρου μέχρι κάποιο φύλλο επομένως κωδικοποιεί έναν πιθανό υπολογισμό. Επίσης, το δένδρο προφανώς θα περιέχει όλους τους πιθανούς υπολογισμούς που μπορεί να κάνει ο αλγόριθμος. Αποδεικνύεται, ότι χωρίς βλάβη της γενικότητας, μπορούμε να υποθέσουμε ότι το δένδρο είναι δυαδικό, πλήρες και γεμάτο όλα τα φύλλα του είναι στο ίδιο επίπεδο (βλέπε σχήμα

Επίσης, έχει ενδιαφέρον το μήκος του υπολογιστικού μονοπατιού από την ρίζα μέχρι το φύλλο να είναι πολυωνυμικό ως προς το μήκος της εισόδου (να αντιστοιχεί δηλαδή το κάθε μονοπάτι σε κάποιον "εύκολο", δηλαδή πολυωνυμικό, υπολογισμό). Θεωρώντας το παραπάνω μοντέλο, θα ορίσουμε μερικές από τις γνωστές κλάσεις υπολογισμού, καθώς και μερικές καινούριες. Πιο συγκεκριμένα, θα χρησιμοποιήσουμε ποσοδείκτες $(\exists, \forall)$ στα μονοπάτια. Επειδή εννοείται πάντοτε ο περιορισμός του μήκους των μονοπατιών, θα γράφουμε π.χ. $\exists y$, αντί για $\exists y : |y| \leq p(|x|)$, όπου y: μεταβλητή για τα μονοπάτια, x: μεταβλητή για την είσοδο, p: πολυώνυμο.

Παρακάτω, σε κάθε περίπτωση χρησιμοποιούμε ένα κατηγόρημα R που υπολογίζεται σε πολυωνυμικό χρόνο. Στην προκειμένη περίπτωση συμβολίζουμε με $R(x, y)$, το γεγονός ότι με είσοδο x, αν ακολουθήσουμε το μονοπάτι y θα έχουμε αποτέλεσμα "ναι" και αντίστοιχα με $\forall y \neg R(x, y)$, ότι με είσοδο x όλα τα μονοπάτια δίνουν αποτέλεσμα "όχι", κτλ.

Για παράδειγμα, η κλάση P μπορεί να οριστεί ως εξής:

$$L \in \mathtt{P} \Leftrightarrow \exists R \in \mathtt{P} : \begin{cases} \forall y R(x,y), & x \in L \\ \forall y \neg R(x,y), & x \notin L \end{cases}$$

Περισσότερο ενδιαφέρον έχει η κλάση NP που μπορεί να οριστεί ως εξής:

$$L \in \mathtt{NP} \Leftrightarrow \exists R \in \mathtt{P} : \begin{cases} \exists y R(x,y), & x \in L \\ \forall y \neg R(x,y), & x \notin L \end{cases}$$

Δηλαδή, αν $x \in L$ υπάρχει τουλάχιστον ένας υπολογισμός που αποδέχεται, ενώ αν $x \notin L$ κανένας υπολογισμός δεν αποδέχεται.

Παρομοίως, η κλάση coNP ορίζεται ως εξής:

$$L \in \mathtt{coNP} \Leftrightarrow \exists R \in \mathtt{P} : \begin{cases} \forall y R(x,y), & x \in L \\ \exists y \neg R(x,y), & x \notin L \end{cases}$$

Παρατηρούμε ότι οι ποσοδείκτες που χρησιμοποιούνται και αντιστοιχούν στο $x \in L$ και στο $x \notin L$ καθορίζουν πλήρως την αντίστοιχη κλάση πολυπλοκότητας. Έτσι, εισαγάγουμε τον παρακάτω συμβολισμό:

$$\mathtt{P} = (\forall, \forall), \quad \mathtt{NP} = (\exists, \forall), \quad \mathtt{coNP} = (\forall, \exists).$$

3.5 Η κλάση UP

Έχοντας ορίσει τι είναι το δέντρο υπολογισμού για μία TM, μπορούμε να ορίσουμε την κλάση πολυπλοκότητας UP, μία κλάση που έχει άμεση σχέση με την κρυπτογραφία.

Ορισμός 3.13. Μία TM χαρακτηρίζεται ως μονοσήμαντη, αν έχει την παρακάτω ιδιότητα: Για κάθε x υπάρχει το πολύ ένα υπολογιστικό μονοπάτι αποδοχής. UP είναι η κλάση των γλωσσών που γίνονται αποδεκτές από μονοσήμαντες TM.

Προφανώς ισχύει $\mathtt{P} \subseteq \mathtt{UP} \subseteq \mathtt{NP}$. Το επόμενο θεώρημα δείχνει την άμεση σχέση που έχει η κλάση UP με τις συναρτήσεις μονής κατεύθυνσης.

Θεώρημα 3.14. $\mathtt{UP} \neq \mathtt{P}$ *αν και μόνον αν υπάρχουν συναρτήσεις μονής κατεύθυνσης.*

Απόδειξη. "⇐": Ας υποθέσουμε ότι η f είναι μια συνάρτηση μονής κατεύθυνσης. Ορίζουμε την εξής γλώσσα:$L_f = \{(x,y) :$ υπάρχει ένα z τέτοιο ώστε $f(z) = y$

και $z \leq x\}$, όπου με $z \leq x$ εννοούμε, είτε ότι το z έχει μικρότερο μήκος από το x είτε ότι το z προηγείται λεξικογραφικά από το x, αν τα δούμε ως ακεραίους ίδιου μήκους. Π.χ. $0 < 1 < 00 < 01 < 10 < 11 < 000 < \ldots$. Ισχυριζόμαστε ότι $L_f \in \mathtt{UP} - \mathtt{P}$. Θα κατασκευάσουμε μία μονοσήμαντη μηχανή Turing U που να αποδέχεται την L_f. Έστω ότι θέλουμε να ελέγξουμε μη ντετερμινιστικό αν $(x, y) \in L_f$. Τότε η U μαντεύει ένα z μήκους το πολύ $|y|^k$ και ελέγχει αν $y = f(z)$. Αν ισχύει και αν $z \leq x$, τότε αποδέχεται το (x, y). Καθώς η f είναι εξ' ορισμού 1-1, είναι προφανές ότι η U είναι μονοσήμαντη.

Για να δείξουμε ότι $L_f \notin \mathtt{P}$ θα υποθέσουμε ότι υπάρχει πολυωνυμικός αλγόριθμος που να αποδέχεται την L_f. Τότε θα δούμε ότι θα μπορούσαμε να αντιστρέψουμε αποδοτικά την f εφαρμόζοντας δυαδική αναζήτηση. Δεδομένου ενός y ρωτάμε αν $(1^{|y|^k}, y) \in L_f$. Αν πάρουμε αρνητική απάντηση, τότε βλέπουμε ότι δεν υπάρχει x με $f(x) = y$, διότι αν υπήρχε θα έπρεπε να ήταν λεξικογραφικά μικρότερο από $1^{|y|^k}$, αφού από ορισμό μονόδρομων συναρτήσεων $|y| \geq |x|^{1/k}$. Αν η απάντηση ήταν θετική, τότε θα ρωτούσαμε αν $(1^{|y|^k-1}, y) \in L_f$, μετά αν $(1^{|y|^k-2}, y) \in L_f$, μέχρι να πάρουμε μια αρνητική απάντηση και να προσδιορίσουμε το μήκος $l \leq |y|^k$ του x. Τότε μπορούμε να προσδιορίσουμε τα ψηφία του x ένα ένα, ρωτώντας αν $(01^{|y|^l-1}, y) \in L_f$ και μετά αναλόγως αν πάρουμε θετική ή αρνητική απάντηση ρωτάμε αν $(001^{|y|^l-2}, y) \in L_f$ ή $(101^{|y|^l-2}, y) \in L_f$ αντίστοιχα. Έτσι μπορούμε σε πολυωνυμικό χρόνο να αντιστρέψουμε την f.

"$\Rightarrow$": Υποθέτουμε ότι υπάρχει μία γλώσσα $L \in \mathtt{UP}-\mathtt{P}$. Θα δούμε πως μπορούμε να φτιάξουμε μία συνάρτηση μονής κατεύθυνσης. Έστω U η μονοσήμαντη NTM που αντιστοιχεί στην L και x ο κωδικός ενός μονοπατιού αποδοχής της U με είσοδο το y. Ορίζουμε σε αυτήν την περίπτωση $f(x) = 1y$, ενώ σε κάθε άλλη περίπτωση $f(x) = 0x$. Το πρώτο bit τις τιμής $f(x)$, το λέμε flag και μας δείχνει αν το x είναι κωδικός υπολογιστικού μονοπατιού αποδοχής.

Θα δείξουμε ότι η f είναι συνάρτηση μονής κατεύθυνσης. Καταρχάς αφού το x είναι υπολογιστικό μονοπάτι περιέχει το y, οπότε μπορούμε να υπολογίσουμε την τιμή $f(x)$ σε πολυωνυμικό χρόνο. Επίσης, τα μήκη των x και $f(x)$ συσχετίζονται πολυωνυμικά αφού η U τρέχει σε πολυωνυμικό χρόνο. Η f είναι 1-1, αφού η U είναι μονοσήμαντη και χρησιμοποιούμε flag, δηλαδή $f(x) = f(x')$ σημαίνει $x = x'$. Τέλος αν μπορούσαμε να αντιστρέψουμε την f σε πολυωνυμικό χρόνο τότε θα μπορούσαμε να αποφασίσουμε για την L σε πολυωνυμικό χρόνο, διότι από το $f^{-1}(1y)$ θα μπορούσαμε να δούμε αν η U αποδέχεται το y ή όχι. □

3.6 Τυχαιότητα (Randomness)

Στο τμήμα αυτό θα ορίσουμε κλάσεις πολυπλοκότητας που βασίζονται στις πιθανότητες, με βάση τυχαίες επιλογές. Αυτή η προσέγγιση είναι πολύ χρήσιμη από

πρακτική άποψη, αφού σε πολλές εφαρμογές, είναι ικανοποιητικός ένας αλγόριθμος ο οποίος κάνοντας κάποιες τυχαίες επιλογές, δίνει στις περισσότερες των περιπτώσεων το σωστό αποτέλεσμα. Ένας πιθανοτικός αλγόριθμος είναι συνήθως πιο απλός στην διατύπωσή του και στην πράξη πιο αποδοτικός από έναν αντίστοιχο ντετερμινιστικό που επιλύει το ίδιο πρόβλημα. Για παράδειγμα, απλοί πιθανοτικοί αλγόριθμοι για τον έλεγχο αν ένας αριθμός είναι πρώτος υπάρχουν από την δεκαετία του 1970 και χρησιμοποιούνται στην πράξη έναντι πιο περίπλοκων ντετερμινιστικών τύπου AKS.

Εδώ θα ασχοληθούμε με κλάσεις πολυπλοκότητας πιθανοτικών αλγορίθμων που τρέχουν σε πολυωνυμικό χρόνο. Πρέπει να προσθέσουμε ότι, αφού μιλάμε για πιθανοτικούς αλγόριθμους, υπάρχουν τρία δυνατά αποτελέσματα σαν έξοδο της TM. Καταρχάς η TM μπορεί να απαντήσει "ναι" ή "όχι", αναλόγως αν αποδέχεται ή όχι την είσοδο, ενώ μπορεί να απαντήσει και "δεν ξέρω", που σημαίνει ότι η TM δεν μπορεί να αποφασίσει ούτε "ναι" ούτε "όχι".

Ορισμός 3.15 (BPP)**.** Είναι τα αρχικά της κλάσης Bounded Probabilistic Polynomial στην οποία ανήκουν οι γλώσσες L για τις οποίες υπάρχει μία NTM M που τρέχει σε πολυωνυμικό χρόνο, τέτοια ώστε: $\exists \epsilon > 0, \forall x$:

i) Αν $x \in L$, τότε $Pr(M(x) =$"ναι"$) \geq 1/2 + \epsilon$, δηλαδή η πιθανότητα να αποδεχθεί την είσοδο x η μηχανή Turing είναι $1/2 + \epsilon$ και

ii) αν $x \notin L$, τότε $Pr(M(x) =$"όχι"$) \geq 1/2 + \epsilon$.

Οι αλγόριθμοι αυτοί ονομάζονται και Monte-Carlo ή αλλιώς bounded two-sided error, επειδή ανεξάρτητα από το αποτέλεσμα (ναι ή όχι), υπάρχει κάποια πιθανότητα λάθους.

Ορισμός 3.16 (PP)**.** Ο ορισμός της μοιάζει με αυτόν της BPP, μόνο που εδώ έχουμε εναλλαγή ποσοδεικτών. Δηλαδή, είναι η κλάση στην οποία ανήκουν οι γλώσσες L για τις οποίες υπάρχει μία NTM M που τρέχει σε πολυωνυμικό χρόνο, τέτοια ώστε: $\forall x, \exists \epsilon > 0$:

i) Αν $x \in L$, τότε $Pr(M(x) =$"ναι"$) \geq 1/2 + \epsilon$, δηλαδή η πιθανότητα να αποδεχθεί την είσοδο x η μηχανή Turing είναι $1/2 + \epsilon$ και

ii) αν $x \notin L$, τότε $Pr(M(x) =$"όχι"$) \geq 1/2 + \epsilon$.

Για προφανείς λόγους η κλάση αυτή χαρακτηρίζεται ως unbounded error και two-sided, διότι σε έναν τέτοιο αλγόριθμο μπορούμε να έχουμε λάθος απάντηση ανεξάρτητα από το αποτέλεσμα.

Ορισμός 3.17 (RP)**.** Είναι τα αρχικά της κλάσης Randomized Polynomial. Για κάθε γλώσσα της κλάσης αυτής υπάρχει μία NTM M τέτοια ώστε: $\exists \epsilon > 0, \forall x$:

i) Αν $x \in L$, τότε $Pr(M(x) =$"ναι"$) \geq 1/2 + \epsilon$, δηλαδή η πιθανότητα να αποδεχθεί την είσοδο x η μηχανή Turing είναι $1/2 + \epsilon$ και

ii) αν $x \notin L$, τότε $Pr(M(x) =$"όχι"$) = 1$.

Σε αυτήν την κλάση, αν ο αντίστοιχος RP αλγόριθμος δώσει απάντηση "ναι", είμαστε σίγουροι ότι $x \in L$. Αντίθετα, η απάντηση "όχι" του RP αλγορίθμου δεν είναι "σίγουρη". Η συμπληρωματική της, η coRP , ορίζεται κατά τον ίδιο τρόπο, μόνο που η πιθανότητα σφάλματος βρίσκεται στην απάντηση "όχι". Αντίστοιχα, μόνο αν ο αλγόριθμος απαντήσει "όχι" θα είμαστε σίγουροι ότι η απάντηση είναι σωστή. Χαρακτηριστικά παραδείγματα αλγορίθμων που ανήκουν στην coRP είναι αυτοί των Solovay-Strassen, όπως, και των Miller-Rabin που αφορούν το PRIMALITY . Αυτές οι δύο κλάσεις ονομάζονται bounded one sided error, διότι κανουν λάθος μόνο για τη μία απάντηση, σε αντίθεση με τις δύο προηγούμενες κλάσεις. Τέλος ας δούμε και μια κλάση που αφορά αλγόριθμους, που όταν απαντήσουν δεν κάνουν ποτέ λάθος.

Ορισμός 3.18 (ZPP)**.** Το όνομα προέρχεται από το Zero-error Probabilistic Polynomial. Στην κλάση αυτή ανήκουν οι γλώσσες για τις οποίες υπάρχει μία NTM M, τέτοια ώστε: $\exists \epsilon > 0, \forall x$:

i) Αν $x \in L$, τότε $Pr(M(x) =$"όχι"$) = 0$, δεν υπάρχει περίπτωση να απορρίψει την είσοδο η TM,

ii) αν $x \notin L$, τότε $Pr(M(x) =$"ναι"$) = 0$ δεν υπάρχει περίπτωση να αποδεχτεί την είσοδο η TM,

iii) $Pr(M(x) =$ "δεν ξέρω" $) < \epsilon$, δηλαδή υπάρχει μια μικρή πιθανότητα η M να μην μπορεί να αποφασίσει αν αποδέχεται την είσοδο ή όχι.

Ένας άλλος τρόπος για να καταλάβουμε την ZPP είναι να την ορίσουμε ως την τομή των κλάσεων RP και coRP, η ZPP $=$ RP $\cap$ coRP. Μπορεί εύκολα να δειχτεί ότι ένα πρόβλημα είναι στο ZPP αν υπάρχει πιθανοτικός αλγόριθμος ο οποίος τρέχει σε αναμενόμενο πολυωνυμικό χρόνο και δίνει πάντοτε σωστή απάντηση. Πράγματι, αν ένα πρόβλημα είναι στο ZPP, σημαίνει ότι έχουμε ένα RP και έναν coRP αλγόριθμο για αυτό, οπότε αρκεί να τρέχουμε εναλλακτικά τους δύο αλγορίθμους, μέχρι ο ένας να δώσει την σίγουρή του απάντηση. Βέβαια, μπορεί να χρειαστεί να τρέχουμε εναλλακτικά τους δύο αλγορίθμους για πάρα πολλές φορές, αλλά με μεγάλη πιθανότητα θα έχουμε μία "σίγουρη" απάντηση, μετά από μερικές επαναλήψεις. Εναλλακτικά, μπορούμε να πούμε ότι ένας ZPP αλγόριθμος έχει τρεις εξόδους: "ναι" ή "όχι" (για τις "σίγουρες" απαντήσεις), και "δεν ξέρω" (για τις όχι "σίγουρες").

Οι αλγόριθμοι στο ZPP ονομάζονται Las Vegas.

Δεδομένου ότι υπάρχουν αρκετοί πιθανοτικοί αλγόριθμοι ευρείας χρήσης για πρακτικά προβλήματα, πολλοί τοποθετούν τους εφικτούς (feasible) υπολογισμούς πάνω από το P, στις πιθανοτικές κλάσεις BPP, RP, ZPP.

Πάντως, δεν γνωρίζουμε αν υπάρχουν πλήρη προβλήματα για τις κλάσεις (BPP, RP, ZPP).

3.7 Διαλογική αλληλεπίδραση (interactivity)

Σε αυτήν την ενότητα θα ορίσουμε κλάσεις με την βοήθεια δύο TM που αλληλεπιδρούν (ανταλλάσσουν μηνύματα μεταξύ τους). Συνήθως, ο ένας, ο αποδείκτης (Prover), προσπαθεί να αποδείξει στον άλλο, τον επαληθευτή (Verifier), ότι μία συμβολοσειρά ανήκει σε μία γλώσσα. Ένα εργαλείο που χρησιμοποιείται είναι η τυχαιότητα (randomness).

Διαλογικά συστήματα αποδείξεων (IP)

Τα διαλογικά συστήματα αποδείξεων έχουν επιλέξει έναν διαφορετικό τρόπο εκτέλεσης ενός υπολογισμού. Αντί να εκτελείται από μία οντότητα (μία μηχανή Turing), εκτελείται από δύο οντότητες οι οποίες συνεργάζονται για την εκτέλεση του ανταλλάσσοντας μηνύματα. Τυπικά αυτές οι οντότητες ονομάζονται αποδείκτης (prover), ο οποίος προσπαθεί να αποδείξει την αλήθεια μίας πρότασης του τύπου "$x \in L$" σε κάποιον άλλο, που τον ονομάζουμε επαληθευτή (verifier). Ο αποδείκτης είναι παντοδύναμος, με την έννοια ότι είναι ένας αλγόριθμος χωρίς περιορισμούς στο μέγεθος των πόρων που χρησιμοποιεί (χρόνος, χώρος). Αντίθετα, ο επαληθευτής είναι απλώς ένας πιθανοτικός αλγόριθμος πολυωνυμικού χρόνου. Ο επαληθευτής και ο αποδείκτης συμμετέχουν σε ένα πρωτόκολλο επικοινωνίας στέλνοντας μηνύματα. Ανάλογα με τα μηνύματα που λαμβάνει ο $\mathcal{V}$ από τον $\mathcal{P}$, αποδέχεται την απόδειξη, αλλιώς την απορρίπτει. Ο αποδείκτης μπορεί να μην είναι έντιμος, και να θέλει να πείσει τον επαληθευτή ότι $x \in L$, ακόμη και για x για τα οποία $x \notin L$. Ο επαληθευτής, απέναντι στον παντοδύναμο αποδείκτη, μπορεί να χρησιμοποιήσει εκτός του πολυωνυμικού χρόνου, κυρίως την τυχαιότητα που διαθέτει. Η κλάση IP ορίστηκε από τους Goldwasser, Micali, Rackoff το 1985.

Ορισμός 3.19. $L \in$ IP:

- $x \in L \Rightarrow$ υπάρχει αποδείκτης (prover) P, ώστε ο επαληθευτής (verifier) V αποδέχεται με πιθανότητα τουλάχιστον 2/3,

- $x \notin L \Rightarrow$ για κάθε αποδείκτη (prover) P, ο επαληθευτής V αποδέχεται με πιθανότητα το πολύ 1/3.

Όπως θα δείξουμε και αργότερα, ο ορισμός αυτός είναι ισοδύναμος με τον εξής:

- $x \in L \Rightarrow$ υπάρχει αποδείκτης (prover) P, ώστε ο επαληθευτής (verifier) V πάντοτε να αποδέχεται,
- $x \notin L \Rightarrow$ για κάθε αποδείκτη (prover) P, ο επαληθευτής V απορρίπτει με συντριπτική πιθανότητα.

Πρέπει, όμως να τονίσουμε ότι οι πιθανότητες αυτές εξαρτώνται μόνο από τα τυχαία bits που χρησιμοποιεί ο επαληθευτής V και δεν μπορεί να τις επηρεάσει ο αποδείκτης P. Μπορούμε να υποθέσουμε, δηλαδή, ότι ο επαληθευτής V κρατάει κρυφά τα τυχαία bits που χρησιμοποιεί από τον αποδείκτη P. Έχοντας κάνει τις παρατηρήσεις αυτές μπορούμε να δούμε ότι η πρώτη συνθήκη μας λέει ότι αν $x \in L$ τότε ο V αποδέχεται πάντα, ενώ αν $x \notin L$, τότε δεν υπάρχει στρατηγική για τον P που να του δίνει το οποιοδήποτε πλεονέκτημα για να πείσει τον V ότι $x \in L$.

Με $IP[k]$ συμβολίζουμε την κλάση των γλωσσών με διαλογικό σύστημα απόδειξης k κινήσεων, όπου μία κίνηση είναι η αποστολή ενός μηνύματος. Για συντομία συμβολίζουμε $IP = IP[p(|x|)]$, όπου $p(|x|)$ οποιοδήποτε πολυώνυμο.

Ας θεωρήσουμε το *πρόβλημα μη ισομορφισμού γράφων*: "Δίνονται δύο γράφοι. Είναι μη ισόμορφοι;". Αυτό το πρόβλημα ανήκει στο coNP.[2] Θα δώσουμε ένα πρωτόκολλο για το πρόβλημα μη ισομορφισμού γράφων, που θα δείχνει ότι το πρόβλημα είναι στο IP.

Αρχικά, ο επαληθευτής έχει τους δύο γράφους G_1 και G_2. Επιλέγει τυχαία έναν από τους δύο, έστω των G_i, και υπολογίζει έναν τυχαίο ισόμορφο γράφο του G_i, έστω τον H (αυτό γίνεται διαλέγοντας τυχαία μία μετάθεση των n κορυφών του γράφου G_i). Στέλνει τον γράφο H στον αποδείκτη, ζητώντας ένα j τέτοιο ώστε ο G_j να είναι ισόμορφος του H. Ο αποδείκτης απαντά με ένα $j \in \{1, 2\}$. Ο επαληθευτής αποδέχεται αν όντως $i = j$, αλλιώς απορρίπτει.

Στην περίπτωση που όντως οι G_1, G_2 είναι μη ισόμορφοι, ο P, αφού είναι παντοδύναμος, βρίσκει με ποιον (μοναδικό) γράφο είναι ισόμορφος ο H που του έστειλε ο V και δίνει την σωστή τιμή για να αποδεχθεί ο V. Αν τώρα οι G_1, G_2 είναι ισόμορφοι, ο P αδυνατεί να συμπεράνει από ποιον γράφο προήλθε ο ισομορφισμός H, άρα δεν μπορεί να κάνει κάτι καλύτερο από το να στείλει τυχαία ένα από τα $\{1, 2\}$ στον V. Έτσι, αν οι δύο γράφοι είναι μη ισόμορφοι ο V δεν αποδέχεται με πιθανότητα $1/2$.

[2] Το συμπληρωματικό του πρόβλημα, το πρόβλημα ισομορφισμού γράφων, είναι στο NP, αλλά δεν φαίνεται να είναι NP-πλήρες.

Τα παραπάνω σκιαγραφούν μία απόδειξη ότι το πρόβλημα μη ισομορφισμού γράφων ανήκει στην IP.

Στην πραγματικότητα, κάθε γλώσσα στην πολυωνυμική ιεραρχία έχει πρωτόκολλο IP. Μάλιστα, έχει αποδειχθεί το ακόμη ισχυρότερο αποτέλεσμα:

Θεώρημα 3.20 (Shamir)**.** $\texttt{IP} = \texttt{PSPACE}$.

Τι γίνεται όμως στην περίπτωση που ο επαληθευτής μπορεί να διαλέγεται με δύο ή περισσότερους αποδείκτες; Αν οι αποδείκτες επικοινωνούν μεταξύ τους, τότε παραμένουμε στην κλάση IP (πρακτικά, ένας αποδείκτης, ως παντοδύναμος αλγόριθμος, μπορεί να εξομοιώνει οσουσδήποτε άλλους). Αν όμως, οι αποδείκτες δεν έχουν επικοινωνία μεταξύ τους, τότε προκύπτει η ισχυρότερη κλάση MIP (Multi IP). Μάλιστα ισχύει: $\texttt{MIP} = \texttt{NEXP}$.

3.8 Ασκήσεις

1. Αποδείξτε ότι αν για κάποιο πρόβλημα απόφασης έχουμε έναν πιθανοτικό αλγόριθμο πολυωνυμικού χρόνου με μονόπλευρη πιθανότητα λάθους το πολύ ίση με κάποια σταθερά c, $0 < c < 1$, τότε για οποιαδήποτε πολυωνυμική σταθέρα $c', 0 < c' < c < 1$ μπορούμε να σχεδιάσουμε πιθανοτικό αλγόριθμο πολυωνυμικού χρόνου με πιθανότητα λάθους το πολύ c'.

2. Επαναλάβετε την παραπάνω απόδειξη για για πιθανοτικούς αλγορίθμους πολυωνυμικού χρόνου με $c' = 1/c^{p(n)}$, όπου p κάποιο πολυώνυμο και n το μέγεθος (μήκος αναπαράστασης) της εισόδου.

3. Μπορείτε να βρείτε μέθοδο ελάττωσης της πιθανότητας λάθους ενός πιθανοτικού αλγορίθμου πολυωνυμικού χρόνου, με αμφίπλευρη πιθανότητα λάθους; Υπόδειξη: δοκιμάστε επαναλήψεις και επιλογή της συχνότερης απάντησης (υποθέστε απαντήσεις 'ναι' ή 'όχι' και ότι η αρχική πιθανότητα είναι μικρότερη από $c < 1/2$).

4. Έστω f μία συνάρτηση μονής κατεύθυνσης. Ορίζουμε την γλώσσα L:

$$L = \{(a, b) \mid \exists x : f(x) = a \ \wedge \ b \text{ είναι suffix του } x\}$$

   (α) Αποδείξτε ότι $L \in UP$.

   (β) Αποδείξτε ότι $L \in UP \setminus P$.

5. Ισχύουν τα (α) και (β) στην παραπάνω άσκηση αν ο ορισμός της L ζητάει το b να είναι απλώς substring του x;

3.9 Ηλεκτρονικό Υλικό

- Διαδραστικές Παρουσιάσεις - Video
  - Διαλέξεις του Avi Wigderson σχετικά με υπολογιστική πολυπλοκοτητα, κρυπτογραφία και τυχαιότητα.
  - Εισαγωγή στις μηχανές Turing

Βιβλιογραφία

[1] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[2] Oded Goldreich. Computational complexity: A conceptual perspective. *SIGACT News*, 39(3):35–39, September 2008.

[3] Christos M. Papadimitriou. *Computational complexity*. Addison-Wesley, Reading, Massachusetts, 1994.

[4] St. Zachos and Aris Pagourtzis. Στοιχεία Θεωρίας Αριθμών και Εφαρμογές στην Κρυπτογραφία. Πανεπιστημιακές Σημειώσεις, 2015.

Κεφάλαιο 4

Υπολογιστικά Προβλήματα και Αλγόριθμοι στην Κρυπτογραφία

Στο κεφάλαιο αυτό θα περιγράψουμε βασικούς αλγόριθμους που σχετίζονται με έννοιες της Θεωρίας Αριθμών και έχουν άμεση εφαρμογή στην κρυπτογραφία.

4.1 Αλγόριθμος Επαναλαμβανόμενου Τετραγωνισμού

Θέλουμε να υπολογίσουμε το $a^m \bmod n$. Έστω ότι η δυαδική αναπαράσταση του m είναι $m = b_{k-1}2^{k-1} + \ldots b_1 2 + b_0$. Αυτό μπορεί να γίνει με τον παρακάτω αλγόριθμο:

function power:
input: $a, b_0, b_1, \ldots, b_{k-1}$
  $x := a$
  $y := 1$
  for i:=0 **to** k-1 **do**
  begin
    if $b_i = 1$ **then** $y := (y \cdot x) \bmod n$
    $x := x^2 \bmod n$
  end
return y

Σχήμα 4.1: Αλγόριθμος Επαναλαμβανόμενου Τετραγωνισμού

Ο παραπάνω αλγόριθμος πραγματοποιεί το πολύ $2\lceil \log_2 m \rceil$ πολλαπλασιασμούς.

4.2 Αλγόριθμος του Ευκλείδη

Η εύρεση του ΜΚΔ δύο φυσικών αριθμών a, b είναι απλή διαδικασία καθώς μπορεί να χρησιμοποιηθεί ο αλγόριθμος του Ευκλείδη, που δίνει αποτέλεσμα με $O(\log a)$ διαιρέσεις($O(\log^3 a)$) bit operations). Για οποιαδήποτε a, b λοιπόν με διαδοχικές διαιρέσεις έχουμε:

$$\begin{array}{rll} a & = bq_1 + r_1 & 0 < r_1 < b \\ b & = r_1q_2 + r_2 & 0 < r_2 < r_1 \\ r_1 & = r_2q_3 + r_3 & 0 < r_3 < r_2 \\ & \vdots & \vdots \\ r_{j-2} & = r_{j-1}q_j + r_j & 0 < r_j < r_{j-1} \\ r_{j-1} & = r_jq_{j+1} & \end{array}$$

Μέχρι να βρούμε ένα r_j που να διαιρεί ακριβώς το προηγούμενο υπόλοιπο r_{j-1}. Τότε το r_j είναι ο ΜΚΔ. Για παράδειγμα:

Παράδειγμα 6.

$$\begin{array}{rcl} 1742 & = & 3 \cdot 494 + 260 \\ 494 & = & 1 \cdot 260 + 234 \\ 260 & = & 1 \cdot 234 + 26 \\ 234 & = & 9 \cdot 26 \end{array} \qquad \begin{array}{rcl} 132 & = & 3 \cdot 35 + 27 \\ 35 & = & 1 \cdot 27 + 8 \\ 27 & = & 3 \cdot 8 + 3 \\ 8 & = & 2 \cdot 3 + 2 \\ 3 & = & 1 \cdot 2 + 1 \\ 2 & = & 2 \cdot 1 \end{array}$$

Επομένως $(1742, 494) = 26$ *και* $(132, 35) = 1$.

Άρα ο αλγόριθμος του Ευκλείδη έχει ως εξής:

function gcd:
input: a, b
  if $a > b$ **then**
    if $b|a$ **then** gcd:=b **else** gcd:=gcd(a mod b, b)
  else
    gcd:=gcd(b, a)
  return gcd

Σχήμα 4.2: Αλγόριθμος Ευκλείδη

Η ορθότητα του αλγόριθμου του Ευκλείδη στηρίζεται στην παρακάτω πρόταση:

Πρόταση 4.1. *Αν* $a, b \in \mathbb{Z}$ *τότε ισχύει ότι* $(a, b) = (b, a \bmod b)$.

Είναι φανερό ότι μία από τις πιο σημαντικές χρήσεις του αλγορίθμου του Ευκλείδη είναι η απόφαση για το αν δύο αριθμοί είναι σχετικά πρώτοι, όπου μπορεί να δώσει απάντηση σε πολυωνυμικό χρόνο.

4.3 Εκτεταμένος Αλγόριθμος του Ευκλείδη - Αντίστροφοι

Πρόταση 4.2 (Bezout). *Αν* $d = (a, b)$ *τότε* $\exists x, y \in \mathbb{Z} : d = xa + yb$

Δηλαδή ο ΜΚΔ μπορεί να γραφεί ως γραμμικός συνδυασμός των a, b. Μάλιστα αυτό μπορεί να γίνει σε πολυωνυμικό χρόνο ($O(\log^3(a)$ bit operations). Ο αλγόριθμος που υπολογίζει τα x, y στηρίζεται στον αλγόριθμο του Ευκλείδη κάνοντας την αντίστροφή διαδικασία π.χ. στο προηγούμενο παράδειγμα για να υπολογίσουμε το 26 (το 1) ως γραμμικό συνδυασμό των 1742 και 494 (των 132 και 35) κάνουμε τα εξής:

26	=	260 - 234	1	=	3 - 2
	=	260 -(494 - 260)		=	3 - (8 - 2 · 3)
	=	2· 260 - 494		=	3 · 3 - 8
	=	2(1742-3· 494)-494		=	3(27-3· 8) - 8
	=	2· 1742-7· 494		=	3· 27 - 10 · 8
				=	3· 27 - 10(35-27)
				=	13· 27 -10 · 35
				=	13(132-3· 35) - 10· 35
				=	13· 132 - 49· 35

4.4 Primality - Factoring

Παρακάτω θα ασχοληθούμε με δύο παρεμφερή προβλήματα με εντελώς διαφορετικές όμως ιδιότητες. Το πρώτο είναι το περίφημο `PRIMALITY` :

Ορισμός 4.3 (`PRIMALITY`). Δίνεται ένας αριθμός, είναι πρώτος;

Το δεύτερο είναι το `FACTORING` :

Ορισμός 4.4 (`FACTORING`). Δίνεται ένας αριθμός, να βρεθούν οι πρώτοι παράγοντες του.

Και τα δύο προβλήματα είναι πολύ σημαντικά στην κρυπτογράφηση, αποκρυπτογράφηση, αλλά κυρίως για την κρυπτανάλυση ενός κρυπτογραφικού συστήματος.

4.5 Αλγόριθμοι Ελέγχου Πρώτων

Ιστορική Αναδρομή

Μία από τις παλιότερες λύσεις του προβλήματος PRIMALITY (primality tests) είναι το *Κόσκινο του Ερατοσθένη*.

Αφού ορίσουμε ένα ανώτατο όριο (αριθμό) k γράφουμε όλους τους αριθμούς $2, 3, 4, 5, \ldots, k$. Στη συνέχεια, παίρνουμε τον πρώτο αριθμό (εδώ το 2), τον μαρκάρουμε σαν πρώτο και διαγράφουμε όλα τα πολλαπλάσιά του που υπάρχουν στη λίστα $(4, 6, 8, \ldots)$. Στη συνέχεια παίρνουμε τον επόμενο αριθμό που δεν έχει διαγραφεί, τον μαρκάρουμε ως πρώτο και διαγράφουμε όλα τα πολλαπλάσιά του που υπάρχουν στη λίστα κ.ο.κ. Τελικά θα παραμείνουν μόνο οι πρώτοι που είναι μικρότεροι ίσοι του k.

Το 17ο αιώνα έγινε ένα νέο βήμα πάνω στο ανοιχτό πρόβλημα με το μικρό θεώρημα Fermat 2.38:

Θεώρημα 4.5 (Μικρό Θεώρημα Fermat)**.**

$$\forall a \in \mathbb{Z},\ \forall \text{prime } p \not| a: \quad a^{p-1} \equiv 1 \pmod{p}$$

Ο παραπάνω έλεγχος μπορεί να μετατραπεί σε ένα πιθανοτικό αλγόριθμο που θα μπορεί να αποφασίζει αν ένας αριθμός είναι πρώτος ή όχι, με πολύ απλό τρόπο. Αρκεί να κάνουμε τον έλεγχο για ένα ικανοποιητικό πλήθος από a. Αν έστω και μια φορά ο έλεγχος αποτύχει είμαστε σίγουροι ότι ο αριθμός δεν είναι πρώτος κάτι το οποίο δεν ισχύει και αντιστρόφως. Υπάρχουν δηλαδή αριθμοί που περνούν τον παραπάνω έλεγχο για κάθε a χωρίς να είναι πρώτοι. Επομένως, το μικρό Θεώρημα του Fermat δε μας απαντά με βεβαιότητα στο πρόβλημα.

Το 1976 ο Miller επινόησε ένα ντετερμινιστικό αλγόριθμο που μπορούσε να δώσει απάντηση στο πρόβλημα και μάλιστα σε πολυωνυμικό χρόνο. Ο αλγόριθμος βασιζόταν στην εκτεταμένη υπόθεση του Riemann. Επομένως ο Miller έδειξε ότι το πρόβλημα βρίσκεται στο P αν η υπόθεση του Riemann είναι σωστή. Το τελευταίο είναι ένα από τα πλέον γνωστά ανοιχτά προβλήματα που μένουν άλυτα εδώ και 100 περίπου χρόνια χωρίς να μπορεί να αμφισβητηθεί σοβαρά.

Ένα χρόνο μετά τον Miller, οι Solovay και Strassen δημοσίευσαν ένα νέο πιθανοτικό αλγόριθμο που έδινε απάντηση στο πρόβλημα. Η πιθανότητα λάθους μπορούσε να περιοριστεί κατά βούληση, κάτι που έθετε το πρόβλημα στην κλάση BPP 3.15. Πιο συγκεκριμένα, αυτό που έδειξαν οι Solovay και Strassen είναι ότι το πρόβλημα βρίσκεται στην κλάση $co - RP$ 3.6. Έτι ο αλγόριθμός τους, αναγνώριζε όλους τους πρώτους σωστά και με πιθανότητα λάθους αυθαίρετα μικρή, απαντούσε αν ένας αριθμός είναι σύνθετος.

Λίγο αργότερα, ο Rabin τροποποιεί τον αλγόριθμο του Miller σε έναν επίσης πιθανοτικό πολυωνυμικό αλγόριθμο, ο οποίος επίσης αποδείκνυε ότι το πρόβλημα βρίσκεται στο $co-RP$. Ο τελευταίος αλγόριθμος και ελαφρά τροποποιημένος από τον Knuth είναι γνωστός ως έλεγχος Miller-Rabin και είναι ο πλέον διαδεδομένος.

Το 1983 οι Adleman, Pomerance και Rumely παρουσίασαν για πρώτη φορά μια νέα μέθοδο που αποδεχόταν το πρόβλημα των πρώτων σε χρόνο $(\log n)^{O(\log\log\log n)}$. Αν και πρακτικά σχεδόν πολυωνυμικός χρόνος, η προσπάθειά τους δεν επέτρεπε στους θεωρητικούς να θέσουν το πρόβλημα στο P. Πάραυτα ήταν η πρώτη πολύ καλή προσπάθεια που έδινε απάντηση σε αποτελεσματικό χρόνο και πάνω από όλα με απουσία τυχαιότητας. Η εφαρμογή του όμως δεν είναι ιδιαίτερα απλή. Οι Cohen και Lenstra βοήθησαν στη θεωρητική και αλγοριθμική απλοποίηση της μεθόδου η οποία όμως δεν έπαψε να απαιτεί από ισχυρούς υπολογιστές μερικά λεπτά για να αποφασίσει για αριθμούς με αναπαράσταση της τάξης των εκατοντάδων ψηφίων. Πάντα δε, υπήρχε και το πρόβλημα της υλοποίησης, που δεν επέτρεπε την ορθή μεταφορά σε ένα κώδικα χωρίς λάθη.

Το 1986 οι Goldwasser και Killian προτείνουν ένα νέο αλγόριθμο με αναμενόμενο πολυωνυμικό χρόνο απόφασης που βασιζόταν σε ελλειπτικές καμπύλες. Ο αλγόριθμος υπέβαλε ένα πιστοποιητικό για πρώτους αριθμούς που έδινε απάντηση για σχεδόν όλους τους αριθμούς σε αποτελεσματικό χρόνο. Στην πράξη ο αλγόριθμος μπορεί να χαρακτηριστεί αναποτελεσματικός, κάτι που έκανε τον Atkin να αναπτύξει μία διαφοροποιημένη μέθοδο γνωστή και ως έλεγχος $ECPP$[1]. Ο τελευταίος αλγόριθμος χρησιμοποιήθηκε για να πιστοποιηθούν πρώτοι με τάξη μεγέθους αναπαράστασης άνω των 1000 ψηφίων στο δεκαδικό σύστημα.

Οι Adleman και Huang το 1992, τροποποίησαν τον αλγόριθμο των Goldwasser και Killian σε μια πιθανοτική μέθοδο που σε πολυωνυμικό χρόνο πιστοποιούσε πρώτους θέτοντας το πρόβλημα των πρώτων στο RP. Με τη βοήθεια παλαιότερων αποτελεσμάτων το τελευταίο συμπέρασμα έθετε το πρόβλημα στην κλάση ZPP 3.18.

Οριστικό τέλος δόθηκε με την εργασία των Agrawal, Kayal και Saxena που εκδόθηκε τις πρώτες μέρες του Αυγούστου του 2002 θέτοντας το πρόβλημα στο P (γνωστός πλέον και ως αλγόριθμος AKS). Παρά την αρχική πολύπλοκη μορφή της απόδειξης της ορθότητας του αλγορίθμου τους, δεν πέρασε πολύ καιρός για να πάρει μια πιο απλή και αποτελεσματική μορφή μετά από παρατηρήσεις του Lenstra. Για την ακρίβεια η πρώτη δυσνόητη και με μικρά λάθη εργασία αντικαταστάθηκε λίγους μήνες μετά την πρώτη δημοσίευση από μια νέα που απαιτούσε από τον αναγνώστη "στοιχειώδη", όπως χαρακτηριστικά σχολιάστηκε, γνώση μαθηματικών και άλγεβρας ενώ κατέβασε παράλληλα και την πολυπλοκότητα του αλγορίθμου. Αξιοσημείωτο είναι επίσης το γεγονός ότι πριν τη δημοσίευση της

[1] Elliptic Curve Primality Proving algorithm.

εργασίας και χωρίς να υπάρχει αυστηρή απόδειξη, ο Agrawal είχε πιστέψει στον αλγόριθμό του αποδεικνύοντας τον αρχικά με υπόθεση τη γενικευμένη υπόθεση Riemann.

Στη συνέχεια θα εξετάσουμε τους αλγορίθμους Solovay - Strassen και Miller - Rabin καθώς και τον νέο αλγόριθμο (AKS).

Τεστ Solovay-Strassen

Το τεστ Solovay-Strassen βασίζεται στο κριτήριο του Euler:
$b^{\frac{n-1}{2}} = (\frac{b}{n}) \pmod n$ αν ο n είναι πρώτος και $b \in U(\mathbb{Z}_n)$.
Η παραπάνω σχέση όμως ισχύει και για περιττό, μη πρώτο n για τους μισούς αριθμούς του $U(\mathbb{Z}_n)$, ενώ δεν ισχύει για τους άλλους μισούς. Έτσι παίρνουμε τον παρακάτω πιθανοτικό αλγόριθμο:

1. Διαλέγουμε k ακεραίους $0 < k < n$ τυχαία.
2. Για κάθε $i \in \{1, \ldots, k\}$ υπολογίζουμε το $b^{\frac{n-1}{2}} \bmod n$ και το σύμβολο Jacobi $\left(\frac{b}{n}\right)$.
3. Αν τα δύο μέρη δεν είναι ισότιμα $modulo\ n$, τότε ανακοινώνεται ότι το n δεν είναι πρώτος (με βεβαιότητα) και ο αλγόριθμος σταματά.
4. Αν έχουν γίνει k επαναλήψεις ανακοινώνεται ότι ο n περνά το τεστ (δηλ. είναι πιθανότατα πρώτος).

Με βάση την παραπάνω παρατήρηση, αν το n περάσει το τεστ και για τους k ακεραίους, τότε η πιθανότητα να μην είναι πρώτος είναι το πολύ $1/2^k$.

Τεστ Miller-Rabin

1. Έστω ότι το n είναι μεγάλος, θετικός, περιττός αριθμός.
2. Διαλέγουμε τυχαία b, όπου $0 < b < n$. Αν $\gcd(b, n) \neq 1$ ή $b^{n-1} \neq 1 \pmod n$ τότε το n είναι σύνθετος (με βεβαιότητα) και η εκτέλεση σταματά. Αλλιώς:
3. Γράφουμε $n-1 = 2^s t$, με t περιττό και υπολογίζουμε το $b^t \bmod n$. Αν είναι ± 1, τότε το n περνά το τεστ.
4. Αλλιώς, υψώνουμε στο τετράγωνο το $b^t \bmod n$, έπειτα το ξαναϋψώνουμε στο τετράγωνο mod$n$ κ.ο.κ. για $s-1$ φορές το πολύ, έως ότου πάρουμε ± 1.

5. Αν πήραμε -1 τότε το n περνάει το τεστ (πιθανόν πρώτος).

6. Αν πήραμε $+1$, τότε το n δεν περνά το τεστ (σύνθετος, με βεβαιότητα) και η εκτέλεση σταματά.

7. Επαναλαμβάνουμε τα παραπάνω βήματα 2–6 k φορές, με διαφορετικό b κάθε φορά. Αν το n περάσει το τεστ και τις k φορές ανακοινώνεται ότι ο n περνάει το τεστ (δηλ. είναι πρώτος με πολύ μεγάλη πιθανότατα).

Αποδεικνύεται παρακάτω ότι το πολύ τα μισά b περνούν το τεστ για n σύνθετο. Συνεπώς, αν το n περάσει το τεστ για k τυχαίες επιλογές του b, τότε η πιθανότητα να μην είναι πρώτος είναι $1/2^k$.

Θεώρημα 4.6. *Αν n πρώτος, τότε περνάει τον έλεγχο Miller-Rabin πάντοτε (για όλα τα b). Αν n σύνθετος, τότε για τα μισά τουλάχιστον b του $U(\mathbb{Z}_n)$ δεν περνάει τον έλεγχο.*

Απόδειξη. Για το πρώτο (n πρώτος), αρκεί να θυμηθούμε ότι $\forall b \in \mathbb{Z}_n \backslash \{0\}, b^{n-1} = 1 \pmod{n}$ (Θ. Fermat). Στη συνέχεια του τεστ, κάθε b που θα επιλεγεί θα δώσει ακολουθία

$\langle b^t, b^{2t}, \ldots, b^{2^i t}, \ldots b^{2^s t} \rangle \pmod{n}$

που είτε ξεκινάει με ισοτιμία $1 \pmod{n}$, είτε εμφανίζει $-1 \pmod{n}$ ακριβώς πριν από την πρώτη εμφάνιση του $1 \pmod{n}$. Αυτό συμβαίνει γιατί έχουμε διαδοχικούς τετραγωνισμούς, και οι μόνες δυνατές ρίζες του 1 modulo n είναι ± 1 όταν ο n είναι πρώτος.

Η απόδειξη για το δεύτερο (n σύνθετος) βασίζεται στο Θεώρημα του Lagrange. Συγκεκριμένα, αποδεικνύουμε ότι οι αριθμοί b για τους οποίους ο n περνά το τεστ, περιέχονται σε υποομάδα του $U(\mathbb{Z}_n)$. Θα αποδείξουμε επίσης ότι υπάρχει τουλάχιστον ένας αριθμός b ώστε το n δεν περνάει το τεστ. Επομένως, από Θ. Lagrange προκύπτει ότι το πλήθος των b που περνούν το τεστ είναι το πολύ $|U(\mathbb{Z}_n)|/2$.

Καταρχήν παρατηρούμε ότι το σύνολο $A = \{b | b^{n-1} = 1 \pmod{n}\}$ είναι κλειστό ως προς τον πολλαπλασιασμό, άρα αποτελεί υποομάδα $U(\mathbb{Z}_n)$. Επομένως, αν υπάρχει έστω και ένα $b \in U(\mathbb{Z}_n) \setminus A$, τότε το A είναι γνήσια υποομάδα του $U(\mathbb{Z}_n)$ και άρα η τάξη της είναι το πολύ $|U(\mathbb{Z}_n)|/2$. Αν λοιπόν για κάθε σύνθετο n υπήρχε κάποιο $b \in U(\mathbb{Z}_n)$, τ.ώ. $b^{n-1} \neq 1 \pmod{n}$, τότε θα υπήρχαν πολλά b τα οποία θα έδειχναν την "συνθετότητα" του n ήδη από το Βήμα 2 του παραπάνω αλγορίθμου. Με άλλα λόγια, το τεστ του Fermat θα ήταν ένας καλός πιθανοτικός αλγόριθμος για το `PRIMALITY` . Όμως, υπάρχουν κάποιοι σύνθετοι αριθμοί, οι αριθμοί Carmichael, που έχουν την ιδιότητα $A = U(\mathbb{Z}_n)$. Για αυτούς τους αριθμούς χρειάζονται τα επόμενα βήματα του τεστ Miller-Rabin. Στη συνέχεια θα ασχοληθούμε μόνο με αριθμούς Carmichael.

Χρησιμοποιούμε την απεικόνιση $b \mapsto seq(b) = \langle b^t, b^{2t}, \ldots, b^{2^i t}, \ldots b^{2^s t} \rangle \pmod{n}$. Λέμε ότι η $seq(b)$ είναι *παραγοντοποιητική ακολουθία* (factoring sequence) αν είναι της μορφής

$$\langle \neq \pm 1, \ldots, \neq \pm \mathbf{1}, = \mathbf{1}, \ldots = 1 \rangle \pmod{n}$$

για τον λόγο ότι η ύπαρξή της συνεπάγεται ότι ο n παραγοντοποιείται (γιατί;). Προσέξτε ότι σε αυτήν την περίπτωση το τεστ Miller-Rabin επιστρέφει 'Σύνθετος'.

Προφανώς, οι ακολουθίες $seq(b)$ που περιέχουν σε οποιοδήποτε σημείο το -1 δεν είναι παραγοντοποιητικές. Ας ορίσουμε ως M το σύνολο των b που απεικονίζονται σε τέτοιες ακολουθίες:

$$M = \{b \in U(\mathbb{Z}_n) \mid -1 \in seq(b)\}$$

Προσέξτε ότι το M δεν είναι κενό: περιέχει τουλάχιστον το -1. Επομένως, μπορούμε να ονομάσουμε u το στοιχείο που δίνει την ακολουθία που έχει το -1 πιο "δεξιά" από όλα τα στοιχεία του M, έστω στη θέση j, δηλαδή ισχύει ότι $u^{2^j t} \equiv -1 \pmod{n}$.

Τέλος, ας ορίσουμε ως B το σύνολο των στοιχείων που παράγουν ακολουθίες που στην θέση j έχουν ισοτιμία $\pm 1 \pmod{n}$:

$$B = \{b \in U(\mathbb{Z}_n) \mid b^{2^j t} \equiv \pm 1 \pmod{n}\}$$

Είναι φανερό ότι $M \subseteq B$, αλλά και ότι όλα τα άλλα στοιχεία με μη παραγοντοποιητική ακολουθία ανήκουν στο B (γιατί;).

Παρατηρούμε τώρα ότι το B είναι κλειστό ως προς πολλαπλασιασμό $\pmod{n}$. Επομένως είναι υποομάδα του $U(\mathbb{Z}_n)$. Θα δείξουμε ότι το B είναι γνήσιο υποσύνολο του $U(\mathbb{Z}_n)$.

Πράγματι, έστω $n = n_1 n_2, \gcd(n_1, n_2) = 1$ (η περίπτωση $n = p^e, e > 1$, θα εξεταστεί παρακάτω). Τότε, λόγω Κινέζικου Θεωρήματος Υπολοίπων (CRT) υπάρχει στοιχείο $w \in U(\mathbb{Z}_n)$ τ.ώ.

$$w = u \pmod{n_1} \quad w = 1 \pmod{n_2}$$

Από τις παραπάνω σχέσεις προκύπτει:

$$w^{2^j t} = -1 \pmod{n_1} \quad w^{2^j t} = 1 \pmod{n_2}$$

Επομένως:

$$w^{2^j t} \neq \pm 1 \pmod{n}$$

Οπότε, επειδή το B είναι γνήσια υποομάδα του $U(\mathbb{Z}_n)$, και (λόγω Θ. Lagrange) η πληθικότητά της διαιρεί την πληθικότητά της $U(\mathbb{Z}_n)$, θα έχουμε αναγκαστικά:

$$|B| \leq \frac{|U(\mathbb{Z}_n)|}{2}$$

Επειδή το B περιέχει *όλα* τα στοιχεία του $U(\mathbb{Z}_n)$ που δίνουν μη παραγοντοποιητικές ακολουθίες (ενδεχομένως και μερικά που δίνουν παραγοντοποιητικές), καταλαβαίνουμε ότι με πιθανότητα τουλάχιστον $1/2$ μια τυχαία επιλογή του b θα ανήκει στο $U(\mathbb{Z}_n) \setminus B$ και επομένως θα δώσει παραγοντοποιητική ακολουθία, αποκαλύπτοντας με βεβαιότητα ότι ο n είναι σύνθετος.

Μένει να εξετάσουμε την περίπτωση $n = p^e, e > 1$. Αποδεικνύεται ότι, επειδή η $U(\mathbb{Z}_n)$ σε αυτή την περίπτωση είναι κυκλική (από γνωστό θεώρημα της Θεωρίας Αριθμών) δεν μπορεί να ισχύει $b^{n-1} \equiv 1$ για b που είναι γεννήτορας της $U(\mathbb{Z}_n)$, καθώς θα έπρεπε $\phi(p^e)|p^e - 1$, που οδηγεί σε αντίφαση (άσκηση: συμπληρώστε τις λεπτομέρειες). Επομένως δεν μπορεί το n να είναι αριθμός Carmichael, άρα καλύπτεται από τον έλεγχο Fermat, όπως εξηγήσαμε νωρίτερα. □

Ο αλγόριθμος Agrawal Kayal Saxena Primality Test (AKS)

Η σπουδαιότητα του AKS έγκειται στο συνδυασμό του ντετερμινισμού με την απόδοση.

Η βασική ιδέα για τον αλγόριθμο AKS

Η βασική ιδέα για τον αλγόριθμο είναι ένας έλεγχος για πρώτους στον οποίο βασίστηκε και ένας παλαιότερος πιθανοτικός αλγόριθμος των Agrawal και Biswas (1999). Πρόκειται για μια γενίκευση του μικρού θεωρήματος Fermat.

Λήμμα 4.7. *Έστω a, p σχετικά πρώτοι μεταξύ τους. Τότε τα επόμενα είναι ισοδύναμα :*

1. *Ο p είναι πρώτος*
2. $(x - a)^p = (x^p - a) \pmod{p}$

Να τονίσουμε ότι η σχέση *2* του παραπάνω λήμματος αναφέρεται στην ισότητα των συντελεστών των πολυωνύμων με μεταβλητή x, και κατ' επέκταση και στα

ίδια τα πολυώνυμα. Η απόδειξή του είναι πολύ απλή και βασίζεται στο γνωστό διώνυμο του Νεύτωνα:

$$(x-a)^p = \sum_{i=0}^{p} \binom{p}{i} x^i (-a)^{p-i}.$$

Το Λήμμα 4.7 μας επιτρέπει να έχουμε ένα κριτήριο για τον έλεγχο των πρώτων αριθμών. Αν λοιπόν ρωτάμε αν ο p είναι πρώτος, αρκεί για κατάλληλο a να απαντήσουμε αν ισχύει η σχέση $(x-a)^p = (x^p - a) \pmod p$. Θεωρώντας ότι το πρόβλημά μας έχει μήκος εισόδου n, τότε είναι πιθανό να χρειαστεί να κάνουμε n ελέγχους διαιρετότητας (για κάθε όρο του παραπάνω πολυωνύμου). Αυτό θα σήμαινε ότι η πολυπλοκότητα θα φράσσει το από κάτω από τη συνάρτηση $f(n) = n$ και έτσι ένας πολυλογαριθμικός[2] αλγόριθμος θα έπρεπε να έχει ξεχαστεί. Αντιθέτως μπορούμε να τροποποιήσουμε τον παραπάνω έλεγχο έτσι ώστε να πετύχουμε το άνω φράγμα στην πολυπλοκότητα με το να μειώσουμε την τάξη των πολυωνύμων $(x-a)^p,\ x^p - a$. Αυτό θα επιτευχθεί υπολογίζοντας τα δύο πολυώνυμα $\bmod(x^r - 1)$, για κατάλληλο r. Ο νέος έλεγχος θα έχει τη μορφή

$$(x-a)^p =_? (x^p - a) \pmod{x^r - 1, p}$$

για ένα κατάλληλο πλήθος από a που δεν είναι παραπάνω από πολυλογαριθμικό. Σκεφτείτε ότι πολυλογαριθμικό για την είσοδό μας σημαίνει πολυωνυμικό για το μέγεθος της αναπαράστασής της. Επίσης θα δείξουμε ότι και το επιθυμητό r φράσσεσαι κατάλληλα. Βεβαίως δεν είναι σαφής η πλήρης ισοδυναμία μεταξύ των δύο παραπάνω ελέγχων αλλά επίσης δεν είναι και αλήθεια. Ενώ λοιπόν στο Λήμμα 4.7 μπορούμε να έχουμε ένα αυστηρό κριτήριο για τον έλεγχο πρώτων αριθμών, δεν ισχύει το ίδιο για το νέο κριτήριο. Από το Λήμμα 4.7 είναι προφανές ότι ένας πρώτος p ικανοποιεί τη σχέση $(x-a)^p = (x^p - a) \bmod (x^r - 1, p)$, ωστόσο η τελευταία σχέση δεν ικανοποιείται μόνο για πρώτους αριθμούς. Τα κατάλληλα (όπως χαρακτηρίστηκαν) a και r, τα οποία συνεισφέρουν στην μείωση της πολυπλοκότητας είναι τέτοια που να μπορούν να μας εξασφαλίσουν την ισοδυναμία για τη σχέση $(x-a)^p = (x^p - a) \bmod (x^r - 1, p) \ \Leftrightarrow \ p$ πρώτος.

Ο αλγόριθμος AKS και η ορθότητά του

Η ορθότητα του αλγορίθμου βασίζεται μεταξύ άλλων και σε ένα γνωστό αποτέλεσμα που αποδίδεται στον Chebyshev:

Λήμμα 4.8. *Έστω $LCM(m)$ το ελάχιστο κοινό πολλαπλάσιο των m πρώτων φυσικών αριθμών. Τότε για $m \geq 7$ ισχύει $LCM(m) \geq 2^m$.*

[2] Σύνθεση του λογαρίθμου με ένα πολυώνυμο, ή αλλιώς συνάρτηση μεγέθους $O(\log^k n)$ για κάποιο $k > 0$.

Πριν παρουσιάσουμε τον αλγόριθμο να τονίσουμε ότι πρόκειται για τη μορφή που πήρε μετά από φιλικές επισημάνσεις του Lenstra. Η αρχική μορφή του ήταν σίγουρα πιο πολύπλοκη και με μικρά λάθη τα οποία αν και αναγνωρίστηκαν γρήγορα δε μπορούσαν να αποτρέψουν τα συγχαρητήρια της κοινότητας προς τους τρεις Ινδούς. Η ορθότητα του αλγορίθμου είναι τετριμμένη προς τη μία κατεύθυνση και απαιτεί γνώσεις από τη θεωρία Galois για περαιτέρω κατανόηση για την άλλη κατεύθυνση. Τέλος να αναφέρουμε ότι δεν πρόκειται για την πιο εξελιγμένη μορφή του αλγορίθμου (κυρίως ως προς την πολυπλοκότητα για να μην αναφέρουμε ακόμα συνδυαστικές εφαρμογές του αλγορίθμου με άλλα συστήματα), πρόκειται όμως για μια βατή μορφή που προσφέρεται για κατανόηση της κεντρικής ιδέας σε θεωρητικό επίπεδο.

Ο αλγόριθμος

---

input: ακέραιος $n > 1$

1. **if** $(n = a^b$, για κάποια $a, b \in \mathbb{N},\ b > 1)$, **output: COMPOSITE**
2. Βρες το μικρότερο r για το οποίο $o_r(n) > 4\log^2 n$
3. **if** $1 < (a, n) < n$ για κάποιο $a < r$, **output: COMPOSITE**
4. **if** $n < r$, **output: PRIME**
5. **for** $a = 1$ **to** $\lfloor 2\sqrt{\phi(\mathrm{r})}\mathrm{logn} \rfloor$ **do**

   if $(\mathrm{x} - a)^{\mathrm{n}} \neq (\mathrm{x}^{\mathrm{n}} - a) \mathrm{mod}(\mathrm{x}^{\mathrm{r}} - 1, \mathrm{n})$, **output: COMPOSITE**
6. **output: PRIME**

Θεώρημα 4.9. *Ο παραπάνω αλγόριθμος επιστρέφει **PRIME** αν και μόνον αν ο ακέραιος n είναι πρώτος.*

Με την απόδειξη του θεωρήματος θα ασχοληθούμε περιληπτικά στη συνέχεια. Καταρχήν εύκολα μπορούμε να αποδείξουμε τη μια φορά της ισοδυναμίας.

Λήμμα 4.10. *Αν ο n είναι πρώτος, τότε ο αλγόριθμος AKS επιστρέφει **PRIME**.*

Απόδειξη. Είναι προφανές ότι αν ο n είναι πρώτος, τότε δεν υπάρχουν $a, b \in \mathbb{N},\ b > 1)$, με $n = a^b$, οπότε το πρώτο βήμα δε μπορεί να επιστρέψει **COMPOSITE**. Το 3ο βήμα επίσης δε μπορεί να επιστρέψει λάθος αποτέλεσμα. Επίσης από το Λήμμα 4.7 (και την παρατήρηση στη γενίκευση του) αν ο p είναι πρώτος, ισχύει ότι $(x-a)^p = (x^p - a) \bmod (x^r - 1, p)$, και επομένως η συνθήκη του πέμπτου βήματος του αλγορίθμου δε μπορεί να ικανοποιηθεί. Ώστε ο αλγόριθμος επιστρέφει **PRIME** έστω και στο τελευταίο βήμα □

Θα επικεντρωθούμε τώρα στη σκιαγράφηση του αντιστρόφου, δηλαδή να αποδείξουμε ότι αν ο αλγόριθμος επιστρέψει **PRIME**, τότε ο αριθμός n είναι πρώτος. Αρχικά να παρατηρήσουμε ότι αν στο βήμα 4 ο αλγόριθμος επιστρέψει **PRIME** τότε θα έχει ικανοποιηθεί η συνθήκη $n \leq r$, δηλαδή στο βήμα 3 θα έχουν γίνει αρκετοί έλεγχοι για εύρεση παράγοντα οι οποίοι θα έχουν αποτύχει, κάτι το οποίο μας επιτρέπει να χαρακτηρίσουμε το n πρώτο. Ώστε μένει να εξετάσουμε το βήμα 6 του αλγορίθμου και να δείξουμε ότι αν επιστραφεί **PRIME**, τότε πράγματι ο n είναι πρώτος. Στη συνέχεια θα υποθέτουμε ότι ο αλγόριθμος έχει αποφανθεί **PRIME** στο 6ο βήμα και στόχος μας θα είναι να δείξουμε πράγματι ότι ο αριθμός της εισόδου είναι πρώτος.

Ο αλγόριθμος έχει δύο ιδιαίτερα σημαντικά στάδια, τα βήματα 2 και 5. Στο δεύτερο βήμα ο αλγόριθμος μας παρέχει έναν ακέραιο r, ο οποίος στη συνέχεια χρησιμοποιείται για να μειώσει τον αρχικά δαπανηρό έλεγχο του βήματος 5. Η ύπαρξη ενός τέτοιου αριθμού εξασφαλίζεται με το παρακάτω λήμμα.

Λήμμα 4.11. *Υπάρχει r, $r \leq \lceil 16 \log^5 n \rceil$ έτσι ώστε $o_r(n) > 4 \log^2 n$.*

Απόδειξη. Ας υποθέσουμε ότι $r_1, r_2, \ldots, r_t$ είναι όλοι οι αριθμοί με την ιδιότητα $o_{r_i}(n) \leq 4 \log^2 n$. Έστω $o_{r_i}(n) = k_i$. Τότε το r_i έχει την ιδιότητα να διαιρεί το $n^{k_i} - 1$. Μια και $k_i \leq 4 \log^2 n$ έπεται ότι κάθε ένα από τα r_i διαιρεί το γινόμενο

$$\prod_{i=1}^{\lfloor 4 \log^2 n \rfloor} (n^i - 1)$$

.

Εύκολα παρατηρούμε ότι

$$\begin{aligned}
&\textstyle\prod_{i=1}^{\lfloor 4 \log^2 n \rfloor} (n^i - 1) = \\
&(n-1)(n^2-1)\ldots(n^{\lfloor 4 \log^2 n \rfloor} - 1) = \\
&O(n^{1+2+\ldots \lfloor 4 \log^2 n \rfloor}) = \\
&O(n^{16 \log^4 n}) = O(2^{16 \log^5 n})
\end{aligned}$$

Από το Λήμμα 4.8 έχουμε ότι το ΕΚΠ των $\lceil 16 \log^5 n \rceil$ πρώτων αριθμών είναι τουλάχιστον $2^{\lceil 16 \log^5 n \rceil}$. Άρα όλοι οι r_i που έχουν την ιδιότητα $o_{r_i}(n) \leq 4 \log^2 n$

διαιρούν κάτι μικρότερο από $2^{16\log^5 n}$ που είναι το κάτω φράγμα για το ελάχιστο κοινό πολλαπλάσιο των $\lceil 16\log^5 n\rceil$ πρώτων αριθμών. Αν όλοι οι αριθμοί r μέχρι και τον $\lceil 16\log^5 n\rceil$ είχαν την ιδιότητα $o_r(n) \leq 4\log^2 n$ θα έπρεπε να διαιρούν κάτι μικρότερο από το ελάχιστο κοινό πολλαπλάσιό τους. Επομένως υπάρχει $r \leq \lceil 16\log^5 n\rceil$ έτσι ώστε $o_r(n) > 4\log^2 n$. □

Αξίζει να σημειωθεί ότι η αξία του λήμματος έχει να κάνει και με τη γρήγορη εύρεση του r κάτι που θα σχολιαστεί αργότερα και στην ανάλυση της πολυπλοκότητας. Το σίγουρο μέχρι στιγμής είναι ότι ένας τέτοιος αριθμός μπορεί τελικά να βρεθεί σε πολυωνυμικό αριθμό (σε σχέση με την αναπαράσταση της εισόδου) βημάτων.

Ας επανέλθουμε στον αλγόριθμο και ας θυμηθούμε ότι στο 5ο βήμα γίνονται $\lfloor 2\sqrt{\phi(r)}\log n\rfloor$ επαναλήψεις κατά τις οποίες ποτέ δεν ικανοποιείται η υπάρχουσα συνθήκη. Για ευκολία ας συμβολίσουμε με $l = \lfloor 2\sqrt{\phi(r)}\log n\rfloor$. Επομένως

$$(x-a)^n = (x^n - a) \bmod (x^r - 1, n),\ \forall a\ : 1 \leq a \leq l. \tag{4.1}$$

Έχοντας υποθέσει ότι p είναι ένας πρώτος παράγοντας του n, η παραπάνω σχέση γίνεται $(x-a)^n = (x^n - a) \bmod (x^r - 1, p)$, για κάθε a με $1 \leq a \leq l$. Ακόμα, από το Λήμμα 4.7 έχουμε ότι

$$(x-a)^p = (x^p - a) \bmod (x^r - 1, p),\ \forall a\ :\ 1 \leq a \leq l, \tag{4.2}$$

μια και από το 3ο βήμα έχουμε Ότις $(a, p) = 1$.

Ώστε οι n και p από τις σχέσεις (1) και (2) έχουν την ίδια συμπεριφορά. Η ιδιότητα αυτή απασχόλησε ιδιαίτερα τους τρεις Ινδούς, γι' αυτό και ονόμασαν τέτοιους αριθμούς για δεδομένα πολυώνυμα, ενδοσκοπικούς (introspective).

Να παρατηρήσουμε ότι από τις προηγούμενες σχέσεις είναι προφανές ότι οι p και n είναι ενδοσκοπικοί για τα πολυώνυμα $x - a$, $\forall a$ με $1 \leq a \leq l$. Για αυτούς τους αριθμούς υπάρχουν ενδιαφέρουσες ιδιότητες όπως το ότι είναι κλειστοί για πολλαπλασιασμό και ως προς τους ακέραιους αλλά και για τα πολυώνυμα. Αυτές οι ιδιότητες μας επιτρέπουν να εξάγουμε χρήσιμα συμπεράσματα.

Ας θεωρήσουμε τα σύνολα $I = \{n^i p^j | i, j \geq 0\}$ και $P = \{\prod_{a=1}^{l}(x-a)^{e_a} | e_a \geq 0\}$. Κάθε στοιχείο του πρώτου είναι ενδοσκοπικό για κάθε στοιχείο του δεύτερου. Στη συνέχεια θα ορίσουμε δύο νέες ομάδες που βασίζονται στα προηγούμενα σύνολα και που θα παίξουν σημαντικό ρόλο στην απόδειξη.

Η πρώτη ομάδα θα είναι υποομάδα της Z_r^* και θα είναι τα στοιχεία του I modulo r. Ας συμβολίσουμε αυτή την ομάδα με G και ας είναι $|G| = t$. Για αυτή την ομάδα αποδεικνύεται το παρακάτω λήμμα:

Λήμμα 4.12. *Η τάξη t της ομάδας G είναι μεγαλύτερη από $4\log^2 n$.*

Για να ορίσουμε τη δεύτερη ομάδα χρειαζόμαστε θεωρία που αφορά τα κυκλοτομικά πολυώνυμα πάνω σε πεπερασμένα σώματα. Σαν μια βοηθητική εισαγωγή να υπενθυμίσουμε ότι οι n-οστές ρίζες της μονάδας αποτελούν μια κυκλική ομάδα τάξεως n. Αν λοιπόν z είναι μια n-οστή ρίζα της μονάδας έτσι ώστε η παραπάνω ομάδα να παράγεται από το z, το z θα ονομάζεται αρχική ρίζα της μονάδας. Προφανώς τότε το τυχαίο στοιχείο z^k της ομάδας θα παράγει και αυτό την ομάδα μόνο αν το k είναι σχετικά πρώτο με την τάξη της ομάδας. Τότε το στοιχείο z^k θα είναι και αυτό αρχική ρίζα της μονάδας. Επομένως υπάρχουν ακριβώς $\phi(n)$ αρχικές ρίζες της μονάδας, όπου ϕ η συνάρτηση του Euler.

Ορισμός 4.13. Αν $z_1, z_2, \ldots, z_{\phi(n)}$ είναι όλες οι διακεκριμένες αρχικές ρίζες της μονάδας, τότε το πολυώνυμο $Q_n(x) = (x - z_1)(x - z_2)\cdots(x - z_{\phi(n)})$ θα ονομάζεται n-οστό κυκλοτομικό πολυώνυμο.

Είναι σαφές τότε ότι το $Q_r(x)$ διαιρεί το πολυώνυμο $x^r - 1$, και μάλιστα επί του σώματος Z_p το αναλύει σε πρώτους παράγοντες τάξης $o_r(p)$. Ας είναι $h(x)$ ένας τέτοιος παράγοντας. Η δεύτερη ομάδα που θα μας χρειαστεί θα είναι τα πολυώνυμα του

$$\{\prod_{a=1}^{l}(x-a)^{e_a} | e_a \geq 0\} \bmod (h(x), p)$$

Θα συμβολίζουμε στο εξής με $\mathcal{G}$ αυτή την ομάδα. Στο σώμα $F = F_p[x]/h(x)$, η ομάδα $\mathcal{G}$ παράγεται από τα πολυώνυμα

$$x + 1, x + 2, \ldots, x + l$$

και είναι υποομάδα της πολλαπλασιαστικής ομάδας F. Το ενδιαφέρον είναι ότι η τάξη της ομάδας $\mathcal{G}$ φράσσεται κατάλληλα από κάτω όπως και από πάνω κάτω από ορισμένες συνθήκες, σε σχέση με την τάξη της G. Για την ακρίβεια ισχύει ότι :

Λήμμα 4.14. *Η τάξη της $\mathcal{G}$ είναι τουλάχιστον $\binom{t+l-2}{t-1}$.*

Λήμμα 4.15. *Αν το n δεν είναι δύναμη του p, τότε $|\mathcal{G}| < \frac{n^{2\sqrt{t}}}{2}$.*

Είμαστε έτοιμοι τώρα να δείξουμε πολύ εύκολα και την δεύτερη φορά του Θεωρήματος 4.9, που ολοκληρώνει την ορθότητα του αλγορίθμου.

Λήμμα 4.16. *Αν ο αλγόριθμος επιστρέψει **PRIME** τότε ο αριθμός n της εισόδου είναι πρώτος.*

Απόδειξη. Υποθέτουμε ότι ο αλγόριθμος επιστρέφει **PRIME**. Από το Λήμμα 4.14 έχουμε ότι αν $|G| = t$ και $l = \lfloor 2\sqrt{\phi(r)}\log n \rfloor$, τότε $|\mathcal{G}| \geq \binom{t+l-2}{t-1}$.
Έχουμε να παρατηρήσουμε ότι

1. $t > 2\sqrt{t}\log n$
2. $l = \lfloor 2\sqrt{\phi(r)}\log n \rfloor \geq \lfloor 2\sqrt{t}\log n \rfloor$ (αφού η G είναι υποομάδα της Z_r^*).
3. $2\sqrt{t}\log n \geq 3$
4. Από γνωστή πρόταση, για $x > 3$ ισχύει $\binom{2x-1}{x} > 2^x$.

□

Άρα

$$\begin{aligned} |\mathcal{G}| &\geq^1 \binom{l-1+\lfloor 2\sqrt{t}\log n \rfloor}{\lfloor 2\sqrt{t}\log n \rfloor} \\ &\geq^2 \binom{2\lfloor 2\sqrt{t}\log n \rfloor - 1}{\lfloor 2\sqrt{t}\log n \rfloor} \\ &\geq^3_4 2^{\lfloor 2\sqrt{t}\log n \rfloor} \geq \frac{n^{2\sqrt{t}}}{2}. \end{aligned}$$

Από το Λήμμα 4.15 και με την προϋπόθεσή ότι ο n δεν είναι δύναμη του p ισχύει ότι $|\mathcal{G}| < \frac{n^{2\sqrt{t}}}{2}$. Επομένως ο n είναι δύναμη του p δηλαδή υπάρχει k με $p^k = n$. Τότε όμως θα πρέπει $k = 1$ γιατί αλλιώς το 1ο βήμα του αλγορίθμου θα επέστρεφε **COMPOSITE**. Ώστε $p = n$ δηλαδή ο n είναι πρώτος.

4.6 Αλγόριθμοι Παραγοντοποίησης

Εισαγωγή

Το πρόβλημα της παραγοντοποίησης ενός αριθμού έχει κρίσιμη θέση στη μοντέρνα κρυπτογραφία καθώς αρκετά κρυπτογραφικά πρωτόκολλα στηρίζουν την ασφάλεια τους στη δυσκολία επίλυσης αυτού του προβλήματος (με πιο γνωστό και διαδεδομένο το RSA). Η εύρεση των παραγόντων ενός αριθμού έχει απασχολήσει από πολύ παλιά σε αμιγώς αριθμοθεωρητικό περιβάλλον πολλούς ερευνητές όπως ο Fermat κ.α. Οι σημερινές εφαρμογές δίνουν μια νέα διάσταση και η μελέτη του προβλήματος είναι αναγκαία ώστε να καθορίζονται οι παράμετροι ασφαλείας των διάφορων πρωτοκόλλων. Σε αυτήν την ενότητα θα δούμε μερικά παραδείγματα αλγορίθμων για την επίλυση του προβλήματος αυτού καθώς και το θεωρητικό τους υπόβαθρο.

Το πρόβλημα της παραγοντοποίησης (το οποίο θα μπορούσε να εκφραστεί και με τον αν η συνάρτηση του πολλαπλασιασμού είναι μονόδρομη) είναι το εξής: Δοθέντος ενός αριθμού n που υπάρχει υπόνοια ότι δεν είναι πρώτος, δηλ. έχει αποτύχει σε κάποιο primality test, να βρεθεί ένας (μη τετριμμένος) παράγοντας του n.

Μια πρώτη προσέγγιση του προβλήματος είναι η μέθοδος trial division η οποία συνίσταται σε διαδοχικές δοκιμές αριθμών από το 2 έως το $\lceil \sqrt{n} \rceil$ μέχρις ότου βρεθεί κάποιος διαιρέτης του n. Η μέθοδος αυτή στηρίζεται στο ότι κάθε σύνθετος n πρέπει να έχει ένα διαιρέτη μικρότερο του $\sqrt{n}$. Η χρονική πολυπλοκότητα της μεθόδου αυτής είναι $O(\sqrt{n})$ και καθώς το μέτρο της χρονικής πολυπλοκότητας σε προβλήματα θεωρίας αριθμών είναι τα bits του n (η ισοδύναμα το $O(\log_2 n)$) είναι εκθετική. Στα παρακάτω θα δούμε κάποιες άλλες πιο αποδοτικές κλασικές μεθόδους (που παραμένουν φυσικά εκθετικού χρόνου).

Μέθοδος ρ

Μια μέθοδος που προτάθηκε από τον J.M. Pollard το 1975 είναι η μέθοδος ρ. Ήταν η πρώτη μέθοδος που πετύχαινε σημαντικά καλύτερα αποτελέσματα από την trial division.

Η μέθοδος στηρίζεται στην ύπαρξη τυχαίων μεταθέσεων στο $\mathbb{Z}_n$ και αν και δεν έχει αποδειχθεί αυτό δίνει καλά αποτελέσματα στην πράξη.

Η μέθοδος ρ: Πρώτα επιλέγουμε μια τυχαία μετάθεση f στο $\mathbb{Z}_n$ συνήθως ένα πολυώνυμο με βαθμό μεγαλύτερο του 1 όπως το $f(x) = x^2+1$ και μια αρχική τιμή x_0 συνήθως 1 ή 2. Υπολογίζουμε διαδοχικά όρους της ακολουθίας $x_{j+1} = f(x_j)$ έως ότου βρούμε ένα ζευγάρι τιμών j, k που να ικανοποιεί την εξής συνθήκη:

$$k > j \quad \wedge \quad x_j \neq x_k \pmod{n} \quad \wedge \quad (x_k - x_j, n) > 1$$

οπότε θα έχουμε βρει και ένα διαιρέτη του n.

Αυτό σημαίνει ότι για κάθε k θα πρέπει να ελέγχουμε όλα τα προηγούμενα j κάτι που αυξάνει σημαντικά τη χρονική πολυπλοκότητα του αλγόριθμου. Με χρήση του επόμενου λήμματος μπορούμε να φτιάξουμε ένα πιο αποτελεσματικό αλγόριθμο:

Λήμμα 4.17. *Αν το r είναι κάποιος διαιρέτης του n τότε αν $x_{k_0} = x_{j_0} \pmod{r}$ και $k = k_0 + m, j = j_0 + m$ έχουμε ότι $x_k = x_j \pmod{r}$.*

Απόδειξη. Η απόδειξη είναι άμεση από τον ορισμό της ακολουθίας x_j. □

Ο αλγόριθμος για τη μέθοδο ρ είναι ο εξής λοιπόν:

Για κάθε k με $2^h \leq k < 2^{h+1}$ που θέλουμε να εξετάσουμε αν ισχύει η συνθήκη 4.6 για κάποιο προηγούμενο j, εξετάζουμε μόνο την περίπτωση $j = 2^h - 1$ και αν δεν επαληθεύεται προχωράμε στο $k+1$. Από αυτά είναι σαφές ότι υπάρχει περίπτωση κάποιο ζευγάρι k_0, j_0 να ικανοποιεί τη συνθήκη 4.6 και να μην το εξετάσουμε, πάραυτα από το λήμμα 4.17 ξέρουμε ότι υπάρχουν και άλλα ζευγάρια που την ικανοποιούν. Πράγματι το ζευγάρι k, j με $j = 2^h - 1$ και $k = j + (k_0 - j_0)$ θα ικανοποιεί

τη σχέση 4.6 αφού ισχύει $k = k_0 + (j - j_0)$ και $j = j_0 + (j - j_0)$ (χρησιμοποιώντας το λήμμα 4.17). Το ζεύγος k, j σίγουρα θα εξετασθεί από τον αλγόριθμο αφού το j έχει την κατάλληλη μορφή και σχετικά με το πόσο θα "περιμένουμε" για να βρούμε αυτό το k μπορούμε να δούμε εύκολα ότι $k \leq 4k_0$. Μπορεί να αποδειχθεί ότι η πολυπλοκότητα αυτού του αλγορίθμου είναι $O(\sqrt[4]{n}\log_2^3(n))$.

Μέθοδος Βάσεων Παραγοντοποίησης

Τώρα θα εστιάσουμε το ενδιαφέρον σε μια μέθοδο που είναι πολύ πιο αποδοτική από τις παραπάνω και οι περισσότεροι από τους τελευταίους αλγόριθμους παραγοντοποίησης είναι παραλλαγές αυτής της μεθόδου.

Η μέθοδος αυτή στηρίζεται στο παρακάτω σκεπτικό: Αν υποθέσουμε ότι $n = ab$ τότε μπορούμε να βρούμε t, s τέτοια ώστε $n = t^2 - s^2$ θέτοντας $t = \frac{a+b}{2}, s = \frac{a-b}{2}$. Επίσης είναι προφανές ότι αν ξέρουμε t, s με $n = t^2 - s^2$ τότε μπορούμε να παραγοντοποιήσουμε το n.

Έτσι αν υποθέσουμε ότι τα a, b είναι σχετικά 'κοντά' μεταξύ τους, μπορούμε να παραγοντοποιήσουμε το n με τον εξής αλγόριθμο: Θέτουμε $t = \lceil \sqrt{n} \rceil$ και αυξάνουμε την τιμή του t κατά 1 έως ότου η τιμή $(t^2 - n)$ να είναι τέλειο τετράγωνο. Τότε έχουμε ότι $s^2 = t^2 - n$ δηλ. $n = t^2 - s^2$. Η χρονική πολυπλοκότητα του αλγόριθμου αυτού εξαρτάται από το πόσο 'κοντά' είναι οι δύο παράγοντες και είναι εύκολο να δει κανείς ότι η κλάση των αριθμών στην οποία αυτός ο αλγόριθμος είναι πολυωνυμικός είναι πολύ μικρή. Πάραυτα το παραπάνω είναι ένα κριτήριο για τι είδους πρώτους p, q πρέπει να επιλέξουμε για το n του συστήματος RSA. Επίσης σε μια ανάλογη ιδέα στηρίζεται και η μέθοδος που θα αναπτύξουμε παρακάτω.

Η μέθοδος αυτή αναζητά t, s που να ικανοποιούν την εξής συνθήκη:

$$t^2 = s^2 \pmod{n} \text{ και } t \neq \pm s \pmod{n}$$

Είναι φανερό ότι αν τα t, s ικανοποιούν την παραπάνω συνθήκη αυτό σημαίνει ότι $n \mid t^2 - s^2$ δηλ. $n \mid (t-s)(t+s)$ αλλά $n \not| (t-s)$ και $n \not| (t+s)$ έτσι τα $(t-s)$ και $(t+s)$ έχουν μη τετριμμένο κοινό παράγοντα με το n. Το ζητούμενο θα είναι η εύρεση δύο αριθμών a, b που να ικανοποιούν την ιδιότητα αυτή. Στα παρακάτω θα δώσουμε τους κατάλληλους ορισμούς με τους οποίους θα κατασκευάσουμε έναν αλγόριθμο που με μεγάλη πιθανότητα θα δίνει δύο τέτοιους αριθμούς.

Ορισμός 4.18. Βάση παραγοντοποίησης θα λέμε ένα σύνολο $\mathcal{B} = \{p_1, \ldots, p_k\}$ όπου τα p_i είναι πρώτοι αριθμοί με εξαίρεση το p_1 που μπορεί να έχει και την τιμή -1.

Ενας αριθμός b θα λέγεται $\mathcal{B}$-αριθμός για κάποιο n αν το τετράγωνό του $modulo\, n$ μπορεί να γραφεί σα γινόμενο στοιχείων της βάσης $\mathcal{B}$ δηλ.

$b^2 \ (\bar{\mathrm{mod}}\ n) = \prod_{i=1}^{k} p_i^{a_i}$ με $p_i \in \mathcal{B}$ [3].
$\mathcal{B}$-διάνυσμα ενός $\mathcal{B}$-αριθμού θα λέμε το διάνυσμα $\langle a_1 \bmod 2, a_2 \bmod 2, \ldots, a_k \bmod 2\rangle$. Οι πράξεις στα $\mathcal{B}$-διανύσματα θα γίνονται $modulo\ 2$.

'Ετσι αν βρούμε κάποιους $\mathcal{B}$-αριθμούς που το άθροισμα των $\mathcal{B}$-διανυσμάτων τους είναι το μηδενικό διάνυσμα τότε ξέρουμε ότι το γινόμενο των τετραγώνων τους είναι τέλειο τετράγωνο. Εστω ότι οι $b_1, b_2, \ldots, b_l$ είναι $\mathcal{B}$-αριθμοί δηλαδή αν για $j = 1, \ldots, l$ ισχύει ότι $b_j^2 \bar{\mathrm{mod}} n = \prod_{i=1}^{k} p_i^{a_{i,j}}$. Αν το άθροισμα των $\mathcal{B}$-διανυσμάτων τους είναι το μηδενικό έχουμε ότι

$$\prod_{j=1}^{l} (b_j^2 \bar{\mathrm{mod}} n) = \prod_{i=1}^{k} p_i^{\sum_{j=1}^{l} a_{i,j}}$$

και ότι το $\sum_{j=1}^{l} a_{i,j}$ είναι άρτιος αριθμός. Τώρα θέτουμε $b = \prod_{i=1}^{l} b_i \bar{\mathrm{mod}} n$ και $c = \prod_{i=1}^{k} (p_i^{\frac{1}{2}\sum_{j=1}^{l} a_{i,j}} \bar{\mathrm{mod}} n)$. Τότε έχουμε ότι $b^2 = c^2(n)$. Πρέπει να σημειωθεί ότι η κατασκευή των b, c είναι αρκετά διαφορετική κάτι το οποίο δείχνει ότι η πιθανότητα να ισχύει ότι $b \neq \pm c(n)$ είναι "μεγάλη". Στο παραπάνω επιχείρημα συνηγορεί το γεγονός ότι αν πάρουμε μια τυχαία ρίζα του $b^2 \bmod n$ θα έχουμε από την πρόταση 2.44 ότι με πιθανότητα ίση με $1/2$ θα βρούμε την b ή $-b$.

Ένα πρόβλημα που προκύπτει στην υλοποίηση των παραπάνω είναι η επιλογή του $\mathcal{B}$ και των $\mathcal{B}$-αριθμών. Η επιλογή για το $\mathcal{B}$ γίνεται συνήθως από μικρούς διαδοχικούς πρώτους (συνήθως με το -1) ενώ για τους $\mathcal{B}$-αριθμούς η επιλογή γίνεται είτε τυχαία είτε είναι της μορφής $\lceil\sqrt{kn}\rceil, \lceil\sqrt{kn}\rceil + 1$ για $k = 1, 2, \ldots$. Βέβαια υπάρχουν πολύ πιο μελετημένοι και πολύπλοκοι τρόποι για την επιλογή αυτή και σε τέτοιες βελτιώσεις βασίζονται πολλοί από τους τελευταίους αλγόριθμους παραγοντοποίησης τους οποίους θα αναφέρουμε επιγραμματικά στο τέλος της ενότητας.

Ένα άλλο πρόβλημα που προκύπτει είναι πόσους $\mathcal{B}$-αριθμούς θα χρειαστεί να διαλέξουμε ώστε να βρούμε ένα υποσύνολο τους ώστε τα $\mathcal{B}$-διανύσματα τους να έχουν άθροισμα το μηδενικό διάνυσμα. Η απάντηση αυτή προκύπτει εύκολα αν σκεφτούμε το σύνολο των $\mathcal{B}$-διανυσμάτων σαν διανυσματικό χώρο, οπότε το ερώτημα μεταφράζεται στο πόσα διανύσματα πρέπει να έχουμε ώστε να είναι βέβαιο ότι είναι γραμμικά εξαρτημένα. Προφανώς η απάντηση θα είναι $k + 1 =\mid \mathcal{B} \mid +1$.

Τώρα θα δώσουμε τα βήματα του αλγόριθμου βάσεων παραγοντοποίησης.

Input(n)
Choose y approx. $\lceil \frac{\log_2 n}{16} \rceil$ bits;

[3] Το $(\bar{\mathrm{mod}}\ n)$ ορίζεται σαν πράξη όπως το mod με τη διαφορά ότι για τα στοιχεία a, για τα οποία $a \bmod n > \frac{n-1}{2}$ δίνει σαν αποτέλεσμα το $-n + (a \bmod n)$.

$\mathcal{B} := \{-1\} \cup \{p \text{ prime less or equal to } y\}$;
i:=1;
repeat
  Choose b_i;
  if b_i is a $\mathcal{B}$-number **then** i:=i+1
until $i =\mid \mathcal{B} \mid +1$;
calculate all $\mathcal{B}$-vectors;
check for a set of b_i where $\mathcal{B}$-vectors have sum zero and
    calculate b, c ;
if $b = \pm c \pmod{n}$ **then repeat** the whole procedure **else**
    factor n.

Από την ανάλυση αυτού του αλγορίθμου την οποία δε θα τη δούμε εδώ προκύπτει ότι η πολυπλοκότητα του είναι $O(e^{C\sqrt{r \log r}})$ όπου r είναι τα bits του n, δηλ. $O(e^{C\sqrt{\log n \log\log n}})$ για κάποια σταθερά C.

Η μέθοδος Quadratic Sieve του C. Pomerance που ήταν η πιο δημοφιλής μέθοδος παραγοντοποίησης κατά τη δεκαετία του 80 βασίζεται στην παραπάνω μέθοδο με μια πιο "έξυπνη" επιλογή της βάσης $\mathcal{B}$ και των $\mathcal{B}$-αριθμών. Η πολυπλοκότητα αυτής της μεθόδου είναι αντίστοιχη της παραπάνω μόνο που η σταθερά C μπορεί να είναι πολύ κοντά στο 1. Για αρκετό καιρό ήταν κοινή πίστη των ερευνητών ότι το $O(e^{C\sqrt{\log n \log\log n}})$ ήταν ένα "φυσικό" κάτω φράγμα στην πολυπλοκότητα του προβλήματος της παραγοντοποίησης. Όμως το 1993, η μέθοδος Number Field Sieve του Lenstra άλλαξε ριζικά την κατάσταση καθώς η (αναμενόμενη) πολυπλοκότητα της είναι $O(e^{(\log n)^{\frac{1}{3}}(\log\log n)^{\frac{2}{3}}})$. Η μέθοδος αυτή θεωρείται (δικαιολογημένα) ότι είναι η πιο πολύπλοκη που έχει προταθεί ποτέ για το πρόβλημα της παραγοντοποίησης.

4.7 Αλγόριθμοι για Τετραγωνικά Υπόλοιπα

Πρόταση 4.19. *Για τα παρακάτω υπάρχουν 'αποδοτικοί' αλγόριθμοι:*

1. *Δίνεται m, n, να υπολογιστεί το $(\frac{m}{n})$.*
2. *Δίνεται p, να βρεθεί a με $(\frac{a}{p}) = 1$.*
3. *Δίνεται a, p με $(\frac{a}{p}) = 1$, να βρεθεί x με $x^2 = a \pmod{p}$.*
4. *Δίνεται a, p, q, να βρεθούν οι τετραγωνικές ρίζες του $a \bmod n$, όπου $n = pq$.*

Απόδειξη. 1. Μπορούμε είτε να χρησιμοποιήσουμε την πρόταση 2.48 αν το n είναι πρώτος, είτε πιο γενικά μπορεί να βρεθεί αλγόριθμος που χρησιμοποιεί το νόμο της τετραγωνικής αντιστροφής. Και στις δύο περιπτώσεις η πολυπλοκότητα είναι $O(\log_2(n)^3)$.

2. Έχουμε από την πρόταση 2.46 ότι η πιθανότητα ένας τυχαίος αριθμός στο $U(\mathbb{Z}_p)$ να είναι τετραγωνικό υπόλοιπο είναι 1/2. Έτσι μπορούμε να κατασκευάσουμε ένα πιθανοτικό αλγόριθμο που να δίνει ένα τετραγωνικό υπόλοιπο modulo p κάνοντας δοκιμές.

3. Υπάρχει αποδοτικός (πιθανοτικός) αλγόριθμος που σε πολυωνυμικό χρόνο απαντά σε αυτό το πρόβλημα [1].

4. μπορούμε να χρησιμοποιήσουμε τον προηγούμενο αλγόριθμο και να συνδυάσουμε τις λύσεις με το Κινέζικο Θεώρημα Υπολοίπων όπως στην πρόταση 2.44. Προσοχή, αν οι p, q δεν είναι γνωστοί τότε το πρόβλημα είναι δύσκολο.

□

4.8 Αλγόριθμοι Διακριτού Λογαρίθμου

Προτού ορίσουμε το πρόβλημα υπενθυμίζουμε ότι είναι εύκολο 4.1 για κάποιον να υπολογίσει το b^x για κάποιο μεγάλο x σε σχετικά μικρό χρόνο. Αν όμως μας δώσουν έναν αριθμό y, ο οποίος είναι της μορφής b^x (θεωρούμε το b γνωστό) είναι εύκολο να υπολογίσουμε το μοναδικό x τέτοιο ώστε $y = b^x$, δηλαδή το $x = \log_b y$; Η επίλυση της παραπάνω εξίσωσης στο $\mathbb{Z}_p$, για p πρώτο, είναι το *Πρόβλημα Διακριτού Λογαρίθμου* (Discrete Logarithm Problem (DLOG)) και η απάντηση στο παραπάνω ερώτημα είναι αρνητική με βάση όσα γνωρίζουμε μέχρι τώρα, αν επιλέξουμε κατάλληλα το σώμα στο οποίο εργαζόμαστε, δηλαδή αν το p έχει τουλάχιστον 150 ψηφία και το $p-1$ έχει τουλάχιστον έναν «μεγάλο» πρώτο παράγοντα.

Ορισμός 4.20. Έστω G μία πεπερασμένη κυκλική ομάδα τάξης n,α ένας γεννήτορας της G και $\beta \in G$. Ο *Διακριτός Λογάριθμος* του β στη βάση α, που συμβολίζεται $log_\alpha \beta$, είναι ο μοναδικός ακέραιος x, $0 \leq x \leq n-1$, τέτοιος ώστε $\beta = \alpha^x$.

Παράδειγμα 7. *Για $p = 97$, η $\mathbb{Z}_{97}^*$ (θεωρούμενη ως κυκλική ομάδα με πράξη τον πολλαπλασιασμό) είναι κυκλική ομάδα τάξης $n = 96$. Ένας γεννήτορας της $\mathbb{Z}_{97}^*$ είναι ο $\alpha = 5$. Αφού $5^{32} = 35$, έχουμε ότι $\log_5 35 = 32$ στο $\mathbb{Z}_{97}^*$.*

Ορισμός 4.21. Το *Πρόβλημα Διακριτού Λογαρίθμου* (DLOG) είναι το παρακάτω:
Δίνονται: ένας πρώτος αριθμός p, ένας γεννήτορας α του $\mathbb{Z}_p^*$ και ένα στοιχείο $\beta \in \mathbb{Z}_p^*$.
Ζητείται: Να βρεθεί ακέραιος x, $0 \leq x \leq p-2$, τέτοιος ώστε

$$\alpha^x = \beta \pmod{p} \tag{4.3}$$

Παρατήρηση 13. Καταχρηστικά, θα χρησιμοποιούμε τον όρο DLOG και για το γενικό πρόβλημα Διακριτού Λογαρίθμου σε πεπερασμένη κυκλική ομάδα $(G, \cdot)$, όπου δίνεται ένας γεννήτορας α της G και ένα στοιχείο $\beta \in G$, και ζητείται $x < ord(G)$ ώστε $\alpha^x = \beta$.

Πρόταση 4.22. *Η δυσκολία του DLOG είναι ανεξάρτητη από την επιλογή του γεννήτορα α του $\mathbb{Z}_p^*$.*

Απόδειξη. Έστω α και γ δύο γεννήτορες του $\mathbb{Z}_p^*$, και $\beta \in \mathbb{Z}_p^*$.
Έστω $x = log_\alpha \beta, y = log_\gamma \beta$ και $z = log_\alpha \gamma$. Τότε $\alpha^x = \beta = \gamma^y = (\alpha^z)^y$, δηλαδή $x = zy \pmod{(p-1)}$. Αλλά τότε $y = xz^{-1} \pmod{(p-1)}$ ή $log_\gamma \beta = (log_\alpha \beta)(log_\alpha \gamma)^{-1} \pmod{(p-1)}$
Επομένως αν μπορούμε να υπολογίσουμε το *διακριτό λογάριθμο* σε μία βάση α τότε μπορούμε να τον υπολογίσουμε σε οποιαδήποτε βάση γ, όπου α,γ γεννήτορες του $\mathbb{Z}_p^*$. □

Στη συνέχεια θα αναφέρουμε κάποιους αλγορίθμους για την εύρεση του *Διακριτού Λογαρίθμου*. Για το σκοπό αυτό θεωρούμε στην υπόλοιπη παράγραφο ότι το p θα είναι πρώτος αριθμός και το α πρωταρχικό στοιχείο modulo p, δηλαδή $\alpha^{(p-1)} = 1 \pmod{p}$. (Υπενθυμίζουμε ότι αν δουλεύουμε στην κυκλική πολλαπλασιαστική ομάδα $\mathbb{Z}_p^*$ τότε το α θα είναι και ο γεννήτορας της.) Επίσης θεωρούμε τα p και α δοσμένα (σταθερά). Έτσι το DLOG γίνεται:

Δίνεται $\beta \in \mathbb{Z}_p^*$. Να βρεθεί x, $0 \leq x \leq p-2$, τέτοιος ώστε $\alpha^x = \beta \pmod{p}$.

Ένας προφανής αλγόριθμος για την επίλυση του παραπάνω προβλήματος είναι με την εξαντλητική μέθοδο αναζήτησης στο $\mathbb{Z}_p^*$. ενός x που να ικανοποιεί την εξίσωση (4.21). Η πολυπλοκότητά του θα είναι προφανώς $\widetilde{O}(p)$.

Αλγόριθμος Shanks (Baby Step - Giant Step)

Ένας άλλος αλγόριθμος για την επίλυση του DLOG, που ανήκει στην κατηγορία των αλγορίθμων ανταλλαγής χρόνου-μνήμης (time-memory trade-off algorithms), λειτουργεί ως εξής: υπολογίζει εκ των προτέρων όλα τα πιθανά α^x και ταξινομώντας τα ζεύγη (x, α^x) βάσει της δεύτερης συντεταγμένης (δηλαδή του α^x), επιλύει το DLOG σε χρόνο $\widetilde{O}(1)$ με $\widetilde{O}(p)$ χρόνο προεπεξεργασίας και χρησιμοποιώντας

$\widetilde{O}(p)$ μνήμη. Μια βελτίωση αυτού αποτελεί ο αλγόριθμος του Shanks που φαίνεται στο σχήμα 4.8. Η πολυπλοκότητα του αλγορίθμου του Shanks είναι $\widetilde{O}(\sqrt{p})$ σε χρόνο και $\widetilde{O}(\sqrt{p})$ σε χώρο.

Αλγόριθμος Shanks

1. $m := \lceil \sqrt{p-1} \rceil$
2. Υπολόγισε το $\alpha^{mj} \bmod p, \forall j : 0 \leq j \leq m-1$
3. Ταξινόμησε τα m διατεταγμένα ζεύγη $(j, \alpha^{mj} \bmod p)$ βάσει της δεύτερης συντεταγμένης (δηλαδή του $\alpha^{mj} \bmod p$), ώστε να προκύψει μια ταξινομημένη λίστα L_1
4. Υπολόγισε το $\beta\alpha^{-i} \bmod p, \forall i : 0 \leq i \leq m-1$
5. Ταξινόμησε τα m διατεταγμένα ζεύγη $(i, \beta\alpha^{-i} \bmod p)$ βάσει της δεύτερης συντεταγμένης (δηλαδή του $\beta\alpha^{-i} \bmod p$), ώστε να προκύψει μια ταξινομημένη λίστα L_2
6. Αναζήτησε ζεύγος $(j, y) \in L_1$ τέτοιο ώστε $(i, y) \in L_2$, δηλαδή δύο ζεύγη που να έχουν την ίδια τεταγμένη)
7. $(\log_\alpha \beta =) x := mj + i \bmod (p-1)$

Σχήμα 4.3: Ο Αλγόριθμος του Shanks

Πρόταση 4.23. *Ο αλγόριθμος του Shanks διαθέτει την ιδιότητα της ορθότητας, δηλαδή ισχύει ότι*

$$\log_\alpha \beta : x = mj + i \bmod (p-1)$$

όπου j, i αυτά που βρίσκει ο αλγόριθμος.

Απόδειξη. Πράγματι αν $(j, y) \in L_1$ και $(i, y) \in L_2$ τότε $\alpha^{mj} = y = \beta\alpha^{-i} \pmod p$ επομένως $\alpha^{mj+i} = \beta \Rightarrow \alpha^x = \beta \Rightarrow x = \log_\alpha \beta$ όπου όλες οι ισοτιμίες θεωρούνται modulo p. □

Παράδειγμα 8. Έστω ότι $p = 809$, και θέλουμε να υπολογίσουμε τον $\log_3 525$ ($\alpha = 3, \beta = 525$). Το $m := \lceil \sqrt{808} \rceil = 29$

Πρώτα υπολογίζουμε τα διατεταγμένα ζεύγη $(j, 99^j \pmod{809})$, για $0 \leq j \leq 28$ δημιουργώντας έτσι τη λίστα:

(0,1)	(5,329)	(10,644)	(15,727)	(20,528)	(25,586)
(1,99)	(6,211)	(11,654)	(16,781)	(21,496)	(26,575)
(2,93)	(7,664)	(12,26)	(17,464)	(22,564)	(27,295)
(3,308)	(8,207)	(13,147)	(18,632)	(23,15)	(28,81)
(4,559)	(9,268)	(14,800)	(19,275)	(24,676)	

από την οποία με ταξινόμηση θα προκύψει η L_1.

Στη συνέχει δημιουργούμε μία λίστα από τα διατεταγμένα ζεύγη (i,$525 \times 3^{-i} \bmod 809$), $0 \leq i \leq 28$:

(0,525)	(5,132)	(10,440)	(15,388)	(20,754)	(25,356)
(1,175)	(6,44)	(11,686)	(16,399)	(21,521)	(26,658)
(2,328)	(7,554)	(12,768)	(17,133)	(22,713)	(27,489)
(3,379)	(8,724)	(13,256)	(18,314)	(23,777)	(28,163)
(4,396)	(9,511)	(14,355)	(19,644)	(24,259)	

από την οποία με ταξινόμηση θα προκύψει η L_2.

Αν τώρα διασχίσουμε ταυτόχρονα τις δύο λίστες, θα βρούμε τα στοιχεία $(10, 644) \in L_1$ και $(19, 644) \in L_2$. Έτσι υπολογίζουμε το

$x = \log_3 525 = 29 \times 10 + 19 = 309$ Πράγματι μπορούμε εύκολα να επαληθεύσουμε ότι $3^{309} = 525 \pmod{809}$

Αλγόριθμος Pohlig-Hellman

Οι Pohlig και Hellman παρατήρησαν ότι το DLOG σε μια *οποιαδήποτε* αντιμεταθετική ομάδα G είναι τόσο δύσκολο όσο το ίδιο πρόβλημα στην μεγαλύτερη υποομάδα της G με τάξη πρώτο αριθμό.

Συγκεκριμένα έστω ότι η τάξη της ομάδας n παραγοντοποιείται ως εξής:

$$n = \prod p_i^{e_i}$$

Αν μπορέσουμε να βρούμε τον διακριτό λογάριθμο x modulo $p_i^{e_i}$ τότε θα μπορέσουμε να συνδυάσουμε το αποτέλεσμα χρησιμοποιώντας το Κινέζικο Θεώρημα των Υπολοίπων για όλους τους παράγοντες ώστε να τον βρούμε modulo n.

Έστω G_{p^e} η κυκλική ομάδα με τάξη p^e, όπου p πρώτος. Για να βρούμε το x modulo p^e το γράφουμε ως εξής:

$$x = x_0 + x_1 p + \cdots + x_{e-1} p^{e-1}$$

Στόχος μας είναι να βρούμε τα $x_0, x_1, \cdots, x_{e-1}$. Αυτό μπορεί να γίνει σε βήματα. Έστω ότι το ξέρουμε μέχρι το p^{t-1} για κάποιο $t < e$. Δηλαδή:

$$x' = x_0 + x_1 p + \cdots + x_{t-1} p^{t-1}$$

Το επόμενο βήμα είναι ο υπολογισμός του x_t. Έχουμε:

$$x = x' + p^t x''$$

και:

$$\beta = \alpha^x = \alpha^{x'} \alpha^{p^t x''}$$

Θέτουμε: $\beta' = \beta \cdot \alpha^{-x}$ και $\alpha' = \alpha^{p^t}$ και έχουμε:

$$\beta' = \alpha'^{x''}$$

Το στοιχείο α' έχει τάξη p^{e-t}. Υψώνουμε την παραπάνω σχέση στην $s = p^{e-t-1}$ και έχουμε:

$$\beta'^s = \alpha'^{x''^s} \Rightarrow \beta'' = \alpha''^{x_t}$$

Το παραπάνω πρόβλημα είναι ένα πρόβλημα διακριτού λογαρίθμου στην G_p το οποίο μπορούμε να επιλύσουμε χρησιμοποιώντας έναν αλγόριθμο όπως ο 4.8.

Αλγόριθμος Index-Calculus

Στην ενότητα αυτή θα περιγράψουμε μία κατηγορία *υποεκθετικών* αλγορίθμων, οι οποίοι συνδέουν το πρόβλημα παραγοντοποίησης με το πρόβλημα του διακριτού λογαρίθμου σε πεπερασμένα σώματα.

Έστω ότι θέλουμε να λύσουμε το πρόβλημα του διακριτού λογαρίθμου στο $\mathbb{F}_p^*$, δηλαδή έχουμε $\alpha, \beta \in \mathbb{F}_p^*$ για τα οποία ισχύει: $\beta = \alpha^x$.

Διαλέγουμε μία βάση παραγοντοποίησης $\mathcal{B}$ στοιχείων, συνήθως μικρών πρώτων. Χρησιμοποιώντας μία από τις τεχνικές παραγοντοποίησης δημιουργούμε σχέσεις της μορφής:

$$\prod_{p_i \in \mathcal{B}} p_i^{e_i} = 1 \pmod{p}$$

οι οποίες δίνουν:

$$\sum_{p_i \in \mathcal{B}} e_i log_\alpha(p_i) = 0 \pmod{p-1}$$

Με τον τρόπο αυτόν μπορούμε να βρούμε τον διακριτό λογάριθμο σε σχέση με τη βάση παραγοντοποίησης:

$$x_i = log_\alpha(p_i)$$

Η παραπάνω τιμή λέγεται και δείκτης (index) του p_i σε σχέση με το α. Άρα τα βήματα της μεθόδου είναι:

- Παραγοντοποιούμε $\beta = \prod p_i \in \mathcal{B} p_i^{\beta_i} \pmod p$
- Υπολογίζουμε $T = \beta \cdot \prod p_i \in \mathcal{B} p_i^{f_i} \pmod p$
- Αν $T = \prod p_i \in \mathcal{B} p_i^{\alpha_i}$
- Τότε $T = \prod p_i \in \mathcal{B} p_i^{\alpha_i - f_i} \pmod p$
- Έτσι μπορούμε να υπολογίσουμε τον διακριτό λογάριθμο x ως εξής:

$$x = log_\alpha(\beta) = log_\alpha(\prod p_i \in \mathcal{B} p_i^{\beta_i}) = \sum_{p_i \in \mathcal{B}} \beta_i log_\alpha(p_i)$$
$$\pmod{p-1} = \sum_{p_i \in \mathcal{B}} \beta_i x_i \pmod{p-1}$$

Παρατήρηση 14 (Ασφάλεια των Bits των Διακριτών Λογαρίθμων:). Ας υποθέσουμε ότι υπάρχει ένα στιγμιότυπο $I = (p, \alpha, \beta)$ του προβλήματος διακριτού λογαρίθμου, όπου τα $\alpha, \beta \in \mathbb{Z}_p^*$, το p είναι πρώτος αριθμός και το α είναι ένα πρωταρχικό στοιχείο του $\mathbb{Z}_p^*$. Μπορεί να αποδειχθεί (Βλέπε [2]), ότι το πρόβλημα της εύρεσης του τελευταίου bit (LSB) του διακριτού λογαρίθμου $\log_\alpha \beta$ είναι εύκολο, δηλαδή μπορεί να λυθεί με αποδοτικό τρόπο.

Από την άλλη μεριά, η εύρεση του i-οστού bit από το τέλος του διακριτού λογαρίθμου $\log_\alpha \beta$ όπου $i \geq 2$, (δηλαδή κάθε άλλου bit εκτός από το τελευταίο) είναι τόσο δύσκολη όσο και η επίλυση του προβλήματος διακριτού λογαρίθμου (εύρεση του $\log_\alpha \beta$).

4.9 Ασκήσεις

1. Υπολογίστε το $3^{23} \bmod 13$ χρησιμοποιώντας τον αλγόριθμο επαναλαμβανόμενου τετραγωνισμού.
2. Υπολογίστε το $102^{7.200.000.023} \bmod 35$ με όσο το δυνατόν λιγότερες πράξεις.
3. Σχεδιάστε έναν αλγόριθμο, ο οποίος με τεχνική επαναλαμβανόμενου τετραγωνισμού να υπολογίζει τη δύναμη ενός αριθμού x διαβάζοντας τα bits του εκθέτη y από τα αριστερά προς τα δεξιά (σε αντίθεση με τον αλγόριθμο που βρίσκεται στις σημειώσεις).
4. Σχεδιάστε αλγόριθμο πολυωνυμικού χρόνου για τον υπολογισμό του $x^{y^z} \bmod p$, με είσοδο x, y, z, p, για p πρώτο αριθμό. Εξηγήστε την πολυπλοκότητά του.

5. Εφαρμόστε τη μέθοδο παραγοντοποίησης ρ για παραγοντοποίηση του αριθμού 77. Ποια είναι η πολυπλοκότητά της.

6. Βρείτε τους παράγοντες του αριθμού 143, εφαρμόζοντας ξεχωριστά την μέθοδο παραγοντοποίησης ρ και την μέθοδο των βάσεων παραγοντοποίησης.

7. Χρησιμοποιώντας τον αλγόριθμο βάσεων παραγοντοποίησης με σύνολο $B = \{-1, 2, 3, 5\}$ και b_i στο σύνολο $\{6, 8, 9, 12, 14, 15\}$ να βρεθεί ένας παράγοντας του αριθμού 33.

8. Περιγράψτε δύο αποδοτικούς τρόπους εύρεσης πολλαπλασιαστικού αντιστρόφου του $\alpha \bmod n$ όπου p, q γωστοί πρώτοι αριθμοί που δεν διαιρούν το α. Εφαρμόστε τους για να υπολογίσετε το $28^{-1} \bmod 51$ (δείξτε τις πράξεις αναλυτικά).

9. Θεωρήστε γνωστό το εξής λήμμα: "Αν $(\frac{k}{N}) = k^{\frac{N-1}{2}} \pmod N$ για κάθε $k \in UZ_N$, τότε ο N είναι πρώτος". Δείξτε ότι:

   - Αν $a \in UZ_N$, τέτοιο ώστε $(\frac{a}{N}) \neq a^{\frac{N-1}{2}} \pmod N$ και $b \in UZ_N$, τέτοιο ώστε $(\frac{b}{N}) = b^{\frac{N-1}{2}} \pmod N$, τότε $(\frac{ab}{N}) \neq (ab)^{\frac{N-1}{2}} \pmod N$.
   - Αν N δεν είναι πρώτος, τότε η πιθανότητα να διαλέξουμε κάποιο τυχαίο $k \in UZ_N$ το οποίο να περνάει το τεστ των Solovay-Strassen, είναι το πολύ 1/2. (Υπόδειξη: Θεωρήστε το σύνολο $B = \{b_1, \ldots, b_m\}$ των στοιχείων του UZ_N που περνάνε το τεστ. και βρείτε ένα άνω φράγμα για το μέγεθός του.)

10. Έστω μια πεπερασμένη υποομάδα $G =< g >$ της πολλαπλασιαστικής ομάδας Z_p^* (όπου p πρώτος) και ένας θετικός ακέραιος n έτσι ώστε $g^n \equiv 1 \pmod p$ και ο n είναι ο μικρότερος θετικός ακέραιος με αυτήν την ιδιότητα. Έστω h στοιχείο της Z_p^*. Υπάρχει ακέραιος x τέτοιος ώστε $g^x \equiv h \pmod p$;. Βρείτε ένα τρόπο να απαντηθεί αυτή η ερώτηση με αποδοτικό αλγόριθμο (πολυωνυμικού χρόνου ως προς $\log p$, ενδεχομένως πιθανοτικό).

11. Έστω $n = pq$ και έστω ότι γνωρίζετε έναν ακέραιο k που είναι πολλαπλάσιο του $(p-1)$ αλλά όχι του $(q-1)$. Βρείτε αποδοτικό αλγόριθμο παραγοντοποίησης του n.

12. Δίνονται p πρώτος και g ένας γεννήτορας της πολλαπλασιαστικής ομάδας Z_p^*. Αν μας δώσουν ένα στοιχείο h, την τάξη του d και ένα τυχαίο στοιχείο a, πως μπορούμε να διαπιστώσουμε (αποδοτικά) αν το a ανήκει στην υποομάδα που παράγει το h; Αιτιολογήστε.

13. Δίνονται p πρώτος και ένα στοιχείο h της πολλαπλασιαστικής ομάδας Z_p^*. Η τάξη του h είναι $r = 2^m 3^k$ (r γνωστό).

- Τι μορφής είναι ο πρώτος p; Βρείτε ένα παράδειγμα τέτοιας ομάδας.
- Δώστε αποδοτικό αλγόριθμο που να λύνει το πρόβλημα του διακριτού λογαρίθμου στην υποομάδα $\langle h \rangle$.

14. Δείξτε πώς μπορεί να επιλυθεί σε πολυωνυμικό χρόνο το πρόβλημα του Διακριτού Λογαρίθμου στην υποομάδα του $\mathbb{Z}_p^*$, με γεννήτορα g τάξης $k = 3^m$ (p, g, k γνωστά).

4.10 Ηλεκτρονικό Υλικό

- Διαδραστικές Παρουσιάσεις - Video
  - Έλεγχος Πρώτων
- Διαδραστικές Υλοποιήσεις
  - Διαδραστικός Υπολογιστής Αλγόριθμου Ευκλείδη και χρήση του για εύρεση αντιστρόφου
  - Παραγοντοποίηση με χρήση Javascript
  - Αλγόριθμοι Διακριτού Λογαρίθμου
- Κώδικας
  - Κώδικας C++ για έλεγχο πρώτων κατά AKS
  - Έλεγχος πρώτων Miller - Rabin σε διάφορες γλώσσες προγραμματισμού

Βιβλιογραφία

[1] Neal Koblitz. *A course in number theory and cryptography*, volume 114. Springer Science and Business Media, 1994.

[2] Douglas R Stinson. *Cryptography: theory and practice*. CRC press, 2005.

Κεφάλαιο 5

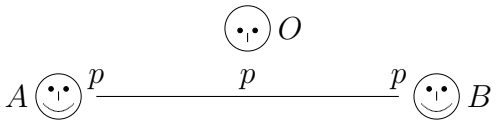
Συμμετρικά Κρυπτοσυστήματα

5.1 Εισαγωγή

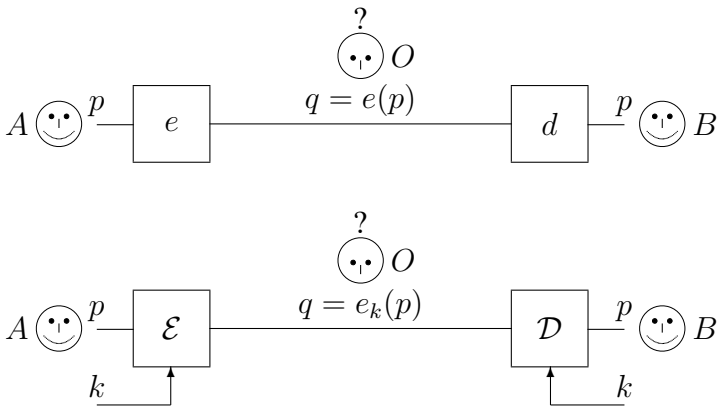
5.1.1 Το πρόβλημα

Όπως αναφέραμε στην εισαγωγή 1.1, ένα από τα προβλήματα που καλείται να λύσει η σύγχρονη κρυπτογραφία (και το οποίο είναι και το ιδρυτικό του κλάδου) είναι η διαφύλαξη της ασφάλειας της επικοινωνίας δύο μερών (Σχήμα 5.1): Ο πομπός A θέλει να μεταδώσει στο δέκτη B ένα μήνυμα p διαμέσου κάποιου καναλιού. Αλλά ο εχθρός O έχει και αυτός πρόσβαση στο κανάλι, κι έτσι μπορεί να υποκλέψει το p και να το επεξεργαστεί με οποιονδήποτε τρόπο επιθυμεί. Ψάχνουμε μεθόδους για να δυσχεράνουμε την προσπάθεια του εχθρού

Δεχόμαστε ότι ο εχθρός (είναι υγιής και) μπορεί να αντιληφθεί τη μετάδοση του μηνύματος, μπορεί να το υποκλέψει και μπορεί ακολούθως να το επεξεργαστεί με οποιονδήποτε τρόπο. Στο πλαίσιο αυτό, η καλύτερη ιδέα μοιάζει να είναι η εξής (Σχήμα 5.1.1): Ανάμεσα σ' αυτόν και στο κανάλι, καθένας από τους πομπό και δέκτη τοποθετεί ένα «κουτί». Το κουτί του πομπού υπολογίζει κάποιον αντιστρέψιμο μετασχηματισμό $e : \mathcal{P} \to \mathcal{Q}$ από το χώρο $\mathcal{P}$ όλων των δυνατών μηνυμάτων σε κάποιον νέο χώρο $\mathcal{Q}$ των μετασχηματισμένων μηνυμάτων, ενώ το κουτί του δέκτη,



Σχήμα 5.1: Το βασικό πρόβλημα: ο πομπός A στέλνει στο δέκτη B το μήνυμα p, αλλά ο εχθρός O μπορεί στην πορεία να το υποκλέψει.



που δέχεται την είσοδό του από το κανάλι, υπολογίζει τον αντίστροφο μετασχηματισμό $d = e^{-1}$ του e. Για να μεταδώσει το p, ο πομπός το αφήνει στην είσοδο του κουτιού του. Το μετασχηματισμένο μήνυμα $q = e(p)$ διασχίζει το κανάλι μέχρι την είσοδο του κουτιού του δέκτη, ο οποίος και παραλαμβάνει το αρχικό μήνυμα $d(q) = e^{-1}\big(e(p)\big) = p$. Στο ενδιάμεσο, ο εχθρός μπορεί να κλέψει το μετασχηματισμένο μήνυμα q, αλλά δεν μπορεί να υπολογίσει το αρχικό μήνυμα p, αν δεν ξέρει τον αντίστροφο μετασχηματισμό d.

Η βελτίωση στην παραπάνω διαδικασία που προτάθηκε από τον Kerckhoffs χρησιμοποιεί το κλειδί ώστε ο πομπός να υλοποιεί μια ολόκληρη οικογένεια μετασχηματισμών

$\mathcal{E} = \{e_k \mid k \in \mathtt{K}\}$, όπου $\mathtt{K}$ είναι ένα σύνολο δεικτών ή κλειδιών (keys) και, για κάθε $k \in \mathtt{K}$, $e_k : \mathcal{P} \to \mathcal{Q}$. Αντίστοιχα, το κουτί του δέκτη υλοποιεί την οικογένεια των αντίστροφων μετασχηματισμών $\mathcal{D} = \{d_k \mid k \in \mathtt{K}\}$, με $d_k = e_k^{-1} : \mathcal{Q} \to \mathcal{P}$. Έχοντας προσυμφωνήσει ένα κοινό κλειδί $k \in \mathtt{K}$, πομπός και δέκτης τροφοδοτούν τα κουτιά τους με αυτό, ώστε να επιλέξουν τους μετασχηματισμούς e_k και d_k αντίστοιχα από τις δύο οικογένειες $\mathcal{E}$ και $\mathcal{D}$. Το υπόλοιπο της επικοινωνίας γίνεται όπως και πριν.

Η γενική ιδέα της λύσης του Σχήματος 5.1.1 χρησιμοποιείται σήμερα ευρύτατα και αναφέρεται ως κρυπτογραφία ιδιωτικού κλειδιού (private key cryptography) ή κρυπτογραφία ενός κλειδιού (one-key cryptography) ή συμμετρική κρυπτογραφία (symmetric cryptography). Τα ονόματα προκύπτουν από την βασική απαίτηση ότι ένα κοινό κλειδί πρέπει να έχει προσυμφωνηθεί ανάμεσα στον πομπό και το δέκτη πριν αρχίσει η επικοινωνία τους διαμέσου του ανασφαλούς καναλιού. Αυτό δεν είναι πάντα απλό να γίνει και τότε αναδεικνύεται ως ένα πολύ σοβαρό μειονέκτημα της μεθόδου.

Πράγματι, για να συμφωνήσουν στο κοινό κλειδί, ο πομπός και ο δέκτης πρέπει με κάποιον τρόπο να επικοινωνήσουν πριν αρχίσει η κυρίως επικοινωνία τους, για την ασφάλεια της οποίας γίνεται όλη η συζήτηση. Οπωσδήποτε, για αυτήν

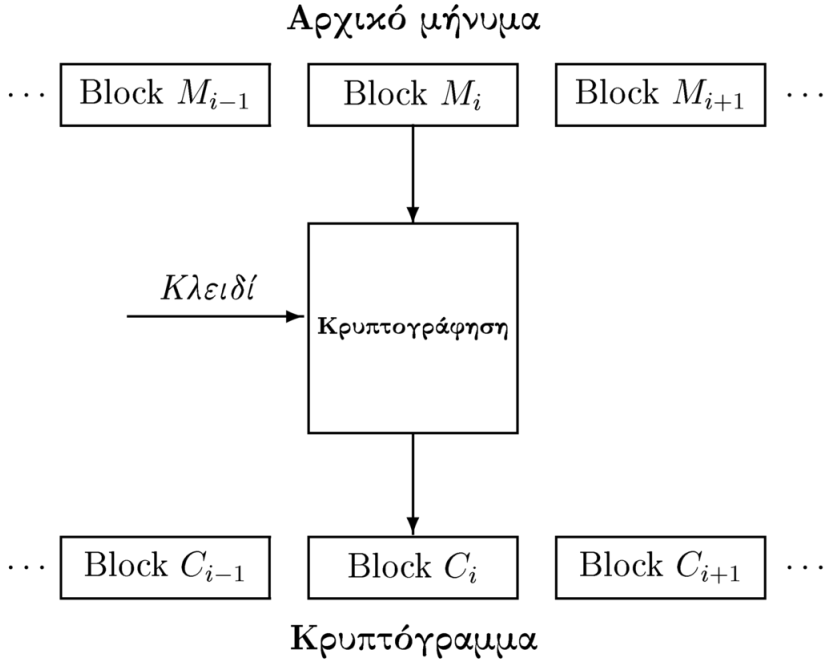
την προκαταρκτική επικοινωνία δεν μπορούν να χρησιμοποιήσουν το ανασφαλές κανάλι που πρόκειται να χρησιμοποιήσουν στην κυρίως επικοινωνία, γιατί έτσι ο εχθρός θα μάθει κι αυτός το κλειδί. Επομένως, θα πρέπει να χρησιμοποιήσουν ένα ασφαλές κανάλι, στο οποίο ο εχθρός να μην έχει πρόσβαση [1].

5.2 Κρυπτοσυστήματα τμήματος

Σε ένα κρυπτοσύστημα τμήματος (block cipher), το αρχικό μήνυμα M χωρίζεται σε διαδοχικά τμήματα $M_1, M_2, \ldots$ ίσου μεγέθους. Κάθε τμήμα M_i κρυπτογραφείται ξεχωριστά, δίνοντας ως αποτέλεσμα ένα τμήμα του κρυπτογράμματος C_i, δηλαδή $C_i = \texttt{Encrypt}_K(M_i)$ για $i = 1, 2, \ldots$. Η αποκρυπτογράφηση στον παραλήπτη γίνεται επίσης κατά τμήματα, δηλαδή $M_i = \texttt{Decrypt}_K(C_i)$ για κάθε i. Αυτή η διαδικασία κρυπτογράφησης απεικονίζεται στο Σχήμα 5.2. Μία τυπική τιμή για το μέγεθος του τμήματος σε αλγορίθμους αυτής της κατηγορίας είναι 128 bits. Στην ειδική περίπτωση που το μήκος ενός μηνύματος δεν είναι πολλαπλάσιο του μεγέθους του τμήματος, κατάλληλο πλήθος ψηφίων προστίθεται σ' αυτό σύμφωνα με κάποια προεπιλεγμένη σύμβαση.

Η λειτουργία αυτών των αλγορίθμων είναι επαναληπτική, υπό την έννοια ότι το αρχικό τμήμα μηνύματος M_i κρυπτογραφείται μέσα από διάφορα διαδοχικά *στάδια (rounds)*, όπου σε κάθε στάδιο συντελείται ακριβώς ο ίδιος κρυπτογραφικός μετασχηματισμός, προκειμένου να σχηματιστεί το τελικό τμήμα κρυπτογράμματος C_i. Για κάθε στάδιο, χρησιμοποιείται διαφορετικό τμήμα του κλειδιού. Η παραπάνω βασική λειτουργία των αλγορίθμων τμήματος, όπως απεικονίζεται στο Σχήμα 5.2, καλείται *Ηλεκτρονικό Βιβλίο Κωδικών* Electronic Code Book (ECB). Υπάρχουν και άλλοι τρόποι λειτουργίας για αυτούς τους αλγόριθμους - για παράδειγμα η *Αλυσιδωτή Κρυπτογράφηση Τμήματος Cipher Block Chaining Mode (CBC)*, όπου σε κάθε τμήμα M_i του μηνύματος, πριν κρυπτογραφηθεί, προστίθεται modulo 2 το τμήμα C_{i-1}.

Σε όλους τους αλγόριθμους τμήματος υπεισέρχεται στη λειτουργία τους μία δομική μονάδα που καλείται *μονάδα αντικατάστασης (Substitution Box ή S-Box)*, η οποία πραγματοποιεί αντικαταστάσεις bits με μη γραμμικό τρόπο. Αποτέλεσμα αυτής της λειτουργίας κατά την κρυπτογράφηση είναι το να υπάρχει εν τέλει μία σύνθετη σχέση μεταξύ των bits του κλειδιού και των bits του κρυπτογράμματος. Αυτή η ιδιότητα καλείται *σύγχυση (confusion)* και έχει οριστεί από τον Shannon στο [42] ως αναγκαία συνθήκη για τον χαρακτηρισμό ενός συστήματος ως ασφαλές. Κατά συνέπεια, οι ιδιότητες του S-Box είναι πολύ σημαντικές για την ασφάλεια του αλγορίθμου στο σύνολό του. Από μαθηματική άποψη, κάθε S-box με m εισόδους και n εξόδους μπορεί να θεωρηθεί ως συνάρτηση $f : \mathbb{F}_2^m \to \mathbb{F}_2^n$ ή, ισοδύναμα, ως συλλογή n λογικών συναρτήσεων $f_i : \mathbb{F}_2^m \to \mathbb{F}_2$, $i = 1, 2, \ldots, n$ (όπου



Σχήμα 5.2: Διάγραμμα βασικής λειτουργίας ενός αλγορίθμου τμήματος

$\mathbb{F}_2 = \{0, 1\}$).

Μία άλλη σημαντική ιδιότητα που όρισε ο Shannon ως απαραίτητη για ένα κρυπτογραφικό σύστημα είναι η *διάχυση (diffusion)*, η οποία υποδηλώνει ότι ένα bit του μηνύματος πρέπει να επηρεάζει πολλά bits του κρυπτογράμματος. Για την ικανοποίηση αυτής της ιδιότητας, οι σύγχρονοι αλγόριθμοι τμήματος περιέχουν δομικές μονάδες που επιτελούν αντιμεταθέσεις bits (*μονάδες αντιμετάθεσης (Permutation-Box ή P-Box)*) ή γραμμικούς μετασχηματισμούς. Πολλοί αλγόριθμοι τμήματος βασίζονται σε αλληλουχία ενεργειών αντικατάστασης και αντιμετάθεσης, βασισμένοι ακριβώς στις θεωρητικές αρχές του Shannon: αυτοί οι αλγόριθμοι καλούνται *δίκτυα αντικατάστασης/αντιμετάθεσης (Substitution - Permutation Networks (SPN))*.

Παράδειγμα 9. *Για παράδειγμα, έστω ότι το μέγεθος τμήματος είναι $n = 8$ και ότι το μήνυμα είναι η ακολουθία των $N = 37$ ψηφίων*

$$10000000 \quad 10011111 \quad 10011110 \quad 10100100 \quad 10011.$$

Για να κάνουμε το μήκος πολλαπλάσιο του n, επιθέτουμε στο μήνυμα 3 επιπλέον (μηδενικά) ψηφία, ώστε $N' = 40$, και χωρίζουμε το αποτέλεσμα σε πακέτα:

$$10000000 \quad 10011111 \quad 10011110 \quad 10100100 \quad 10011000 \cdots$$

Ακολούθως, κρυπτογραφούμε κάθε πακέτο χωριστά και ανεξάρτητα από τα υπόλοιπα, παράγοντας πέντε νέα πακέτα, π.χ. τα

$$01001100 \qquad 01101111 \qquad 01110110 \qquad 01100101 \qquad 01010101$$

ώστε το κρυπτογραφημένο μήνυμα είναι η ακολουθία

$$0100110001101111011101100110010101010101\cdots$$

Με βάση τα παραπάνω μπορούμε να εξειδικεύσουμε τον ορισμό του κρυπτοσυστήματος πακέτου 1.3 για τα κρυπτοσυστήματα τμήματος θέτοντας $\mathtt{K} = \mathbb{B}^m$ και $\mathtt{M} = \mathtt{C} = \mathbb{B}^n$ με $n > m$ και $\mathbb{B} = \{0, 1\}$. Επεται ότι για κάθε $k \in \mathtt{K}$ η συνάρτηση $\mathtt{Encrypt}_k$ είναι μια αντιστοιχία του $\mathtt{M}$ στο $\mathtt{C}$, δηλαδή μια μετάθεση του $\mathbb{B}^n$ και η $\mathtt{Decrypt}_k$ είναι ακριβώς η αντίστροφη μετάθεση. Από τα παραπάνω προκύπτουν εύκολα οι παρακάτω προτάσεις:

Πρόταση 5.1. *Υπάρχει μια αντιστοιχία ανάμεσα στο σύνολο όλων των κρυπτοσυστημάτων πακέτου με διαστάσεις $n \times m$ και στο σύνολο όλων των μήκους 2^m ακολουθιών μεταθέσεων του $\mathbb{B}^n$.*

Πόρισμα 5.2. *Το πλήθος των διαφορετικών κρυπτοσυστημάτων πακέτου με διαστάσεις $n \times m$ είναι*

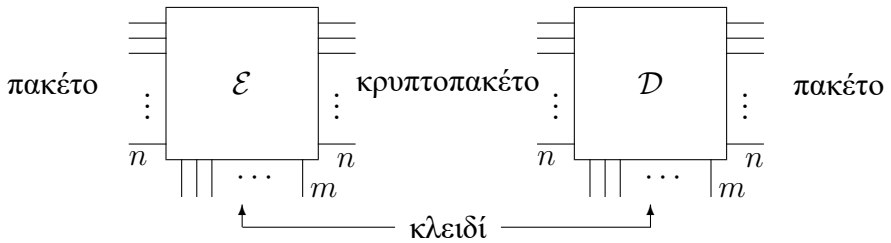
$$(2^n!)^{2^m}.$$

Κατασκευή

Παρόλο που η συνολοθεωρητική περιγραφή της προηγούμενης ενότητας μάλλον δεν το έχει κάνει σαφές, το να φτιάξει κανείς ένα κρυπτοσύστημα πακέτου με διαστάσεις $n \times m$ είναι κάτι μόνο λίγο περισσότερο από το να φτιάξει δύο «κουτιά», `Encrypt` και `Decrypt`, καθένα από τα οποία δέχεται στην είσοδό του $m + n$ δυαδικά ψηφία (m ψηφία του κλειδιού και n ψηφία ενός πακέτου) και παράγει στην έξοδο άλλα n ψηφία (ένα άλλο πακέτο), όπως στο Σχήμα 5.2. Το κάτι περισσότερο συνίσταται στο ότι η κατασκευή θα πρέπει να τηρεί τις παρακάτω τέσσερις προδιαγραφές, κατά γνησίως φθίνουσα σειρά σπουδαιότητας:

Αντιστρεψιμότητα Κάθε κρυπτογραφική πράξη που εκτελείται μέσα στο κουτί `Encrypt` πρέπει να είναι αντιστρέψιμη, ώστε το κουτί `Decrypt` να μπορεί τελικά, όταν τροφοδοτείται με το ίδιο κλειδί και το κρυπτοκείμενο, να ανακτά το αρχικό κείμενο εφαρμόζοντας τις αντίστροφες πράξεις. Διαφορετικά, τα δύο κουτιά δεν συνιστούν κρυπτοσύστημα.

Ασφάλεια Οι ακολουθίες των πράξεων που εφαρμόζονται μέσα στα κουτιά πρέπει να είναι τέτοιες ώστε να γίνεται όσο το δυνατόν δυσκολότερη η προσπάθεια του εχθρού να υπολογίσει το κλειδί που χρησιμοποιείται. Διαφορετικά, το κρυπτοσύστημα των δύο κουτιών δεν είναι ισχυρό.

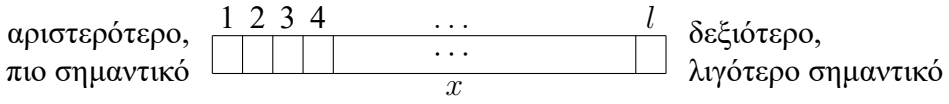


Ταχύτητα Οι ακολουθίες των πράξεων μέσα στα δύο κουτιά πρέπει να μπορούν να εκτελεστούν σε αρκετά μικρό χρόνο, ώστε να μπορούν να επιτευχθούν αρκετά υψηλοί ρυθμοί κρυπτογράφησης. Διαφορετικά, το κρυπτοσύστημα δεν είναι πρακτικό.

Απλότητα Οι ακολουθίες των πράξεων μέσα στα δύο κουτιά πρέπει να είναι συνολικά όσο το δυνατόν απλούστερες. Διαφορετικά, το κρυπτοσύστημα δεν είναι φθηνό.

Οι στοιχειώδεις πράξεις που χρησιμοποιούνται για την κατασκευή των κουτιών δεν είναι πάρα πολλές, και αυτό που διαφοροποιεί τα κρυπτοσυστήματα πακέτου μεταξύ τους δεν είναι τόσο το ποιες πράξεις χρησιμοποιούν όσο το πώς συνδυάζουν αυτές τις πράξεις μεταξύ τους. Στην απαρίθμηση που ακολουθεί, βολεύει να θεωρήσουμε το κλειδί σαν μια σταθερά, παράμετρο του κάθε κουτιού, και όχι σαν μια μεταβλητή στην είσοδό του. Μιλάμε λοιπόν για τις πράξεις που χρησιμοποιούνται στην κατασκευή των συναρτήσεων $\texttt{Encrypt}_k$ και $\texttt{Decrypt}_k$, για $k \in \texttt{K} = \mathbb{B}^m$.

Εξάλλου, για κάθε $l \geq 1$ και για κάθε ακολουθία ψηφίων $x \in \mathbb{B}^l$, θεωρούμε τα ψηφία της x αριθμημένα έτσι ώστε το αριστερότερο (πιο σημαντικό) ψηφίο να έχει



αύξοντα αριθμό 1 και το δεξιότερο (λιγότερο σημαντικό) να έχει αύξοντα αριθμό l. Παρόμοια, η αρίθμηση των γραμμών και των στηλών όλων των πινάκων και των διανυσμάτων ξεκινάει από το 1. Τέλος, όπου δεν δημιουργείται σύγχυση, δεν θα κάνουμε τη διάκριση ανάμεσα σε έναν ακέραιο και την ακολουθία δυαδικών ψηφίων που τον αναπαριστά στο δυαδικό σύστημα - π.χ., επιτρέπεται να μιλάμε για τη διαφορά $x - 3$, όταν $x \in \mathbb{B}^l$.

5.2.1 Βασικές Πράξεις

Στην ενότητα αυτή θα ασχοληθούμε με τις βασικές πράξεις που χρησιμοποιούνται στην κατασκευή συμμετρικών κρυπτοσυστημάτων.

Δυαδικές πράξεις (bitwise operators)

Οι πιο συνηθισμένες τέτοιες πράξεις είναι η (αποκλειστική) διάζευξη και οι περιστροφές.

Η διάζευξη είναι η γνωστή συνάρτηση δύο μεταβλητών

$$\oplus : \mathbb{B}^l \times \mathbb{B}^l \to \mathbb{B}^l,$$

για κάποιο $l \geq 1$, που σε κάθε δύο $x, y \in \mathbb{B}^l$ επιστρέφει ως $x \oplus y$ την ακολουθία που έχει ως i-οστό ψηφίο ($i = 1, \ldots, l$) το άθροισμα (modulo 2) των i-οστών ψηφίων των x και y. Είναι εύκολο να επαληθεύσει κανείς ότι με αυτήν την πράξη το $\mathbb{B}^l$ γίνεται αβελιανή ομάδα με ουδέτερο στοιχείο την ακολουθία 0^l των l μηδενικών και με αντίστροφο κάθε ακολουθίας την ίδια την ακολουθία. Αυτό πρακτικά σημαίνει ότι μπορούμε, π.χ., την εξίσωση

$$\big((x_3 \oplus x_4) \oplus (x_1 \oplus x_2)\big) \oplus x_5 = y_1 \oplus (y_2 \oplus y_3)$$

να την γράψουμε κατευθείαν και ισοδύναμα ως

$$x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 = y_1 \oplus y_2 \oplus y_3$$

(δηλαδή να διώξουμε όλες τις παρενθέσεις και να αλλάξουμε τη σειρά των μεταβλητών) αλλά και ως

$$x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus y_1 \oplus y_2 \oplus y_3 = 0^l$$

(δηλαδή να φέρουμε όλες τις μεταβλητές στο ένα μέλος και να αφήσουμε στο άλλο μέλος το 0^l).

Συχνά, η πράξη εμφανίζεται με το ένα από τα ορίσματά της σταθερό, δηλαδή ως η συνάρτηση μιας μεταβλητής $\oplus_u : \mathbb{B}^l \to \mathbb{B}^l, \quad$ με $\quad \oplus_u(x) = x \oplus u,$ για κάποιο $u \in \mathbb{B}^l$, και τότε βέβαια είναι μια αντιστοιχία με αντίστροφή της το εαυτό της. Επιπλέον, τις περισσότερες φορές η σταθερά u είναι συνάρτηση του κλειδιού k (συνηθέστερα, μια επιλογή ψηφίων του k). Τότε λέμε ότι η $\oplus_u$ είναι μια συνάρτηση λεύκανσης (whitening).[1]

[1] Η ονομασία προέρχεται από το ότι η διάζευξη με ψηφία από το κλειδί κάνει το όρισμα να φαίνεται περισσότερο τυχαίο, και άρα να μοιάζει περισσότερο με τον *λευκό* θόρυβο.

Οι περιστροφές είναι οι γνωστές πράξεις $<<<$: $\mathbb{B}^l \times \{0,\ldots,l-1\} \to \mathbb{B}^l$ και $>>>$: $\mathbb{B}^l \times \{0,\ldots,l-1\} \to \mathbb{B}^l$, για κάποιο $l \geq 1$, που για κάθε $x \in \mathbb{B}^l$ και κάθε $i = 0,\ldots,l-1$ επιστρέφουν ως $x <<< i$ και $x >>> i$ τις ακολουθίες που προκύπτουν από την κυκλική περιστροφή της x κατά i ψηφία προς τα αριστερά ή δεξιά, αντίστοιχα. Όταν το μήκος l των ακολουθιών δεν είναι σαφές, θα γράφουμε τις $<<<$, $>>>$ και ως $<<<_l$, $>>>_l$. Είναι προφανές ότι οι περιστροφές είναι αντιστοιχίες και μάλιστα $(<<<_l)^{-1} =>>>_l$.

Στις περισσότερες περιπτώσεις, οι περιστροφές χρησιμοποιούνται με το δεύτερο από τα ορίσματά τους σταθερό και μόνο πρόσφατα[2] αυτό το όρισμα άρχισε να εξαρτάται από τις εισόδους των συναρτήσεων $\texttt{Encrypt}_k$ και $\texttt{Decrypt}_k$. Πρόκειται για μια σημαντική καινοτομία, μια και, όπως θα δούμε σε επόμενες ενότητες, η χρήση της μπορεί να κάνει τους αλγόριθμους πολύ πιο απλούς, και μάλιστα χωρίς συνέπειες στην ασφάλεια και την ταχύτητα.[3] Τότε μιλάμε για εξαρτημένες περιστροφές (data dependent rotations) — σε αντιδιαστολή με τις σταθερές περιστροφές(fixed rotations) — και το δεύτερο όρισμα μπορεί να είναι οποιαδήποτε ακολουθία δυαδικών ψηφίων, αλλά μόνο τα $\lg l$ δεξιότερα (λιγότερο σημαντικά) ψηφία της (ως αναπαράσταση μιας τιμής στο $\{0,\ldots,l-1\}$) λαμβάνονται υπόψη.

Άλλες δυαδικές πράξεις (π.χ., η συμπλήρωση) δεν είναι δημοφιλείς. Βεβαίως, για τις περισσότερες από αυτές (απλή διάζευξη, σύζευξη, ολισθήσεις, κλπ.), ο λόγος που δεν χρησιμοποιούνται είναι το γεγονός ότι δεν είναι 1–1 συναρτήσεις, και έτσι εμποδίζουν την αντιστρεψιμότητα του αποτελέσματος.

Αριθμητικές πράξεις

Κυρίως χρησιμοποιείται η συνήθης πρόσθεση modulo 2^l

$$+ : \mathbb{B}^l \times \mathbb{B}^l \to \mathbb{B}^l,$$

για κάποιο $l \geq 1$. Όταν το ένα από τα δύο ορίσματά της είναι σταθερό, η πράξη είναι βεβαίως μια αντιστοιχία ως προς το μεταβλητό όρισμα και η αντίστροφή της είναι η αφαίρεση (modulo 2^l, πάλι) κατά το σταθερό όρισμα.

Συναρτήσεις απόσπασης

Μια ειδική περίπτωση επιλογής είναι η πράξη που από μια ακολουθία 32 ψηφίων ξεχωρίζει και επιστρέφει την τρίτη (από αριστερά προς τα δεξιά) τετράδα. Είναι

[2] Κατά τον Rivest, πρώτα στο αλγόριθμο του [25] και μετά στο RC5 [35], οπότε και στο RC6 [36]. Επίσης, στον MARS [5].

[3] Είναι σημαντικό ότι σε πολλούς νέους επεξεργαστές, η πράξη της περιστροφής έχει σταθερό κόστος ακόμη και όταν το δεύτερο όρισμά της είναι μεταβλητό (δηλαδή άγνωστο κατά τη μεταγλώττιση).

φανερό ότι πρόκειται για την επιλογή $sel_{[9,10,11,12]}$.

Στη γενική περίπτωση, έχουμε να κάνουμε με την πράξη που χωρίζει μια ακολουθία μήκους n σε n/m διαδοχικές υπακολουθίες μήκους m, για κάποιο m που διαιρεί το n, και επιστρέφει τη j-οστή από αυτές (από τα αριστερά προς τα δεξιά), για κάποιο $j \in \{1, 2, \ldots, \frac{n}{m}\}$. Τότε πρόκειται για την επιλογή sel_A, με

$$A = [(j-1)m + 1, (j-1)m + 2, \ldots, jm].$$

Συμβολίζουμε αυτή την πράξη με το σύμβολο $\cdot_{j,m}$, και λέμε ότι είναι μια *απόσπαση*. Ώστε, για το συγκεκριμένο διάνυσμα A, $\cdot_{j,m} : \mathbb{B}^n \to \mathbb{B}^m$, με $x_{j,m} = sel_A(x)$.

Ειδικές περιπτώσεις απόσπασης αποτελούν οι πράξεις που επιστρέφουν το αριστερό ή το δεξί μισό μιας ακολουθίας άρτιου μήκους. Τις συμβολίζουμε με $\cdot^L$ και $\cdot^R$ αντίστοιχα. Δηλαδή, για άρτιο n, $\cdot^L = \cdot_{1,n/2} : \mathbb{B}^n \to \mathbb{B}^{n/2}$ και $\cdot^R = \cdot_{2,n/2} : \mathbb{B}^n \to \mathbb{B}^{n/2}$.

Ανταλλαγή μισών

Μια ειδική περίπτωση μετάθεσης είναι η πράξη που, δεδομένης μια ακολουθίας x άρτιου μήκους, επιστρέφει την ακολουθία που προκύπτει όταν αλλάξουμε θέση στα δύο μισά (αριστερό και δεξί) της x. Συμβολίζουμε αυτην τη πράξη με το σύμβολο $\ominus$. Ώστε, για άρτιο n, $\ominus : \mathbb{B}^n \to \mathbb{B}^n$ με $\ominus x = sel_A(x)$, όπου $A = [\frac{n}{2} + 1, \ldots, n, 1, \ldots, \frac{n}{2}]$. Και προφανώς $\ominus^{-1} = \ominus$.

Παράθεση

Η παράθεση είναι η πράξη της συνένωσης δυο δυαδικών ακολουθιών σε μία καινούρια, δηλαδή η συνάρτηση $|| : \mathbb{B}^{l_1} \times \mathbb{B}^{l_2} \to \mathbb{B}^{l_1+l_2}$ με $x||y = xy$, για οποιαδήποτε $l_1, l_2 \geq 1$.

Αντικαταστάσεις

Αν $l_1, l_2, l_3 \geq 1$, τότε κάθε $2^{l_1} \times 2^{l_2}$ πίνακας A με στοιχεία από το $\{0, \ldots, 2^{l_3} - 1\}$ ορίζει τη συνάρτηση

$$sb_A : \mathbb{B}^{l_1+l_2} \to \mathbb{B}^{l_3}$$

που, για κάθε $x \in \mathbb{B}^{l_1}$ και $y \in \mathbb{B}^{l_2}$, επιστρέφει για την ακολουθία xy την ακολουθία

$$sb_A(xy) = A[x + 1, y + 1].$$

Π.χ., αν $l_1 = 1$, $l_2 = 2$, $l_3 = 3$ και $A = \begin{pmatrix} 001 & 011 & 111 & 101 \\ 101 & 010 & 000 & 110 \end{pmatrix}$, τότε $sb_A(110) = 000$. Προφανώς, η sb_A είναι αντιστρέψιμη μόνο αν $l_1 + l_2 = l_3$,

όμως αυτή του παραδείγματος δεν είναι. Τον πίνακα A τον ονομάζουμε και *πίνακα αντικατάστασης*(substitution box, s-box).

Οι πίνακες αντικατάστασης ενός κρυπτοσυστήματος μπορούν να είναι σταθεροί ή να εξαρτώνται από το κλειδί k (*εξαρτημένοι πίνακες αντικατάστασης*(key dependent s-boxes))[4] και συνιστούν ένα από τα πιο σημαντικά είδη πράξεων. Όταν είναι καλά σχεδιασμένοι, αποτελούν τη βασική άμυνα του κρυπτοσυστήματος απέναντι σε πολύ ισχυρές τακτικές κρυπτανάλυσης, όπως η διαφορική. Όμως κάποια νέα συστήματα τους έχουν ήδη εγκαταλείψει, στηρίζοντας πλέον την ασφάλειά τους στις εξαρτημένες περιστροφές που, παρότι πολύ απλούστερες, (μοιάζουν να) είναι τουλάχιστον το ίδιο ανθεκτικές απέναντι στην κρυπτανάλυση.

5.2.2 Δίκτυα Feistel (Feistel networks)

Τώρα που ξέρουμε τους στοιχειώδεις τελεστές που έχουμε στη διάθεσή μας, μπορούμε να δούμε πώς αυτοί συνδυάζονται για δώσουν πιο πολύπλοκους μετασχηματισμούς.

Μια από τις πιο παλιές τεχνικές για αυτή τη δουλειά είναι η τεχνική των Feistel δικτύων, που πήραν το όνομά τους από τον άνθρωπο που τα πρωτοπεριέγραψε (δημόσια, τουλάχιστον) [10]. Με κατάλληλη επιλογή των παραμέτρων τους μπορούν να εξασφαλίσουν πολύ μεγάλη ασφάλεια και ταχύτητα, ενώ παράλληλα καθιστούν το σύστημα εξαιρετικά απλό, υπό την έννοια ότι οι αλγόριθμοι κρυπτογράφησης και αποκρυπτογράφησης είναι σχεδόν όμοιοι. Κάθε τέτοιο δίκτυο έχει τρεις βασικές ακέραιες παραμέτρους, $n, l, r \geq 1$, ο ρόλος των οποίων θα εξηγηθεί στη συνέχεια. Η n πρέπει να είναι άρτια.

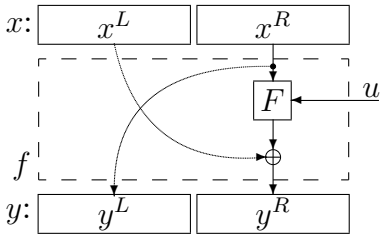
Το πρώτο πράγμα που πρέπει να έχει κανείς στη διάθεσή του για να κατασκευάσει ένα Feistel δίκτυο είναι μια συνάρτηση

$$F : \mathbb{B}^{n/2} \times \mathbb{B}^l \to \mathbb{B}^{n/2},$$

όπου n, l οι επιλεγμένοι παράμετροι για το δίκτυο. Πέρα από το να είναι γρήγορα υπολογίσιμη, η F δεν υποχρεούται να έχει καμιά άλλη ιδιότητα (π.χ., μονομορφικότητα, αντιστρεψιμότητα). Επειδή συχνά θα θεωρούμε για ευκολία το δεύτερο όρισμά της ως παράμετρο, εισάγουμε από τώρα το συμβολισμό F_u για τη συνάρτηση $x \mapsto F(x, u)$ και για οποιοδήποτε $u \in \mathbb{B}^l$.

Με την F διαθέσιμη, μπορούμε να κατασκευάσουμε το βασικό συστατικό του δικτύου: ένα *γύρο Feistel* Feistel round (Σχήμα 5.3). Ο γύρος δέχεται σαν είσοδο ένα πακέτο x και μια εξαρτώμενη από το κλειδί ακολουθία l ψηφίων, το *υποκλειδί*subkey u. Η έξοδός του είναι ένα νέο πακέτο y, που υπολογίζεται ως εξής: Το αριστερό του μισό y^L είναι το δεξί μισό x^R του πακέτου εισόδου. Και το δεξιό

[4] Για παράδειγμα, είναι σταθεροί στο DES, αλλά εξαρτημένοι στο Twofish.



Σχήμα 5.3: Ένας γύρος Feistel πάνω από τη συνάρτηση F.

του μισό y^R είναι η διάζευξη του αριστερού μισού x^L του πακέτου εισόδου με την τιμή που επιστρέφει η συνάρτηση F για το υποκλειδί u και το δεξί μισό x^R του x. Επομένως, ένας γύρος Feistel είναι η συνάρτηση

$$f : \mathbb{B}^n \times \mathbb{B}^l \to \mathbb{B}^n$$

που για κάθε $x \in \mathbb{B}^n$ και $u \in \mathbb{B}^l$ επιστρέφει την τιμή

$$f(x, u) = x^R || x^L \oplus F(x^R, u).$$

Και πάλι, θα συμφέρει να εμφανίζουμε το δεύτερο όρισμα της f ως παράμετρο, οπότε ονομάζουμε από τώρα f_u τη συνάρτηση $x \mapsto f(x, u)$, ώστε για κάθε $u \in \mathbb{B}^l$ είναι $f_u : \mathbb{B}^n \to \mathbb{B}^n$, με $f_u(x) = x^R || x^L \oplus F_u(x^R)$. Και είναι εύκολη η επόμενη παρατήρηση.

Πρόταση 5.3. *Για κάθε $u \in \mathbb{B}^l$, η f_u είναι μια μετάθεση του $\mathbb{B}^n$. Η αντίστροφη μετάθεση δίνεται από τον τύπο*

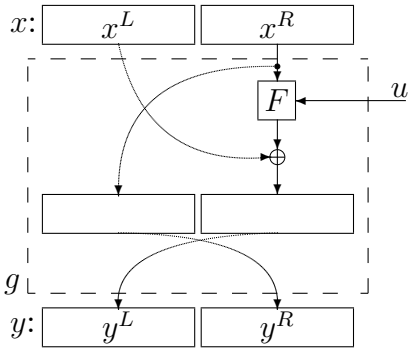
$$f_u^{-1}(y) = y^R \oplus F_u(y^L) || y^L. \tag{5.1}$$

Απόδειξη. Έστω ότι $x_1 \neq x_2$ και $y_1 = f_u(x_1)$, $y_2 = f_u(x_2)$. Αν $x_1^R \neq x_2^R$, τότε $y_1^L \neq y_2^L$ και επομένως $y_1 \neq y_2$. Αν αντίθετα $x_1^R = x_2^R$, τότε $F_u(x_1^R) = F_u(x_2^R)$ και $x_1^L \neq x_2^L$, άρα $x_1^L \oplus F_u(x_1^R) \neq x_2^L \oplus F_u(x_2^R)$, δηλαδή $y_1^R \neq y_2^R$, και επομένως ξανά $y_1 \neq y_2$. Άρα η f_u είναι μονομορφισμός.

Εξάλλου, για το τυχόν $y \in \mathbb{B}^n$, υπάρχει το πακέτο $x = y^R \oplus F_u(y^L) || y^L$ και είναι εύκολο να επιβεβαιώσουμε ότι $f_u(x) = y$. Οπότε η f_u είναι και επιμορφική.

Άρα η f_u είναι μια αντιστοιχία του $\mathbb{B}^n$ στο $\mathbb{B}^n$, δηλαδή μια μετάθεσή του. Και, από τον τρόπο που υπολογίσαμε το x της προηγούμενης παραγράφου, έπεται ότι η αντίστροφη μετάθεση έχει τον τύπο (5.3). □

Αλλά ο τύπος της f_u^{-1} μοιάζει αρκετά με τον τύπο της f_u. Αυτό δεν είναι τυχαίο και μπορεί να γίνει πιο συγκεκριμένο.



Σχήμα 5.4: Δίκτυο Feistel ενός γύρου.

Πρόταση 5.4. *Για κάθε $u \in \mathbb{B}^l$, ισχύει ότι*

$$f_u^{-1} \circ \ominus = \ominus \circ f_u.$$

Απόδειξη. Για κάθε $y \in \mathbb{B}^n$, υπολογίζουμε

$$\begin{aligned} & f_u^{-1}(\ominus y) = (\ominus y)^R \oplus F_u\big((\ominus y)^L\big)||(\ominus y)^L \\ & = y^L \oplus F_u(y^R)||y^R \\ & = \ominus\big(y^R||y^L \oplus F_u(y^R)\big) \\ & = \ominus\big(f_u(y)\big)\ , \end{aligned}$$

το οποίο επιβεβαιώνει τον ισχυρισμό.

□

Αυτό σημαίνει ότι, για κάθε $u \in \mathbb{B}^l$, η συνάρτηση $g_u = \ominus \circ f_u$, που περιγράφεται στο Σχήμα 5.4, είναι αντίστροφη του εαυτού της. Γιατί

$$g_u^{-1} = (\ominus \circ f_u)^{-1} = f_u^{-1} \circ \ominus^{-1} = f_u^{-1} \circ \ominus = \ominus \circ f_u = g_u.$$

Αν λοιπόν σε ένα κρυπτοσύστημα πακέτου με διαστάσεις $n \times l$ η συνάρτηση κρυπτογράφησης είναι η $\texttt{Encrypt}_k = g_k$, τότε η συνάρτηση αποκρυπτογράφησης είναι η $\texttt{Decrypt}_k = \texttt{Encrypt}_k^{-1} = \texttt{Encrypt}_k$ και πάλι. Δηλαδή το ίδιο πρόγραμμα (ή το ίδιο ολοκληρωμένο κύκλωμα) θα κάνει και την κρυπτογράφηση και την αποκρυπτογράφηση. Αυτό σίγουρα είναι ένα αποφασιστικό βήμα προς την απλότητα του κρυπτοσυστήματος συνολικά. Και είναι σημαντικό το ότι ισχύει, όποια κι είναι η συνάρτηση F που έχουμε διαλέξει αρχικά.

Όμως είναι εύκολο να διαπιστώσουμε ότι μια συνάρτηση σαν αυτή του Σχήματος 5.4 δεν μπορεί να προσφέρει την ασφάλεια που ζητάμε, αν η συνάρτηση F δεν γίνει εξαιρετικά πολύπλοκη. Ευτυχώς, υπάρχει τρόπος να διατηρήσουμε την

απλότητα χωρίς να χάσουμε σε ασφάλεια. Και αυτός βασίζεται σε μια γενίκευση της μέχρι τώρα ιδέας.

Πρόταση 5.5. *Για κάθε $r \geq 1$ και κάθε $u_1, \ldots, u_r \in \mathbb{B}^l$, ισχύει ότι*

$$f_{u_r}^{-1} \circ \cdots \circ f_{u_1}^{-1} \circ \ominus = \ominus \circ f_{u_r} \circ \cdots \circ f_{u_1}.$$

Απόδειξη. Επαγωγικά στο r. Αν $r = 1$, το αποτέλεσμα είναι η Πρόταση 5.4.
Αν $r > 1$, τότε

$$\begin{aligned} &f_{u_r}^{-1} \circ \cdots \circ f_{u_2}^{-1} \circ f_{u_1}^{-1} \circ \ominus = f_{u_r}^{-1} \circ \cdots \circ f_{u_2}^{-1} \circ \ominus \circ f_{u_1} \qquad \text{(από την Πρόταση 5.4)} \\ &= \ominus \circ f_{u_r} \circ \cdots \circ f_{u_2} \circ f_{u_1}, \text{ (επαγωγική υπόθεση)} \end{aligned}$$

κι έτσι το ζητούμενο ισχύει.

□

Επομένως, για κάθε $u_1, \ldots, u_r \in \mathbb{B}^l$, η συνάρτηση του Σχήματος 5.5

$$g_{u_1,\ldots,u_r} = \ominus \circ f_{u_r} \circ \cdots \circ f_{u_1},$$

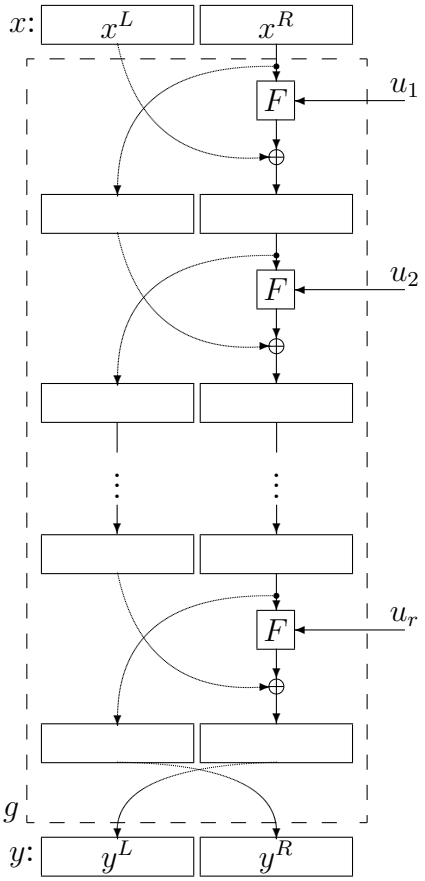
έχει την ιδιότητα ότι

$$\begin{aligned} g_{u_1,\ldots,u_r}^{-1} &= (\ominus \circ f_{u_r} \circ \cdots \circ f_{u_1})^{-1} \\ &= f_{u_1}^{-1} \circ \cdots \circ f_{u_r}^{-1} \circ \ominus^{-1} \\ &= f_{u_1}^{-1} \circ \cdots \circ f_{u_r}^{-1} \circ \ominus \\ &= \ominus \circ f_{u_1} \circ \cdots \circ f_{u_r} \\ &= g_{u_r,\ldots,u_1}. \end{aligned}$$

Δηλαδή, για να αντιστρέψουμε την $g_{u_1,\ldots,u_r}$ αρκεί να αντιστρέψουμε τη σειρά των παραμέτρων της. Δηλαδή τη σειρά με την οποία δίνουμε τις τιμές $u_1, \ldots, u_r$ στα r αντίγραφα της F μέσα στο υπολογιστικό διάγραμμα της συνάρτησης στο Σχήμα 5.5.

Αν λοιπόν σε ένα κρυπτοσύστημα πακέτου με διαστάσεις $n \times m$ η συνάρτηση κρυπτογράφησης είναι η $\texttt{Encrypt}_k = g_{u_1,\ldots,u_r}$, όπου τα $u_1, \ldots, u_r \in \mathbb{B}^l$ είναι συναρτήσεις του κλειδιού k, τότε η συνάρτηση αποκρυπτογράφησης είναι απλώς η $\texttt{Decrypt}_k = \texttt{Encrypt}_k^{-1} = g_{u_r,\ldots,u_1}$, και μπορεί να υπολογιστεί από το ίδιο πρόγραμμα (ή ολοκληρωμένο κύκλωμα) που υπολογίζει την $\texttt{Encrypt}_k$, με μόνη τροποποίηση την αντιστροφή της σειράς με την οποία τροφοδοτούνται οι παράμετροι στα F-κουτιά (Σχήμα 5.6).

Τώρα μπορούμε να κάνουμε πιο συγκεκριμένο το τι ακριβώς έχουμε ορίσει και το τι έχουμε αποδείξει για αυτό.



Σχήμα 5.5: Δίκτυο Feistel r γύρων.

Ορισμός 5.6. Θεωρούμε τυχόντες ακέραιους $n, l \geq 1$, με το n άρτιο, και τυχούσα συνάρτηση

$$F : \mathbb{B}^{n/2} \times \mathbb{B}^l \to \mathbb{B}^{n/2}.$$

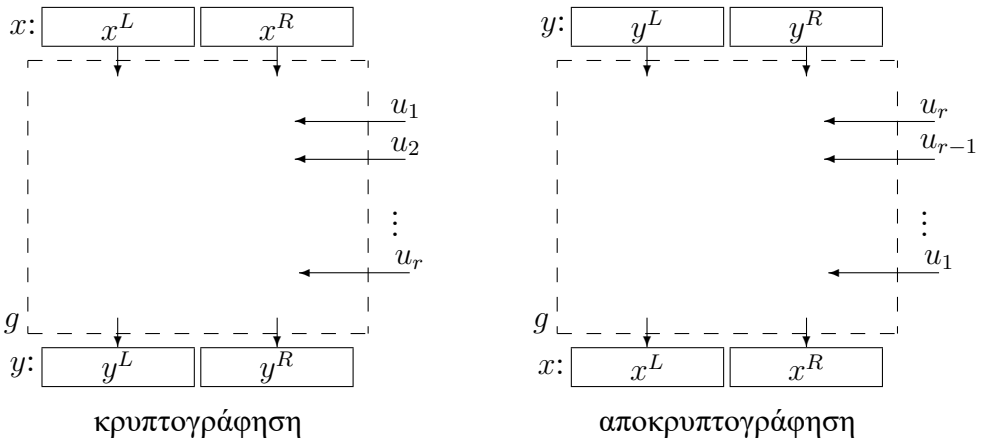
Η συνάρτηση $f : \mathbb{B}^n \times \mathbb{B}^l \to \mathbb{B}^n$, με $f(x, u) = x^R || x^L \oplus F(x^R, u)$ λέγεται *Feistel γύρος πάνω από την* F (Σχήμα 5.3). Επιπλέον, για κάθε $u \in \mathbb{B}^l$, λέμε ότι η συνάρτηση $f_u : \mathbb{B}^n \to \mathbb{B}^n$, με $f_u(x) = f(x, u)$ είναι η *Feistel μετάθεση πάνω από την* F *που ορίζει το* u.

Ορισμός 5.7. Θεωρούμε τυχόντες ακέραιους $n, l, r \geq 1$, με το n άρτιο, και τυχούσα συνάρτηση

$$F : \mathbb{B}^{n/2} \times \mathbb{B}^l \to \mathbb{B}^{n/2}.$$

Λέμε *Feistel δίκτυο* r *γύρων πάνω από την* F τη συνάρτηση

$$\mathcal{F} : \mathbb{B}^n \times (\mathbb{B}^l)^r \to \mathbb{B}^n$$



Σχήμα 5.6: Κρυπτογράφηση και αποκρυπτογράφηση με ένα δίκτυο Feistel r γύρων.

που για κάθε $x \in \mathbb{B}^n$ και $u_1, \ldots, u_r \in \mathbb{B}^l$ επιστρέφει την τιμή

$$\mathcal{F}(x, u_1, \ldots, u_r) = (\ominus \circ f_{u_r} \circ \cdots \circ f_{u_1})(x),$$

όπου f_{u_i} είναι η Feistel μετάθεση πάνω από την F που ορίζει το u_i, για κάθε $i = 1, \ldots, r$ (Σχήμα 5.5). Λέμε ότι το $\mathcal{F}$ έχει διαστάσεις $n \times l \times r$.

Επιπλέον, για κάθε $u_1, \ldots, u_r$ στο $\mathbb{B}^l$, λέμε ότι η συνάρτηση $\mathcal{F}_{u_1,\ldots,u_r} : \mathbb{B}^n \to \mathbb{B}^n$, με $\mathcal{F}_{u_1,\ldots,u_r}(x) = \mathcal{F}(x, u_1, \ldots, u_r)$ είναι η *Feistel πολυμετάθεση πάνω από την F που ορίζουν τα $u_1, \ldots, u_r$*.

Λήμμα 5.8. *Κάθε Feistel μετάθεση f_u του Ορισμού 5.6 είναι μια μετάθεση του $\mathbb{B}^n$, που ικανοποιεί την*

$$f_u^{-1} \circ \ominus = \ominus \circ f_u.$$

Και κάθε Feistel πολυμετάθεση $\mathcal{F}_{u_1,\ldots,u_r}$ του Ορισμού 5.7 είναι μια μετάθεση του $\mathbb{B}^n$, που αντιστρέφεται από την Feistel πολυμετάθεση με την αντίστροφη σειρά ορισμάτων,

$$(\mathcal{F}_{u_1,\ldots,u_r})^{-1} = \mathcal{F}_{u_r,\ldots,u_1}.$$

Απόδειξη. Η συζήτηση που προηγήθηκε. □

5.3 Data Encryption Standard (DES)

Όπως είδαμε στην Ενότητα 5.2, ένα κρυπτοσύστημα πακέτου μπορεί να έχει οποιεσδήποτε διαστάσεις. Επίσης, οι στοιχειώδεις πράξεις που μπορούν να χρησιμοποι-

ηθούν στην κατασκευή του είναι πάρα πολλές και οι στοιχειώδεις τρόποι για να τις οργανώσουμε είναι επίσης πολλοί. Είναι λοιπόν επόμενο καθένας που ξεκινά να κατασκευάσει ένα τέτοιο κρυπτοσύστημα να καταλήγει και σε ένα διαφορετικό, με πιθανότατα άλλες διαστάσεις από αυτά των άλλων κατασκευαστών, και σχεδόν πάντα με άλλα χαρακτηριστικά όσον αφορά την ασφάλεια, την ταχύτητα και την απλότητα.

Όμως, για να μπορέσουν να επικοινωνήσουν, δύο μέρη πρέπει να χρησιμοποιήσουν *το ίδιο* κρυπτοσύστημα. Και αυτό γίνεται πολύ δύσκολο όταν δεν υπάρχει ένα, μοναδικό και καθιερωμένο. Το ρόλο αυτό κλήθηκε να παίξει το πρότυπο που αναπτύχθηκε από την IBM και την NSA στη δεκαετία του '70 και τέθηκε σε ισχύ το 1977 από το αμερικανικό Υπουργείο Εμπορίου [8], με σκοπό την κρυπτογραφική προστασία *ευαίσθητων αλλά όχι απόρρητων* δεδομένων.

5.3.1 Περιγραφή

Το Data Encryption Standard (DES) είναι ένα κρυπτοσύστημα πακέτου με διαστάσεις 64×56. Βασικό δομικό στοιχείο των συναρτήσεων κρυπτογράφησης και αποκρυπτογράφησης είναι ένα Feistel δίκτυο $\mathcal{F}$, διαστάσεων $64 \times 48 \times 16$.

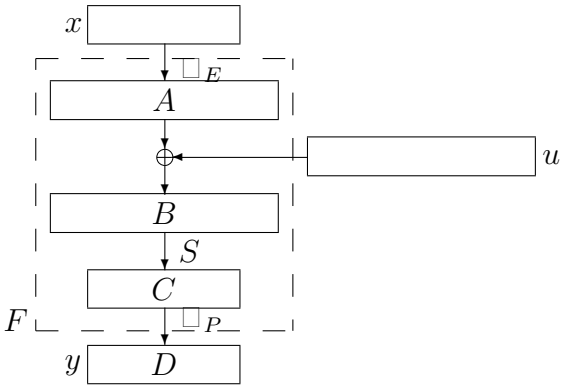
Η συνάρτηση F

Η συνάρτηση πάνω από την οποία ορίζεται το $\mathcal{F}$ είναι η $F : \mathbb{B}^{32} \times \mathbb{B}^{48} \to \mathbb{B}^{32}$, με $F(x,u) = exp_P\Big(S\big(exp_E(x) \oplus u\big)\Big)$, που φαίνεται και στο Σχήμα 5.7. Περιγράφουμε τις συναρτήσεις exp_P, exp_E και S αμέσως μετά - πρώτα σημειώνουμε την παραμετρική εκδοχή της F: $F_u : \mathbb{B}^{32} \to \mathbb{B}^{32}$, με $F_u(x) = exp_P\Big(S\big(\oplus u\big(exp_E(x)\big)\big)\Big)$,

για κάθε $u \in \mathbb{B}^{48}$.

P			
16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Πίνακας 5.1: Το διάνυσμα P. Το πάνω αριστερά στοιχείο είναιτο 1ο στοιχείο του διανύσματος και το κάτω δεξιά το 32ο.



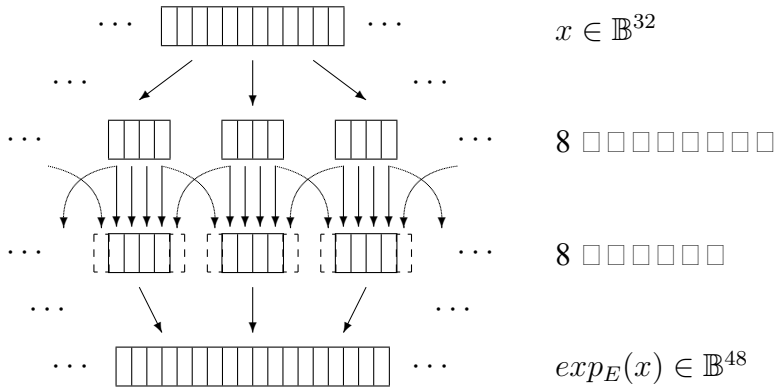
Σχήμα 5.7: Η συνάρτηση F πάνω από την οποία ορίζεται το Feistel δίκτυο $\mathcal{F}$ του DES.

Η συνάρτηση exp_P είναι η μετάθεση (των 32 ψηφίων) που ορίζεται από το διάνυσμα P του Πίνακα 5.1.

Η συνάρτηση exp_E είναι η επέκταση (των 32 ψηφίων σε 48) που ορίζεται από το διάνυσμα E του Πίνακα 5.2. Πρακτικά, το $exp_E(x)$ υπολογίζεται ως εξής (Σχήμα 5.8): Η 32-ψήφια ακολουθία x χωρίζεται σε 8 τετράδες. Καθεμιά από αυτές επεκτείνεται σε εξάδα, αντιγράφοντας απλώς το δεξιότερο ψηφίο της προηγούμενής της και το αριστερότερο ψηφίο της επόμενής της (ιδιαιτέρως, ως προηγούμενη της πρώτης τετράδας λογίζεται η τελευταία). Οι 8 εξάδες που προκύπτουν συνενώνονται με την ίδια σειρά για να δώσουν την επιστρεφόμενη 48-ψήφια ακολουθία.

E					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Πίνακας 5.2: Το διάνυσμα E. Το πάνω αριστερά στοιχείο είναι το 1ο στοιχείο του διανύσματος και το κάτω δεξιά το 48ο.



Σχήμα 5.8: Υπολογισμός του $exp_E(x)$.

Τέλος, η συνάρτηση $S : \mathbb{B}^{48} \to \mathbb{B}^{32}$ ορίζεται με βάση τους οχτώ πίνακες αντικατάστασης (s-boxes) $S_1, \ldots, S_8$ του Πίνακα 5.5 (σελίδα 140): Αν $x \in \mathbb{B}^{48}$, για να υπολογίσουμε την $S(x)$ πρώτα χωρίζουμε τη x σε 8 εξάδες ψηφίων. Για κάθε $j = 1, \ldots, 8$, η j-οστή εξάδα (από αριστερά), έστω $b_1b_2b_3b_4b_5b_6$, ορίζει το στοιχείο $S_j[b_1b_6 + 1, b_2b_3b_4b_5 + 1]$ του πίνακα S_j (ώστε τα δύο εξωτερικά ψηφία της εξάδας προσδιορίζουν μια γραμμή και τα τέσσερα εσωτερικά μια στήλη του S_j). Έτσι παράγονται 8 τετράδες ψηφίων, που συνενωμένες με την ίδια σειρά δίνουν την 32-ψήφια ακολουθία $S(x)$. Με την ορολογία της Ενότητας1.1, η συνάρτηση

S έχει τύπο

$$S(x) = ||_{j=1}^{8} sub_{S_j}\Big(exp_Z\big(x_{j,6}\big)\Big), \tag{5.2}$$

όπου sub_{S_j} είναι η συνάρτηση αντικατάστασης που ορίζει ο πίνακας S_j και exp_Z η μετάθεση που ορίζει το διάνυσμα $Z = [1, 6, 2, 3, 4, 5]$ (ώστε τα ψηφία κάθε εξάδας $x_{j,6}$ να μπαίνουν στη σωστή σειρά πριν ερμηνευθούν από την sub_{S_j}). Για ευκολία, θα συμβολίζουμε τη σύνθεση $sub_{S_j} \circ exp_Z$ με $sub_{S'_j}$, ώστε

$$S(x) = ||_{j=1}^{8} sub_{S'_j}(x_{j,6}).$$

Όταν δεν υπάρχει σύγχυση για το ποιες ακριβώς είναι οι είσοδοι x και u, θα συμβολίζουμε τα αποτελέσματα των ενδιάμεσων υπολογισμών όπως στο Σχήμα 5.7, δηλαδή

$$A = exp_E(x), \qquad B = \oplus u(A), \qquad C = S(B), \qquad D = P(C). \tag{5.3}$$

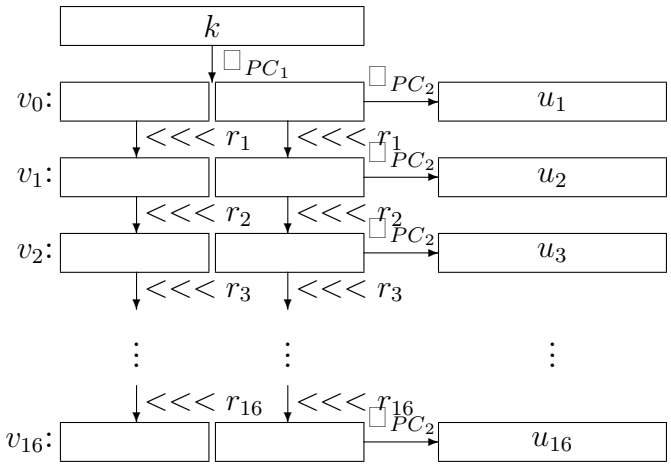
5.3.2 Υποκλειδιά

Λέμε ότι καθεμιά από τις 16 48-ψήφιες παράμετρους του Feistel δικτύου που περιγράψαμε αποτελεί το *υποκλειδί* (subkey) του αντίστοιχου γύρου του δικτύου. Και όλα τα υποκλειδιά παράγονται από το κλειδί $k \in \mathbb{B}^{56}$ σύμφωνα με τη διαδικασία που φαίνεται στο Σχήμα 5.9 και εξηγούμε αμέσως τώρα.[5]

[5] Το ακριβές είναι ότι το πρότυπο θεωρεί πως κάθε κλειδί έχει 64 ψηφία, με τα ψηφία 8, 16, 24, 32, 40, 48, 56 και 64 να είναι *ψηφία ισοτιμίας*, δηλαδή να έχουν απαραίτητα τέτοιες τιμές ώστε καθεμιά από τις ομάδες ψηφιών 1–8, 9–16, 17–24, 25–32, 33–40, 41–48, 49–56 και 57–64 να έχει άρτιο πλήθος μονάδων. Αυτό βεβαίως σημαίνει ότι τα συγκεκριμένα 8 ψηφία έχουν τιμές που καθορίζονται πλήρως από τα υπόλοιπα, κι έτσι δεν προσφέρουν στην ασφάλεια του κρυπτοσυστήματος.

Εδώ έχουμε αγνοήσει αυτή την τεχνική λεπτομέρεια και θεωρούμε από την αρχή ότι το κλειδί έχει μόνο τα υπόλοιπα $64 - 8 = 56$ ψηφία. Συνέπεια αυτού είναι και το γεγονός ότι τα διανύσματα PC_1 και PC_2 που φαίνονται στον Πίνακα 5.3 δεν είναι τα διανύσματα PC-1 και PC-2 που περιέχονται στο πρότυπο, αλλά κατάλληλες τροποποιήσεις τους. Γι' αυτό και η μικρή αλλαγή στο συμβολισμό.

[H]



Σχήμα 5.9: Η διαδικασία παραγωγής των 16 υποκλειδιών από το κλειδί.

Αρχικά, τα ψηφία του κλειδιού μετατίθενται κατά τη μετάθεση exp_{PC_1}, όπου PC_1 το διάνυσμα του Πίνακα 5.3. Το αποτέλεσμα χωρίζεται σε δύο κομμάτια, που υφίστανται ανεξάρτητα 16 διαδοχικές περιστροφές, κατά μία ή δύο θέσεις. Μετά από την i-οστή περιστροφή ($i = 1, \ldots, 16$), η σύμπτυξη exp_{PC_2} (όπου PC_2 το διάνυσμα του Πίνακα 5.3) διαλέγει 48 από τα 56 ψηφία των δύο κομματιών, παράγοντας έτσι το i-οστό υποκλειδί, u_i.

Πιο αναλυτικά, η διαδικασία υπολογίζει την ακολουθία $v = (v_i)_{0 \le i \le 16}$ που ορίζουν οι σχέσεις $\begin{array}{l} v_0 = exp_{PC_1}(k), \\ v_i = v_{i-1}^L <<< r_i || v_{i-1}^R <<< r_i, \end{array}$ όταν οι σταθερές $r_1, \ldots, r_{16}$ που αποφασίζουν το ποσό της κάθε περιστροφής δίνονται από τον πίνακα

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
r_i	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

.

Στη συνέχεια, για κάθε $i = 1, \ldots, 16$, το i-οστό υποκλειδί είναι το

$$u_i = exp_{PC_2}(v_i).$$

Η παραπάνω διαδικασία είναι χρήσιμη όταν θέλουμε πραγματικά να *υπολογίσουμε* τα υποκλειδιά, γι' αυτό και περιγράφεται από το πρότυπο. Αν όμως μας ενδιαφέρει (όπως εδώ) μόνο να τα *ορίσουμε*, τότε υπάρχει απλούστερος τρόπος: Παρακολουθώντας την πορεία κάθε ψηφίου του κλειδιού δια μέσου των διαδοχικών περιστροφών και επιλογών, μπορούμε να βρούμε ποιο ψηφίο του κλειδιού αντιγράφεται στο κάθε ψηφίο του κάθε υποκλειδιού. Έτσι, υπολογίζουμε τα διανύσματα $R_1, \ldots, R_{16}$ των Πινάκων 5.6 και 5.7 (σελίδες 141 και 142), για τα οποία

PC_1							PC_2					
50	43	36	29	22	15	8	14	17	11	24	1	5
1	51	44	37	30	23	16	3	28	15	6	21	10
9	2	52	45	38	31	24	23	19	12	4	26	8
17	10	3	53	46	39	32	16	7	27	20	13	2
56	49	42	35	28	21	14	41	52	31	37	47	55
7	55	48	41	34	27	20	30	40	51	45	33	48
13	6	54	47	40	33	26	44	49	39	56	34	53
19	12	5	25	18	11	4	46	42	50	36	29	32

Πίνακας 5.3: Τα διανύσματα PC_1 και PC_2.

ισχύει ότι

$$u_i = exp_{R_i}(k),$$

για κάθε $i = 1, \ldots, 16$. Στο εξής θα θυμόμαστε μόνο αυτό για την παραγωγή των υποκλειδιών.

5.3.3 Οι συναρτήσεις κρυπτογράφησης και αποκρυπτογράφησης

Έχοντας ήδη περιγράψει το Feistel δίκτυο $\mathcal{F}$ και τον τρόπο κατασκευής των υποκλειδιών $u_1, \ldots, u_{16}$, μπορούμε τώρα να δούμε τον ορισμό της συνάρτησης κρυπτογράφησης. Σ' αυτόν συμμετέχουν η μετάθεση exp_{IP} και η αντίστροφή της, $exp_{IP}^{-1} = exp_{IP^{-1}}$, όπου IP (initial permutation) και IP^{-1} τα διανύσματα του Πίνακα 5.4.

Για το τυχόν κλειδί $k \in \mathbb{B}^{56}$ λοιπόν, η κρυπτογράφηση γίνεται μέσω της συνάρτησης (Σχήμα 5.10)

$$e_k = exp_{IP^{-1}} \circ \mathcal{F}_{exp_{R_1}(k),\ldots,exp_{R_{16}}(k)} \circ exp_{IP}.$$

Δηλαδή, το τυχόν $p \in \mathbb{B}^{64}$ περνάει διαδοχικά από τις μεταθέσεις exp_{IP}, $\mathcal{F}_{exp_{R_1}(k),\ldots,exp_{R_{16}}(k)}$ και $exp_{IP^{-1}}$. Για το ίδιο υποκλειδί, η συνάρτηση αποκρυπτογράφησης πρέπει βεβαίως να είναι η

IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

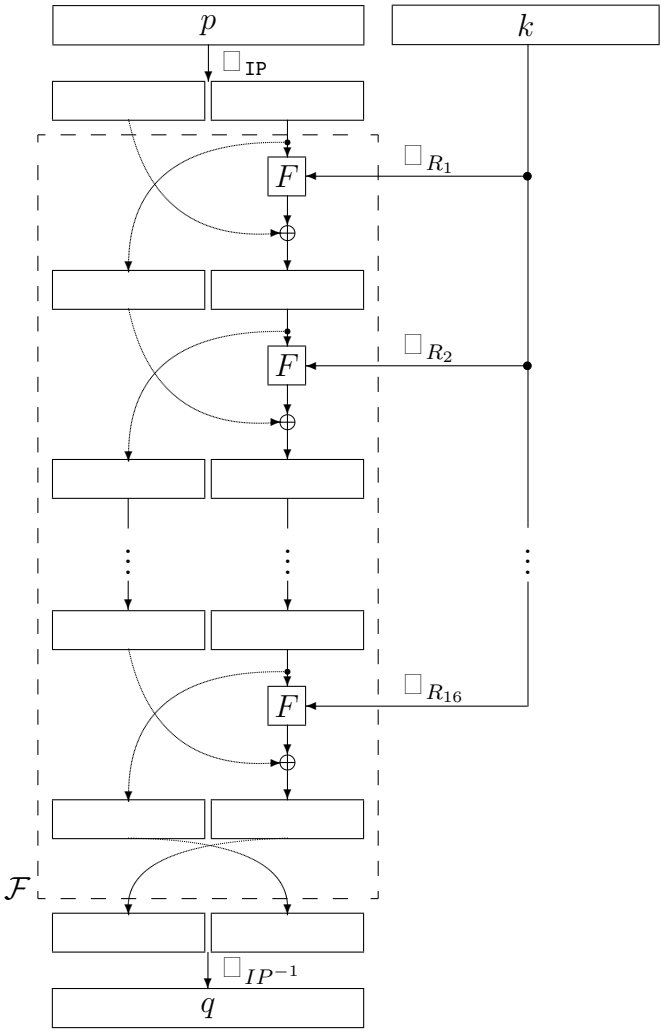
IP^{-1}

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Πίνακας 5.4: Τα διανύσματα IP και IP^{-1}.

$$
\begin{aligned}
&d_k = e_k^{-1} \\
&= (exp_{IP^{-1}} \circ \mathcal{F}_{exp_{R_1}(k),\ldots,exp_{R_{16}}(k)} \circ exp_{IP})^{-1} \\
&= exp_{IP}^{-1} \circ (\mathcal{F}_{exp_{R_1}(k),\ldots,exp_{R_{16}}(k)})^{-1} \circ exp_{IP^{-1}}^{-1} \quad \text{(Λήμμα1.1λη:Feistel-ιδιότητα)} \\
&= exp_{IP^{-1}} \circ \mathcal{F}_{exp_{R_{16}}(k),\ldots,exp_{R_1}(k)} \circ exp_{IP}.
\end{aligned}
$$

Οπότε, σύμφωνα προς τη φιλοσοφία των δικτύων Feistel, η αποκρυπτογράφηση γίνεται όπως και η κρυπτογράφηση (Σχήμα 5.10), με μόνη διαφορά την αντιστροφή της σειράς των υποκλειδιών.



Σχήμα 5.10: Η συνάρτηση κρυπτογράφησης του DES.

Τελικά, είμαστε σε θέση να ορίσουμε το DES.

Ορισμός 5.9. Το Data Encryption Standard είναι το κρυπτοσύστημα πακέτου $(\mathbb{B}^{64}, \mathbb{B}^{64}, \mathbb{B}^{56}, \mathcal{E}, \mathcal{D})$, όπου

$$\begin{aligned} \mathcal{E} &= \{e_k \mid k \in \mathbb{B}^{56}\}, \qquad \text{με} \quad e_k = exp_{IP^{-1}} \circ \mathcal{F}_{exp_{R_1}(k),\ldots,exp_{R_{16}}(k)} \circ exp_{IP}, \\ \mathcal{D} &= \{d_k \mid k \in \mathbb{B}^{56}\}, \qquad \text{με} \quad d_k = exp_{IP^{-1}} \circ \mathcal{F}_{exp_{R_{16}}(k),\ldots,exp_{R_1}(k)} \circ exp_{IP}, \end{aligned}$$

τα διανύσματα $IP, IP^{-1}, R_1, \ldots, R_{16}$ δίνονται στους Πίνακες 5.4, 5.6, 5.7 και $\mathcal{F}$ είναι το 16 γύρων Feistel δίκτυο πάνω από την συνάρτηση F της Ενότητας 5.3.1.

5.3.4 Τρόποι λειτουργίας

Απλή λειτουργία

Όπως εξηγήσαμε στην Ενότητα1.1packet, για να κρυπτογραφήσουμε μια δυαδική ακολουθία x μήκους N με ένα κρυπτοσύστημα πακέτου διαστάσεων $n \times m$, πρέπει πρώτα να την σπάσουμε σε $v = \frac{N}{n}$ πακέτα των n ψηφίων το καθένα,[6]

$$x = p_1 p_2 \cdots p_v,$$

και ακολούθως να κρυπτογραφήσουμε κάθε πακέτο χωριστά. Έτσι, το κρυπτογραφημένο μήνυμα είναι η ακολουθία κρυπτοπακέτων

$$y = q_1 q_2 \cdots q_v,$$

με

$$q_i = \texttt{Encrypt}_k(p_i) \text{ για κάθε } i = 1, \ldots, v,$$

και η αποκρυπτογράφησή του γίνεται με την προφανή χρήση της d_k,

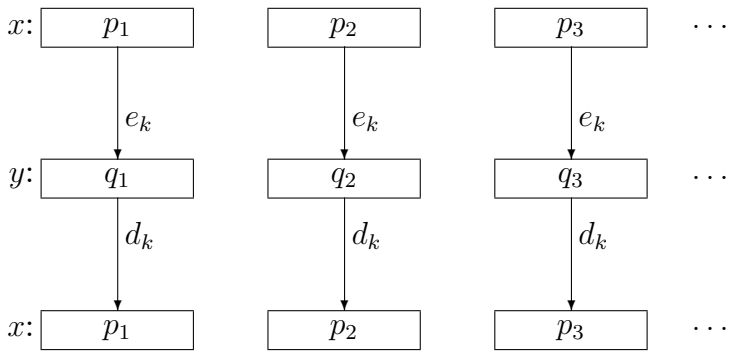
$$p_i = \texttt{Decrypt}_k(q_i), \qquad \text{για κάθε } i = 1, \ldots, v.$$

Τα ίδια ισχύουν και για το DES.

Αυτός όμως είναι μόνο ένας (ο απλούστερος και ταχύτερος) από τους τρόπους με τους οποίους μπορούμε να χρησιμοποιήσουμε ένα κρυπτοσύστημα πακέτου. Και ονομάζεται *λειτουργία ηλεκτρονικού βιβλίου κωδικών* ECB, ακριβώς γιατί χρησιμοποιεί το κρυπτοσύστημα ως ένα βιβλίο-«πίνακα αντιστοίχησης» των πακέτων σε κρυπτοπακέτα. Υπάρχουν και άλλοι τρόποι λειτουργίας, που έχουν ειδικά χαρακτηριστικά για να εξυπηρετούν συγκεκριμένες εφαρμογές. Τέσσερις από αυτούς

[6] Υποθέτουμε ότι το μήκος N είναι πολλαπλάσιο του n που δε βλάπτει τη γενικότητα, όπως έχουμε ήδη εξηγήσει.

έχουν τυποποιηθεί [33] για να χρησιμοποιούνται με το DES και ο πρώτος είναι αυτός που ήδη αναφέραμε (Σχήμα 5.11).



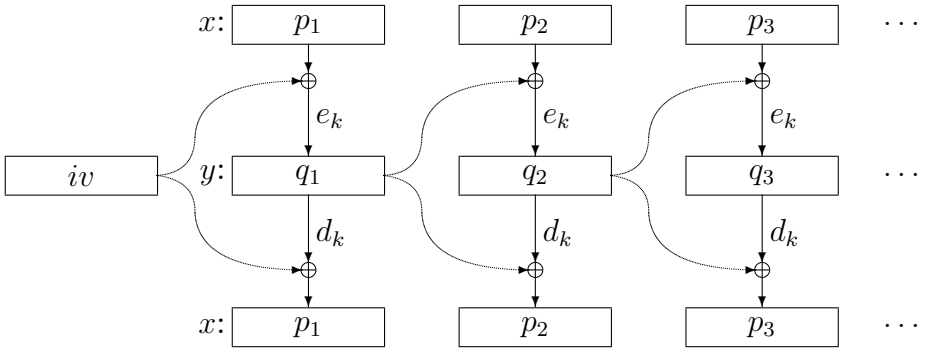
Σχήμα 5.11: Χρήση του DES ως ηλεκτρονικού βιβλίου κωδικών. Το ένα από τα δύο κομμάτια δείχνει την κρυπτογράφηση (πάνω) και το άλλο την αποκρυπτογράφηση (κάτω).

Ο δεύτερος τυποποιημένος τρόπος λειτουργίας του DES λέγεται *λειτουργία με Αλύσιδωτή Κρυπτογράφηση Τμημάτων πακέτων* Cipher Block Chain mode, CBC mode και φαίνεται στο Σχήμα 5.12. Πρόκειται για τροποποίηση της ECB λειτουργίας, ως προς το ότι, πριν το διαβάσει η συνάρτηση κρυπτογράφησης, το πακέτο υφίσταται μια αποκλειστική διάζευξη με το προηγούμενο κρυπτοπακέτο — για το πρώτο από τα πακέτα, p_0, η διάζευξη σχηματίζεται με κάποιο αυθαίρετα προεπιλεγμένο και κοινό για τον πομπό και το δέκτη πακέτο, που το ονομάζουμε iv. Πιο αναλυτικά, η ακολουθία των κρυπτοπακέτων είναι η

$$\begin{aligned} q_1 &= \texttt{Encrypt}_k(p_1 \oplus iv), \\ q_i &= \texttt{Encrypt}_k(p_i \oplus q_{i-1}), \qquad \text{για κάθε } i = 2, \ldots, v, \end{aligned}$$

και από αυτά παίρνουμε πάλι πίσω τα πακέτα ως εξής:

$$\begin{aligned} p_1 &= \texttt{Decrypt}_k(q_1) \oplus iv, \\ p_i &= \texttt{Decrypt}_k(q_i) \oplus q_{i-1}, \qquad \text{για κάθε } i = 2, \ldots, v. \end{aligned}$$



Σχήμα 5.12: Χρήση του DES με αλυσωτή σύνδεση των πακέτων. Το ένα από τα δύο κομμάτια δείχνει την κρυπτογράφηση (πάνω) και το άλλο την αποκρυπτογράφηση (κάτω).

Λειτουργία με ανάδραση

Οι άλλοι δύο τυποποιημένοι τρόποι λειτουργίας έχουν σχεδιαστεί με σκοπό την κρυπτογράφηση πακέτων με μήκος μικρότερο από $n = 64$. Η αφορμή προκύπτει από το γεγονός ότι πολλές φορές το μήνυμα που θέλουμε να κρυπτογραφήσουμε είναι ήδη μια σειρά από πεπερασμένες δυαδικές ακολουθίες σταθερού μήκους, έστω l, μόνο που αυτό το μήκος είναι λιγότερο από 64. Σε κάποιες από αυτές τις περιπτώσεις θέλουμε να μπορούμε να κρυπτογραφήσουμε με αυτές τις ακολουθίες ως στοιχειώδεις μονάδες κρυπτογράφησης και όχι με τις 64-άδες που επιβάλλει το DES. Το πιο χαρακτηριστικό τέτοιο παράδειγμα συμβαίνει όταν το μήνυμα είναι μια ακολουθία από ASCII χαρακτήρες, οπότε $l = 8$.

Για ένα μήνυμα λοιπόν που αποτελείται από $\lambda = \frac{N}{l}$ πακέτα μήκους $l \in \{1, \ldots, 64\}$ το καθένα,

$$x = p_1 p_2 \cdots p_\lambda,$$

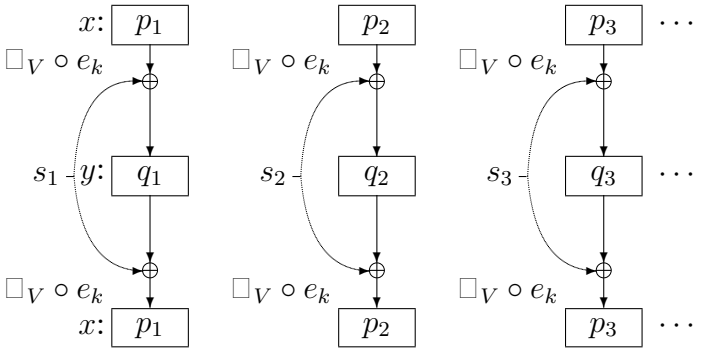
παραγάγουμε, χρησιμοποιώντας το DES, ένα ρεύμα $(s_i)_{1 \le i \le \lambda}$ από 64-ψήφια πακέτα και κρυπτογραφούμε το μήνυμα σύμφωνα με την εξίσωση

$$q_i = p_i \oplus exp_V\big(e_k(s_i)\big), \qquad \text{για κάθε } i = 1, \ldots, \lambda,$$

όπου $V = [1, 2, \ldots, l]$. Δηλαδή, κάθε φορά κρυπτογραφούμε με το DES το αντίστοιχο πακέτο του ρεύματος και με τα l πιο σημαντικά ψηφία του αποτελέσματος εφαρμόζουμε μια διάζευξη στο πακέτο του μηνύματος. Αντίστοιχα, για την αποκρυπτογράφηση ο δέκτης παράγει το ίδιο ρεύμα $(s_i)_{1 \le i \le}$ και ανακτά το μήνυμα εκτελώντας την ίδια απλή διάζευξη,

$$p_i = q_i \oplus exp_V\big(e_k(s_i)\big), \qquad \text{για κάθε } i = 1, \ldots, \lambda.$$

Το Σχήμα 5.13 εξηγεί καλύτερα τι συμβαίνει.



Σχήμα 5.13: Χρήση του DES με ανάδραση.

Μένει λοιπόν να καθορίσουμε πώς παράγεται το ρεύμα των 64-ψήφιων πακέτων. Θα δείξουμε δύο τρόπους γι' αυτό, κι έτσι θα έχουμε ορίσει τελικά (μαζί και με τα παραπάνω) τους δύο τυποποιημένους τρόπους λειτουργίας που απέμειναν. Σε κάθε περίπτωση το πρώτο πακέτο του ρεύματος είναι μια αυθαίρετα επιλεγμένη τιμή και καθένα από τα επόμενα πακέτα παράγεται από το αμέσως προηγούμενό του μέσω μιας DES κρυπτογράφησης. Αυτό στην πράξη υλοποιείται με μια ανάδραση από την έξοδο στην είσοδο του DES, γι' αυτό και οι αντίστοιχοι τρόποι λειτουργίας λέγονται *λειτουργίες με ανάδραση*feedback modes, FB modes.

Στη *λειτουργία με ανάδραση από την έξοδο* Output FeedBack Mode (OFB), το πρώτο πακέτο s_1 του ρεύματος προκύπτει από μια αυθαίρετα προεπιλεγμένη δυαδική ακολουθία iv μήκους $L \leq 64$, όταν τη συμπληρώσουμε (αν χρειάζεται) με $64 - L$ μηδενικά στις πιο σημαντικές θέσεις. Για να φτιάξουμε καθένα από τα επόμενα πακέτα, s_{i+1}, ολισθαίνουμε το προηγούμενό του, s_i, κατά l θέσεις προς τα αριστερά και τις l λιγότερες σημαντικές θέσεις του, που αδειάζουν, τις αναπληρώνουμε με τα l πιο σημαντικά ψηφία του κρυπτοπακέτου που προκύπτει από την κρυπτογράφηση του s_i. Δηλαδή,

$$s_1 = 00\cdots0||iv,$$
$$s_{i+1} = exp_W(s_i)||exp_V\big(e_k(s_i)\big), \qquad \text{για κάθε } i = 2,\ldots,\lambda,$$

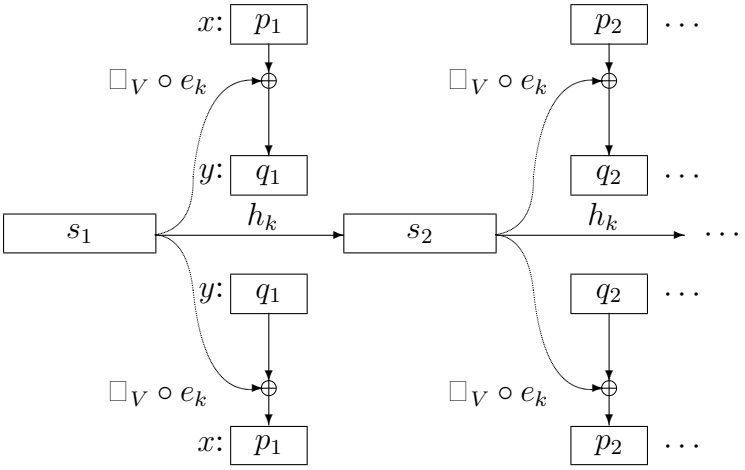
όπου $W = [l+1, l+2, \ldots, 64]$ το διάνυσμα για την επιλογή των $64-l$ λιγότερο σημαντικών ψηφίων.

Στη *λειτουργία με ανάδραση από τα κρυπτοκείμενα*Cipher BFeedBack Mode (CFB) το πρώτο πακέτο του ρεύματος κατασκευάζεται όπως και στην OFB λειτουργία. Η διαφορά βρίσκεται στην κατασκευή των υπόλοιπων πακέτων, στην οποία συμμετέχουν και τα παραγόμενα l-ψήφια κρυπτοκείμενα, σύμφωνα με τη δεύτερη από τις παρακάτω εξισώσεις:

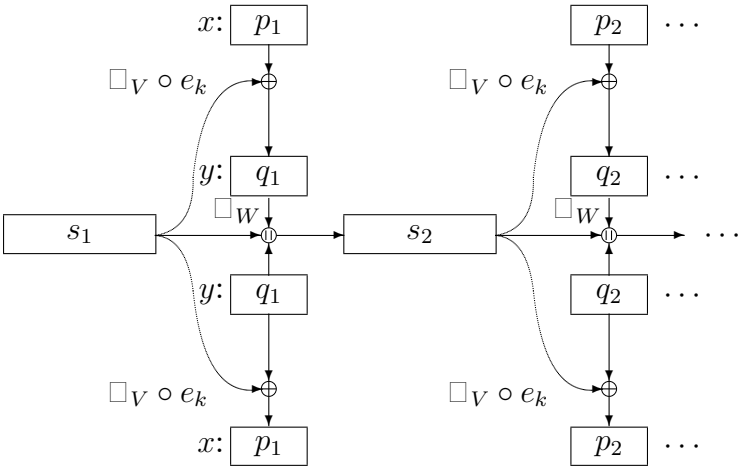
$$s_1 = 00\cdots0||iv,$$
$$s_{i+1} = exp_W(s_i)||q_i, \qquad \text{για κάθε } i = 2,\ldots,\lambda.$$

Δηλαδή, για να φτιάξουμε το s_{i+1}, εφαρμόζουμε στο s_i μια αριστερή ολίσθηση κατά l και μετά αναπληρώνουμε τις l λιγότερο σημαντικές θέσεις με το πιο πρόσφατα παραχθέν κρυπτοπακέτο.

Τα Σχήματα 5.14 και 5.15 εξηγούν καλύτερα τι συμβαίνει.

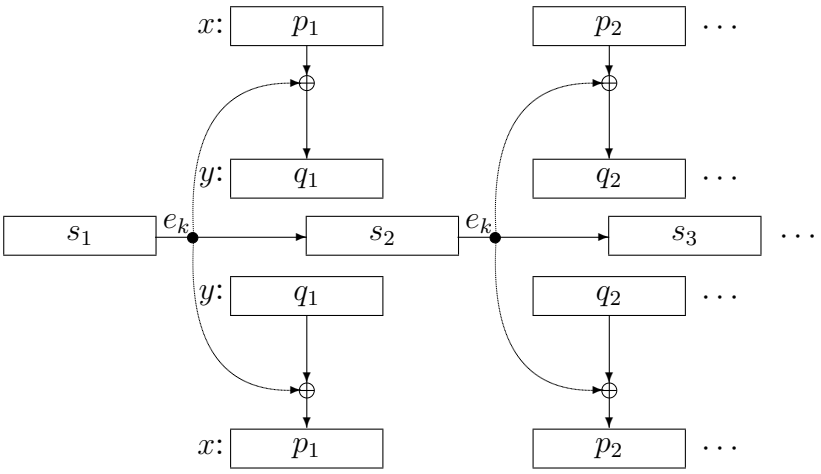


Σχήμα 5.14: Χρήση του DES με ανάδραση από την έξοδο. Όπου $h_k = || \circ (exp_W, exp_V \circ e_k)$.

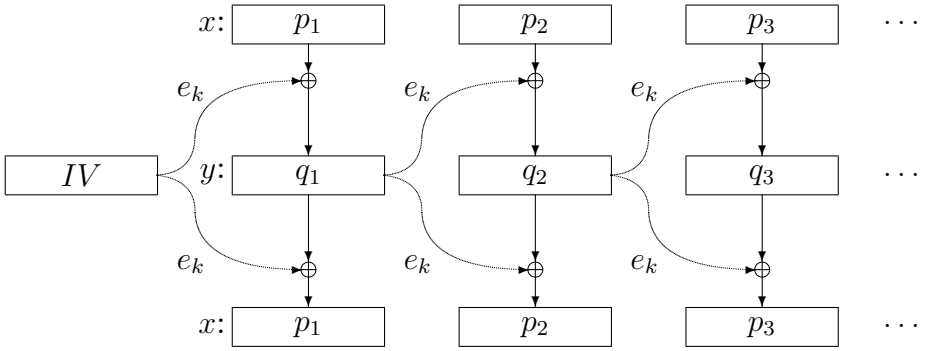


Σχήμα 5.15: Χρήση του DES με ανάδραση από τα κρυπτοκείμενα.

Εξάλλου, είναι συχνή και χρήσιμη η εφαρμογή των λειτουργιών ανάδρασης ακόμη και στην περίπτωση που $l = 64$, δηλαδή τα πακέτα στα οποία είναι ήδη χωρισμένο το μήνυμα είναι πράγματι 64-ψήφια. Τότε τα Σχήματα 5.14 και 5.15 απλοποιούνται κατά πολύ, και αυτές τις απλοποιημένες εκδοχές δείχνουμε στα Σχήματα 5.16 και 5.17.



Σχήμα 5.16: Η λειτουργία με ανάδραση από την έξοδο, όταν $l = 64$.



Σχήμα 5.17: Η λειτουργία με ανάδραση από τα κρυπτοκείμενα, όταν $l = 64$.

5.3.5 Εξέλιξη: Επιθέσεις και Βελτιώσεις

Με την εξέλιξη της κρυπτογραφίας αλλά και της τεχνολογίας των υπολογιστικών συστημάτων, άρχισαν να εμφανίζονται διάφορα είδη επιθέσεων εναντίον του DES. Οι περισσότερες από τις επιθέσεις αυτές είναι της μορφής KPA , δηλαδή ο αντίπαλος έχει στην κατοχή του ένα σύνολο από ζεύγη μηνυμάτων και κρυπτοκειμένων $(m_i, c_i = \texttt{Encrypt}(K, m_i))$. Στόχος του είναι η αποκάλυψη στοιχείων γύρω από το κοινό κλειδί κρυπτογράφησης K των μηνυμάτων.

Εξαντλητική Αναζήτηση Ο πρώτος τρόπος επίθεσης KPA σε οποιοδήποτε κρυπτοσύστημα είναι η δοκιμή όλων των πιθανών κλειδιών (brute force). Αποδεικνύεται σχετικά εύκολα ότι για τον σκοπό αυτό αρκούν μόνο 2 ζεύγη μηνυμάτων και των κρυπτογραφήσεών τους με το δεδομένο κλειδί, καθώς η απόσταση μοναδικότητας (unicity distance), δηλαδή το πλήθος μηνυμάτων το οποίο ακυρώνει τυχαία κλειδιά, είναι 8.2 bytes [32]. Μάλιστα στην περίπτωση του DES, μπορεί να δειχθεί [4], ότι ακόμα και ένα τέτοιο ζεύγος προσδιορίζει μοναδικά το κλειδί με πιθανότητα $99,5\%$.

Είναι φανερό ότι στη χειρότερη περίπτωση θα χρειαστούν 2^{56} δοκιμές - καθώς τα 8 από τα 64 bits του κλειδιού αποτελούν bits ισοτιμίας, κάτι που ακόμα και στην δεκαετία του 1990 ήταν μη αποδεκτό - πόσο μάλλον στις μέρες μας. Ακόμα χειρότερα παρουσιάστηκαν 2 επιθέσεις που μειώνουν σημαντικά τη χειρότερη περίπτωση εξανλητικών δοκιμών, χρησιμοποιώντας όμως περισσότερα τέτοια ζεύγη μηνυμάτων.

Γραμμική Κρυπτανάλυση Η γραμμική κρυπτανάλυση προτάθηκε από τους Matsui και Yamaguro στο [30] για το κρυπτοσύστημα FEAL και σύντομα εφαρμόστηκε και στο DES [29]. Είναι και αυτή μια επίθεση KPA.

Η γραμμική κρυπτανάλυση προσπαθεί να εκφράσει την έξοδο ενός κρυπτοσυστήματος ως γραμμικό συνδυασμό της εισόδου. Αν αυτό επιτευχθεί, χρησιμοποιώντας αρκετά ζεύγη μηνυμάτων - κρυπτοκειμένων, η εύρεση του κλειδιού ανάγεται στην επίλυση ενός γραμμικού συστήματος, η οποία μπορεί να γίνει πολύ απλά χρησιμοποιώντας τη μέθοδο του Gauss. Το πρόβλημα βέβαια, είναι ότι κάθε κρυπτοσύστημα περιέχει και μη γραμμικά στοιχεία, όπως για παράδειγμα τα S-Boxes στο DES, που καθιστούν αδύνατο τον παραπάνω στόχο. Η γραμμική κρυπτανάλυση προσπαθεί με διαδοχικές δοκιμές να βρει την καλύτερη δυνατή γραμμική προσέγγιση (πλέον) της εισόδου και της εξόδου και έτσι να καταλήξει στο κλειδί.

Συγκεκριμένα ψάχνει να βρει k bits από την είσοδο m και l από την έξοδο c για τα οποία να ισχύει:

$$m_{i_1} \oplus \cdots \oplus m_{i_k} \oplus c_{j_1} \oplus j_l = 0$$

S_1

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

S_2

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

S_3

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

S_4

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

S_5

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

S_6

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

S_7

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

S_8

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

Πίνακας 5.5: Οι 8 πίνακες αντικατάστασης. (Αναπαριστούμε τις τετραψήφιες δυαδικές ακολουθίες με τους αντίστοιχους ακεραίους.)

					R_1						
9	45	30	53	43	15	29	50	2	8	17	37
3	31	23	22	39	51	52	1	32	24	16	36
20	25	35	48	33	4	42	27	5	47	21	26
54	19	34	56	14	18	40	13	12	55	49	28
					R_2						
2	38	23	46	36	8	22	43	52	1	10	30
53	24	16	15	32	44	45	51	50	17	9	29
13	18	28	41	26	56	35	20	25	40	14	19
47	12	27	49	7	11	33	6	5	48	42	21
					R_3						
45	24	9	32	22	51	8	29	38	44	53	16
39	10	2	1	43	30	31	37	36	3	52	15
54	4	14	27	12	42	21	6	11	26	55	5
33	25	13	35	48	56	19	47	18	34	28	7
					R_4						
31	10	52	43	8	37	51	15	24	30	39	2
50	53	45	44	29	16	17	23	22	46	38	1
40	49	55	13	25	28	7	47	56	12	41	18
19	11	54	21	34	42	5	33	4	20	14	48
					R_5						
17	53	38	29	51	23	37	1	10	16	50	45
36	39	31	30	15	2	3	9	8	32	24	44
26	35	41	54	11	14	48	33	42	25	27	4
5	56	40	7	20	28	18	19	49	6	55	34
					R_6						
3	39	24	15	37	9	23	44	53	2	36	31
22	50	17	16	1	45	46	52	51	43	10	30
12	21	27	40	56	55	34	19	28	11	13	49
18	42	26	48	6	14	4	5	35	47	41	20
					R_7						
46	50	10	1	23	52	9	30	39	45	22	17
8	36	3	2	44	31	32	38	37	29	53	16
25	7	13	26	42	41	20	5	14	56	54	35
4	28	12	34	47	55	49	18	21	33	27	6
					R_8						
32	36	53	44	9	38	52	16	50	31	8	3
51	22	46	45	30	17	43	24	23	15	39	2
11	48	54	12	28	27	6	18	55	42	40	21
49	14	25	20	33	41	35	4	7	19	13	47

Πίνακας 5.6: Τα διανύσματα $R_1, \ldots, R_8$.

R_9											
50	29	46	37	2	31	45	9	43	24	1	53
44	15	39	38	23	10	36	17	16	8	32	52
4	41	47	5	21	20	54	11	48	35	33	14
42	7	18	13	26	34	28	56	55	12	6	40
R_{10}											
36	15	32	23	45	17	31	52	29	10	44	39
30	1	50	24	9	53	22	3	2	51	43	38
49	27	33	18	7	6	40	56	34	21	19	55
28	48	4	54	12	20	14	42	41	25	47	26
R_{11}											
22	1	43	9	31	3	17	38	15	53	30	50
16	44	36	10	52	39	8	46	45	37	29	24
35	13	19	4	48	47	26	42	20	7	5	41
14	34	49	40	25	6	55	28	27	11	33	12
R_{12}											
8	44	29	52	17	46	3	24	1	39	16	36
2	30	22	53	38	50	51	32	31	23	15	10
21	54	5	49	34	33	12	28	6	48	18	27
55	20	35	26	11	47	41	14	13	56	19	25
R_{13}											
51	30	15	38	3	32	46	10	44	50	2	22
45	16	8	39	24	36	37	43	17	9	1	53
7	40	18	35	20	19	25	14	47	34	4	13
41	6	21	12	56	33	27	55	54	42	5	11
R_{14}											
37	16	1	24	46	43	32	53	30	36	45	8
31	2	51	50	10	22	23	29	3	52	44	39
48	26	4	21	6	5	11	55	33	20	49	54
27	47	7	25	42	19	13	41	40	28	18	56
R_{15}											
23	2	44	10	32	29	43	39	16	22	31	51
17	45	37	36	53	8	9	15	46	38	30	50
34	12	49	7	47	18	56	41	19	6	35	40
13	33	48	11	28	5	54	27	26	14	4	42
R_{16}											
16	52	37	3	50	22	36	32	9	15	24	44
10	38	30	29	46	1	2	8	39	31	23	43
27	5	42	55	40	11	49	34	12	54	28	33
6	26	41	4	21	25	47	20	19	7	56	35

Πίνακας 5.7: Τα διανύσματα $R_9, \ldots, R_{16}$.

Στην ιδανική περίπτωση που η τυχαιοποίηση του κρυπτοσυστημάτος λειτουργεί σωστά, η πιθανότητα να βρεθεί μία τέτοια σχέση είναι $\frac{1}{2}$. Οποιαδήποτε απόκλιση από την τιμή αυτή υπονοεί κάποια αδυναμία του κρυπτοσυστήματος, οπότε έχουμε *γραμμικές προσεγγίσεις*. Για να κατασκευαστεί μία τέτοια σχέση σε κρυπτοσυστήματα τα οποία περιέχουν μη-γραμμικά στοιχεία, πρέπει με κάποιο τρόπο να αλληλοακυρωθούν τα ενδιάμεσα αποτελέσματά τους. Έτσι μπορούμε να βρούμε ποιες γραμμικές προσεγγίσεις των S-boxes συμβαίνουν πιο συχνά και στη συνέχεια να εφαρμόσουμε την λειτουργία του κρυπτοσυστήματος (μεταθέσεις κτλ.) ώστε να ισχύουν για όλο το κρυπτοσύστημα. Τελικά η δυσκολία της επίθεσης ανάγεται στο μέγεθος των υποκλειδιών που χρησιμοποιούνται στους ενδιάμεσους γύρους και όχι στο μέγεθος του συνολικού κλειδιού του κρυπτοσυστήματος.

Διαφορική Κρυπτανάλυση Η διαφορική κρυπτανάλυση είναι μία ισχυρότατη επίθεση τύπου CPA, δηλαδή ο αντίπαλος μπορεί να αποκτήσει κρυπτογραφήσεις μηνυμάτων της επιλογής του. Δεν αφορά αποκλειστικά το DES, αλλά είναι γενικότερη και έχει εφαρμοστεί σε κρυπτοσυστήματα ροής αλλά και συναρτήσεις σύνοψης. Η βασική ιδέα της μεθόδου αυτής προτάθηκε από τους Shamir και Biham στο [3]. Αργότερα [7], αποδείχθηκε ότι ήταν γνωστή στους σχεδιαστές της IBM, οι οποίοι μάλιστα προσπάθησαν να κάνουν το DES ανθεκτικό σε τέτοιο είδους επιθέσεις, αποκρύπτοντας το γεγονός αυτό από την NSA.

Η βασική ιδέα της διαφορικής κρυπτανάλυσης είναι η εξής: Ο αντίπαλος αποκτά κρυπτοκείμενα που αντιστοιχούν σε μηνύματα της επιλογής του τα οποία έχουν μία σταθερή διαφορά μεταξύ τους. Στη συνέχεια αναλύει τα κρυπτοκείμενα μεταξύ τους, υπολογίζοντας την ίδια διαφορά, προσπαθώντας να βρει στατιστικές σχέσεις μεταξύ τους. Ένα είδος διαφοράς είναι η πράξη XOR ($\oplus$). Η εύρεση των πιο πιθανών διαφορών για ένα είδος S-Box, μπορεί οδηγεί στην ανάκτηση κάποιων τμημάτων των υποκλειδιών και εν συνεχεία στην μείωση του χώρου εξαντλητικής αναζήτησης.

Double DES και Triple DES Η πληθώρα λοιπόν αυτή των επιθέσεων εναντίον του DES, είχε ως αποτέλεσμα να θεωρείται επίσημα σπασμένο από το 1997. Νωρίτερα όμως και προτού βρεθεί ο διάδοχός του υπήρξαν ενδιάμεσες προτάσεις με στόχο να βελτιώσουν προσωρινά την ασφάλειά του

Μία πρόταση η οποία δεν εφαρμόστηκε καθώς αποδείχτηκε ευάλωτη ήταν το *Double DES*, το οποίο συνδυάζει 2 κλασσικά κλειδιά DES 56-bit, ελπίζοντας να πετύχει ασφάλεια 112 bit. Δηλαδή:

$$\texttt{Encrypt}(k_1, k_2, m) = \texttt{Encrypt}(k_1, \texttt{Encrypt}(k_2, m))$$

Δυστυχώς ο τρόπος σχεδιασμού αυτός το κάνει ευάλωτο στην επίθεση Meet in

The Middle (MITM), η οποία μπορεί να οδηγήσει στην αποκάλυψη των κλειδιών ως εξής:

- Ο αντίπαλος με δεδομένο ένα κρυπτοκείμενο c ψάχνει να βρει (k_1, k_2) ώστε $c = \texttt{Encrypt}(k_1, \texttt{Encrypt}(k_2, m))$
- ή ισοδύναμα: $\texttt{Decrypt}(k_1, c) = \texttt{Encrypt}(k_2, m)$
- Για κάθε έναν από τους 2^56 συνδυασμούς ενός κλειδιού k μπορούν να υπολογιστούν και να αποθηκευτούν οι τιμές $\texttt{Encrypt}(k, m)$ ταξινομημένες σε κάποιον πίνακα $\mathcal{T}$.
- Στην συνέχεια μπορεί να χρησιμοποιηθεί δυαδική αναζήτηση ώστε να βρεθεί στον πίνακα αυτό η τιμή $\texttt{Decrypt}(k', c)$ για κάποιο k' .
- Όταν θα συμβεί αυτό θα ξέρουμε ότι $(k', k) = (k_1, k_2)$

Με απλά λόγια, η επίθεση αυτή ανάγει την αναζήτηση σε χρόνου τάξεως 2^{112} σε αναζήτηση χρόνου $2^{56} \cdot \log(2^{56}) + 2^{56} \cdot \log(2^{56})$ για την ταξινομήση και την δυαδική αναζήτηση για κάθε δυνατή τιμή στον πίνακα χώρου $2^5 6$.

Για τον παραπάνω λόγο πρακτικά χρησιμοποιήθηκε τελικά το *Triple-DES*, το οποίο ορίζεται ως εξής:

$$\texttt{Encrypt}(k_1, k_2, k_3, m) = \texttt{Encrypt}(k_1, \texttt{Decrypt}(k_2, \texttt{Encrypt}(k_3, m)))$$

Μία άλλη παραλλαγή που χρησιμοποιείται είναι το *DES-X*, το οποίο ορίζεται ως εξής:

$$\texttt{Encrypt}_(m) = k_2 \oplus E_{k_3}(m \oplus k_1)$$

5.4 Advanced Encryption Standard (AES)

Τον Ιανουάριο του 1997 το NIST ανακοίνωσε την πρόθεση για την ανάπτυξη ενός νέου κρυπτογραφικού προτύπου, του Advanced Encryption Standard (AES) , που θα αντικαθιστούσε το DES. Τον Σεπτέμβριο του 1997 ανακοινώθηκε η έναρξη ανοιχτού διαγωνισμού. Το NIST δεν θα προχωρούσε σε αξιολόγηση της ασφάλειας και της αποδοτικότητας των υποψηφίων κρυπτοσυστημάτων, αλλά κάλεσε την κρυπτολογική κοινότητα να πραγματοποιήσει επιθέσεις. Τα σχετικά αποτελέσματα θα ανακοινώνονταν σε μια σειρά συνεδρίων. Πάντως το AES είχε από την αρχή τις εξής βασικές προδιαγραφές:

- να είναι κρυπτοσύστημα πακέτου,

- να υποστηρίζει πακέτα των 128 bit, και
- να υποστηρίζει κλειδιά των 128, 192 και 256 bit.

Οι υποψήφιοι αλγόριθμοι που πληρούσαν τις αρχικές απαιτήσεις του διαγωνισμού ήταν 15, από τους οποίους οι 9 ήταν προτάσεις προερχόμενες εκτός ΗΠΑ. Στα πλαίσια του πρώτου γύρου που έληξε τον Αύγουστο του 1999, διοργανώθηκαν δύο συνέδρια με θέμα τα χαρακτηριστικά των υποψηφίων κρυπτοσυστημάτων. Με το τέλος του πρώτου γύρου τα υποψήφια κρυπτοσυστήματα μειώθηκαν σε 5 (MARS, RC6, Rijndael, Serpent και Twofish).

Μέσα στους 14 μήνες που διήρκησε ο δεύτερος γύρος μεσολάβησε ένα ακόμα συνέδριο και τελικά τον Οκτώβριο του 2000 το NIST ανακοίνωσε [34] πως ο Rijndael, των Βέλγων Joan Daemen και Vincent Rijmen, χωρίς ουσιαστικές τροποποιήσεις, θα αποτελούσε το AES. Η επιλογή έγινε τόσο με βάση την ασφάλεια — αντοχή στις επιθέσεις που πραγματοποιήθηκαν, αλλά και θεωρητική άμυνα απέναντι σε μεθόδους γραμμικής και διαφορικής κρυπτανάλυσης — όσο και με βάση την ταχύτητα, την απλότητα και την ευελιξία του Rijndael.

Το Rijndael είναι ένα κρυπτοσύστημα πακέτου με μεταβλητό μέγεθος πακέτου και μεταβλητό μέγεθος κλειδιού. Το μέγεθος πακέτου και το μέγεθος κλειδιού μπορούν ανεξάρτητα να καθοριστούν σε οποιοδήποτε πολλαπλάσιο του 32 μεταξύ 128 και 256 bit.

Το μοναδικό σημείο διαφοροποίησης του AES από το Rijndael είναι ο περιορισμός του μεγέθους του κλειδιού και του πακέτου, καθώς το σταθεροποιεί στα 128 bit και υποστηρίζει μόνο κλειδιά των 128,192 ή 256 bit. Τέθηκε σε ισχύ στις 26 Μαΐου 2002 από το αμερικάνικο Υπουργείο Εμπορίου.

Στο Rijndael, θεωρούμε τα byte ως πολυώνυμα πάνω από το $GF(2)$ με τις πράξεις που ορίστηκαν στο 2.2.2, όπου για να ορίσουμε τον πολλαπλασιασμό χρησιμοποιούμε το ανάγωγο πολυώνυμο

$$m(x) = x^8 + x^4 + x^3 + x + 1.$$

Μ"αυτόν τον τρόπο κατασκευάζουμε μια αναπαράσταση για το $GF(2^8)$. Κάθε byte είναι έτσι στοιχείο του $GF(2^8)$.

Στην περιγραφή του Rijndael, οι στήλες των 4 byte θεωρούνται πολυώνυμα πάνω από το $GF(2^8)$ με βαθμό μικρότερο του 4. Για να ορίσουμε τον πολλαπλασιασμό χρησιμοποιούμε το πολυώνυμο

$$l(x) = x^4 + 1.$$

Εδώ το πολυώνυμο αναγωγής που έχουμε δεν είναι ανάγωγο αφού στο $GF(2^8)$

$$x^4 + 1 = (x + 1)^4.$$

Επομένως δεν έχει κάθε πολυώνυμο πολλαπλασιαστικό αντίστροφο. Καθώς στον αλγόριθμο πάντα ο ένας από τους δύο παράγοντες θα είναι ένα σταθερό πολυώνυμο, αρκεί αυτό να επιλεγεί έτσι ώστε να μη διαιρείται με το $x+1$, προκειμένου η πράξη να αντιστρέφεται.

5.4.1 Είσοδος και Έξοδος

Για την κρυπτογράφηση η είσοδος είναι ένα πακέτο απλού κειμένου και ένα κλειδί και η έξοδος ένα πακέτο κρυπτοκειμένου. Για την αποκρυπτογράφηση η είσοδος είναι ένα πακέτο κρυπτοκειμένου και ένα κλειδί και η έξοδος ένα πακέτο απλού κειμένου. Για την επεξεργασία της εισόδου χρησιμοποιείται ένας βοηθητικός πίνακας που ονομάζεται *State*.

Ο State είναι ένας $4 \times N_b$ πίνακας από byte, όπου N_b το πηλίκο του μεγέθους πακέτου προς 32. Έστω το πακέτο απλού κειμένου:

$$p_0 p_1 p_2 \dots p_{4 \cdot N_b - 1},$$

όπου p_0 το πρώτο byte, p_1 το δεύτερο κτλ. Παρόμοια ένα πακέτο κρυπτοκειμένου μπορεί να παρασταθεί ως:

$$c_0 c_1 c_2 \dots c_{4 \cdot N_b - 1}.$$

Έστω $a_{i,j}$, $0 \leq< 4$, $0 \leq j < N_b$ το byte που βρίσκεται στην i γραμμή και την j στήλη του State. Στην κρυπτογράφηση η είσοδος (πακέτο απλού κειμένου) απεικονίζεται (5.8):

$$a_{i,j} = p_{i+4j} \ , \ 0 \leq i < 4, \ 0 \leq j < N_b.$$

Αντίστοιχα στην αποκρυπτογράφηση:

$$a_{i,j} = c_{i+4j} \ , \ 0 \leq i < 4, \ 0 \leq j < N_b.$$

Σύμφωνα με τα παραπάνω στο τέλος της κρυπτογράφησης το κρυπτοκείμενο εξάγεται από τον State ως εξής:

$$c_i = a_{imod4,i/4} \ , \ 0 \leq i < 4N_b.$$

Αντίστοιχα στο τέλος της αποκρυπτογράφησης το απλό κείμενο εξάγεται από τον State:

$$p_i = a_{imod4,i/4} \ , \ 0 \leq i < 4N_b.$$

Με παρόμοιο τρόπο το κλειδί απεικονίζεται σ"ένα $4 \times N_k$ πίνακα K, όπου N_k το πηλίκο του μεγέθους του κλειδιού προς 32.

Αν $z_0 z_1 z_2 \dots z_{4 \cdot N_b - 1}$ το κλειδί και $k_{i,j}$ το τυχόν στοιχείο του K, τότε

$$k_{i,j} = z_{i+4j} \ \ 0 \leq i < 4, \ 0 \leq j < N_k.$$

Για το πρότυπο AES έχουμε $N_b = 4$ και $N_k = 4, 6, 8$.

p_0	p_4	p_8	p_{12}
p_1	p_5	p_9	p_{13}
p_2	p_6	p_{10}	p_{14}
p_3	p_7	p_{11}	p_{15}

Πίνακας 5.8: Ο πίνακας State για $N_b = 4$.

5.4.2 Βασικοί Μετασχηματισμοί

Ο αλγόριθμος, για την κρυπτογράφηση εκτελεί έναν προκαθορισμένο (από το μήκος του κλειδιού και το μέγεθος του πακέτου) αριθμό επαναλήψεων τεσσάρων βασικών βημάτων (`SubBytes`, `ShiftRows`, `MixColumns`, `AddRoundKey`). Κάθε τέτοια επανάληψη συνιστά ένα *γύρο* (`Round`) του αλγορίθμου. Το πλήθος των εκτελούμενων γύρων ισούται με N_r, με τον τελευταίο γύρο να διαφέρει από τους υπόλοιπους στο ότι δεν περιλαμβάνει την `MixColumns`. Ανάλογα ισχύουν και για την αποκρυπτογράφηση, όπου εκτελούνται με την αντίστροφη σειρά οι αντίστροφες διαδικασίες (`InvSubBytes`, `InvShiftRows`, `InvMixColumns`, `AddRoundKey`). Στα Σχήματα 2 και 3 φαίνεται ο αντίστοιχος ψευδοκώδικας.

Το πλήθος των εκτελούμενων γύρων N_r εξαρτάται από τα N_b, N_k και δίνεται στον Πίνακα 5.9, όπου για το πρότυπο AES ενδιαφέρουν μόνο οι έντονα τυπωμένες τιμές.

	N_b				
N_k	**4**	5	6	7	8
4	**10**	11	12	13	14
5	11	11	12	13	14
6	**12**	12	12	13	14
7	13	13	13	13	14
8	**14**	14	14	14	14

Πίνακας 5.9: Το πλήθος N_r των εκτελούμενων γύρων συναρτήσει των N_b, N_k.

Ο Μετασχηματισμός `SubBytes`

Η διαδικασία `SubBytes` είναι ο μόνος μη γραμμικός μετασχηματισμός του κρυπτοσυστήματος. Ουσιαστικά πρόκειται για έναν *πίνακα αντικατάστασης* (S-box), τον οποίο θα ονομάζουμε S_{RD}, που δρα στα byte του State (Σχήμα 5.20).

Ο S_{RD} έχει προκύψει από τη σύνθεση δύο απεικονίσεων:

Έστω $a = a_7a_6a_5a_4a_3a_2a_1a_0 \in GF(2^8)$ (το a όπως έχουμε πεί είναι byte). Τότε

```
Cipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
begin
   byte state[4,Nb]

   state = in

   AddRoundKey(state, w[0, Nb-1])

   for round = 1 step 1 to Nr-1
      SubBytes(state)
      ShiftRows(state)
      MixColumns(state)
      AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
   end for

   SubBytes(state)
   ShiftRows(state)
   AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])

   out = state
end
```

Σχήμα 5.18: Ψευδοκώδικας για την κρυπτογράφηση.

θεωρούμε την απεικόνιση που αντιστρέφει τα στοιχεία του $GF(2^8) \setminus 0$ και στέλνει το 0 στον εαυτό του:

$$g : a \to b = \begin{cases} a^{-1} & a \neq 0 \\ 0 & , \ a = 0 \end{cases}$$

και τον αφφινικό μετασχηματισμό:

```
InvCipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
begin
   byte state[4,Nb]

   state = in

   AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])

   for round = Nr-1 step -1 downto 1
      InvShiftRows(state)
      InvSubBytes(state)
      AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
      InvMixColumns(state)
   end for

   InvShiftRows(state)
   InvSubBytes(state)
   AddRoundKey(state, w[0, Nb-1])

   out = state
end
```

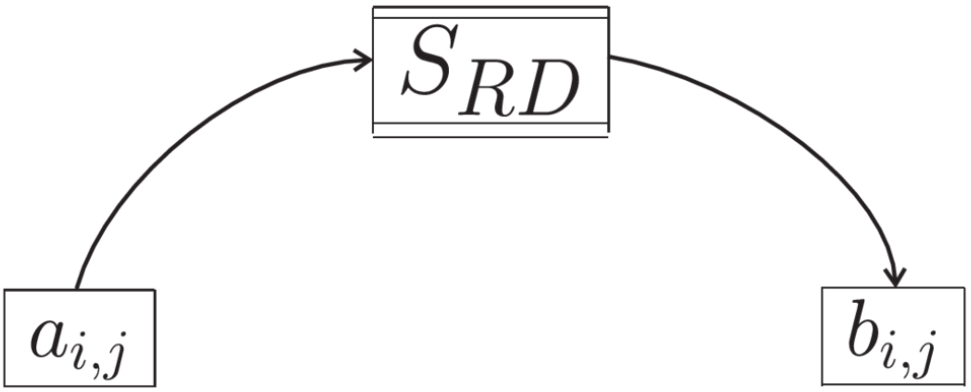
Σχήμα 5.19: Ψευδοκώδικας για την αποκρυπτογράφηση.

$$f : a \to b = \begin{vmatrix} b_7 \\ b_6 \\ b_5 \\ b_4 \\ b_3 \\ b_2 \\ b_1 \\ b_0 \end{vmatrix} = \begin{vmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{vmatrix} \times \begin{vmatrix} a_7 \\ a_6 \\ a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{vmatrix} \oplus \begin{vmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{vmatrix}.$$

Τότε ορίζουμε

$$S_{RD}[a] = f(g(a)).$$

Η υλοποίηση της `SubBytes` γίνεται με την απ'ευθείας χρήση του πίνακα S_{RD} (Πίνακας 5.10).



Σχήμα 5.20: Η `SubBytes` δρα σε κάθε ένα byte του State χωριστά.

Ο αντίστροφος μετασχηματισμός. Η `InvSubBytes` είναι ο αντίστροφος μετασχηματισμός της `SubBytes` . Είναι φανερό ότι

$$S_{RD}^{-1}[a] = g^{-1}(f^{-1}(a)) = g(f^{-1}(a)).$$

Ο αφφινικός μετασχηματισμός f^{-1} είναι:

$$f^{-1} : a \to b = \begin{vmatrix} b_7 \\ b_6 \\ b_5 \\ b_4 \\ b_3 \\ b_2 \\ b_1 \\ b_0 \end{vmatrix} = \begin{vmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{vmatrix} \times \begin{vmatrix} a_7 \\ a_6 \\ a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{vmatrix} \oplus \begin{vmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{vmatrix}.$$

Όπως και στην `SubBytes` , έτσι κι εδώ η υλοποίηση της `InvSubBytes` γίνεται με την απ"ευθείας χρήση του πίνακα S_{RD}^{-1} (Πίνακας 5.11).

Ο Μετασχηματισμός `ShiftRows`

Στον μετασχηματισμό `ShiftRows`, η γραμμή i του State μετατίθενται κυκλικά προς τα αριστερά κατά C_i θέσεις.

Συγκεκριμένα, αν $a_{i,j}$ κάποιο στοιχείο του State, τότε

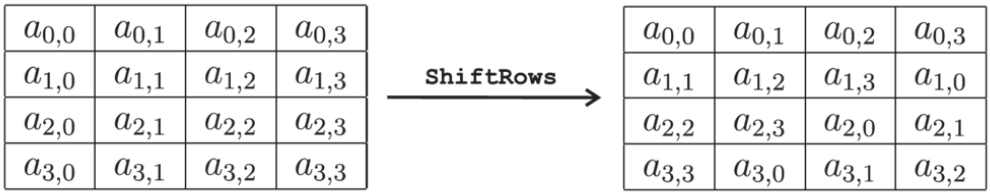
$$a_{i,j} \longrightarrow a'_{i,j} = a_{i,(j+C_i) \mod N_b}.$$

Οι τιμές των C_i για τις διάφορες τιμές του N_b φαίνονται παρακάτω. Φυσικά για το πρότυπο AES ενδιαφέρει μόνο η περίπτωση που $N_b = 4$:

N_b	C_0	C_1	C_2	C_3
4	**0**	**1**	**2**	**3**
5	0	1	2	3
6	0	1	2	3
7	0	1	2	4
8	0	1	3	4

Ο αντίστροφος μετασχηματισμός. Ο αντίστροφος μετασχηματισμός `InvShiftRows` είναι προφανής, αφού αρκεί να μεταθέσουμε κυκλικά προς τα δεξιά κατά C_i θέσεις την γραμμή i του State, δηλαδή:

$$a_{i,j} \longrightarrow a'_{i,j} = a_{i,(j-C_i)} \mod N_b.$$



Σχήμα 5.21: Η δράση της `ShiftRows` πάνω στον State για $N_b = 4$.

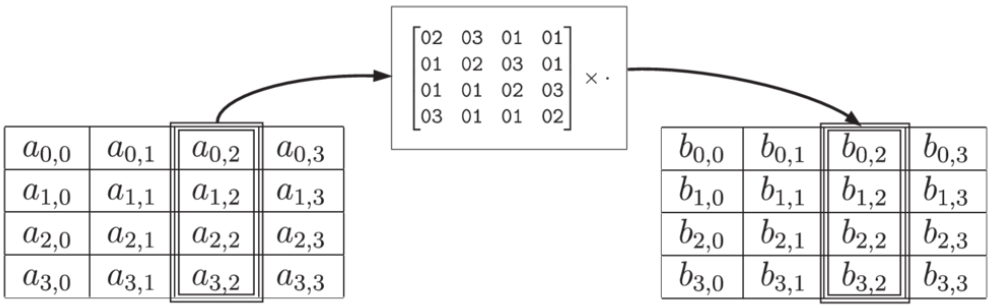
Ο Μετασχηματισμός `MixColumns`

Στον μετασχηματισμό `MixColumns` οι στήλες του State θεωρούνται πολυώνυμα πάνω από το $GF(2^8)$ και πολλαπλασιάζονται modulo $x^4 + 1$ με ένα σταθερό πολυώνυμο $c(x)$. Το σταθερό πολυώνυμο είναι το

$$c(x) = 03 \cdot x^3 + 01 \cdot x^2 + 01 \cdot x + 02,$$

το οποίο δεν διαιρείται με το $x + 1$ και επομένως είναι αντιστρέψιμο (βλέπε Υποενότητα 2.2.2). Έστω $b(x) = c(x) \cdot a(x) \pmod{x^4 + 1}$. Όπως ήδη έχουμε δει, μπορούμε να χρησιμοποιήσουμε πολλαπλασιασμό πινάκων:

$$\begin{vmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{vmatrix} = \begin{vmatrix} & 02030101 & & \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{vmatrix} \times \begin{vmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{vmatrix}.$$



Σχήμα 5.22: Η `MixColumnes` δρα σε κάθε μία στήλη του State.

Ο αντίστροφος μετασχηματισμός. Ο αντίστροφος μετασχηματισμός της `InvMixColumns`, όπως είναι φανερό από την έως τώρα θεώρηση μας, θα είναι ο πολλαπλασιασμός των στηλών του State με ένα πολυώνυμο $d(x)$ τέτοιο ώστε:

$$(03 \cdot x^3 + 01 \cdot x^2 + 01 \cdot x + 02) \times d(x) \equiv 01 \pmod{x^4 + 1}.$$

Ισχύει ότι

$$d(x) = c^{-1}(x) = 0b \cdot x^3 + 0d \cdot x^2 + 09 \cdot x + 0e,$$

και συνεπώς, γραμμένη ως πολλαπλασιασμός πινάκων, η `InvMixColumns` μετασχηματίζει τις στήλες ως εξής:

$$\begin{vmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{vmatrix} = \begin{vmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0e & 09 & 0e \end{vmatrix} \times \begin{vmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{vmatrix}.$$

Υλοποίηση. Για την αποδοτική υλοποίηση των `MixColumns` και `InvMixColumns` χρησιμοποιείται ένας 16×16 πίνακα, ton xtimes, του οποίου το ij στοιχείο ($0 \leq i, j \leq 15$) είναι το γινόμενο των byte 02 και $i_{16}j_{16}$ στο $GF(2^8)$, όπου i_{16}, j_{16} τα αντίστοιχα των i, j δεκαεξαδικά ψηφία. Το byte 02 αναπαριστά το πολυώνυμο $\beta(x) = x$, δηλαδή ο xtimes είναι ο πίνακας πολλαπλασιασμού κάθε στοιχείου του $GF(2^8)$ με το $x \in GF(2^8)$. Έτσι ο πολλαπλασιασμός στο $GF(2^8)$ ανάγεται στον υπολογισμό γινομένων με δυνάμεις του x και την XOR πρόσθεσή τους.

Παράδειγμα 10. Ισχύει ότι

$$\begin{aligned} 0e &= 08 \oplus 04 \oplus 02 \\ &= 02^3 \oplus 04^2 \oplus 02. \end{aligned}$$

Έτσι ο πολλαπλασιασμός οποιουδήποτε byte α με το $0e$ γίνεται:

$$0e \times \alpha = (02^3 \times \alpha) \oplus (04^2 \times \alpha) \oplus (02 \times \alpha)$$
$$= \text{xtimes(xtimes(xtimes}(\alpha))) \oplus \text{xtimes(xtimes}(\alpha)) \oplus \text{xtimes}((\alpha).$$

Ο Μετασχηματισμός `AddRoundKey`

Στον μετασχηματισμό `AddRoundKey`, ένα *Κλειδί Γύρου* (Round Key) προστίθεται στον State με bitwise XOR. Κάθε Κλειδί Γύρου αποτελείται από N_b λέξεις των τεσσάρων byte από το διάνυσμα key schedule το οποίο θα συμβολίζουμε με w (βλέπε Υποενότητα 5.4.3). Έτσι κατά τον i γύρο για την j στήλη του State έχουμε: $\left| \; a_{0,j}, a_{1,j}, a_{2,j}, a_{3,j} \; \right| \oplus \; w_{i \cdot N_b + j} \; = \left| \; a'_{0,j}, a'_{1,j}, a'_{2,j}, a'_{3,j} \; \right|$.

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$

$+$

$k_{0,0}$	$k_{0,1}$	$k_{0,2}$	$k_{0,3}$
$k_{1,0}$	$k_{1,1}$	$k_{1,2}$	$k_{1,3}$
$k_{2,0}$	$k_{2,1}$	$k_{2,2}$	$k_{2,3}$
$k_{3,0}$	$k_{3,1}$	$k_{3,2}$	$k_{3,3}$

$=$

$b_{0,0}$	$b_{0,1}$	$b_{0,2}$	$b_{0,3}$
$b_{1,0}$	$b_{1,1}$	$b_{1,2}$	$b_{1,3}$
$b_{2,0}$	$b_{2,1}$	$b_{2,2}$	$b_{2,3}$
$b_{3,0}$	$b_{3,1}$	$b_{3,2}$	$b_{3,3}$

Σχήμα 5.23: Η `AddRoundKey` για $N_b = 4$.

Ο αντίστροφος μετασχηματισμός. Είναι γνωστό ότι $a \oplus a = 0$, όπου $\oplus$ η αποκλειστική διάζευξη (XOR). Επομένως η αντίστροφη της `AddRoundKey` είναι η ίδια η `AddRoundKey`.

5.4.3 Η Επέκταση Κλειδιού

Η διαδικασία Key Expansion δέχεται σαν είσοδο το αρχικό κλειδί, K, που δίνει ο χρήστης και το χρησιμοποιεί για να παράξει ένα κλειδί για κάθε γύρο. Ο αλγόριθμος χρειάζεται αρχικά ένα σύνολο $4N_b$ byte (για την πρώτη `AddRoundKey`) και στη συνέχεια κάθε ένας από τους N_r γύρους χρειάζεται $4N_b$ byte για το αντίστοιχο κλειδί. Έτσι η Key Expansion παράγει το διάνυσμα w (key schedule) μήκους $N_b(N_r + 1)$, του οποίου κάθε συνιστώσα $[w_i]$, με $0 \leq i < N_b(N_r + 1)$, είναι μία λέξη τεσσάρων byte.

Ο ψευδοκώδικας για την επέκταση του αρχικού κλειδιού στο key schedule φαίνεται στο Σχήμα 5.24.

Η συνάρτηση `SubWord` δέχεται σαν όρισμα μία λέξη τεσσάρων byte και στέλνει κάθε ένα από αυτά στην εικόνα του μέσω του πίνακα αντικατάστασης S_{RD}, παράγοντας έτσι μια νέα λέξη.

Η συνάρτηση `RotWord` δέχεται μία λέξη $[a_0, a_1, a_2, a_3]$ και επιστρέφει την κυκλική της μετάθεση $[a_1, a_2, a_3, a_0]$.

Το διάνυσμα `Rcon[i]`, με $0 \leq i \leq N_r$, ισούται με $[x^{i-1}, 00, 00, 00]$, όπου το x^{i-1} παριστάνει το αντίστοιχο στοιχείο του $GF(2^8)$. Για παράδειγμα,

$$\begin{aligned} \texttt{Rcon[5]} &= [x^4, 00, 00, 00] \\ &= [10, 00, 00, 00]. \end{aligned}$$

Οι πρώτες N_k λέξεις του επεκτεταμένου κλειδιού καταλαμβάνονται από το αρχικό κλειδί. Κάθε επόμενη λέξη, $[w_i]$, ισούται με την XOR πρόσθεση της προηγούμενης, $[w_{i-1}]$, μετασχηματισμένης, και της λέξης που προηγείται κατά N_k θέσεις, $[w_{i-N_k}]$. Στην περίπτωση που το i είναι πολλαπλάσιο του N_k, η $[w_{i-1}]$ μετασχηματίζεται πριν την XOR πρόσθεση σε:

$$\texttt{temp} = \texttt{SubWord}\ (\texttt{RotWord}\ ([w_{i-1}])) \oplus \texttt{Rcon[i]},$$

ενώ όταν $N_k > 6$ και το $i-4$ είναι πολλαπλάσιο του N_k, η $[w_{i-1}]$ μετασχηματίζεται σε:

$$\texttt{temp} = \texttt{SubWord}\ ([w_{i-1}]).$$

Σε κάθε άλλη περίπτωση το $[w_{i-1}]$ παραμένει αμετάβλητο πριν την XOR πρόσθεση με το $[w_{i-N_k}]$.

5.4.4 Ισοδύναμη Αποκρυπτογράφηση

Όπως αναφέρθηκε και στην αρχή της Παραγράφου 5.4.2, η προφανής αποκρυπτογράφηση είναι να εφαρμόσουμε πάνω στο κρυπτοπακέτο τους αντίστροφους μετασχηματισμούς με την αντίστροφη σειρά. Μπορούμε εντούτοις να απλοποιήσουμε τη διαδικασία της αποκρυπτογράφησης παρατηρώντας τα εξής:

- Η `InvSubBytes` και η `InvShiftRows` αντιμετατίθενται.
- Η `AddRoundKey` και η `InvMixColumns` αντιμετατίθενται, αν τροποποιήσουμε κατάλληλα το Κλειδί Γύρου.

Το πρώτο συμβαίνει γιατί η `InvShiftRows` αλλάζει τη διάταξη των byte στον State χωρίς να επηρεάζει τις τιμές τους, ενώ η `InvSubBytes` αλλάζει τις τιμές τους ανεξάρτητα από τη θέση τους.

Η ορθότητα της δεύτερης παρατήρησης προκύπτει από τη γραμμικότητα της

```
KeyExpansion(byte key[4*Nk], word w[Nb*(Nr+1)], Nk)
begin
   word temp

   i = 0

   while (i < Nk)
      w[i] = word(key[4*i], key[4*i+1], key[4*i+2], key[4*i+3])
      i = i+1
   end while

   i = Nk

   while (i < Nb * (Nr+1)]
      temp = w[i-1]
      if (i mod Nk = 0)
         temp = SubWord(RotWord(temp)) xor Rcon[i/Nk]
      else if (Nk > 6 and i mod Nk = 4)
         temp = SubWord(temp)
      end if
      w[i] = w[i-Nk] xor temp
      i = i + 1
   end while
end
```

Σχήμα 5.24: Ψευδοκώδικας για την επέκταση κλειδιού.

$$\texttt{InvMixColumns}: \texttt{InvMixColumns(State} \oplus \texttt{RoundKey)} = \\ = \texttt{InvMixColumns(State)} \oplus \texttt{InvMixColumns(RoundKey)}.$$

Αρκεί λοιπόν να χρησιμοποιήσουμε ως *ισοδύναμο Κλειδί Γύρου* το αποτέλεσμα της εφαρμογής της `InvMixColumns` στο Κλειδί Γύρου.

Με βάση τα παραπάνω, η αποκρυπτογράφηση μπορεί να μετασχηματιστεί σε μία *ισοδύναμη αποκρυπτογράφηση*, η οποία έχει την ίδια δομή με την κρυπτογράφηση (Σχήμα 5.25). Το γεγονός αυτό είναι ουσιαστικό για την απλότητα του κρυπτοσυστήματος και για την αποδοτικότητά του τόσο σε software όσο και σε hardware.

```
EqInvCipher(byte in[4*Nb], byte out[4*Nb], word dw[Nb*(Nr+1)])
begin
   byte  state[4,Nb]

   state = in

   AddRoundKey(state, dw[Nr*Nb, (Nr+1)*Nb-1])

   for round = Nr-1 step -1 downto 1
      InvSubBytes(state)
      InvShiftRows(state)
      InvMixColumns(state)
      AddRoundKey(state, dw[round*Nb, (round+1)*Nb-1])
   end for

   InvSubBytes(state)
   InvShiftRows(state)
   AddRoundKey(state, dw[0, Nb-1])

   out = state
end
```

Για την ισοδύναμη αποκρυπτογράφηση, ο ακόλουθος ψευδοκώδικας προστίθεται στο τέλος της Key Expansion

```
   for i = 0 step 1 to (Nr+1)*Nb-1
      dw[i] = w[i]
   end for

   for round = 1 step 1 to Nr-1
      InvMixColumns(dw[round*Nb, (round+1)*Nb-1]) // note change of type
   end for
```

Σχήμα 5.25: Ψευδοκώδικας για την ισοδύναμη αποκρυπτογράφηση.

5.5 Πίνακες Αντικατάστασης

Παρακάτω φαίνονται οι πίνακες αντικατάστασης S_{RD} και S_{RD}^{-1} που χρησιμοποιούμε για την `SubBytes` και την `InvSubBytes` .

		y														
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Πίνακας 5.10: Ο πίνακας S_{RD}. Το στοιχείο $S_{RD}(i, j)$ είναι η εικόνα του ij.

		y														
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
	1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
	2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
	3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
	4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
	5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
	6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
	7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
	8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
	9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
	a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
	b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
	c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
	d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
	e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
	f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Πίνακας 5.11: Ο πίνακας S_{RD}^{-1}. Το στοιχείο $S_{RD}^{-1}(i, j)$ είναι η εικόνα του ij.

5.6 Παράδειγμα του Rijndael για ένα block

Παρακάτω φαίνεται βήμα προς βήμα η εκτέλεση του αλγορίθμου με είσοδο ένα πακέτο 128 bit απλού κειμένου και κλειδί του ίδιου μήκους. Χρησιμοποιούμε δεκαεξαδικό συμβολισμό. Για τις συντομογραφίες της αριστερής στήλης έχουμε για τη CIPHER (κρυπτογράφηση):

input:	πακέτο απλού κειμένου
start:	το περιεχόμενο του State στην αρχή του r γύρου
s_box:	το περιεχόμενο του State μετά την `SubBytes`
s_row:	το περιεχόμενο του State μετά την `ShiftRows`
m_col:	το περιεχόμενο του State μετά την `MixColumns`
k_sch:	η τιμή του key schedule για τον γύρο r
output:	πακέτο κρυπτοκειμένου,

για την INVERSE CIPHER (αποκρυπτογράφηση):

iinput:	πακέτο κρυπτοκειμένου
istart:	το περιεχόμενο του State στην αρχή του r γύρου
is_box:	το περιεχόμενο του State μετά την `InvSubBytes`
is_row:	το περιεχόμενο του State μετά την `InvShiftRows`
ik_sch:	η τιμή του key schedule για τον γύρο r
ik_add:	το περιεχόμενο του State μετά την `AddRoundKey`
ioutput:	πακέτο απλού κειμένου

και για την EQUIVALENT INVERSE CIPHER (ισοδύναμη αποκρυπτογράφηση):

iinput:	πακέτο κρυπτοκειμένου
istart:	το περιεχόμενο του State στην αρχή του r γύρου
is_box:	το περιεχόμενο του State μετά την `InvSubBytes`
is_row:	το περιεχόμενο του State μετά την `InvShiftRows`
im_col:	το περιεχόμενο του State μετά την `InvMixColumns`
ik_sch:	η τιμή του key schedule για τον γύρο r
ioutput:	πακέτο απλού κειμένου,

όπου ο αριθμός γύρου r=0 έως 10.

Πακέτο απλού κειμένου: $00112233445566778899aabbccddeeff$
Κλειδί: $000102030405060708090a0b0c0d0e0f$

```
CIPHER (ENCRYPT):
```

```
round[ 0].input  00112233445566778899aabbccddeeff
round[ 0].k_sch  000102030405060708090a0b0c0d0e0f
round[ 1].start  00102030405060708090a0b0c0d0e0f0
round[ 1].s_box  63cab7040953d051cd60e0e7ba70e18c
round[ 1].s_row  6353e08c0960e104cd70b751bacad0e7
round[ 1].m_col  5f72641557f5bc92f7be3b291db9f91a
round[ 1].k_sch  d6aa74fdd2af72fadaa678f1d6ab76fe
round[ 2].start  89d810e8855ace682d1843d8cb128fe4
round[ 2].s_box  a761ca9b97be8b45d8ad1a611fc97369
round[ 2].s_row  a7be1a6997ad739bd8c9ca451f618b61
round[ 2].m_col  ff87968431d86a51645151fa773ad009
round[ 2].k_sch  b692cf0b643dbdf1be9bc5006830b3fe
round[ 3].start  4915598f55e5d7a0daca94fa1f0a63f7
round[ 3].s_box  3b59cb73fcd90ee05774222dc067fb68
round[ 3].s_row  3bd92268fc74fb735767cbe0c0590e2d
round[ 3].m_col  4c9c1e66f771f0762c3f868e534df256
round[ 3].k_sch  b6ff744ed2c2c9bf6c590cbf0469bf41
round[ 4].start  fa636a2825b339c940668a3157244d17
round[ 4].s_box  2dfb02343f6d12dd09337ec75b36e3f0
round[ 4].s_row  2d6d7ef03f33e334093602dd5bfb12c7
round[ 4].m_col  6385b79ffc538df997be478e7547d691
round[ 4].k_sch  47f7f7bc95353e03f96c32bcfd058dfd
round[ 5].start  247240236966b3fa6ed2753288425b6c
round[ 5].s_box  36400926f9336d2d9fb59d23c42c3950
round[ 5].s_row  36339d50f9b539269f2c092dc4406d23
round[ 5].m_col  f4bcd45432e554d075f1d6c51dd03b3c
round[ 5].k_sch  3caaa3e8a99f9deb50f3af57adf622aa
round[ 6].start  c81677bc9b7ac93b25027992b0261996
round[ 6].s_box  e847f56514dadde23f77b64fe7f7d490
round[ 6].s_row  e8dab6901477d4653ff7f5e2e747dd4f
round[ 6].m_col  9816ee7400f87f556b2c049c8e5ad036
round[ 6].k_sch  5e390f7df7a69296a7553dc10aa31f6b
round[ 7].start  c62fe109f75eedc3cc79395d84f9cf5d
round[ 7].s_box  b415f8016858552e4bb6124c5f998a4c
round[ 7].s_row  b458124c68b68a014b99f82e5f15554c
round[ 7].m_col  c57e1c159a9bd286f05f4be098c63439
round[ 7].k_sch  14f9701ae35fe28c440adf4d4ea9c026
round[ 8].start  d1876c0f79c4300ab45594add66ff41f
round[ 8].s_box  3e175076b61c04678dfc2295f6a8bfc0
round[ 8].s_row  3e1c22c0b6fcbf768da85067f6170495
round[ 8].m_col  baa03de7a1f9b56ed5512cba5f414d23
round[ 8].k_sch  47438735a41c65b9e016baf4aebf7ad2
round[ 9].start  fde3bad205e5d0d73547964ef1fe37f1
round[ 9].s_box  5411f4b56bd9700e96a0902fa1bb9aa1
round[ 9].s_row  54d990a16ba09ab596bbf40ea111702f
round[ 9].m_col  e9f74eec023020f61bf2ccf2353c21c7
round[ 9].k_sch  549932d1f08557681093ed9cbe2c974e
round[10].start  bd6e7c3df2b5779e0b61216e8b10b689
round[10].s_box  7a9f102789d5f50b2beffd9f3dca4ea7
```

```
round[10].s_row  7ad5fda789ef4e272bca100b3d9ff59f
round[10].k_sch  13111d7fe3944a17f307a78b4d2b30c5
round[10].output 69c4e0d86a7b0430d8cdb78070b4c55a


INVERSE CIPHER (DECRYPT):

round[ 0].iinput  69c4e0d86a7b0430d8cdb78070b4c55a
round[ 0].ik_sch  13111d7fe3944a17f307a78b4d2b30c5
round[ 1].istart  7ad5fda789ef4e272bca100b3d9ff59f
round[ 1].is_row  7a9f102789d5f50b2beffd9f3dca4ea7
round[ 1].is_box  bd6e7c3df2b5779e0b61216e8b10b689
round[ 1].ik_sch  549932d1f08557681093ed9cbe2c974e
round[ 1].ik_add  e9f74eec023020f61bf2ccf2353c21c7
round[ 2].istart  54d990a16ba09ab596bbf40ea111702f
round[ 2].is_row  5411f4b56bd9700e96a0902fa1bb9aa1
round[ 2].is_box  fde3bad205e5d0d73547964ef1fe37f1
round[ 2].ik_sch  47438735a41c65b9e016baf4aebf7ad2
round[ 2].ik_add  baa03de7a1f9b56ed5512cba5f414d23
round[ 3].istart  3e1c22c0b6fcbf768da85067f6170495
round[ 3].is_row  3e175076b61c04678dfc2295f6a8bfc0
round[ 3].is_box  d1876c0f79c4300ab45594add66ff41f
round[ 3].ik_sch  14f9701ae35fe28c440adf4d4ea9c026
round[ 3].ik_add  c57e1c159a9bd286f05f4be098c63439
round[ 4].istart  b458124c68b68a014b99f82e5f15554c
round[ 4].is_row  b415f8016858552e4bb6124c5f998a4c
round[ 4].is_box  c62fe109f75eedc3cc79395d84f9cf5d
round[ 4].ik_sch  5e390f7df7a69296a7553dc10aa31f6b
round[ 4].ik_add  9816ee7400f87f556b2c049c8e5ad036
round[ 5].istart  e8dab6901477d4653ff7f5e2e747dd4f
round[ 5].is_row  e847f56514dadde23f77b64fe7f7d490
round[ 5].is_box  c81677bc9b7ac93b25027992b0261996
round[ 5].ik_sch  3caaa3e8a99f9deb50f3af57adf622aa
round[ 5].ik_add  f4bcd45432e554d075f1d6c51dd03b3c
round[ 6].istart  36339d50f9b539269f2c092dc4406d23
round[ 6].is_row  36400926f9336d2d9fb59d23c42c3950
round[ 6].is_box  247240236966b3fa6ed2753288425b6c
round[ 6].ik_sch  47f7f7bc95353e03f96c32bcfd058dfd
round[ 6].ik_add  6385b79ffc538df997be478e7547d691
round[ 7].istart  2d6d7ef03f33e334093602dd5bfb12c7
round[ 7].is_row  2dfb02343f6d12dd09337ec75b36e3f0
round[ 7].is_box  fa636a2825b339c940668a3157244d17
round[ 7].ik_sch  b6ff744ed2c2c9bf6c590cbf0469bf41
round[ 7].ik_add  4c9c1e66f771f0762c3f868e534df256
round[ 8].istart  3bd92268fc74fb735767cbe0c0590e2d
round[ 8].is_row  3b59cb73fcd90ee05774222dc067fb68
round[ 8].is_box  4915598f55e5d7a0daca94fa1f0a63f7
round[ 8].ik_sch  b692cf0b643dbdf1be9bc5006830b3fe
round[ 8].ik_add  ff87968431d86a51645151fa773ad009
```

```
round[ 9].istart  a7be1a6997ad739bd8c9ca451f618b61
round[ 9].is_row  a761ca9b97be8b45d8ad1a611fc97369
round[ 9].is_box  89d810e8855ace682d1843d8cb128fe4
round[ 9].ik_sch  d6aa74fdd2af72fadaa678f1d6ab76fe
round[ 9].ik_add  5f72641557f5bc92f7be3b291db9f91a
round[10].istart  6353e08c0960e104cd70b751bacad0e7
round[10].is_row  63cab7040953d051cd60e0e7ba70e18c
round[10].is_box  00102030405060708090a0b0c0d0e0f0
round[10].ik_sch  000102030405060708090a0b0c0d0e0f
round[10].ioutput 00112233445566778899aabbccddeeff

EQUIVALENT INVERSE CIPHER (DECRYPT):

round[ 0].iinput  69c4e0d86a7b0430d8cdb78070b4c55a
round[ 0].ik_sch  13111d7fe3944a17f307a78b4d2b30c5
round[ 1].istart  7ad5fda789ef4e272bca100b3d9ff59f
round[ 1].is_box  bdb52189f261b63d0b107c9e8b6e776e
round[ 1].is_row  bd6e7c3df2b5779e0b61216e8b10b689
round[ 1].im_col  4773b91ff72f354361cb018ea1e6cf2c
round[ 1].ik_sch  13aa29be9c8faff6f770f58000f7bf03
round[ 2].istart  54d990a16ba09ab596bbf40ea111702f
round[ 2].is_box  fde596f1054737d235febad7f1e3d04e
round[ 2].is_row  fde3bad205e5d0d73547964ef1fe37f1
round[ 2].im_col  2d7e86a339d9393ee6570a1101904e16
round[ 2].ik_sch  1362a4638f2586486bff5a76f7874a83
round[ 3].istart  3e1c22c0b6fcbf768da85067f6170495
round[ 3].is_box  d1c4941f7955f40fb46f6c0ad68730ad
round[ 3].is_row  d1876c0f79c4300ab45594add66ff41f
round[ 3].im_col  39daee38f4f1a82aaf432410c36d45b9
round[ 3].ik_sch  8d82fc749c47222be4dadc3e9c7810f5
round[ 4].istart  b458124c68b68a014b99f82e5f15554c
round[ 4].is_box  c65e395df779cf09ccf9e1c3842fed5d
round[ 4].is_row  c62fe109f75eedc3cc79395d84f9cf5d
round[ 4].im_col  9a39bf1d05b20a3a476a0bf79fe51184
round[ 4].ik_sch  72e3098d11c5de5f789dfe1578a2cccb
round[ 5].istart  e8dab6901477d4653ff7f5e2e747dd4f
round[ 5].is_box  c87a79969b0219bc2526773bb016c992
round[ 5].is_row  c81677bc9b7ac93b25027992b0261996
round[ 5].im_col  18f78d779a93eef4f6742967c47f5ffd
round[ 5].ik_sch  2ec410276326d7d26958204a003f32de
round[ 6].istart  36339d50f9b539269f2c092dc4406d23
round[ 6].is_box  2466756c69d25b236e4240fa8872b332
round[ 6].is_row  247240236966b3fa6ed2753288425b6c
round[ 6].im_col  85cf8bf472d124c10348f545329c0053
round[ 6].ik_sch  a8a2f5044de2c7f50a7ef79869671294
round[ 7].istart  2d6d7ef03f33e334093602dd5bfb12c7
round[ 7].is_box  fab38a1725664d2840246ac957633931
round[ 7].is_row  fa636a2825b339c940668a3157244d17
round[ 7].im_col  fc1fc1f91934c98210fbfb8da340eb21
```

```
round[ 7].ik_sch  c7c6e391e54032f1479c306d6319e50c
round[ 8].istart  3bd92268fc74fb735767cbe0c0590e2d
round[ 8].is_box  49e594f755ca638fda0a59a01f15d7fa
round[ 8].is_row  4915598f55e5d7a0daca94fa1f0a63f7
round[ 8].im_col  076518f0b52ba2fb7a15c8d93be45e00
round[ 8].ik_sch  a0db02992286d160a2dc029c2485d561
round[ 9].istart  a7be1a6997ad739bd8c9ca451f618b61
round[ 9].is_box  895a43e485188fe82d121068cbd8ced8
round[ 9].is_row  89d810e8855ace682d1843d8cb128fe4
round[ 9].im_col  ef053f7c8b3d32fd4d2a64ad3c93071a
round[ 9].ik_sch  8c56dff0825dd3f9805ad3fc8659d7fd
round[10].istart  6353e08c0960e104cd70b751bacad0e7
round[10].is_box  0050a0f04090e03080d02070c01060b0
round[10].is_row  0010203040506070809 0a0b0c0d0e0f0
round[10].ik_sch  000102030405060708090a0b0c0d0e0f
round[10].ioutput 00112233445566778899aabbccddeeff
```

5.7 Κρυπτοσυστήματα Ροής

5.7.1 Εισαγωγή

Τα *κρυπτοσυστηματα ροής (stream ciphers)* είναι μία σημαντική κατηγορία κρυπτογραφικών αλγορίθμων. Λειτουργούν κρυπτογραφώντας μεμονωμένα δυαδικά ψηφία του μηνύματος. Είναι αρκετά γρήγορα στην εκτέλεση και μπορούν να υλοποιηθούν με σχετικά απλό υλικό. Τα κρυπτοσυστηματα ροής *έχουν μνήμη*, με την έννοια ότι το αποτέλεσμα της κρυπτογράφησης ενός δυαδικού ψηφίου, μπορεί να εξαρτάται από την κρυπτογράφηση των προηγουμένων δυαδικών ψηφίων.

Βασική πηγή έμπνευσης για τα κρυπτοσυστηματα ροής αποτελεί το one-time pad (κρυπτόγραμμα Vernam) (βλ. 1.2.2). Όπως είδαμε στην ενότητα 1.4.3, η κρυπτογράφηση είναι απολύτως ασφαλής χωρίς όρους, εφόσον το μέγεθους του κλειδιού είναι τουλάχιστον ίσο με το μέγεθος του μηνύματος, με δεδομένο βέβαια ότι τα bits του κλειδιού της κρυπτογράφησης επιλέγονται τυχαία και ανεξάρτητα. Με αυτή την προυπόθεση το one-time pad, παρέχει απόλυτη ασφάλεια.

Δυστυχώς όμως η παραπάνω απαίτηση για το μέγεθος του κλειδιού, καθιστά ένα τέτοιο σύστημα μη εφαρμόσιμο στην πράξη, λόγω δυσκολιών στην δημιουργία, διαχείριση και διανομή των κρυπτογραφικών κλειδιών. Τα κρυπτοσυστήματα ροής προσπαθούν να παρακάμψουν αυτή την δυσχέρεια με τον εξής τρόπο: Χρησιμοποιούν ένα κλειδί το οποίο παράγεται με τυχαιο τρόπο και έχει πολύ μικρό μέγεθος, ανεξάρτητο από το μήνυμα προς κρυπτογράφηση. Προσπαθούν όμως να το επεκτείνουν ώστε να έχει το μέγεθος του μηνύματος. Φυσικά, η επέκταση αυτή παράγει μια ακολουθία bits (κλειδοροή) η οποία δεν είναι πραγματικά τυχαία. Στόχος των διαφόρων αλγορίθμων που χρησιμοποιούνται είναι η κλειδοροή

να φαίνεται τυχαία σε έναν αντίπαλο με περιορισμένους υπολογιστικούς πόρους. Κατά συνέπεια, τα κρυπτοσυστήματα αυτά δεν προσφέρουν απόλυτη ασφάλεια, αλλά μόνο υπολογιστική. Οι ιδιότητες της ακολουθίας που χρησιμοποιείται ως κλειδοροή αποτελούν κρίσιμο παράγοντα για την ασφάλεια ενός συστήματος της κατηγορίας stream ciphers. Αντικείμενο αυτού του κεφαλαίου αποτελεί η μελέτη κρυπτογραφικών ιδιοτήτων των ακολουθιών, καθώς επίσης και των συστημάτων παραγωγής αυτών. Έμφαση θα δοθεί στις ακολουθίες που παράγονται από καταχωρητές ολίσθησης των οποίων η συνάρτηση ανάδρασης είναι γραμμική: οι γραμμικοί καταχωρητές έχουν μελετηθεί σε μεγάλο βαθμό στη βιβλιογραφία, αφενός λόγω των πολύ καλών μαθηματικών ιδιοτήτων που τους διέπουν και, αφετέρου, λόγω της ευκολίας υλοποίησής τους.

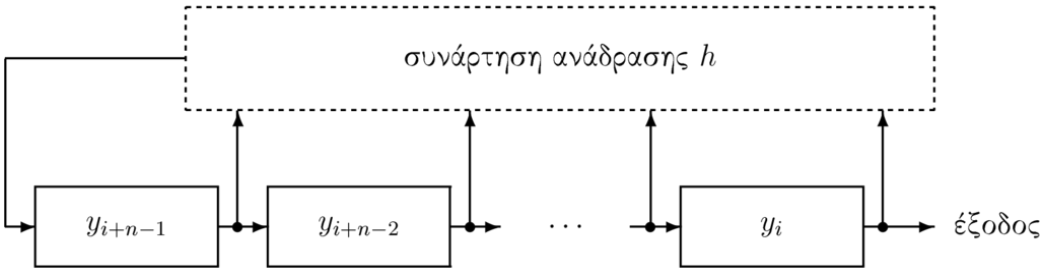
Εισάγεται επίσης η έννοια της (γραμμικής/μη γραμμικής) πολυπλοκότητας ως σημαντικό κρυπτογραφικό κριτήριο για μία ακολουθία και περιγράφονται τεχνικές οι οποίες, κάνοντας χρήση μη γραμμικών λογικών (boolean) συναρτήσεων, οδηγούν στη δημιουργία ακολουθιών μεγάλης γραμμικής πολυπλοκότητας. Αυτές οι τεχνικές στηρίζονται σε μη γραμμικά φίλτρα (ενότητα 5.7.6) και σε μη γραμμικούς συνδυαστές (ενότητα 5.7.7), τα οποία εφαρμόζονται κατάλληλα σε γραμμικούς καταχωρητές. Τέλος, περιγράφονται κάποιες από τις βασικές επιθέσεις κρυπτανάλυσης που έχουν εφαρμοστεί σε συστήματα αυτής της μορφής. Κάθε τέτοια επίθεση έχει ως αποτέλεσμα το να καθορίζονται συγκεκριμένες ιδιότητες που πρέπει να πληρούνται από τις λογικές συναρτήσεις, έτσι ώστε τα συστήματα να είναι ανθεκτικά σε αυτές.

5.7.2 Καταχωρητές ολίσθησης με ανάδραση (FSR)

Ας θεωρήσουμε το πεπερασμένο σώμα $\mathbb{F}_q$ που αποτελείται από q στοιχεία.

Ορισμός 5.10. Μία ακολουθία $y = \{y_i\}_{i\geq 0}$ με στοιχεία στο σώμα $\mathbb{F}_q$ ονομάζεται τελικά περιοδική (ultimately periodic) εάν υπάρχουν ακέραιοι $T > 0$ και $t_0 \geq 0$ τέτοιοι ώστε $y_{i+T} = y_i \; \forall \; i \geq t_0$. Ο μικρότερος ακέραιος T με την παραπάνω ιδιότητα ονομάζεται πρωταρχική περίοδος (fundamental period) της ακολουθίας y ή απλά περίοδος, και ο ακέραιος t_0 εκφράζει την προ-περίοδο (preperiod) της y. Αν $t_0 = 0$, τότε η ακολουθία y καλείται περιοδική (periodic).

Αν μία ακολουθία παίρνει τιμές στο $\mathbb{F}_2 = \{0, 1\}$, τότε καλείται *δυαδική ακολουθία* - αυτή είναι και η συνηθέστερη περίπτωση για κρυπτογραφικές εφαρμογές. Όταν αναφερόμαστε σε μία ακολουθία πεπερασμένου μήκους με N στοιχεία, θα τη συμβολίζουμε με y^N. Για κάθε τέτοια πεπερασμένη ακολουθία $y^N = y_0 y_1 \dots y_{N-1}$ και για κάθε $j \leq N$, συμβολίζουμε με y^j την υπακολουθία $y_0 y_1 \dots y_{j-1}$ που απαρτίζεται από τα πρώτα j στοιχεία της y. Κάθε τέτοια υπακολουθία (substring) καλείται



Σχήμα 5.26: Διάγραμμα ενός καταχωρητή ολίσθησης με ανάδραση (FSR)

πρόθεμα (prefix) της y^N. Στην περίπτωση όπου $j < N$, η υπακολουθία y^j καλείται *γνήσιο πρόθεμα (proper prefix)* της y^N. Ορίζουμε επίσης $y_i^j \triangleq y_i y_{i+1} \dots y_j$ για κάθε $i \leq j$ - συνεπώς, $y^j = y_0^{j-1}$. Αν $j = N-1$, τότε κάθε υπακολουθία y_i^j καλείται *επίθεμα (suffix)* της y^N. Αντίστοιχα, αν για ένα επίθεμα ισχύει $i > 0$, τότε ονομάζεται *γνήσιο επίθεμα (proper suffix)*.

Περιοδικές ή τελικά περιοδικές ακολουθίες παράγονται από *καταχωρητές ολίσθησης με ανάδραση* (FeedBack Shift Register (FSR)) (Σχήμα 5.26).

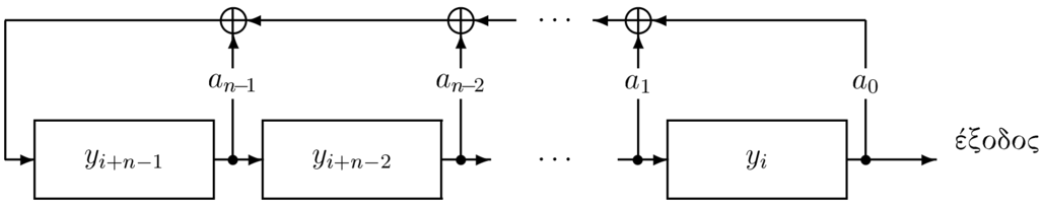
Ένας καταχωρητής μήκους n στο σώμα $\mathbb{F}_q$ αποτελείται από n θέσεις μνήμης ή βαθμίδες, κάθε μία εκ των οποίων μπορεί να περιέχει ένα στοιχείο του σώματος $\mathbb{F}_q$. Σε κάθε παλμό του ρολογιού, το περιεχόμενο της κάθε θέσης μνήμης μετατοπίζεται κατά μία θέση δεξιά, ενώ η τιμή της αριστερότερης βαθμίδας καθορίζεται από την πράξη ανάδρασης h. Συνεπώς, κάθε ακολουθία που παράγεται από έναν τέτοιο καταχωρητή ικανοποιεί την ακόλουθη αναδρομική σχέση

$$y_{i+n} = h(y_{i+n-1}, \dots, y_i), \qquad i \geq 0, \tag{5.4}$$

όπου η συνάρτηση $h : \mathbb{F}_q^n \to \mathbb{F}_q$ είναι στη γενική περίπτωση μη γραμμική - στις περισσότερες περιπτώσεις, ο σταθερός όρος της συνάρτησης h ισούται με 0.

Για κάθε χρονική στιγμή, τα περιεχόμενα των θέσεων μνήμης ορίζουν την *κατάσταση (state)* του FSR. Με άλλα λόγια, αν y είναι η παραγόμενη ακολουθία από έναν FSR, τότε η κατάστασή του για κάθε χρονική στιγμή $i \geq 0$ δίνεται από το διάνυσμα $(y_{i+n-1}\ y_{i+n-2}\ \dots\ y_i)$. Προφανώς, ένας FSR μήκους n μπορεί να περάσει από q^n διαφορετικές καταστάσεις. Συνεπώς, η μέγιστη περίοδος που μπορεί να έχει μία ακολουθία που παράγεται από έναν FSR n βαθμίδων είναι q^n. Μία περιοδική ακολουθία στο $\mathbb{F}_q$ περιόδου q^n, όπου η περίοδός της περιέχει όλες τις πιθανές n-άδες στο $\mathbb{F}_q$, ονομάζεται *ακολουθία De Brujin*. Οι ακολουθίες De Brujin παρουσιάζουν σημαντικό ερευνητικό ενδιαφέρον.

Αν περιοριστούμε στους FSR των οποίων η συνάρτηση ανάδρασης έχει μηδενικό σταθερό όρο, τότε η μέγιστη δυνατή περίοδος της ακολουθίας εξόδου είναι $q^n - 1$, λόγω του ότι αν ο FSR περάσει από τη μηδενική κατάσταση θα παραμείνει για



Σχήμα 5.27: Διάγραμμα ενός γραμμικού καταχωρητή ολίσθησης με ανάδραση (LFSR)

πάντα σε αυτή. Στο υπόλοιπο τμήμα αυτού του κεφαλαίου θα θεωρούμε ότι ο σταθερός όρος της συνάρτησης ανάδρασης είναι 0.

Για την ειδική περίπτωση των δυαδικών ακολουθιών, το διάγραμμα ενός καταχωρητή όπως αυτό του Σχήματος 5.26 αντικατοπτρίζει άμεσα και την υλοποίησή του σε επίπεδο λογικών πυλών - συγκεκριμένα, κάθε θέση μνήμης του καταχωρητή αντιστοιχεί σε ένα flip-flop, ενώ επιπλέον οι προσθέσεις και οι πολλαπλασιασμοί που υπεισέρχονται στη συνάρτηση ανάδρασης h υλοποιούνται ως κλασικοί αθροιστές και πολλαπλασιαστές αντίστοιχα σε επίπεδο bit (στη γενική περίπτωση όπου οι πράξεις γίνονται πάνω σε κάποιο πεπερασμένο σώμα $\mathbb{F}_q$, τότε οι προσθέσεις και οι πολλαπλασιασμοί πάνω στο σώμα έχουν πιο σύνθετη υλοποίηση). Επιπρόσθετα, όταν η παραγόμενη ακολουθία είναι δυαδική, τότε η συνάρτηση ανάδρασης h είναι μία *λογική συνάρτηση (Boolean function)* με n μεταβλητές.

5.7.3 Γραμμικοί καταχωρητές ολίσθησης με ανάδραση - Ακολουθίες μεγίστου μήκους

Γραμμικοί καταχωρητές ολίσθησης με ανάδραση (Linear FeedBack Shift Register (LFSR)) ονομάζονται εκείνοι οι καταχωρητές των οποίων η συνάρτηση ανάδρασης είναι γραμμική, δηλαδή της μορφής

$$y_{i+n} = a_{n-1}y_{i+n-1} + \cdots + a_1 y_{i+1} + a_0 y_i,\ a_j \in \mathbb{F}_q,\ \forall\, j = 1, 2, \ldots, n. \quad (5.5)$$

Ένας LFSR με n βαθμίδες απεικονίζεται στο Σχήμα 5.27. Βασικές ιδιότητες των συστημάτων αυτών έχουν μελετηθεί διεξοδικά στο κλασικό βιβλίο του Golomb [14], αλλά και στο κεφάλαιο 8 του βιβλίου των Lidl-Niederreiter [24]. Το *χαρακτηριστικό πολυώνυμο (characteristic polynomial)* ενός LFSR που δίνεται από το σχήμα 5.27 είναι το πολυώνυμο

$$f(x) = x^n - a_{n-1}x^{n-1} - \cdots - a_0 \in \mathbb{F}_q[x]. \quad (5.6)$$

Στη βιβλιογραφία συχνά χρησιμοποιείται, για την περιγραφή ενός LFSR όπως αυτού του σχήματος 5.27, το πολυώνυμο $f'(x) = 1 - a_{n-1}x - a_{n-2}x^2 - \ldots - a_0x^n$. Αυτό το πολυώνυμο, που επίσης προσδιορίζει μονοσήμαντα έναν LFSR, καλείται *πολυώνυμο ανάδρασης (feedback polynomial)*[16]. Αντίστροφα, αν για μία δοθείσα ακολουθία όλοι οι όροι της ικανοποιούν μία αναδρομική σχέση της μορφής (5.7.3), τότε το πολυώνυμο f που περιγράφεται στην (5.7.3) καλείται *χαρακτηριστικό πολυώνυμο της ακολουθίας*. Στη γενική περίπτωση, μία ακολουθία έχει πολλά χαρακτηριστικά πολυώνυμα (ισοδύναμα, υπάρχουν πολλοί διαφορετικοί LFSRs που παράγουν την ακολουθία).

Ένας LFSR παράγει περιοδική ακολουθία αν και μόνο αν ο σταθερός όρος a_0 του χαρακτηριστικού του πολυωνύμου είναι μη μηδενικός - δηλαδή, αν $f(0) \neq 0$. Στο υπόλοιπο της ενότητας περιοριζόμαστε σε αυτούς τους LFSR. Για οποιονδήποτε LFSR μήκους n με χαρακτηριστικό πολυώνυμο f, αν η αρχική του κατάσταση είναι η $100\ldots0$ τότε η παραγόμενη ακολουθία έχει τη μέγιστη δυνατή περίοδο που μπορεί να έχει οποιαδήποτε ακολουθία παράγεται από αυτόν τον LFSR. Η συγκεκριμένη περιοδική ακολουθία που προκύπτει καλείται *ακολουθία κρουστικής απόκρισης (impulse response sequence)*. Οποιαδήποτε άλλη ακολουθία παράγεται από αυτόν τον LFSR έχει περίοδο που είναι διαιρέτης της περιόδου της κρουστικής απόκρισης ή ίση με αυτή. Με τη σειρά της, η περίοδος της ακολουθίας κρουστικής απόκρισης του LFSR ισούται με την τάξη $\mathrm{ord}(f)$ του χαρακτηριστικού πολυωνύμου f - δηλαδή, είναι ίση με τον μικρότερο ακέραιο k τέτοιον ώστε το f να διαιρεί το $x^k - 1$. Αν το πολυώνυμο f είναι *μη αναγώγιμο (irreducible)* στο $\mathbb{F}_q$ και έχει βαθμό n, τότε $\mathrm{ord}(f) \mid q^n - 1$. Ισχύει $\mathrm{ord}(f) = q^n - 1$ αν και μόνο αν το f είναι *πρωταρχικό πολυώνυμο (primitive polynomial)* στο $\mathbb{F}_q$ [24, pp. 89].

Αν μία ακολουθία $y \in \mathbb{F}_q$ που περιγράφεται από την (5.7.3) έχει χαρακτηριστικό πολυώνυμο f του οποίου οι ρίζες $\alpha_1, \alpha_2, \ldots, \alpha_n$ είναι ανά δύο διαφορετικές μεταξύ τους (ή, ισοδύναμα, η πολλαπλότητα κάθε ρίζας του f είναι 1), τότε υπάρχουν στοιχεία $\beta_1, \beta_2, \ldots, \beta_n$, που εξαρτώνται από τις πρώτες n τιμές της ακολουθίας, τέτοια ώστε

$$y_i = \sum_{j=1}^{n} \beta_j \alpha_j^i, \qquad i = 0, 1, \ldots \tag{5.7}$$

Οι ακριβείς τιμές των $\beta_1, \ldots, \beta_n$ προσδιορίζονται από την επίλυση γραμμικού συστήματος [24]. Ουσιαστικά, καθορίζονται πλήρως από την αρχική κατάσταση εκείνου του LFSR με χαρακτηριστικό πολυώνυμο f που παράγει την ακολουθία. Μπορεί εύκολα να αποδειχτεί πως για την ειδική περίπτωση όπου το χαρακτηριστικό πολυώνυμο f του LFSR είναι μη αναγώγιμο με ρίζες $\alpha, \alpha^q, \alpha^{q^2}, \ldots, \alpha^{q^{n-1}}$, τότε οι συντελεστές $\beta_1, \ldots, \beta_n$ παίρνουν τέτοια μορφή ώστε η (5.7.3) να γράφεται ισοδύναμα ως

$$y_i = \beta\alpha^i + \beta^q\alpha^{qi} + \beta^{q^2}\alpha^{q^2 i} + \ldots + \beta^{q^{n-1}}\alpha^{q^{n-1}i}, \qquad i = 0, 1, \ldots \tag{5.8}$$

Έστω ακολουθία $y \in \mathbb{F}_q$ που ικανοποιεί τη σχέση (5.7.3) για κάθε χρονική στιγμή i. Τότε υπάρχει ένα μοναδικό *μονικό (monic)* πολυώνυμο $m(x) \in \mathbb{F}_q[x]$ (όπου ως μονικό ορίζεται ένα πολυώνυμο του οποίου ο συντελεστής του μεγιστοβάθμιου όρου είναι μονάδα) με την ακόλουθη ιδιότητα: ένα μονικό πολυώνυμο $f(x) \in \mathbb{F}_q[x]$ είναι χαρακτηριστικό πολυώνυμο της ακολουθίας y αν και μόνο αν το πολυώνυμο $m(x)$ διαιρεί το $f(x)$. Το πολυώνυμο $m(x)$ ονομάζεται *ελάχιστο πολυώνυμο (minimal polynomial)* της ακολουθίας y. Με άλλα λόγια, το ελάχιστο πολυώνυμο μίας ακολουθίας περιγράφει τη συνάρτηση ανάδρασης του LFSR με το μικρότερο μήκος που μπορεί να παράγει την ακολουθία. Η περίοδος μίας ακολουθίας με ελάχιστο πολυώνυμο $m(x)$ ισούται με $\text{ord}(m(x))$. Από την προηγούμενη ανάλυση γίνεται σαφές ότι ένας LFSR παράγει πάντα ακολουθίες με τη μέγιστη δυνατή περίοδο $q^n - 1$, ανεξαρτήτως της αρχικής του κατάστασης (πλην της μηδενικής), *αν και μόνο αν* το χαρακτηριστικό του πολυώνυμο είναι πρωταρχικό πολυώνυμο στο σώμα $\mathbb{F}_q$. Σε αυτήν την περίπτωση, το χαρακτηριστικό πολυώνυμο του LFSR είναι προφανώς και το ελάχιστο πολυώνυμο της παραγόμενης ακολουθίας. Αυτοί οι LFSR καλούνται *πρωταρχικοί LFSRs*, ενώ οι ακολουθίες που παράγονται από πρωταρχικούς LFSR ονομάζονται *ακολουθίες μεγίστου μήκους (maximal-length sequences ή m-sequences)*, λόγω της μέγιστης περιόδου που έχουν. Οι ακολουθίες μεγίστου μήκους ικανοποιούν πολλές σημαντικές ιδιότητες. Χαρακτηριστική είναι η *ιδιότητα ολίσθησης-πρόσθεσης (shift-and-add property)*: αν σε μία οποιαδήποτε ακολουθία μεγίστου μήκους y προσθέσουμε μία αυθαίρετη κυκλική ολίσθησή της, τότε η προκύπτουσα ακολουθία είναι επίσης κυκλική ολίσθηση της y. Άλλες σημαντικές ιδιότητες των ακολουθιών αυτών, με κρυπτογραφική χροιά, θα αναλυθούν στην Ενότητα 5.7.4.

Από τους παραπάνω ορισμούς γίνεται φανερό ότι υπάρχει άμεσος τρόπος κατασκευής δυαδικών ακολουθιών De Brujin μέσω των ακολουθιών μεγίστου μήκους. Αν y είναι μία δυαδική ακολουθία μεγίστου μήκους με περίοδο $2^n - 1$, τότε όλες οι πιθανές n-άδες στο $\mathbb{F}_2$ εμφανίζονται σε μία περίοδο της ακολουθίας, πλην της n-άδας που αποτελείται μόνο από μηδενικά- συνεπώς, αν σε μία περίοδο της y εισάγουμε ένα 0 αμέσως μετά την εμφάνιση $n - 1$ διαδοχικών 0, τότε η νέα ακολουθία περιόδου 2^n που θα προκύψει θα είναι De Brujin. Ωστόσο, πρέπει να σημειωθεί ότι δεν παράγονται όλες οι δυαδικές ακολουθίες De Brujin με αυτόν τον τρόπο.

5.7.4 Πολυπλοκότητα ακολουθιών

Όπως είναι προφανές, η ψευδοτυχαιότητα μιας ακολουθίας είναι αναγκαία προϋπόθεση για την κατασκευή ενός ασφαλούς κρυπτογραφικού συστήματος. Το πότε μια ακολουθία μπορεί να χαρακτηριστεί ως ψευδοτυχαία δεν είναι εύκολο ερώτημα. Ένα αυτονόητο κριτήριο που πρέπει να ικανοποιεί μια ψευδοτυχαία ακολου-

θία είναι το να έχει μεγάλη περίοδο. Εκτός της μεγάλης περιόδου, ο Golomb στο [14] πρότεινε τα ακόλουθα τρία κριτήρια ψευδοτυχαιότητας περιοδικών *δυαδικών* ακολουθιών, τα οποία είναι γνωστά ως *κριτήρια ψευδοτυχαιότητας του Golomb (Golomb's randomness postulates)*:

1. Σε μία περίοδο της ακολουθίας, το πλήθος των 0 είναι ίσο ή διαφέρει κατά ένα με το πλήθος των 1. Αυτή η ιδιότητα ονομάζεται *ιδιότητα ισοβαρούς ακολουθίας (balance property)*.

2. Ως *διαδρομή (run)* σε μία δυαδική ακολουθία θεωρούμε ένα τμήμα της $\tilde{y} = xx\ldots x$ που αποτελείται από όμοια στοιχεία x ($x = 0$ ή 1), όπου όμως τόσο το στοιχείο που προηγείται του $\tilde{y}$ όσο και αυτό που έπεται του $\tilde{y}$ είναι διαφορετικά από το x. Το κριτήριο ψευδοτυχαιότητας έχει ως εξής: σε μία περίοδο, το $1/2$ των διαδρομών έχουν μήκος 1, το $1/4 = 1/2^2$ αυτών έχουν μήκος 2, το $1/8 = 1/2^3$ αυτών μήκος 3 κ.ο.κ. μέχρις ώτου το $1/2^k$ του πλήθους των διαδρομών είναι μικρότερο της μονάδας. Αυτή η ιδιότητα καλείται *ιδιότητα διαδρομής (run property)*.

3. Η συνάρτηση αυτοσυσχέτισης $c(\tau) = \sum_{i=0}^{N-1}(-1)^{y_i+y_{i+\tau}}$ της δυαδικής ακολουθίας y, όπου N η περιόδός της, παίρνει δύο τιμές, συγκεκριμένα $c(\tau) = \begin{cases} N, & \tau \equiv 0 (\mathrm{mod}\ N) \\ K, & \tau \not\equiv 0 (\mathrm{mod}\ N) \end{cases}$ όπου K σταθερός ακέραιος. Αυτή η ιδιότητα καλείται *ιδιότητα αυτοσυσχέτισης (autocorrelation property)*.

Οι ακολουθίες μεγίστου μήκους ικανοποιούν και τα τρία κριτήρια ψευδοτυχαιότητας του Golomb [14]. Το γεγονός αυτό, σε συνδυασμό με το ότι οι ακολουθίες μεγίστου μήκους έχουν τη μέγιστη δυνατή περίοδο ως προς το μήκος του καταχωρητή από τον οποίο παράγονται, τις κατέστησε βασικές κρυπτογραφικές ακολουθίες στα πρώτα χρόνια εμφάνισης κρυπτογραφικών εφαρμογών. Μία σημαντική αδυναμία τους όμως είχε σαν αποτέλεσμα να χαρακτηριστούν, αν και διαθέτουν πολύ καλές ιδιότητες, κρυπτογραφικά ακατάλληλες: η αδυναμία τους αυτή είναι η χαμηλή *γραμμική πολυπλοκότητα* που έχουν, η οποία ορίζεται στη συνέχεια.

Ορισμός 5.11. Μη γραμμική πολυπλοκότητα (nonlinear complexity ή nonlinear span) ή απλά πολυπλοκότητα μιας ακολουθίας ορίζεται ως το μήκος του μικρότερου FSR ο οποίος παράγει την ακολουθία. Κάθε FSR ο οποίος παράγει μία ακολουθία και έχει μήκος όσο η πολυπλοκότητά της καλείται ελάχιστος FSR της ακολουθίας. Αντίστοιχα, γραμμική πολυπλοκότητα μιας ακολουθίας (linear complexity ή linear span) ορίζεται ως το μήκος του μικρότερου LFSR ο οποίος παράγει την ακολουθία. Ο ελάχιστος LFSR για μία ακολουθία ορίζεται με όμοιο τρόπο.

Από τον παραπάνω ορισμό γίνεται προφανές ότι αν τα $c(y), \text{lc}(y)$ υποδηλώνουν αντίστοιχα τη μη γραμμική και τη γραμμική πολυπλοκότητα της ακολουθίας y, τότε $c(y) \leq \text{lc}(y)$.

Ορισμός 5.12. Για μία ακολουθία $y^N = y_0 y_1 \dots y_{N-1}$ πεπερασμένου μήκους, το προφίλ πολυπλοκότητας (complexity profile) ορίζεται ως η ακολουθία των ακεραίων αριθμών

$$c(y^1), c(y^2) \dots, c(y^N).$$

Με ανάλογο τρόπο ορίζεται το προφίλ γραμμικής πολυπλοκότητας (linear complexity profile).

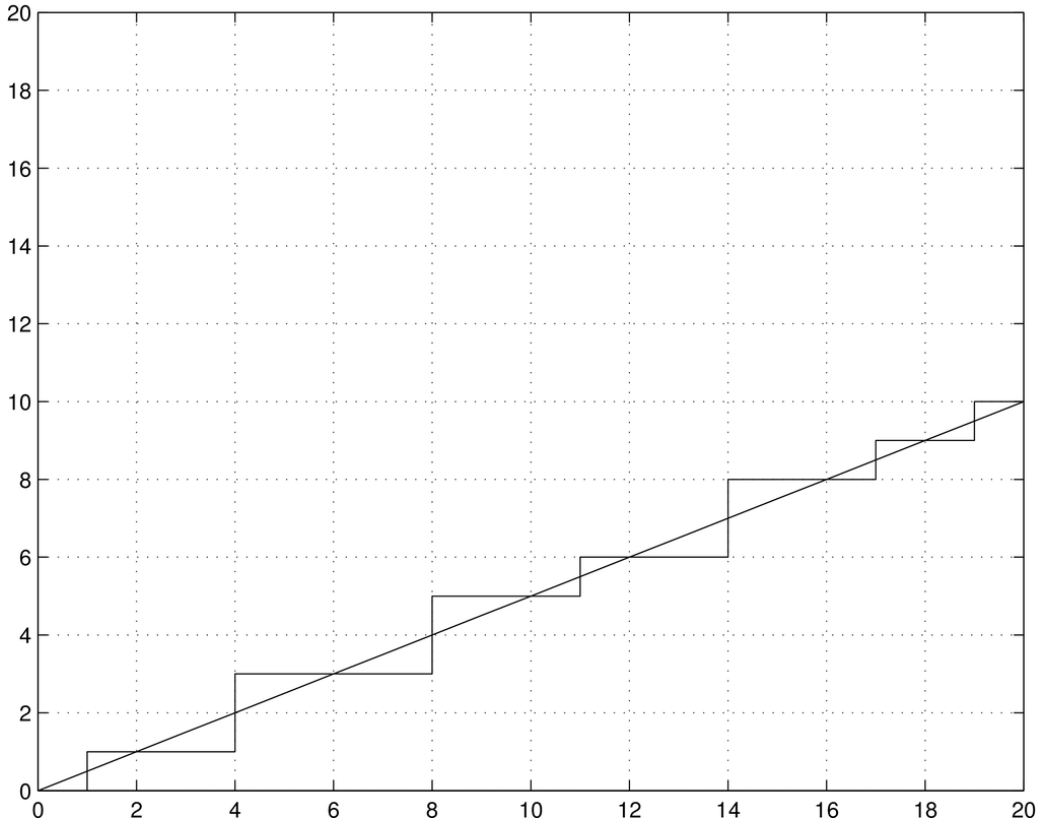
Γραμμική πολυπλοκότητα

Η γραμμική πολυπλοκότητα ακολουθιών έχει μελετηθεί σε πολύ μεγάλο βαθμό στη βιβλιογραφία [6, 15, 21, 18, 26, 27, 28, 40, 41]. Σημαντικές ιδιότητες που χαρακτηρίζουν το προφίλ γραμμικής πολυπλοκότητας ακολουθιών πεπερασμένου μήκους είναι οι ακόλουθες:

1. Αν $j > i$, τότε $\text{lc}(y^j) \geq \text{lc}(y^i)$.
2. Αν $\text{lc}(y^{i+1}) > \text{lc}(y^i)$, τότε $\text{lc}(y^i) \leq \frac{i}{2}$.
3. Αν $\text{lc}(y^{i+1}) > \text{lc}(y^i)$, τότε $\text{lc}(y^{i+1}) + \text{lc}(y^i) = i + 1$.

Από τις παραπάνω ιδιότητες προκύπτει πως, οποτεδήποτε έχουμε αύξηση στην τιμή της γραμμικής πολυπλοκότητας μίας ακολουθίας, τότε η νέα τιμή της γραμμικής πολυπλοκότητας είναι συμμετρική της παλιάς ως προς τη γραμμή $f(N) = \frac{N}{2}$, $N = 1, 2, \dots$. Αυτό αναδεικνύεται στο Σχήμα 5.28 όπου απεικονίζεται το προφίλ γραμμικής πολυπλοκότητας της ακολουθίας πεπερασμένου μήκους $y^{20} = 10010011110001001110$. Η τιμή N/2 ισούται επίσης (κατά προσέγγιση) με την αναμενόμενη τιμή $E(\text{lc}(y^N))$ της γραμμικής πολυπλοκότητας για μια τυχαία ακολουθία y^N, όπως αυτή έχει υπολογιστεί από τον Rueppel στο [40]: με άλλα λόγια, ισχύει $E(\text{lc}(y^N)) \approx \frac{N}{2}$ για μεγάλες τιμές του N. Επίσης, η διασπορά $\text{Var}(\text{lc}(y^N))$ ικανοποιεί τη σχέση $\text{Var}(\text{lc}(y^N)) \approx \frac{86}{81}$ [40].

Η εύρεση του ελάχιστου LFSR που παράγει μία ακολουθία πραγματοποιείται με τον *αλγόριθμο Berlekamp-Massey* (BMA) [2, 26] (σχήμα 5.29). Είναι ένας διαδεδομένος αλγόριθμος που πρωτοεισήχθη για την αποκωδικοποίηση BCH κωδίκων από τον Berlekamp το 1967 [2], ενώ δύο χρόνια αργότερα ο Massey χρησιμοποίησε τον ίδιο αλγόριθμο για την εύρεση του ελάχιστου LFSR που απαιτείται για την παραγωγή μίας ακολουθίας y^N [26]. Ο αλγόριθμος είναι αναδρομικός και



Σχήμα 5.28: Το προφίλ της γραμμικής πολυπλοκότητας για τη δυαδική ακολουθία $y^{20} = 10010011110001001110$

υπολογίζει το πολυώνυμο ανάδρασης του ελάχιστου LFSR για κάθε υπακολουθία y^i, $1 \leq i \leq N$. Για κάθε χρονική στιγμή n, $0 \leq n < N$, γίνεται έλεγχος αν ο τρέχων ελάχιστος LFSR της υπακολουθίας y^n παράγει την υπακολουθία y^{n+1}. Αν ναι, τότε προφανώς ο τρέχων LFSR είναι ο ελάχιστος για την υπακολουθία y^{n+1} και μένει αμετάβλητος (η περίπτωση όπου $d = 0$): διαφορετικά, γίνεται έλεγχος για το αν η γραμμική πολυπλοκότητα της y^{n+1} είναι μεγαλύτερη ή ίση της αντίστοιχης για την y^n. Αν η γραμμική πολυπλοκότητα παραμένει ίση (ισοδύναμα, $2\,\mathrm{lc}(y^n) > n$) ή αυξάνει ($2\,\mathrm{lc}(y^n) \leq n$), τότε μία διορθωτική συνάρτηση προστίθεται στο πολυώνυμο ανάδρασης του ελάχιστου LFSR της y^n προκειμένου να προσδιοριστεί το πολυώνυμο ανάδρασης του ελάχιστου LFSR της y^{n+1}. Αυτή η διορθωτική συνάρτηση είναι πλήρως ορισμένη και εξαρτάται από το πολυώνυμο ανάδρασης του ελάχιστου LFSR της y^j, όπου $j < n$ η πιο πρόσφατη χρονική στιγμή κατά την οποία υπήρξε αύξηση στην γραμμική πολυπλοκότητα. Ο αλγόριθμος Berlekamp-Massey έχει ακόμα πιο απλή μορφή όταν εξετάζουμε

```
b ← 1
k ← 1
B(x) ← 1
n ← 0
L ← 0                                   % linear complexity
c(x) ← 1                                % feedback polynomial
while n < N do
    d ← y_n + Σ_{i=1}^{L} c_i y_{n−i}
    if d ≠ 0 then
        if 2L > N then                  % the linear complexity does not increase
        c(x) ← c(x) − db^{−1} x^k B(x)
        k ← k + 1
        elseif L ≤ n/2 then             % the linear complexity increases
            T(x) ← c(x)
            c(x) ← c(x) − db^{−1} x^k B(x)
            L ← n + 1 − L               % new value of complexity
            b ← d
            B(x) ← T(x)
            k ← 1
        endif
    else                                % d = 0, i.e. no change of LFSR
        x ← x + 1
    endif
    n ← n + 1
endwhile
```

Σχήμα 5.29: Ο αλγόριθμος Berlekamp-Massey

δυαδικές ακολουθίες, γιατί σε αυτήν την περίπτωση οι υπεισερχόμενες προσθέσεις πάνω στο σώμα είναι πράξεις XOR. Για μία δυαδική ακολουθία μήκους N, ο BMA έχει υπολογιστική πολυπλοκότητα $O(N^2)$. Μία ισοδύναμη αλλά διαφορετική περιγραφή του BMA, βασισμένη σε ιδιότητες πινάκων που προκύπτουν από την επίλυση ενός κατάλληλου γραμμικού συστήματος, παρουσιάζεται στο [18]. Επίσης, η ισοδυναμία του BMA με τον κλασικό αλγόριθμο του Ευκλείδη που χρησιμοποιείται για αποκωδικοποίηση BCH κωδίκων αποσαφηνίζεται στο [17].

Μία σημαντική ιδιότητα του προφίλ γραμμικής πολυπλοκότητας, που καθιστά τον αλγόριθμο Berlekamp-Massey ισχυρό εργαλείο κρυπτανάλυσης, είναι η ακόλουθη: ο *ελάχιστος LFSR μίας ακολουθίας* y^N *είναι μοναδικός αν και μόνο αν* $\mathrm{lc}(y^N) \leq \frac{N}{2}$. Με άλλα λόγια, *αν η γραμμική πολυπλοκότητα μίας ακολουθίας είναι* L*, τότε γνωρίζοντας μόνο* $2L$ *διαδοχικά στοιχεία της ακολουθίας μας επιτρέπουν, μέσω του BMA, να υπολογίσουμε ολόκληρη την ακολουθία*. Συνεπώς, ο Berlekamp-Massey αλγόριθμος κατέστησε τη γραμμική πολυπλοκότητα ως σημαντικό κρυπτογραφικό κριτήριο: *μία ακολουθία που χρησιμοποιείται ως κλειδοροή σε έναν*

αλγόριθμο ροής πρέπει να έχει υψηλή γραμμική πολυπλοκότητα. Γίνεται φανερό λοιπόν πια ότι οι ακολουθίες μεγίστου μήκους, παρόλο που πληρούν τα κριτήρια ψευδοτυχαιότητας του Golomb, είναι ακατάλληλες για κρυπτογραφικές εφαρμογές: μία ακολουθία μεγίστου μήκους με περίοδο $2^N - 1$ έχει, προφανώς, γραμμική πολυπλοκότητα ίση με N και, συνεπώς, γνώση μόνο $2N$ διαδοχικών bits της ακολουθίας επιτρέπει τον πλήρη προσδιορισμό όλης της ακολουθίας!

Για περιοδικές ακολουθίες υπάρχουν επίσης πολλά σημαντικά αποτελέσματα που σχετίζονται με τη γραμμική πολυπλοκότητα. Από την ανάλυση της Ενότητας 5.7.3 προκύπτει άμεσα ότι *για περιοδική ακολουθία, η γραμμική της πολυπλοκότητα ισούται με το βαθμό του ελάχιστου πολυωνύμου της*. Αξίζει τέλος να αναφερθεί πως η γραμμική πολυπλοκότητα περιοδικών δυαδικών ακολουθιών, με περίοδο $N = 2^n$ για κάποιον θετικό ακέραιο n, υπολογίζεται πολύ αποδοτικά από τον αλγόριθμο των Games-Chan [11]. Ο αλγόριθμος αυτός είναι γραμμικός ως προς την περίοδο της ακολουθίας. Μειονέκτημά του, εκτός του ότι μπορεί να εφαρμοστεί μόνο σε ακολουθίες συγκεκριμένης περιόδου, είναι το ότι ολόκληρη η περίοδος της ακολουθίας πρέπει να είναι εκ των προτέρων γνωστή: δεν έχει δηλαδή τη "βήμα-προς-βήμα" δομή του BMA. Επίσης, ο αλγόριθμος των Games-Chan δεν είναι κατάλληλος για ακολουθίες με πολύ μεγάλες περιόδους λόγω του ότι απαιτεί την αποθήκευση ολόκληρης της περιόδου προκειμένου να αρχίσει την επεξεργασία της. Ωστόσο παραμένει πολύ σημαντικός αλγόριθμος επειδή αναδεικνύει ξεχωριστές ιδιότητες που διέπουν τις ακολουθίες με περίοδο κάποια δύναμη του 2. Τέτοιες ακολουθίες έχουν προταθεί για εφαρμογή σε κρυπτογραφικές εφαρμογές: για παράδειγμα, οι T-συναρτήσεις (T-functions) [22], κρυπτογραφικές αδυναμίες των οποίων έχουν μελετηθεί στο [23], εμπίπτουν σε αυτήν την περίπτωση. Γίνεται φανερό λοιπόν από τα παραπάνω ότι η κρυπτογραφική πληροφορία που αναδεικνύεται από το φάσμα μίας ακολουθίας είναι η γραμμική της πολυπλοκότητα.

Μη γραμμική πολυπλοκότητα

Η μη γραμμική πολυπλοκότητα έχει μελετηθεί σε σημαντικά μικρότερο βαθμό στη βιβλιογραφία από ό,τι η γραμμική. Η τιμή της $c(y)$ για μία ακολουθία y υπολογίζεται από την ακόλουθη Πρόταση [20]:

Πρόταση 5.13. *Για μία ακολουθία y, έστω L ο μεγαλύτερος ακέραιος αριθμός που ικανοποιεί την ακόλουθη ιδιότητα: υπάρχουν $0 \leq i < j \leq N - 1 - L$ τέτοια ώστε $y_i^{i+L-1} = y_j^{j+L-1}$ και $y_{i+L} \neq y_{j+L}$. Τότε, ισχύει $c(y) = L + 1$.*

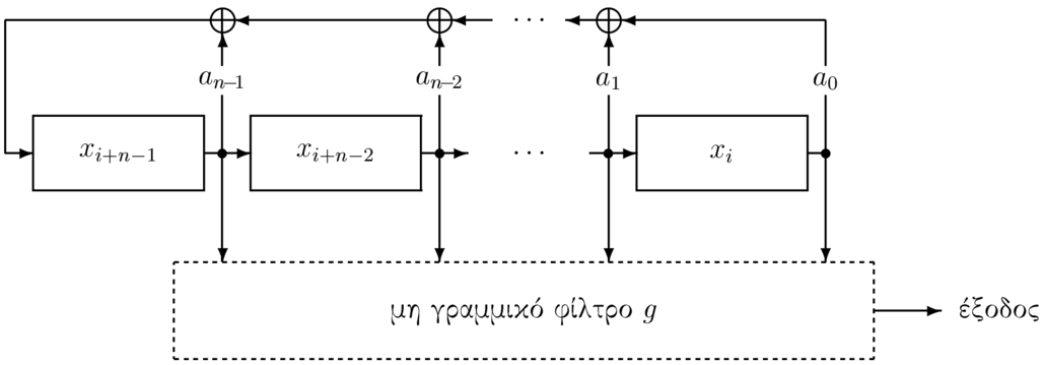
Κατά αναλογία με το ελάχιστο πολυώνυμο μίας ακολουθίας, ορίζουμε ως *ελάχιστο μη γραμμικό πολυώνυμο (nonlinear minimal polynomial)* μίας ακολουθίας y τη συνάρτηση ανάδρασης οποιουδήποτε FSR μήκους $m = c(y)$ που παράγει την

y. Στο [19] αποδεικνύεται πως, για μία ακολουθία y με τιμές σε οποιοδήποτε πεπερασμένο σώμα, ένας κατευθυνόμενος ακυκλικός γράφος λέξεων (directed acyclic word graph - DAWG) μπορεί να χρησιμοποιηθεί έτσι ώστε να προσδιοριστεί το προφίλ μη γραμμικής πολυπλοκότητας της y- οι κόμβοι-λέξεις του γράφου είναι κατάλληλα επιλεγμένες υπακολουθίες της y. Στο [9] δίνεται μία κατανομή (κατά προσέγγιση) της τιμής της μη γραμμικής πολυπλοκότητας για τυχαίες δυαδικές ακολουθίες, έτσι ώστε να μπορεί να χρησιμοποιηθεί ως μέτρο εκτίμησης της ψευδοτυχαιότητάς τους. Στο [38] παρουσιάζεται μία αλγοριθμική τεχνική για τον υπολογισμό ενός ελάχιστου μη γραμμικού FSR που παράγει μία οποιαδήποτε ακολουθία στο $\mathbb{F}_2$. Η τεχνική αυτή βασίζεται σε ιδιότητες που ενυπάρχουν στον πίνακα του ισοδύναμου γραμμικού συστήματος εξισώσεων, η λύση του οποίου προσδιορίζει τη συνάρτηση ανάδρασης του ελάχιστου FSR. Στο [39] μελετάται η ειδική περίπτωση όπου η συνάρτηση ανάδρασης του FSR είναι πολυώνυμο βαθμού το πολύ 2 (δηλαδή, σε κάθε πολλαπλασιαστή υπεισέρχονται το πολύ 2 βαθμίδες) και παρουσιάζεται αλγόριθμος για τον υπολογισμό του ελάχιστου FSR αυτής της μορφής ο οποίος παράγει δοθείσα δυαδική ακολουθία. Τέλος, στο [37] παρουσιάζεται μέθοδος με την οποία, για κάθε επιθυμητή τιμή γραμμικής πολυπλοκότητας, κατασκευάζονται ακολουθίες με τη μέγιστη δυνατή τιμή για τη μη γραμμική πολυπλοκότητα.

5.7.5 Παραγωγή ακολουθιών υψηλής πολυπλοκότητας

Από την προηγούμενη ενότητα κατέστη σαφές πως, λόγω του αλγορίθμου Berlekamp-Massey, υψηλή γραμμική πολυπλοκότητα είναι αναγκαία προϋπόθεση για κρυπτογραφικές ακολουθίες. Είδαμε επίσης πως LFSR με πρωταρχικό χαρακτηριστικό πολυώνυμο παράγουν ακολουθίες με τη μικρότερη δυνατή γραμμική πολυπλοκότητα - τις λεγόμενες ακολουθίες μεγίστου μήκους. Εν τούτοις, οι πολύ καλές λοιπές ιδιότητες αυτών των LFSR, με κύρια τη μεγάλη περίοδο των παραγομένων ακολουθιών, είχαν σαν αποτέλεσμα να μην τεθούν στο περιθώριο. Η πλειοψηφία των αλγορίθμων ροής εξακολουθούν να χρησιμοποιούν LFSR ως δομικά συστατικά της γεννήτριας της κλειδοροής (LFSR-based stream ciphers). Προκειμένου όμως να αυξηθεί η γραμμική πολυπλοκότητα των παραγομένων ακολουθιών, υπεισέρχονται και μη γραμμικές πράξεις οι οποίες επιδρούν με κάποιο τρόπο στην κανονική γραμμική λειτουργία των LFSR. Αυτές οι μη γραμμικές πράξεις κατηγοριοποιούνται ως εξής [32]:

- *Μη γραμμικά φίλτρα (nonlinear filter generators)*. Σε αυτήν την περίπτωση, μία μη γραμμική λογική συνάρτηση δρα στις βαθμίδες ενός LFSR - η παραγόμενη κλειδοροή είναι πια η έξοδος αυτής της συνάρτησης.



Σχήμα 5.30: Εφαρμογή μη γραμμικού φίλτρου σε έναν LFSR

- *Μη γραμμικοί συνδυαστές (nonlinear combination generators)*. Σε αυτήν την περίπτωση, οι έξοδοι πολλών LFSR "τροφοδοτούν" μία μη γραμμική λογική συνάρτηση - η παραγόμενη κλειδοροή προκύπτει από την έξοδο αυτής της συνάρτησης.

- *Γεννήτριες ελεγχόμενες από ρολόι (Clock-controlled generators)*. Αυτή είναι η περίπτωση κατά την οποία ο χρονισμός ενός LFSR (δηλαδή το κάθε πότε αλλάζει κατάσταση) καθορίζεται από την έξοδο ενός άλλου LFSR.

Με άλλα λόγια, οι λογικές συναρτήσεις παίζουν σημαντικό ρόλο στην κατασκευή δυαδικών ακολουθιών μεγάλης γραμμικής πολυπλοκότητας. Μία συγκεντρωτική περιγραφή όλων των κρυπτογραφικών ιδιοτήτων λογικών συναρτήσεων, καθώς και όλων των σημαντικών κατασκευών συναρτήσεων που έχουν προταθεί στη βιβλιογραφία, παρουσιάζεται στο [31].

5.7.6 Μη γραμμικά φίλτρα

Μία τεχνική που οδηγεί σε δυαδικές ακολουθίες μεγάλης γραμμικής πολυπλοκότητας είναι η εφαρμογή μίας μη γραμμικής λογικής συνάρτησης στις βαθμίδες ενός πρωταρχικού LFSR, όπως απεικονίζεται στο Σχήμα 5.30. Η συνάρτηση g καλείται *μη γραμμικό φίλτρο (nonlinear filter function)*. Ιδανικά, το μη γραμμικό φίλτρο θα πρέπει να εξασφαλίζει ομοιόμορφη κατανομή των bits 0 ή 1 στην παραγόμενη κλειδοροή. Στη γενική περίπτωση, οι παραγόμενες ακολουθίες έχουν μεγάλη περίοδο και υψηλή γραμμική πολυπλοκότητα [15]. Ωστόσο, ανοιχτό ερευνητικό πρόβλημα παραμένει ο ακριβής προσδιορισμός της τιμής της γραμμικής πολυπλοκότητας των ακολουθιών που παράγονται από συστήματα αυτής της κατηγορίας. Ένα άνω φράγμα για τη γραμμική πολυπλοκότητα δίνεται από τον Key στο [21]:

συγκεκριμένα, αν n είναι το μήκος του LFSR και k ο βαθμός του μη γραμμικού φίλτρου, τότε η γραμμική πολυπλοκότητα της ακολουθίας που παράγεται είναι το πολύ ίση με $L_k = \sum_{i=1}^{k} \binom{n}{i}$.

Κρυπτανάλυση σε συστήματα μη γραμμικών φίλτρων

Διάφορες κρυπταναλυτικές τεχνικές έχουν αναπτυχθεί για την ανάλυση συστημάτων που βασίζονται σε μη γραμμικά φίλτρα. Στην πλειοψηφία των περιπτώσεων, τόσο το χαρακτηριστικό πολυώνυμο του LFSR όσο και το μη γραμμικό φίλτρο είναι δημοσίως γνωστά- η μυστικότητα της παραγόμενης ακολουθίας-κλειδιού έγκειται στο ότι μένει μυστική η αρχική κατάσταση του LFSR. Κατά συνέπεια, οι κρυπταναλυτικές τεχνικές αποσκοπούν στον προσδιορισμό αυτής της αρχικής κατάστασης.

Επιθέσεις συσχέτισης: Στο [43] αναλύεται μία μέθοδος κρυπτανάλυσης που βασίζεται σε ιδιότητες της συνάρτησης ετεροσυσχέτισης (cross - correlation function) ανάμεσα στην ακολουθία μεγίστου μήκους που παράγεται από τον LFSR και στην ακολουθία εξόδου του συστήματος.

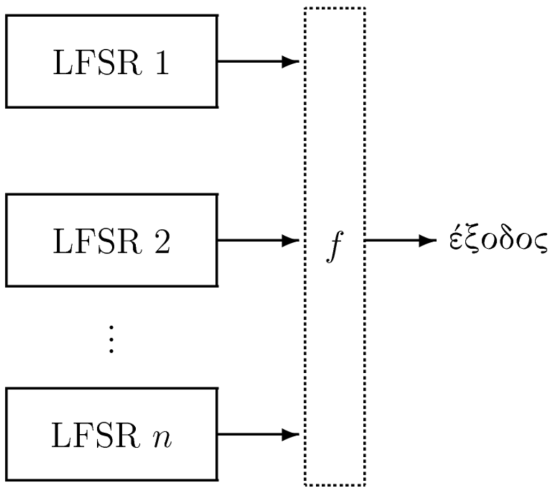
Επιθέσεις αναστροφής: Μία άλλη τεχνική ανάκτησης της αρχικής κατάστασης του LFSR είναι η *επίθεση αναστροφής (inversion attack)*, η οποία προτάθηκε στο [13]. Μία επίθεση αναστροφής μπορεί να εφαρμοστεί στις περιπτώσεις εκείνες όπου η συνάρτηση φίλτρου g γράφεται είτε στη μορφή

$$g(x_1, x_2, \ldots, x_n) = x_1 + h(x_2, x_3, \ldots, x_n)$$

είτε

$$g(x_1, x_2, \ldots, x_n) = x_n + h(x_1, x_2, \ldots, x_{n-1}),$$

όπου n το πλήθος των βαθμίδων του LFSR που υπεισέρχονται στην είσοδο του φίλτρου. Τότε, αν $u = \{u_i\}_{i \geq 0}$ είναι η ακολουθία μεγίστου μήκους που παράγεται από τον LFSR, η ακολουθία εξόδου y περιγράφεται (για την πρώτη περίπτωση) από σχέση της μορφής $y_i = u_{i+\gamma_1} + h(u_{i+\gamma_2}, \ldots, u_{i+\gamma_n})$, όπου $\gamma_1 > \gamma_2 > \ldots > \gamma_n$ θετικοί ακέραιοι που καθορίζονται από το μη γραμμικό φίλτρο. Άρα, αν το bit y_i είναι γνωστό, τότε προσδιορίζεται το bit $u_{i+\gamma_1}$ αν είναι γνωστά τα προηγούμενα $\gamma_n - \gamma_1$ bits $u_{i+\gamma_1+1}, \ldots, u_{i+\gamma_n}$. Κατά συνέπεια, η ποσότητα $\gamma_n - \gamma_1$ πρέπει να είναι μεγάλη - ιδανικά, ίση με $L - 1$, όπου L το μήκος του LFSR. Επίσης, η επίθεση αναστροφής γίνεται ακόμα πιο αποδοτική αν ο μέγιστος κοινός διαιρέτης d των $\gamma_{i+1} - \gamma_i$, $i = 1, 2, \ldots, n-1$ είναι μεγάλος- ιδανικά λοιπόν, θα θέλαμε το μη γραμμικό φίλτρο να ικανοποιεί τη σχέση $d = 1$. Η τεχνική αναστροφής γενικεύτηκε στο [12], έτσι ώστε να μπορεί να εφαρμοστεί σε όλα τα μη γραμμικά φίλτρα χωρίς κανέναν περιορισμό.



Σχήμα 5.31: Εφαρμογή μη γραμμικού συνδυαστή σε πολλούς LFSRs

Αλγεβρικές επιθέσεις: Αν L το μήκος του LFSR, τότε κάθε bit της ακολουθίας κλειδιού μπορεί να γραφεί ως μία συνάρτηση των L bits της αρχικής κατάστασης. Συνεπώς, γνώση N στοιχείων της κλειδοροής επιτρέπει τον προσδιορισμό της αρχικής κατάστασης του LFSR μέσω επίλυσης ενός μη γραμμικού συστήματος N εξισώσεων με L αγνώστους.

5.7.7 Μη γραμμικοί συνδυαστές

Μία δεύτερη τεχνική για την εξάλειψη της εγγενούς γραμμικότητας των LFSRs είναι ο συνδυασμός πολλών LFSR, με τρόπο τέτοιο ώστε οι έξοδοί τους να τροφοδοτούν μία μη γραμμική λογική συνάρτηση, που καλείται συνάρτηση-συνδυαστής (Σχήμα 5.31). Οι LFSRs που επιλέγονται για την κατασκευή τέτοιων συστημάτων είναι πρωταρχικοί, λόγω των καλών ιδιοτήτων και της υψηλής περιόδου που έχουν. Όμοια με την περίπτωση των μη γραμμικών φίλτρων, η συνάρτηση-συνδυαστής πρέπει να είναι ισοβαρής.

5.8 Ασκήσεις

1. Αποδείξτε ότι στο κρυπτοσύστημα DES ισχύει $E_K(M) = C$ αν και μόνον αν $E_{\overline{K}}(\overline{M}) = \overline{C}$.

2. Ένα κλειδί K λέγεται αδύναμο αν για κάθε κείμενο M ισχύει $E_K(M) = M$. Βρείτε τέσσερα αδύναμα κλειδιά του κρυπτοσυστήματος DES.

3. Θεωρήστε την παραλλαγή του DES-X,με 2 κλειδιά k_1, k_2 ,όπου η κρυπτογράφηση ενός απλού κειμένου M γίνεται ως εξής : $\texttt{Encrypt}(k_1, k_2)(M) = \texttt{Encrypt}(k1, (M \oplus k_2)$ όπου `Encrypt` η συνάρτηση κρυπτογράφησης του DES. Έχουμε περισσότερη ασφάλεια από τον κλασσικό DES στο παραπάνω σύστημα; Θεωρήστε ότι ο αντίπαλος έχει δυνατότητα KPA (διαθέτει μερικά ζεύγη απλού κειμένου - κρυπτοκειμένου).

4. Θεωρήστε την επέκταση του DES, με 2 κλειδιά K_1, K_2, όπου η κρυπτογράφηση ενός αρχικού κειμένου M γίνεται ως εξής (δηλαδή πρώτα εφαρμόζεται ο κλασικός DES με κλειδί K_1 και στο αποτέλεσμα εφαρμόζεται XOR με το κλειδί K_2):

$$DES'_{K_1,K_2}(M) = DES_{K_1}(M) \oplus K_2$$

i. Δείξτε ότι, παρ'όλο που υπάρχουν $2^{56} \cdot 2^{64}$ ζεύγη κλειδιών, αν διαθέτουμε μερικά ζεύγη αρχικού κειμένου / κρυπτοκειμένου μπορούμε να σπάσουμε το σύστημα με 2^{56} δοκιμές.

ii. Θα είχαμε περισσότερη ασφάλεια αν χρησιμοποιούσαμε την παρακάτω παραλλαγή;

$DES''_{K_1,K_2}(M) = DES_{K_1}(M \oplus K_2)$

5. Θεωρήστε το εξής κρυπτοσύστημα: για δοσμένο γνωστό πρώτο αριθμό p, επιλέγονται τυχαία δύο αριθμοί $a \in \mathbb{Z}_p^*, k \in \mathbb{Z}_p$. Η κρυπτογράφηση για ένα "κείμενο" $m \in \mathbb{Z}_p$ γίνεται ως εξής:

$$E_{(a,k)}(m) = am + k \mod p$$

i. Δείξτε ότι η κρυπτογράφηση είναι '1-1' και 'επί'.

ii. Αποδείξτε ότι για *μία* κρυπτογράφηση το σύστημα έχει την ιδιότητα της τέλειας μυστικότητας (perfect secrecy).

iii. Αποδείξτε ότι αν το ίδιο κλειδί χρησιμοποιηθεί για να κρυπτογραφήσει δύο κείμενα $m_1, m_2 \in \mathbb{Z}_p$, τότε το σύστημα δεν έχει πλέον perfect secrecy.

iv. Δείξτε με ποιον τρόπο μπορούμε να σπάσουμε το σύστημα αν μας δοθούν δύο διαφορετικά ζεύγη αρχικού κειμένου / κρυπτοκειμένου.

6. Ένα κρυπτοσύστημα λέγεται *malleable (εύπλαστο)* εάν είναι δυνατόν ένας αντίπαλος να φτιάξει, γνωρίζοντας μόνο το κρυπτογράφημα $c = E(m)$ ενός αρχικού κειμένου m, ένα έγκυρο κρυπτογράφημα του αρχικού κειμένου $c' = E(f(m))$, για κάποια πολυωνυμικά αντιστρέψιμη συνάρτηση f της επιλογής του.

- Εξετάστε αν τα κρυπτοσυστήματα Vigenère, Affine cipher είναι malleable. Αιτιολογήστε την απάντησή σας.
- Υποθέστε ότι η συνάρτηση DES έχει την ιδιότητα της μη-διακρισιμότητας μηνύματος (βλέποντας κανείς δύο κρυπτοκείμενα C, C', όπου ισχύει είτε $C = Enc(M), C' = Enc(M')$ είτε $C = Enc(M'), C' = Enc(M)$, δεν μπορεί πρακτικά να ξεχωρίσει από ποιο αρχικό μήνυμα προήλθαν, ακόμη και αν γνωρίζει τα δύο αρχικά μηνύματα M, M').

  Εξετάστε αν υπάρχει ή όχι malleability στους εξής τρόπους λειτουργίας: ECB, CBC, CFB, OFB, CTR.
- Ποιοι από τους παραπάνω τρόπους λειτουργίας διαθέτουν την ιδιότητα ασφάλειας IND-CPA; Τι ισχύει σχετικά με τις IND-CCA και IND-CCA2; Ορισμοί:
- Μπορεί ένα malleable κρυπτοσύστημα να έχει την ιδιότητα IND-CCA2 ή όχι, και γιατί;

5.9 Ηλεκτρονικό Υλικό

- DES Challenges
  - Project Deschall
  - DES Cracker
- Linear Cryptanalysis Tutorial
- Differential Cryptanalysis

Βιβλιογραφία

[1] Χρήστος Καπούτσης. Κρυπτοσυστήματα πακέτου: από το des στο aes. Master's thesis, University Of Athens, Greece, 2000. Διαθέσιμο στο http://www.math.uoa.gr/~mpla/thesis/cak.ps.

[2] E. R. Berlekamp. *Algebraic coding theory*. McGraw-Hill, New York, 1968.

[3] Eli Biham and Adi Shamir. Differential cryptanalysis of des-like cryptosystems. *Journal of CRYPTOLOGY*, 4(1):3–72, 1991.

[4] Dan Boneh. Coursera online cryptography course: Exhaustive search attacks on des, 2012. Διαθέσιμο στο https://www.youtube.com/watch?v=k9LF5O5CCQk.

[5] Carolynn Burwick, Don Coppersmith, Edward D'Avignon, Rosario Gennaro, Shai Halevi, Charanjit Jutla, Stephen M. Matyas, Luke O'Connor, Mohammad Peyravian, David Safford, and Nevenko Zunic. The mars encryption algorithm, 1999.

[6] Agnes Hui Chan, Mark Goresky, and Andrew Klapper. On the linear complexity of feedback registers. *IEEE Transactions on Information Theory*, 36(3):640–644, 1990.

[7] Don Coppersmith. The data encryption standard (des) and its strength against attacks. *IBM journal of research and development*, 38(3):243–250, 1994.

[8] Des. Data encryption standard. In *In FIPS PUB 46, Federal Information Processing Standards Publication*, pages 46–2, 1977.

[9] Diane Erdmann and Sean Murphy. An approximate distribution for the maximum order complexity. *Designs, Codes and Cryptography*, 10(3):325–339, 1997.

[10] H. Feistel. Cryptography and computer privacy. *Scientific American*, 228(5):15–23, 1973.

[11] R. A. Games and A. H. Chan. A fast algorithm for determining the complexity of a binary sequence with period 2^n. *IEEE Trans. Information Theory*, 29:144–146, 1983.

[12] Jovan Golic, Andrew Clark, and Ed Dawson. Generalized inversion attack on nonlinear filter generators. *Computers, IEEE Transactions on*, 49(10):1100–1109, 2000.

[13] Jovan Dj Golic. On the linear complexity of functions of periodic gf (q) sequences. *Information Theory, IEEE Transactions on*, 35(1):69–75, 1989.

[14] Solomon W. Golomb. *Shift Register Sequences*. Aegean Park Press, Laguna Hills, CA, USA, 1981.

[15] El Groth. Generation of binary sequences with controllable complexity. *Information Theory, IEEE Transactions on*, 17(3):288–296, 1971.

[16] T Herlestam. On functions of linear shift register sequences. In *Proc. Of a Workshop on the Theory and Application of Cryptographic Techniques on Advances in cryptology—EUROCRYPT '85*, pages 119–129, New York, NY, USA, 1986. Springer-Verlag New York, Inc.

[17] A. E. Heydtmann and J. M. Jensen. On the equivalence of the Berlekamp-Massey and the Euclidean algorithms for decoding. *Information Theory, IEEE Transactions on*, 46(7):2614–2624, 2000.

[18] Kyoki Imamura and Wataru Yoshida. A simple derivation of the berlekamp-massey algorithm and some applications. *IEEE Trans. Inf. Theor.*, 33(1):146–150, January 1987.

[19] Cees JA Jansen and Dick E Boekee. The shortest feedback shift register that can generate a given sequence. In *Advances in Cryptology—CRYPTO'89 Proceedings*, pages 90–99. Springer, 1990.

[20] Cornelis Johannes Adrianus Jansen. *Investigations on nonlinear streamcipher systems: construction and evaluation methods*. PhD thesis, TU Delft, Delft University of Technology, 1989.

[21] E. Key. An analysis of the structure and complexity of nonlinear binary sequence generators. *IEEE Trans. Inf. Theor.*, 22(6):732–736, September 2006.

[22] Alexander Klimov and Adi Shamir. Cryptographic applications of t-functions. In Mitsuru Matsui and RobertJ. Zuccherato, editors, *Selected Areas in Cryptography*, volume 3006 of *Lecture Notes in Computer Science*, pages 248–261. Springer Berlin Heidelberg, 2004.

[23] Nicholas Kolokotronis. Cryptographic properties of stream ciphers based on t-functions. In *Information Theory, 2006 IEEE International Symposium on*, pages 1604–1608. IEEE, 2006.

[24] R. Lidl and H. Niederreiter. *Introduction to Finite Fields and Their Applications*. Cambridge University Press, Cambridge, 1986.

[25] W E Madryga. A high performance encryption algorithm. In *Proceedings of the 2Nd IFIP International Conference on Computer Security: A Global Challenge*, pages 557–569, 1984.

[26] James L Massey. Shift-register synthesis and bch decoding. *Information Theory, IEEE Transactions on*, 15(1):122–127, 1969.

[27] James L Massey and Shirlei Serconek. A fourier transform approach to the linear complexity of nonlinearly filtered sequences. In *Advances in Cryptology—CRYPTO'94*, pages 332–340. Springer, 1994.

[28] James L Massey and Shirlei Serconek. Linear complexity of periodic sequences: a general theory. In *Advances in cryptology—CRYPTO'96*, pages 358–371. Springer, 1996.

[29] Mitsuru Matsui. Linear cryptanalysis method for des cipher. In *Advances in Cryptology—EUROCRYPT'93*, pages 386–397. Springer, 1994.

[30] Mitsuru Matsui and Atsuhiro Yamagishi. A new method for known plaintext attack of feal cipher. In *Advances in Cryptology—Eurocrypt'92*, pages 81–91. Springer, 1993.

[31] Willi Meier, Enes Pasalic, and Claude Carlet. Algebraic attacks and decomposition of boolean functions. In *Advances in Cryptology-EUROCRYPT 2004*, pages 474–491. Springer, 2004.

[32] Alfred J Menezes, Paul C Van Oorschot, and Scott A Vanstone. *Handbook of applied cryptography*. CRC press, 1996.

[33] US Department of Commerce. Des modes of operation. Technical Report FIPS PUB 81, US Department of Commerce / National Bureau of Standards, December 1998.

[34] Federal Information Processing and Announcing The. Announcing the advanced encryption standard (aes).

[35] Ronald L. Rivest. The rc5 encryption algorithm, 1995.

[36] Ronald L. Rivest, M. J. B. Robshaw, Y.L. Yin, and R. Sidney. The rc6 block cipher, 1998.

[37] Panagiotis Rizomiliotis. Constructing periodic binary sequences with maximum nonlinear span. *IEEE transactions on information theory*, 52(9):4257–4261, 2006.

[38] Panagiotis Rizomiliotis and Nicholas Kalouptsidis. Results on the nonlinear span of binary sequences. *IEEE transactions on information theory*, 51(4):1555–1563, 2005.

[39] Panagiotis Rizomiliotis, Nicholas Kolokotronis, and Nicholas Kalouptsidis. On the quadratic span of binary sequences. *IEEE transactions on information theory*, 51(5):1840–1848, 2005.

[40] Rainer A. Rueppel. *Analysis and Design of Stream Ciphers*. Springer-Verlag New York, Inc., New York, NY, USA, 1986.

[41] Rainer A. Rueppel and Othmar Staffelbach. Products of linear recurring sequence with maximum complexity. In *EUROCRYPT*, pages 30–32, 1986.

[42] C. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal, Vol 28, pp. 656–715*, 1949.

[43] Thomas Siegenthaler. Cryptanalysts representation of nonlinearly filtered ml-sequences. In *Advances in Cryptology—EUROCRYPT'85*, pages 103–110. Springer, 1986.

Κεφάλαιο 6

Κρυπτοσυστήματα Δημοσίου Κλειδιού

6.1 Εισαγωγή

Η ιδέα της κρυπτογραφίας δημοσίων κλειδιών οφείλεται στους Diffie και Hellman (1976) [4], και το πρώτο κρυπτοσύστημα δημοσίου κλειδιού ήταν το RSA, το οποίο προτάθηκε το 1977 από τους Rivest, Shamir και Adlemann ([10]).

Όπως αναφέραμε στην εισαγωγή στην κρυπτογραφία δημοσίου κλειδιού[1], το κλειδί κρυπτογράφησης k είναι δημόσιο, δηλαδή είναι γνωστό στον καθένα, ενώ το κλειδί αποκρυπτογράφησης k' κρατείται απόρρητο. Το βασικό στοιχείο πίσω από την κρυπτογραφία δημοσίου κλειδιού είναι η χρήση συναρτήσεων μονής κατεύθυνσης (one-way functions). Δεν είναι γνωστό όμως αν όντως υπάρχουν συναρτήσεις μονής κατεύθυνσης.[2]

Οι *συναρτήσεις μονής κατεύθυνσης* ορίζονται ως εξής:

Ορισμός 6.1. Έστω Σ_1, Σ_2 δυο πεπερασμένα σύνολα (αλφάβητα), και f μια συνάρτηση $f : \Sigma_1^* \to \Sigma_2^*$. Λέμε ότι η f είναι **συνάρτηση μονής κατεύθυνσης** αν ισχύουν τα παρακάτω:

η f είναι 1-1 και για όλα τα $x \in \Sigma_1^*$, και για την $f(x)$ ισχύει: $|x|^{1/k} \leq |f(x)| \leq |x|^k$ για κάποιο $k > 0$.

Η $f(x)$ μπορεί να υπολογιστεί σε πολυωνυμικό χρόνο.

Για την f^{-1}, την αντίστροφη συνάρτηση της f δεν υπάρχει αλγόριθμος πολυωνυ-

[1] Λέγεται επίσης και κρυπτογραφία μονής κατεύθυνσης

[2] Για την ακρίβεια γνωρίζουμε ότι υπάρχουν συναρτήσεις μονής κατεύθυνσης αν και μόνο αν $P \neq UP$, κάτι όμως που δεν έχει ακόμα αποδειχθεί.

μικού χρόνου, δηλαδή το πρόβλημα του υπολογισμού του $x \in \Sigma_1$ δεδομένου του $f(x) \in \Sigma_2$ είναι απρόσιτο.

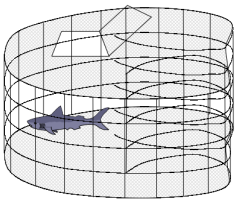
Παρατήρηση 15. Μία συνάρτηση $f(x)$ λέγεται μονής κατεύθυνσης εάν είναι εύκολο να υπολογιστεί η $f(x)$ δεδομένου του x (εύκολο =σε χαμηλό πολυωνυμικό χρόνο, κατά προτίμηση γραμμικό), ενώ ο αντίστροφος υπολογισμός του x δεδομένου του $f(x)$ είναι απρόσιτος (κατά προτίμηση είναι στην κλάση $NP \setminus P$).

Αν μια συνάρτηση f είναι συνάρτηση μονής κατεύθυνσης και ο υπολογισμός της f^{-1} είναι εύκολος όταν δίνεται μια *απόρρητη καταπακτή* (secret trapdoor), τότε η f λέγεται **συνάρτηση καταπακτής (trapdoor function)**.

Ένα παράδειγμα είναι ο πολλαπλασιασμός ακεραίων. Η $f(x, y) = x \cdot y$ είναι μια συνάρτηση που πιστεύεται ότι είναι μονής κατεύθυνσης, αν και δεν έχει αποδειχτεί. Ένα άλλο παράδειγμα είναι η ύψωση σε δύναμη modulo έναν πρώτο αριθμό n (αυτό χρησιμοποιείται στο κρυπτοσύστημα RSA). Η αντίστροφη αυτής της πράξης είναι η εύρεση ρίζας modulo n.

Ας σημειωθεί ότι ακόμα και αν $P \neq NP$, δεν είναι σίγουρο ότι υπάρχουν συναρτήσεις μονής κατεύθυνσης.

Ένα διαισθητικό παράδειγμα είναι το εξής: Έστω το δίχτυ του σχήματος 6.1. Ένα ψάρι μπαίνει μέσα σε αυτό από την τρύπα, αλλά λόγω του σχήματος του διχτυού, δεν μπορεί να βρει ξανά την έξοδο και να βγει. Το ίδιο πράγμα συμβαίνει και με τις συναρτήσεις μονής κατεύθυνσης. Αν όμως δείξουμε στο ψάρι το άνοιγμα στο πάνω μέρος του διχτυού (το αντίστοιχο του trapdoor), τότε το ψάρι θα βγει εύκολα.



Σχήμα 6.1: Δίχτυ μίας κατεύθυνσης (κιούρτος).

6.2 Ασφάλεια Κρυπτοσυστημάτων Δημοσίου Κλειδιού

Για να μοντελοποιήσουμε την ασφάλεια ενός κρυπτοσυστήματος δημοσίου κλειδιού πρέπει να λάβουμε υπόψιν μας ότι ο αντίπαλος $\mathcal{A}$ μπορεί να πειραματιστεί

ελεύθερα με τις κρυπτογραφήσεις μηνυμάτων της αρεσκείας του καθώς διαθέτει το δημόσιο κλειδί του συστήματος. Έτσι έχουν ιδιαίτερο νόημα οι επιθέσεις CPA και CCA από την ενότητα 1.4.2.

Εδώ θα τις περιγράψουμε αυστηρά χρησιμοποιώντας τα παρακάτω παιχνίδια ασφαλείας, τα οποία είναι ουσιαστικά αυτά που παρουσιάσαμε στην ενότητα 1.4.4, τροποποιημένα ώστε να αφορούν ρητά κρυπτοσυστήματα δημοσίου κλειδιού.

CPA

- Ο $\mathcal{C}$ εκτελεί τον αλγόριθμο $\texttt{KeyGen}(1^\eta)$ και παράγει τα κλειδιά κρυπτογράφησης και αποκρυπτογράφησης (pk, sk) και δημοσιοποιεί το πρώτο.
- Ο $\mathcal{A}$ μπορεί να κρυπτογραφεί πολυωνυμικό πλήθος μηνυμάτων χρησιμοποιώντας το δημόσιο κλειδί.
- Τελικά παράγει δύο μηνύματα m_0, m_1 τα οποία στέλνει στον $\mathcal{C}$.
- Ο $\mathcal{C}$ διαλέγει ένα τυχαίο bit $b \in \{0, 1\}$.
- Στην συνέχεια κρυπτογραφεί το μήνυμα m_b παράγοντας το $c \leftarrow \texttt{Encrypt}(pk, m_b)$ και το στέλνει στον $\mathcal{A}$.
- Ο $\mathcal{A}$ συνεχίζει να κρυπτογραφεί πολυωνυμικό πλήθος μηνυμάτων.
- Με βάση τις πληροφορίες που συγκέντρωσε και το c παράγει ένα bit b'.
- Αν $b' = b$ τότε το αποτέλεσμα του παιχνιδιού ορίζεται 1 και ο $\mathcal{A}$ κερδίζει. Σε διαφορετική περίπτωση το αποτέλεσμα ορίζεται 0.

Ορισμός 6.2. IND-CPA Το κρυπτοσύστημα έχει την ιδιότητα IND-CPA αν κάθε PPT $\mathcal{A}$ έχει αμελητέο πλεονέκτημα στον υπολογισμό του b από το να μαντέψει τυχαία.

Για το παιχνίδι ασφαλείας IND-CCA έχουμε τις παρακάτω παραλλαγές:

CCA

- Ο $\mathcal{C}$ εκτελεί τον αλγόριθμο $\texttt{KeyGen}(1^\eta)$ και παράγει τα κλειδιά κρυπτογράφησης και αποκρυπτογράφησης (pk, sk) και δημοσιοποιεί το πρώτο.
- Ο $\mathcal{A}$ μπορεί να κρυπτογραφεί πολυωνυμικό πλήθος μηνυμάτων

- Ο $\mathcal{A}$ χρησιμοποιεί το σύστημα ως *decryption oracle* και μπορεί να αποκρυπτογραφήσει πολυωνυμικό πλήθος μηνυμάτων
- Τελικά παράγει δύο μηνύματα m_0, m_1 τα οποία στέλνει στον $\mathcal{C}$,
- Ο $\mathcal{C}$ διαλέγει ένα τυχαίο bit $b \in \{0, 1\}$.
- Στην συνέχεια κρυπτογραφεί το μήνυμα m_b παράγοντας το $c \leftarrow \texttt{Encrypt}(pk, m_b)$ και το στέλνει στον $\mathcal{A}$.
- Ο $\mathcal{A}$ συνεχίζει να κρυπτογραφεί πολυωνυμικό πλήθος μηνυμάτων και να κάνει οποιονδηποτε άλλο υπολογισμό μπορεί.
- *Προαιρετικά* ο $\mathcal{A}$ μπορει να συνεχίσει να χρησιμοποιεί το *decryption oracle*, αλλά δεν μπορεί να ζητήσει αποκρυπτογράφηση του c.
- Με βάση τις πληροφορίες που συγκέντρωσε και το c παράγει ένα bit b'.
- Αν $b' = b$ τότε το αποτέλεσμα του παιχνιδιού ορίζεται 1 και ο $\mathcal{A}$ κερδίζει. Σε διαφορετική περίπτωση το αποτέλεσμα ορίζεται 0.

Ορισμός 6.3. Ορισμός ασφάλειας Το κρυπτοσύστημα έχει την ιδιότητα IND-CCA1 αν κάθε PPT $\mathcal{A}$ έχει αμελητέο πλεονέκτημα στον υπολογισμό του b από το να μαντέψει τυχαία.

Αν επιπλέον ισχύει το προαιρετικό βήμα το κρυπτοσύστημα είναι IND-CCA2 (adaptive IND-CCA).

6.3 Κρυπτοσυστήματα Δημοσίου Κλειδιού από Προβλήματα NP

Ένα κρυπτοσύστημα δημοσίου κλειδιού μπορεί κατασκευαστεί ακολουθώντας τα ακόλουθα γενικά βήματα:

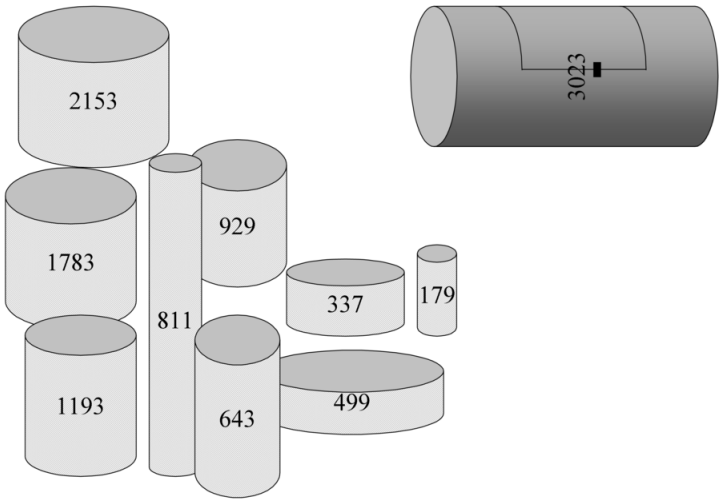
- Διαλέγουμε ένα δύσκολο πρόβλημα Π. Το Π θα πρέπει να είναι υπολογιστικά απρόσιτο. Το Π λέγεται υποκείμενο πρόβλημα.
- Βρίσκουμε ένα εύκολο υποπρόβλημα Π_{easy} του Π. Το Π_{easy} θα πρέπει να χρειάζεται μικρό πολυωνυμικό χρόνο.
- "Ανακατεύουμε" το Π_{easy} με τέτοιο τρόπο ώστε το προκύπτον πρόβλημα $\Pi_{shuffle}$ να μοιάζει με το γενικό πρόβλημα Π.

- Δημοσιεύουμε το $\Pi_{shuffle}$ και τη μέθοδο κρυπτογράφησης. Η μέθοδος ανάκτησης του Π_{easy} από το $\Pi_{shuffle}$ είναι η μυστική καταπακτή. Έτσι ο νόμιμος αποδέκτης θα πρέπει να λύσει το Π_{easy} ενώ ο κρυπταναλυτής βρίσκεται ενάντια στο γενικό πρόβλημα Π.

Ας εφαρμόσουμε την παραπάνω μεθολογία με ένα παράδειγμα:

6.3.1 Το κρυπτοσύστημα σακιδίου Merkle - Hellman

Το κρυπτοσύστημα Merkle - Hellman, είναι ένα κρυπτοσύστημα δημοσίου κλειδιού βασισμένο στο διακριτό πρόβλημα του σακιδίου (Discrete Knapsack Problem -DKP). Στην πραγματικότητα, γι'αυτό το κρυπτοσύστημα δεν χρησιμοποιούμε το πρόβλημα του σακιδίου, αλλά το πρόβλημα του *Αθροίσματος Υποσυνόλων (Subset Sum)*.



Σχήμα 6.2: Διακριτό πρόβλημα σακιδίου

Το πρόβλημα του σακιδίου (για τους σκοπούς αυτού του παραδείγματος) αποτελείται από μια n-άδα $A = (a_1, \ldots, a_n)$ από διαφορετικούς θετικούς ακεραίους, που ονομάζεται *διάνυσμα του σακιδίου* και ένα θετικό ακέραιο k, την χωρητικότητα του σακιδίου. Το πρόβλημα είναι να βρούμε τέτοιους ακεραίους a_i των οποίων το άθροισμα να είναι ίσο με k. Προφανώς πάντα μπορεί να βρεθεί μια λύση ελέγχοντας εξαντλητικά όλα τα 2^n υποσύνολα του A, ψάχνοντας μήπως κάποιο από αυτά έχει άθροισμα το k. Για μεγάλο n αυτός ο υπολογισμός είναι απρόσιτος.

Από το πρόβλημα αυτό μπορούμε να ορίσουμε μια συνάρτηση $f(x)$ ως εξής:

Έστω $\vec{x}$ η δυαδική αναπαράσταση (με n bits) του ακεραίου x, $0 \leq x \leq 2^{n-1} - 1$ (προσθέτοντας μηδενικά στην αρχή, αν είναι απαραίτητο). Το $f(x)$ είναι ίσο με το άθροισμα όλων των a_i τέτοιων ώστε το i-οστό bit του x να είναι άσσος.

Δηλαδή:

$$f(1) = f(0 \ldots 001) = a_n$$
$$f(3) = f(0 \ldots 011) = a_{n-1} + a_n$$

Χρησιμοποιώντας πολλαπλασιασμό διανυσμάτων μπορούμε να συμβολίσουμε: $f(x) = A \cdot B_x$ όπου A είναι το διάνυσμα σακιδίου και B_x είναι η δυαδική αναπαράσταση του x γραμμένη ως διάνυσμα-στήλη.

Με τον παραπάνω ορισμό είναι εύκολο να υπολογίσουμε το $f(x)$ από το x, ενώ το να υπολογίσουμε το x από το $f(x)$ είναι ισοδύναμο με το να λύσουμε το πρόβλημα του σακιδίου, αφού το x αναπαριστά (στη δυαδική του μορφή) τα αντικείμενα του Α που έχουν άθροισμα $f(x)$.

Η συνάρτηση $f(x)$ μπορεί να χρησιμοποιηθεί για κρυπτογράφηση με σχετικά απλό τρόπο:

Το αρχικό κείμενο γράφεται με τη μορφή δυαδικής ακολουθίας αντικαθιστώντας κάθε γράμμα του αλφαβήτου με τον αντίστοιχο αριθμό (π.χ. ASCII) (γραμμένο σε δυαδική μορφή). Κάθε ακουλουθία των n bits κρυπτογραφείται υπολογίζοντας τη συνάρτηση f για το συγκεκριμένο κομμάτι (block).

Η αποκρυπτογράφηση, από την άλλη πλευρά, είναι εξίσου δύσκολη με ένα NP-πλήρες πρόβλημα, όχι μόνο για τον κρυπταναλυτή, αλλά και για τον ίδιο το χρήστη. Αυτό μπορεί να αποφευχθεί σχεδιάζοντας το σύστημα έτσι ώστε ο νόμιμος χρήστης (που γνωρίζει μια μυστική καταπακτή) να χρειάζεται να λύσει ένα εύκολο στιγμιότυπο του προβλήματος του σακιδίου, ενώ ο κρυπταναλυτής να βρίσκεται απέναντι στο γενικό (δύσκολο) πρόβλημα.

Η καταπακτή μπορεί να προκύψει από τα προβλήματα σακιδίου που χρησιμοποιούν υπεραυξητικά διανύσματα A. Ένα διάνυσμα σακιδίου (ή μια n-άδα γενικά) $A = (a_1, \ldots, a_n)$ λέγεται *υπεραυξητική* αν κάθε στοιχείο a_i υπερβαίνει το άθροισμα όλων των προηγούμενων αριθμών:

$$a_j > \sum_{i=1}^{j-1} a_i, \ for\ j = 2, \ldots, n$$

Σε αυτή την περίπτωση το πρόβλημα του σακιδίου μπορεί να λυθεί σε *γραμμικό χρόνο* αφού ένα πέρασμα του διανύσματος του σακιδίου είναι αρκετό. Είναι επίσης προφανές (από τον αλγόριθμο που λύνει το πρόβλημα) ότι το υπεραυξητικό πρόβλημα σακιδίου έχει πάντα *το πολύ μία λύση*.

Εάν χρησιμοποιούμε υπεραυξητική ακολουθία A για ένα κρυπτοσύστημα, δεν πρέπει να τη δημοσιεύσουμε, γιατί θα έκανε την κρυπτανάλυση γραμμική (άρα εύκολη). Αυτό μπορούμε να το αποφύγουμε ανακατεύοντας το A σε ένα διάνυσμα A' που θα μοιάζει με ένα τυχαίο διάνυσμα σακιδίου. Το ανακάτεμα αυτό μπορεί να γίνει με πολλαπλασιασμό με modulo:

Διαλέγουμε έναν ακέραιο (modulus) $m > \sum a_i$ και έναν πολλαπλασιαστή t έτσι ώστε ο t και ο m να μην έχουν κοινούς παράγοντες. Η εκλογή του t εξασφαλίζει την ύπαρξη του t^{-1} (αντίστροφος του t) έτσι ώστε $tt^{-1} = 1 \pmod m$. Υπολογίζουμε τα γινόμενα $a'_i = ta_i$ και ορίζουμε το διάνυσμα $A = (a'_1, \ldots, a'_n)$. Το διάνυσμα $A' = tA$ μπορεί να δημοσιευτεί ως κλειδί κρυπτογράφησης, αλλά οι όροι t, t^{-1}, m πρέπει να κρατηθούν μυστικοί (secret trapdoor).

Ο νόμιμος αποδέκτης, για να αποκρυπτογραφήσει ένα block κρυπτοκειμένου c' (block με n bits) πρέπει πρώτα να υπολογίσει το $c = t^{-1}c' \bmod m$, και μετά να λύσει το εύκολο πρόβλημα σακιδίου βασισμένο στο $A = t^{-1}A' \bmod m$. Η μοναδική λύση είναι το σωστό block αρχικού κειμένου p αφού:

$$c = t^{-1}c' = t^{-1}A'p = t^{-1}tAp = Ap \pmod m$$

που σημαίνει ότι παρόλο που η κρυπτογράφηση έγινε χρησιμοποιώντας το A', η αποκρυπτογράφηση μπορεί να βασιστεί στο A και τους μυστικούς όρους.

Αυτό το κρυπτοσύστημα δημοσίου κλειδιού που βασίζεται στο πρόβλημα του σακιδίου έσπασε από τον Shamir [12] με αλγόριθμο πολυωνυμικού χρόνου. Η κρυπταναλυτική επίθεση βασίζεται στο γεγονός ότι δεν είναι απαραίτητο για τον κρυπταναλυτή να βρει τον ίδιο πολλαπλασιαστή t και modulus m με αυτα που χρησιμοποιεί ο σχεδιαστής του συστήματος. Αρκεί να βρει t' και m' τέτοια ώστε ο πολλαπλασιασμός του δημοσιευμένου διανύσματος με το $(t')^{-1} \pmod{m'}$ να είναι ένα υπεραυξητικό διάνυσμα. Έτσι, ο κρυπταναλυτής μπορεί να σπάσει το σύστημα με προεπεξεργασία, αφού το κλειδί δημοσιευτεί.

Στο κρυπτοσύστημα σακιδίου Π είναι το πρόβλημα σακιδίου (NP-πλήρες), Π_{easy} είναι το πρόβλημα σακιδίου με υπεραυξητικό διάνυσμα σακιδίου και το $\Pi_{shuffle}$ προκύπτει από πολλαπλασισμό modulo όπως περιγράψαμε παραπάνω.

Στην πράξη δεν χρησιμοποιούνται κρυπτοσυστήματα που να βασίζονται σε NP-hard προβλήματα, αλλά κυρίως προβλήματα NP της θεωρίας αριθμών. Ο κύριος λόγος γι' αυτό είναι το γεγονός ότι η δυσκολία NP προκύπτει από τα χειρότερα (worst-case) στιγμιότυπα (instances) των προβλημάτων. Αντίθετα στην κρυπτογραφία χρειαζόμαστε προβλήματα που (σχεδόν) όλα τα στιγμιότυπα να είναι δύσκολα. Δηλαδή χρειαζόμαστε δυσκολία μέσης περίπτωσης (average case complexity). Περισσότερα για τέτοια κρυπτοσυστήματα συζητάμε στην ενότητα 12.4.

6.4 Το κρυπτοσύστημα RSA

6.4.1 Λειτουργία

Το κρυπτοσύστημα RSA [10] προτάθηκε από τους Rivest, Shamir και Adleman και είναι το πρώτο κρυπτοσύστημα δημοσίου κλειδιού. Αποτελείται από τους εξής αλγόριθμους:

- **Δημιουργία Κλειδιών** `KeyGen`
  - Επιλογή δύο μεγάλων πρώτων p, q
  - Υπολογισμός του γινομένου $N = p \cdot q$
  - Υπολογισμός ακεραίου e με $gcd(e, \phi(N)) = 1$
  - Υπολογισμός $d = e^{-1} \pmod{\phi(N)}$ χρησιμοποιώντας τον Εκτεταμένο Αλγόριθμο του Ευκλείδη
  - Δημόσιο κλειδί είναι το (e, N) και ιδιωτικό το (d, p, q)
- **Κρυπτογράφηση** `Encrypt`
  - Κωδικοποίηση μηνύματος m στο $\mathbb{Z}_n$
  - Υπολογισμός $c = m^e \bmod n$
- **Αποκρυπτογράφηση** `Decrypt`
  - Υπολογισμός $m = c^d \bmod n$

Παρατηρήσεις Είναι εύκολο να παρατηρήσουμε ότι το RSA είναι ορθό καθώς:

$$Dec(Enc(m)) = (m^e)^d = m^{ed} = m \pmod{n} \text{ αφού } ed = 1 \pmod{\phi(n)}.$$

Μάλιστα με λίγη σκέψη είναι εύκολο να διαπιστώσει κανείς ότι η ορθότητα ισχύει είτε το μήνυμα κωδικοποιείται ως σχετικά πρώτος με το n είτε όχι.

Αφού το N είναι το γινόμενο των δύο πρώτων αριθμών, p, q θα ισχύει: $\phi(N) = (p-1)(q-1)$. Έτσι, αν κάποιος γνωρίζει τα p και q μπορεί εύκολα να υπολογίσει το $\phi(N)$ και επομένως και το d.

Κάποιες καλές επιλογές για το e είναι οι πρώτοι αριθμοί $3, 17, 65537$, ώστε να μην χρειάζονται πολλοί πολλαπλασιασμοί στον αλγόριθμο Επαναλαμβανόμενου Τετραγωνισμού 4.1.

6.4.2 Ασφάλεια

Καταρχήν σε σχέση με τα παιχνίδια ασφαλείας της ενότητας 6.2, πρέπει να παρατηρήσουμε ότι το RSA όπως το ορίσαμε παραπάνω δεν είναι IND-CPA γιατί είναι ντετερμινιστικό. Πράγματι, αν τα δύο πιθανά μηνύματα που αντιστοιχούν σε ένα κρυπτοκείμενο c που έχει στη διάθεση του ο $\mathcal{A}$ είναι:

- m_0 = "Buy IBM"
- m_1 = "Sell IBM"

τότε ο $\mathcal{A}$ μπορεί να τα κρυπτογραφήσει με το δημόσιο κλειδί και να τα συγκρίνει με το c.

Επιπλέον το παραδοσιακό RSA δεν είναι IND-CCA. Έστω ότι ο $\mathcal{A}$ μπορεί να αποκρυπτογραφήσει μηνύματα της επιλογής του, εκτός από το κρυπτοκείμενο-στόχο c. Τότε μπορεί να κερδίζει το παιχνίδι CCA ως εξής:

- Στόχος: Αποκρυπτογράφηση του $c = m_b^e \pmod n$
- Μπορεί να αποκρυπτογραφήσει το $c' = c_b x^e \pmod n$ όπου το x είναι δικής του επιλογής
- Ανακτά το $m_b = \frac{m'}{x}$
- Αν $m_b = m_0$ επιστρέφει $b^* = 0$ αλλιώς επιστρέφει $b^* = 1$

Αγνοώντας τους παραπάνω 'έμμεσους' τρόπους παραβίασης της ασφάλειας του κρυπτοσυστήματος, ας δούμε τρόπους άμεσης επίθεσης.

Ο αλγόριθμος RSA στηρίζεται στη συνάρτηση $E(x) = x^e \bmod n$ η οποία υποθέτουμε ότι είναι *μονής κατεύθυνσης*, χωρίς αυτό να έχει αποδειχτεί, δηλ. δεν υπάρχει γνωστός αποδοτικός αλγόριθμος για να την αντιστρέψει κανείς. Η αντιστροφή της παραπάνω συνάρτησης αναφέρεται ως πρόβλημα RSA.

Ορισμός 6.4 (RSA Problem - `RSAP`)**.** Δίνονται $N = pq$, e με $gcd(e, \phi(N)) = 1$ και $c \in \mathbb{Z}_N^*$. Να βρεθεί η τιμή $c^{\frac{1}{e}} \pmod n \ (\equiv c^d \pmod n)$, δηλαδή $m \in \mathbb{Z}_N^*$, τ.ώ. $m^e \bmod N = c$.

Το γεγονός ότι ο παραπάνω υπολογισμός της e-οστής ρίζας modulo n είναι δύσκολος ή ισοδύναμα ότι η παραπάνω συνάρτηση είναι δύσκολο να αντιστραφεί ονομάζεται *Υπόθεση RSA*. Φυσικά είναι φανερό ότι η αντιστροφή μπορεί να γίνει εύκολα όταν η πληροφορία καταπακτής (p, q, d) είναι γνωστή.

Βέβαια μία άλλη προσπάθεια παραβίασης δεν αξιοποιεί καθόλου το μήνυμα, προσπαθώντας να ανακτήσει κατευθείαν το ιδιωτικό κλειδί από το δημόσιο.

Ορισμός 6.5 (RSA Key Inversion Problem - $\texttt{RSA}-\texttt{KINV}$)**.** Δίνονται $N = pq$, e με $gcd(e, \phi(N)) = 1$. Να βρεθεί η τιμή $e^{-1} \pmod{\phi(N)} (= d)$

Είναι εύκολο να αποδειχθεί ότι:

Θεώρημα 6.6. $\texttt{RSAP} \leq \texttt{RSA}-\texttt{KINV}$

Απόδειξη. Αν βρεθεί $d = e^{-1}$ υπολογίζεται εύκολα $m = c^d \bmod N$, ώστε $m^e \equiv c^{de} \equiv c \pmod{N}$. □

Εναλλακτικά θα μπορούσαμε να επιτεθούμε στο υπόλοιπο τμήμα της πληροφορίας καταπακτής δηλαδή στην εύρεση των p, q. Αυτό θα μπορούσαμε να το πετύχουμε με 2 τρόπους, μέσω του $\phi(N)$ και του προβλήματος $\texttt{FACTORING}$.

Ορισμός 6.7 (Το πρόβλημα $\texttt{COMPUTE}-\phi(\texttt{N})$)**.** Δίνεται $\phi(N)$ με $N = pq$ όπου p, q πρώτοι. Να βρεθούν τα p, q

Ορισμός 6.8. Το πρόβλημα $\texttt{FACTORING}$ Δίνεται $N = pq$ με p, q πρώτοι. Να βρεθούν τα p, q

Ας δούμε αρχικά την σχέση των δύο παραπάνω προβλημάτων και πιο συγκεκριμένα αν μπορούμε να υπολογίσουμε με κάποιο τρόπο το $\phi(n)$. Τότε όμως θα ήταν πολύ εύκολο να υπολογίσουμε τους αριθμούς p, q λύνοντας τις εξισώσεις:

$$\begin{aligned} n &= pq \\ \phi(n) &= (p-1)(q-1) \end{aligned}$$

Αν αντικαταστήσουμε $q = N/p$ στη δεύτερη εξίσωση, τότε έχουμε μια δευτεροβάθμια εξίσωση για το p:

$$p^2 - (n - \phi(n) + 1)p + n = 0$$

Οι δυο λύσεις αυτής της εξίσωσης θα είναι p και q, οι δύο παράγοντες του n. Παρατηρούμε λοιπόν ότι ο υπολογισμός του $\phi(n)$ είναι ισοδύναμος με την παραγοντοποίηση του n, δηλαδή ισχύει:

Θεώρημα 6.9. $\texttt{COMPUTE}-\phi(\texttt{N}) \equiv \texttt{FACTORING}$.

Είναι φανερό τέλος ότι αν μπορούμε να πετύχουμε παραγοντοποίηση του n, μπορούμε να βρούμε το κλειδί d χρησιμοποιώντας τον εκτεταμένο Ευκλείδειο αλγόριθμο (όπως στη δημιουργία κλειδιών) και να προχωρήσουμε και στην αποκρυπτογράφηση μηνύματος. Δηλαδή

Θεώρημα 6.10. *Το πρόβλημα RSA δεν είναι πιο δύσκολο από το* $\texttt{FACTORING}$. $\texttt{RSAP} \leq \texttt{RSA}-\texttt{KINV} \leq \texttt{FACTORING}$

Δεν είναι γνωστό αν ισχύει το αντίστροφο, δηλαδή αν τα προβλήματα είναι ισοδύναμα. Υπάρχουν όμως ενδείξεις ότι το RSA είναι πιο εύκολο από το FACTORING [11]. Είναι επομένως πιθανό να υπάρχουν άλλοι πιο άμεσοι τρόποι για την κρυπτανάλυση ενός κρυπτοσυστήματος RSA, χωρίς την παραγοντοποίηση του N.

Από την άλλη μεριά, μπορούμε να δείξουμε ότι το σπάσιμο του RSA με υπολογισμό του εκθέτη αποκρυπτογράφησης d είναι μάλλον δύσκολο: αν υπάρχει ένας αλγόριθμος για τον υπολογισμό του d ενός συστήματος RSA, τότε μπορούμε να σχεδιάσουμε έναν πιθανοτικό αλγόριθμο Las Vegas για την παραγοντοποίηση του N. Αυτός ο Las Vegas αλγόριθμος, χρησιμοποιεί τον αλγόριθμο για τον υπολογισμό του εκθέτη d ως *μαντείο (oracle)*. Δηλαδή:

Θεώρημα 6.11. *Ένας αλγόριθμος για τον υπολογισμό του εκθέτη αποκρυπτογράφησης d σε ένα κρυπτοσύστημα RSA μπορεί να μετατραπεί σε πιθανοτικό αλγόριθμο για την παραγοντοποίηση του n.*

Απόδειξη. Ο πιθανοτικός αλγόριθμος παραγοντοποίησης διαλέγει αριθμούς $w \in \mathbb{Z}_n^*$. Είναι προφανές ότι αν μια τυχαία επιλογή του $w < n$ ικανοποιεί τη σχέση $(w, n) > 1$ τότε μπορεί να γίνει αμέσως η παραγοντοποίηση του n. Άρα μπορούμε να υποθέσουμε ότι $(w, n) = 1$.

Αν μπορεί να βρεθεί μια μη τετριμμένη τετραγωνική ρίζα του $1 \pmod n$, τότε επίσης μπορούμε να παραγοντοποιήσουμε εύκολα το n. Μια μη τετριμμένη τετραγωνική ρίζα του $1 \pmod n$ είναι ένας αριθμός u με τις παρακάτω ιδιότητες: $u \neq \pm 1 \pmod n$ και $u^2 = 1 \pmod n$

Η ισοτιμία $x^2 = 1 \pmod p$ έχει δύο λύσεις modulo p: $\pm 1 \pmod p$. Όμοια, η ισοτιμία $x^2 = 1 \pmod q$ έχει δύο λύσεις modulo q: $\pm 1 \pmod q$.

Αφού η σχέση $x^2 = 1 \pmod n$ (όπου $n = pq$) ισχύει αν και μόνο αν $x^2 = 1 \pmod p$ και $x^2 = 1 \pmod q$, έπεται ότι: $x^2 = 1 \pmod n \iff x = \pm 1 \pmod p$ και $x = \pm 1 \pmod q$

Αυτό υποδηλώνει ότι υπάρχουν τέσσερις τετραγωνικές ρίζες modulo n οι οποίες μπορούν να βρεθούν με το Κινέζικο Θεώρημα Υπολοίπων αν γνωρίζουμε τα p, q. Δύο από αυτές είναι οι τετριμμένες ρίζες $x = \pm 1 \pmod n$. Οι άλλες δύο είναι οι μη τετριμμένες ρίζες ($\mp u \pmod N$) οι οποίες είναι αρνητικές η μία της άλλης modulo n.

Το $(u+1)(u-1)$ διαιρείται[3] από το n, ήτοι $n|(u+1)(u-1)$, αλλά οι παράγοντες $u+1$ και $u-1$ δεν διαιρούνται. Συνεπώς, $(u+1, n)$ ισούται είτε με p είτε με q.

Ο αλγόριθμος δίνεται παρακάτω:

Διαλέγουμε ένα τυχαίο $w < n$ και υποθέτουμε ότι $(w, n) = 1$ (αν $(w, n) > 1$ τότε μπορούμε να παραγοντοποιήσουμε το n κατευθείαν). Αφού ο δεδομένος

[3] $(u+1)(u-1) = u^2 - 1 = 0 \pmod n$

αλγόριθμος μπορεί να υπολογίσει τον εκθέτη αποκρυπτογράφησης d, μπορούμε να γράψουμε το $ed-1$ στη μορφή: $ed-1=2^s r,\ s\geq 1$ και r odd.

Επίσης ξέρουμε ότι $ed-1$ είναι πολλαπλάσιο του $\phi(n)$, επομένως έχουμε την ισοτιμία:

$$w^{2^s r}=1 \pmod n$$

Έστω s' ο μικρότερος αριθμός για τον οποίο

$$w^{2^{s'} r}=1 \pmod n \text{ και } 0\leq s'<n.$$

Αν $s'>0$ και $w^{2^{s'-1} r}\neq -1 \pmod n$ τότε έχουμε βρει μια μη τετριμμένη τετραγωνική ρίζα του $1 \pmod n$, και επομένως μπορούμε να παραγοντοποιήσουμε το n όπως εξηγήθηκε παραπάνω.

Προκειμένου να ολοκληρώσουμε την απόδειξη, πρέπει να δείξουμε ότι ο αλγόριθμος επιτυγχάνει (δηλαδή ο τυχαίος αριθμός w ικανοποιεί την ισότητα 6.4.2) με πιθανότητα $\geq \frac{1}{2}$.

Ας υποθέσουμε λοιπόν ότι η σχέση 6.4.2 δεν ικανοποιείται, δηλαδή: $w^r = 1 \pmod n$, ή $w^{2^t r}=-1 \pmod n$, για κάποιο $,t, 0\leq t<s$

Θα δώσουμε ένα άνω όριο για τα w που ικανοποιούν τις συνθήκες στην εξίσωση 6.4.2 ("κακές" τυχαίες επιλογές):

Γράφουμε

$$p-1=2^i a,\ q-1=2^j b, \text{ όπου } a \text{ και } b \text{ είναι περιττοί.}$$

Χωρίς βλάβη της γενικότητας, μπορούμε να υποθέσουμε ότι $i\leq j$. Αφού το $2^s r$ είναι πολλαπλάσιο του $\phi(n)$, και το r είναι ένα πολλαπλάσιο του ab. Έτσι αν $t\geq i$ τότε το $2^t r$ είναι πολλαπλάσιο του $p-1$ και

$$w^{2^t r}=1 \pmod p \Longrightarrow$$
$$w^{2^t r}\neq -1 \pmod p \Longrightarrow w^{2^t r}\neq -1 \pmod n$$

Αυτό σημαίνει ότι οι συνθήκες 6.4.2 δεν ικανοποιούνται ποτέ για $t\geq i$. Αφού $i<s$ μπορούμε να γράψουμε τη συνθήκη 6.4.2 με την επόμενη ισοδύναμη μορφή:

$$w^r=1 \pmod n \text{ ή } w^{2^t r}=-1 \pmod n, \text{ για κάποιο } t, 0\leq t<i$$

Μπορούμε τώρα να εκτιμήσουμε τον αριθμό των w που ικανοποιούν την 6.4.2. Έστω g είναι μια γεννήτρια του $\mathbb{Z}_p^*$ και έστω $w=g^u \pmod p$. Τότε είναι:

$$w^r=1 \pmod p \Longleftrightarrow ur=0 \pmod{p-1}$$

Έτσι οι δύο παραπάνω ισοτιμίες έχουν το ίδιο πλήθος λύσεων για τους αγνώστους u, w. Το πλήθος των λύσεων για την τελευταία ισοτιμία ισούται με $(r, p-1) = a$. Επομένως a είναι το πλήθος των λύσεων και για την πρώτη ισοτιμία.

Με τον ίδιο τρόπο, βλέπουμε ότι το πλήθος των λύσεων για την ισοτιμία $w^r = 1 \pmod{q}$ είναι b. Αυτό υποδηλώνει ότι ab είναι το πλήθος των λύσεων για την ισοτιμία $w^r = 1 \pmod{n}$.[4]

Το πλήθος των w που ικανοποιούν τη δεύτερη συνθήκη στον τύπο 6.4.2 μπορεί να βρεθεί με τον ίδιο τρόπο όπως παραπάνω: Το πλήθος των λύσεων w για την ισοτιμία

$$w^{2^{t+1}r} = 1 \pmod{p} \text{ (αντίστοιχα } w^{2^t r} = 1 \pmod{p})$$

όπου $t+1 \leq i$, ισούται με $(2^{t+1}r, p-1) = 2^{t+1}a$ (αντίστοιχα $(2^t r, p-1) = 2^t a$). Συνεπώς το πλήθος των λύσεων για την ισοτιμία

$$w^{2^t r} = -1 \pmod{p}$$

είναι το πολύ $2^{t+1}a - 2^t a = 2^t a$.

Με τον ίδιο τρόπο το πλήθος των λύσεων για την ισοτιμία

$$w^{2^t r} = -1 \pmod{q}$$

είναι το πολύ $2^{t+1}b - 2^t b = 2^t b$.

Αυτό συνεπάγεται ότι το πλήθος των λύσεων για την ισοτιμία

$$w^{2^t r} = -1 \pmod{n}$$

είναι το πολύ $2^t a \cdot 2^t b = 2^{2t} ab$.

Το άνω όριο για το πλήθος των ανεπιθύμητων ('κακών' τυχαίων επιλογών) w (το πλήθος των w που ικανοποιούν την 6.4.2 ή ισοδύναμα την 6.4.2) μπορεί να βρεθεί με πρόσθεση του πλήθους των λύσεων από τις δύο ισοτιμίες της 6.4.2 (άθροισμα για όλες τις πιθανές τιμές):

$$ab + ab \sum_{t=0}^{i-1} 2^{2t} = ab \left(1 + \sum_{t=0}^{i-1} 4^t \right) =$$

$$ab \left(1 + \frac{4^i - 1}{3} \right) = ab \left(\frac{2}{3} \cdot 2^{2i-1} + \frac{2}{3} \right) \leq$$

[4] Σημειώνουμε εδώ ότι κάθε ζευγάρι λύσεων από τις p- και q- ισοτιμίες φέρει μια λύση για την n- ισοτιμία μέσω του Κινέζικου Θεωρήματος Υπολοίπων. Συνολικά υπάρχουν ab τέτοια ζευγάρια.

$$\leq ab\left(\frac{2}{3}\cdot 2^{i+j-1}+\frac{2}{3}\right)\leq$$

$$ab\cdot 2^{i+j-1}=\phi(N)/2$$

Αφού το $\phi(n)$ είναι το πλήθος όλων των πιθανών w, το πολύ τα μισά από αυτά είναι 'κακές' επιλογές για τον αλγόριθμο παραγοντοποίησης. Αυτό σημαίνει ότι αφού ελεγχθούν k τυχαία επιλεγμένοι αριθμοί w, η πιθανότητα να μην βρεθεί ένα w ώστε ο αλγόριθμος να κάνει την παραγοντοποίηση (δηλαδή η πιθανότητα να μην γίνει "καλή" επιλογή) είναι 2^{-k}. □

Αν δεχθούμε την Επεκτεταμένη Υπόθεση του Riemann, μπορούμε να δείξουμε ότι υπάρχουν πολύ μικροί αριθμοί w οι οποίοι αποτελούν καλές επιλογές για τον παραπάνω αλγόριθμο, και επομένως ο αλγόριθμος Las Vegas που περιγράφηκε προηγουμένως μπορεί να είναι πολύ αποδοτικός.

Επιπλέον ο May στο [8] απλοποιεί την παραπάνω κατάσταση καθώς αποδεικνύει ότι η παραπάνω αναγωγή δεν είναι πιθανοτική αλλά ντετερμινιστική. Δηλαδή η γνώση των (n, e, d) είναι πολυωνυμικά ισοδύναμη με την παραγοντοποίηση του n (όταν $ed \leq n^2$). Δηλαδή η εύρεση του μυστικού εκθέτη είναι ισοδύναμη με την παραγοντοποίηση (όχι όμως και την επίλυση του RSAP). Συνοπτικά λοιπόν έχουμε:

$$\texttt{RSAP} \leq \texttt{RSA}-\texttt{KINV} \equiv \texttt{FACTORING} \equiv \texttt{COMPUTE}-\phi(\texttt{n})$$

6.4.3 Επιθέσεις στο RSA

Επίθεση κοινού γινομένου

Ας θεωρήσουμε ένα περιβάλλον, όπου τα γινόμενα N, όπως επίσης και οι εκθέτες κρυπτογράφησης και αποκρυπτογράφησης, διανέμονται από μια υπηρεσία την οποία εμπιστεύονται όλες οι πλευρές. Ας υποθέσουμε ότι η υπηρεσία γνωστοποιεί το κοινό για όλους γινόμενο n, τα κλειδιά κρυπτογράφησης $\{e_i\}$ και διανέμει στους χρήστες ξεχωριστά απόρρητους εκθέτες αποκρυπτογράφησης d_i.Οι πρώτοι αριθμοί p, q τέτοιοι ώστε $n = pq$ είναι γνωστοί μόνο στην υπηρεσία.

Με αυτό το σενάριο, κάθε χρήστης μπορεί να βρει σε ντετερμινιστικό τετραγωνικό χρόνο το μυστικό εκθέτη αποκρυπτογράφησης ενός άλλου χρήστη **χωρίς να παραγοντοποιήσει** το n.

Πράγματι, έστω δύο χρήστες A και B στο σενάριο που περιγράφεται παραπάνω. Θα δείξουμε ότι ο B μπορεί να βρει το d_A. Για κάποιο k ισχύει ότι

$$e_B d_B - 1 = k\phi(n)$$

Ονομάζουμε α τον μέγιστο αριθμό που διαιρεί το $e_B d_B - 1$ και είναι σχετικά πρώτος με το e_A:

$$\alpha | (e_B d_B - 1) \text{ και } (\alpha, e_A) = 1$$

επιπλέον ορίζουμε $t = \frac{e_B d_B - 1}{\alpha}$
Προκειμένου να υπολογιστεί το α, θέτουμε

$$e_B d_B - 1 = g_0, (g_0, e_A) = h_0.$$

και ορίζουμε επαγωγικά, για $i \geq 1$,

$$g_i = g_{i-1}/h_{i-1}, h_i = (g_i, e_A).$$

Για $h_i \geq 2$ (αν $h_i = 1$ τότε και $\alpha = g_i$), έχουμε $g_{i+1} \leq g_i/2$. Αυτό σημαίνει ότι βρίσκουμε $h_i = 1$ σε γραμμικό αριθμό βημάτων. Ο χρήστης Β μπορεί τώρα να υπολογίσει c και b με τον Εκτεταμένο Ευκλείδειο Αλγόριθμο έτσι ώστε:

$$c\alpha + be_A = 1$$

Σημειώνουμε ότι $\phi(n)$ διαιρεί το α γιατί $\alpha t = k\phi(n)$ και $(t, \phi(n)) = 1$ (αυτό έπεται από το ότι $(e_A, \phi(n)) = 1$ ενώ όλοι οι παράγοντες του t διαιρούν το e_A και άρα το t είναι ένα γινόμενο από αριθμούς εκ των οποίων κανένας δεν έχει κοινό παράγοντα με το $\phi(n)$). Έτσι

$$be_A = 1 \pmod{\phi(n)},$$

και επομένως το $b \pmod{n}$ μπορεί να χρησιμοποιηθεί ως d_A.

Επίθεση μικρού δημόσιου εκθέτη

Όπως αναφέραμε στην εισαγωγή είναι δυνατόν να χρησιμοποιεί ως δημόσιος εκθέτης στο RSA η τιμή $e = 3$ για να μειωθεί το κόστος κρυπτογράφησης. Αυτό όμως έχει την εξής αρνητική συνέπεια η οποία μπορεί να φανεί με το παρακάτω παράδειγμα:
Έστω ότι χρησιμοποιούνται τρία δημόσια κλειδιά $k_1 = (3, n_1), k_2 = (3, n_2), k_3 = (3, n_3)$ για την κρυπτογράφηση του ίδιου μηνύματος m. Δηλαδή

- $c_1 = \texttt{Encrypt}(k_1, m) = m^3 \bmod N_1$
- $c_2 = \texttt{Encrypt}(k_2, m) = m^3 \bmod N_2$

- $c_3 = \texttt{Encrypt}(k_3, m) = m^3 \bmod N_3$

Ο αντίπαλος $\mathcal{A}$ όμως μπορεί να το εκμεταλλευτεί αυτό, ως εξής:

- Αρχικά κάνει χρήση του CRT για υπολογισμό του $X = m^3 \bmod n_1 n_2 n_3$
- Αλλά $m^3 < N_1 N_2 N_3$ που σημαίνει ότι πολύ απλά μπορεί να υπολογίσει το μήνυμα ως $m = \sqrt[3]{X}$.

Επίθεση μικρού ιδιωτικού εκθέτη

Στον αντίποδα της προηγούμενης ενότητας ο Michael Wiener πρότεινε [13] μία επίθεση στο RSA που εκμεταλλεύεται μεγάλες τιμές του e, δηλαδή μικρές τιμές του d. Τέτοιες τιμές έχουν ως στόχο την πιο αποδοτική διενέργεια πράξεων που σχετίζονται με την χρήση του RSA σε συστήματα ψηφιακών υπογραφών (βλ. 7.3). Στην επίθεση αυτή υποθέτουμε ότι $d < \frac{1}{3} n^{\frac{1}{4}}$. Για να την κατανοήσουμε πλήρως παραθέτουμε χωρίς απόδειξη το παρακάτω θεώρημα:

Θεώρημα 6.12. *Έστω $x \in \mathbb{R}$. Αν $|x - \frac{a}{b}| < \frac{1}{2b^2}$ τότε το κλάσμα $\frac{a}{b}$ εμφανίζεται στην προσέγγιση με συνεχή κλάσματα του x.*

Αρχικά παρατηρούμε ότι $n \approx \phi(n)$:

$$n - \phi(n) = pq - (p-1)(q-1) = p + q - 1 < 3\sqrt{n}$$

Όπως είναι γνωστό ο $\mathcal{A}$ γνωρίζει το e και ότι $\exists k : ed = 1 + k\phi(n)$ ή ισοδύναμα ότι:

$$|\frac{e}{\phi(n)} - \frac{k}{d}| = \frac{1}{d}$$

Όμως έχουμε $n \approx \phi(n)$, οπότε μπορούμε να αντικαταστήσουμε:

$$|\frac{e}{n} - \frac{k}{d}| = |\frac{ed - kn}{dn}|$$

Αν προσθαφαιρέσουμε στον παραπάνω αριθμητή την ποσότητα $k\phi(n)$ προκύπτει ότι:

$$|\frac{e}{n} - \frac{k}{d}| = |\frac{1 - k(n - \phi(n))}{dn}| =$$
$$|\frac{1 - k(n - \phi(n))}{dn}| < \frac{3k\sqrt{n}}{dn} = \frac{3k}{d\sqrt{n}}$$

Από την υπόθεση για το μέγεθος του d έχουμε:

$$|\frac{e}{n} - \frac{k}{d}| < \frac{1}{2d^2}$$

Επειδή $gcd(k, d) = 1$ το κλάσμα k/d είναι απλοποιημένο, και κατά συνέπεια θα εμφανίζεται στην προσέγγιση του e/n με συνεχή κλάσματα.

Έτσι μπορούμε να βρούμε το d, κρυπτογραφώντας ένα τυχαίο μήνυμα m και υψώνοντας το κρυπτοκείμενο σε κάθε έναν από τους παρονομαστές της προσέγγισης του e/n με συνεχή κλάσματα για να δούμε αν αποκρυπτογραφείται σωστά.

Κλείνοντας την παραπάνω ενότητα, αξίζει να επισημάνουμε ότι έτσι κι αλλιώς μικρές τιμές του d δεν είναι σε καμία περίπτωση αποδεκτές, γιατί θα έχουν γρήγορη επιτυχία σε επιθέσεις εξαντλητικής αναζήτησης. Η επιθέση που περιγράψαμε, ανεβάζει την ελάχιστη τιμή του d σε $N^{1/4}/3$ ενώ έχουν προταθεί και βελτιώσεις που βασίζονται στη θεωρία των δικτυωτών (lattices), η οποία περιγράφεται στην ενότητα 12.4.

6.4.4 Μερική ανάκτηση πληροφοριών στο RSA

Γενικά ένα κρυπτοσύστημα μπορεί να θεωρείται ασφαλές, υπό την έννοια ότι είναι απρόσιτο να αποκρυπτογραφηθεί ολόκληρο το απλό κείμενο, όμως ένας κρυπταναλυτής ίσως να μπορεί να αποσπάσει κάποια πληροφορία [5] για το απλό κείμενο, (όπως πχ. το τελευταίο bit (parity bit) του κρυπτογραφημένου τμήματος), χωρίς να χρειάζεται να να το αποκρυπτογραφήσει ολόκληρο. Αυτό συνήθως λέγεται *μερική πληροφορία* κάτι που θεωρείται πολύ σημαντικό στην κρυπτογραφία.

Για να αποδείξουμε ότι ένα συγκεκριμένο κρυπτοσύστημα δεν αποκαλύπτει κάποια συγκεκριμένη κρυπταναλυτική πληροφορία, συνήθως αποδεικνύουμε ότι ένας αλγόριθμος που υπολογίζει την εν λόγω μερική πληροφορία, μπορεί να χρησιμοποιηθεί για να σχεδιαστεί ένας άλλος αλγόριθμος ο οποίος σπάει το κρυπτοσύστημα. Αυτό αποδεικνύει ότι οποτεδήποτε διαρρέει από το κρυπτοσύστημα τέτοιου είδους μερική κρυπταναλυτική πληροφορία, το σύστημα μπορεί τελείως να παραβιαστεί (με τον τρόπο αυτό μπορεί κανείς να αποδείξει ότι συγκεκριμένα μέρη του συστήματος είναι τόσο δύσκολο να προσβληθούν όσο και ολόκληρο το σύστημα).

Τα ακόλουθα είναι παραδείγματα μερικής πληροφορίας:

1. δεδομένου ενός κρυπτοκειμένου $y = x^e \bmod N$, να μπορεί κανείς να υπολογίσει τη δυαδική ισοτιμία του απλού κειμένου x, δηλαδή το τελευταίο bit του x. (Με το $parity_{n,e}(y)$ συμβολίζουμε το τελευταίο bit του x.)

[5] Στο σύστημα RSA, για παράδειγμα, τουλάχιστον 1 bit διαρρέει: η τιμή του συμβόλου Jacobi του απλού κειμένου

2. δεδομένου ενός κρυπτοκειμένου $y = x^e \bmod n$, να μπορεί κανείς να καθορίσει αν ισχύει το $0 \leq x \leq n/2$ ή το $N/2 < x \leq n-1$. (Ορίζουμε συνάρτηση θέσης (location function) $loc_{n,e}(y)$ να είναι 0 αν $0 \leq x \leq n/2$ και 1 σε κάθε άλλη περίπτωση.)

Τυπικά, η *συνάρτηση θέσης* και η συνάρτηση δυαδικής ισοτιμίας που αναφέρθηκαν παραπάνω ορίζονται ως εξής:

$$loc_{N,e}(x^e \bmod n) = \begin{cases} 0, & x \leq n/2 \\ 1, & x > n/2 \end{cases}$$
$$parity_{N,e}(x^e \bmod n) = \begin{cases} 0, & x \bmod 2 = 0 \\ 1, & x \bmod 2 = 1 \end{cases}$$

Μπορεί να αποδειχθεί για το σύστημα RSA ότι κάθε αλγόριθμος που δίνει τις παραπάνω μερικές πληροφορίες ($parity_{n,e}$ ή $loc_{n,e}$), μπορεί να χρησιμοποιηθεί ως μαντείο (oracle) για έναν αλγόριθμο που σπάει το κρυπτοσύστημα RSA (υπολογίζει δηλαδή ολόκληρο το απλό κείμενο x).

Πρώτον, ο υπολογισμός του $parity_{n,e}$ είναι πολυωνυμικά ισοδύναμος με τον υπολογισμό του $loc_{N,e}$: (Με το $E_{n,e}(x)$ ορίζουμε την κρυπτογράφηση του x σε y: $y = E_{n,e}(x) = x^e \bmod n$ και με το $D_{n,e}(y) = x$ την αποκρυπτογράφηση)

$$loc_{n,e}(y) = parity_{n,e}(y \times E_{n,e}(2) \bmod n)$$
$$parity_{n,e}(y) = loc_{n,e}(y \times E_{n,e}(2^{-1}) \bmod n)$$

χρησιμοποιώντας την ιδιότητα

$$E_{n,e}(x_1)E_{n,e}(x_2) = E_{n,e}(x_1 x_2).$$

Ο αλγόριθμος για την εύρεση του απλού κειμένου $x = D_{n,e}(y)$ βασίζεται στην τεχνική της δυαδικής αναζήτησης, και χρησιμοποιεί την $loc_{n,e}$ ως μαντείο:

Σε κάθε βήμα του αλγόριθμου υπολογίζουμε:

$$y_i = loc_{n,e}(y_{i-1} \cdot E_{n,e}(2)) = loc_{n,e}(y \cdot (E_{n,e}(2))^i) = loc_{n,e}(E_{n,e}(x \cdot 2^i))$$

όπου $0 \leq i \leq \lfloor log_2 n \rfloor$ και $y_0 = E_{n,e}(x) = y$ (το κρυπτογραφημένο κείμενο, (ciphertext).

Παρατηρούμε ότι

$$loc_{n,e}(E_{n,e}(x)) = 0 \Leftrightarrow x \in \left[0, \tfrac{n}{2}\right)$$
$$loc_{n,e}(E_{n,e}(2x)) = 0 \Leftrightarrow x \in \left[0, \tfrac{n}{4}\right) \cup \left[\tfrac{n}{2}, \tfrac{3n}{4}\right)$$
$$loc_{n,e}(E_{n,e}(4x)) = 0 \Leftrightarrow x \in \left[0, \tfrac{n}{8}\right) \cup \left[\tfrac{n}{4}, \tfrac{3n}{8}\right) \cup \left[\tfrac{n}{2}, \tfrac{5n}{8}\right) \cup \left[\tfrac{3n}{4}, \tfrac{7n}{8}\right)$$

Με τον τρόπο αυτό μπορούμε να υπολογίσουμε το x σε $\lfloor log_2 N \rfloor$ βήματα.

Έτσι, αποδείξαμε το ακόλουθο θεώρημα:

Θεώρημα 6.13 (Goldwasser, Micali, Tong). *(βλ. [7]) Για κάθε στιγμιότυπο N, e του RSA, οι παρακάτω προτάσεις είναι ισοδύναμες:*

1. *Υπάρχει ένας αποδοτικός αλγόριθμος A τέτοιος ώστε $A(x^e \bmod N) = x$, για κάθε $x \in \mathbb{Z}_n^*$*
2. *Υπάρχει ένας αποδοτικός αλγόριθμος που υπολογίζει την συνάρτηση $parity_{N,e}$*
3. *Υπάρχει ένας αποδοτικός αλγόριθμος που υπολογίζει την συνάρτηση $loc_{N,e}$*

6.5 Το κρυπτοσύστημα ElGamal

6.5.1 Ανταλλαγή Κλειδιού Diffie-Hellman

Στην κλασική τους εργασία [4], η οποία έθεσε τα θεμέλια της κρυπτογραφίας δημοσίου κλειδιού, οι Diffie και Hellman, ασχολήθηκαν με το πρόβλημα της ανταλλαγής κλειδιού. Συγκεκριμένα δύο οντότητες **A**, **B** θέλουν να συμφωνήσουν σε ένα κοινό κλειδί, ώστε να επικοινωνήσουν χρησιμοποιώντας κρυπτογραφία (πχ. με ένα συμμετρικό κρυπτοσύστημα σαν το DES. Ο μόνος τρόπος επικοινωνίας τους είναι μέσω ενός δημοσίου καναλιού, στο οποίο μπορεί να υπάρχουν ωτακουστές, άρα δεν μπορούν να απλά να συμφωνήσουν στο κλειδί χωρίς να το μάθει οποιοσδήποτε έχει πρόσβαση στο κανάλι. Το πρόβλημα λύθηκε χρησιμοποιώντας το παρακάτω απλό πρωτόκολλο, το οποίο βασίζεται στο πρόβλημα του Διακριτού Λογαρίθμου (βλ. ορισμό 4.8).

Η ασφάλεια του παραπάνω πρωτοκόλλου βασίζεται στα παρακάτω προβλήματα:

Υπολογιστικό Πρόβλημα των Diffie-Hellman Computational Diffie Hellman Problem (CDH) *Δίνονται*: ένας πρώτος αριθμός p, ένας γεννήτορας g του $\mathbb{Z}_p^*$ και τα στοιχεία $g^\alpha \pmod p, g^\beta \pmod p \in \mathbb{Z}_p^*$.

Ζητείται: Να βρεθεί το $g^{\alpha\beta} \pmod p$

Πρόβλημα Απόφασης των Diffie-Hellman Decisional Diffie Hellman Problem (DDH) *Δίνονται*: ένας πρώτος αριθμός p, ένας γεννήτορας g του $\mathbb{Z}_p^*$ και τα στοιχεία $g^\alpha \pmod p, g^\beta \pmod p, g^c \pmod p$.

Ζητείται: Ισχύει $c = \alpha\beta$

Τα παραπάνω προβλήματα όπως είναι φανερό σχετίζονται με το DLOG. Συγκεκριμένα κανένα δεν είναι πιο δύσκολο από αυτό και ισχύει το παρακάτω θεώρημα:

1. Αρχικά δημοσιεύεται ένας πρώτος αριθμός p, κατάλληλα επιλεγμένος (ώστε να καθίσταται «αδύνατη» η επίλυση του αντίστοιχου DLOG) και ένας γεννήτορας g του $\mathbb{Z}_p^*$.
2. Η **A** επιλέγει έναν τυχαίο ακέραιο $\alpha \in \mathbb{Z}_p^*$ που τον γνωρίζει μόνο αυτή και στέλνει στον **B** το μήνυμα: $y_\alpha = g^\alpha \pmod p$.
3. Ο **B** επιλέγει έναν τυχαίο ακέραιο $\beta \in \mathbb{Z}_p^*$ που τον γνωρίζει μόνο αυτός και στέλνει στην **A** το μήνυμα: $y_\beta = g^\beta \pmod p$.
4. Ο **B** λαμβάνει το $g^\alpha \pmod p$ και υπολογίζει το $K = (g^\alpha)^\beta \pmod p$.
5. Η **A** λαμβάνει το $g^\beta \pmod p$ και υπολογίζει το $K = (g^\beta)^\alpha \pmod p$.
6. Οι **A**, **B** συμφώνησαν στο κοινό κλειδί. K

Σχήμα 6.3: Πρωτόκολλο Ανταλλαγής Κλειδιού Diffie-Hellman

Θεώρημα 6.14. $\texttt{DDHP} \leq \texttt{CDHP} \leq \texttt{DLOG}$

Απόδειξη. Πράγματι, αν θέσω $x = g^\alpha \pmod p$ και $y = g^\beta \pmod p$, τότε $\alpha = \log_g x$ και $\beta = \log_g y$. Αν επομένως μπορούμε να λύσουμε το DLOG, μπορούμε να υπολογίσουμε τα (α, β) άρα και το $g^{\alpha\beta} \pmod p$.

Αν τώρα μπορούμε να λύσουμε το CDHP μπορούμε να υπολογίσουμε το $g^{\alpha\beta} \pmod p$ και να ελέγξουμε αν είναι ίσο με το $g^c \pmod p$. □

Το αντίστροφο δεν έχει αποδειχθεί.

Θεώρημα 6.15. *Αν ισχύει η υπόθεση DDH, τότε το πρωτόκολλο ανταλλαγής κλειδιού Diffie-Hellman είναι ασφαλές.*

Απόδειξη. Όπως αναφέραμε στην εισαγωγή 1.4.4 θα αποδείξουμε κατασκευαστικά το αντιθετοαντίστροφο. Δηλαδή αν το πρωτόκολλο Diffie-Hellman δεν είναι ασφαλές, τότε μπορώ να κατασκευάσω έναν PPT αλγόριθμο που 'σπάει' την υπόθεση DDH.

Αν το πρωτόκολλο Diffie Hellman δεν είναι ασφαλές, σημαίνει ότι υπάρχει ένας PPT αλγόριθμος $\mathcal{A}$ οποίος μπορεί παρακολουθώντας *παθητικά* την ανταλλαγή κλειδιού να αποκτήσει μη αμελητέο πλεονέκτημα στο να μαντέψει μία ιδιότητα του $K = g^{\alpha\beta} \pmod p$ από τα μηνύματα (y_α, y_β). Η ιδιότητα αυτή μπορεί να μοντελοποιηθεί με μία συνάρτηση $pred$, οπότε έχουμε:

$$Prob[\mathcal{A}\ (y_\alpha, y_\beta) = pred(K)] > \tfrac{1}{2} + \text{non-negl}(\eta)$$

όπου η η παράμετρος ασφαλείας του πρωτοκόλλου.

Με βάση αυτόν τον αντίπαλο κατασκευάζουμε τον $\mathcal{B}$ ως εξής:

- Είσοδος: $g^\alpha \pmod p, g^\beta \pmod p, g^c \pmod p$
- Κλήση του $\mathcal{A}$ με εισόδους τα y_α, y_β
- Υπολογισμός και επιστροφή:

$$\begin{cases} 1, & \mathcal{A}(y_\alpha, y_\beta) = pred(y_c) \\ 0, & \mathcal{A}(y_\alpha, y_\beta) \neq pred(y_c) \end{cases}$$

Ο $\mathcal{B}$ έχει μη αμελητέα πιθανότητα να σπάσει την DDH, εφόσον $K = g^{\alpha\beta} \pmod p$. Άρα:
$Prob[\mathcal{B}\ (y_\alpha, y_\beta, K) = 1] = Prob[\mathcal{B}\ (y_\alpha, y_\beta) = pred(K)] > \frac{1}{2} + \text{non-negl}(\eta)$
Αντίθετα αν του δοθεί ένα τυχαίο στοιχείο $g^c \pmod p$ τότε έχουμε:
$Prob[\mathcal{B}\ (y_\alpha, y_\beta, y_c) = 1] = Prob[\mathcal{B}\ (y_\alpha, y_\beta) = pred(y_c)] = \frac{1}{2}$
Η στατιστική απόσταση των δύο κατανομών θα είναι τουλάχιστον:
$\frac{1}{2} + \text{non-negl}(\eta) - \frac{1}{2} = \text{non-negl}(\eta)$, δηλαδή σπάει την DDH με μη αμελητέα πιθανότητα. □

Παράδειγμα Θεωρούμε ότι η Αλίκη και ο Μπομπ θέλουν να συμφωνήσουν σε ένα κοινό κλειδί.

1. Η Αλίκη επιλέγει τα $p = 2357$, έναν γεννήτορα του $\mathbb{Z}_p^*$, τον $\alpha = 2$ και το τυχαίο $x = 135$ και στέλνει στον Μπομπ την τριάδα:

$$\left(2357, 2, 2^{135} \bmod 2357\right) = (2357, 2, 641)$$

2. Ο Μπομπ επιλέγει ένα τυχαίο $y = 111$ και στέλνει στην Αλίκη το μήνυμα:

$$\left(2^{111} \bmod 2357\right) = 1238$$

3. Ο Μπομπ λαμβάνει το μήνυμα της Αλίκης και υπολογίζει το κλειδί:

$$K = 641^{111} \bmod 2357 = 787$$

4. Η Αλίκη λαμβάνει το μήνυμα του Μπομπ και υπολογίζει το κλειδί:

$$K = 1238^{135} \bmod 2357 = 787$$

6.5.2 Κρυπτοσύστημα Δημοσίου Κλειδιού ElGamal

Το κρυπτοσύστημα *ElGamal* [5] μετατρέπει το σύστημα ανταλλαγής κλειδιών Diffie-Hellman σε κρυπτοσύστημα δημοσίου κλειδιού. Είναι ένα από τα πιο γνωστά κρυπτοσυστήματα δημοσίου κλειδιού. Λειτουργεί ως εξής:

- **Δημιουργία Κλειδιών** `KeyGen`
  - Επιλογή δύο μεγάλων πρώτων p, q ώστε $q \mid (p-1)$ και ενός γεννήτορα g της υποομάδας τάξης q του $\mathbb{Z}_p^*$ `G`.
  - Τυχαία επιλογή $x \in_R \mathbb{Z}_q$
  - Υπολογισμός $y = g^x \pmod{p}$
  - Επιστροφή δημοσίου και ιδιωτικού κλειδιού $(pk = y, sk = x)$
- **Κρυπτογράφηση** `Encrypt`
  - Τυχαία επιλογή $r \in_R \mathbb{Z}_q$
  - Υπολογισμός $G = g^r \pmod{p}$
  - Υπολογισμός $M = m \cdot y^r \pmod{p}$
  - Επιστροφή $c = (G, M)$
- **Αποκρυπτογράφηση** `Decrypt`
  - Δίνεται το κρυπτοκείμενο (G, M).
  - Χρήση μυστικού κλειδιού και επιστροφή $M/G^x \pmod{p}$

Στο παράδειγμα που ακολουθεί γίνεται μία επίδειξη της λειτουργίας του κρυπτοσυστήματος ElGamal για μικρές παραμέτρους.

- *Δημιουργία Κλειδιού* `KeyGen`: Η Αλίκη επιλέγει τον $p = 2579$ και τον γεννήτορα $= 2$ του $\mathbb{Z}_{2579}^*$. Στη συνέχεια επιλέγει το ιδιωτικό της κλειδί $x = 765$ και υπολογίζει το:

$$g^x \bmod p = 2^{765} \bmod 2579 = 949.$$

  Το δημόσιο κλειδί της Αλίκης είναι το:

$$(p, g, g^x) = (2579, 2, 949).$$

- *Κρυπτογράφηση* `Encrypt`: Έστω τώρα ότι ο Μπομπ θέλει να στείλει στην Αλίκη το μήνυμα $m = 1299$. Για να το κρυπτογραφήσει επιλέγει ένα τυχαίο k, έστω $k = 853$, και υπολογίζει τα :

$$\gamma = 2^{853} \bmod 2579 = 435.$$

και

$$\delta = 1299 \cdot 949^{853} \bmod 2579 = 2396.$$

Στέλνει λοιπόν στην Αλίκη το μήνυμα $c = (\gamma, \delta) = (435, 2396)$.

- *Αποκρυπτογράφηση* `Decrypt`: Η Αλίκη λαμβάνει το c και υπολογίζει το:

$$\gamma^{p-1-a} = 435^{1813} \bmod 2579 = 1980$$

και τέλος ανακτά το m υπολογίζοντας το:

$$m = (1980 \cdot 2396) \bmod 2579 = 1299.$$

Παρατηρήσεις Η *Συνάρτηση Κρυπτογράφησης* του ElGamal είναι *πιθανοτική (randomized)*. Δηλαδή το αποτέλεσμα εξαρτάται από τον τυχαία επιλεγμένο ακέραιο r. Έτσι από ένα απλό κείμενο (plaintext) μπορούν να προκύψουν πολλά διαφορετικά κρυπτογραφημένα κείμενα (ciphertexts), γεγονός που αυξάνει την ασφάλειά του, αρκεί να επιλέγονται διαφορετικοί τυχαίοι k για κάθε κρυπτογραφημένο κείμενο που θέλουμε να στείλουμε. Πιο συγκεκριμένα αποδεικνύεται το παρακάτω θεώρημα:

Θεώρημα 6.16. *Το ElGamal διαθέτει την ιδιότητα IND-CPA, αν ισχύει η υπόθεση DDH.*

Απόδειξη. Θα προχωρήσουμε με τον ίδιο τρόπο με το 6.15. Έστω ότι το ElGamal δεν διαθέτει την ιδιότητα IND-CPA. Αυτό σημαίνει ότι υπάρχει αντίπαλος $\mathcal{A}$, ο οποίος μπορεί να νικήσει το παιχνίδι 1.4.4 με μη αμελητέα πιθανότητα. Δηλαδή:

$$Prob[CPA(1^\eta) = 1] > \tfrac{1}{2} + \text{non-negl}(\eta)$$

Θα κατασκευάσουμε έναν αντίπαλο $\mathcal{B}$ ο οποίος έχοντας ως είσοδο μία τριάδα, θα μπορεί να ξεχωρίσει με μη αμελητέα πιθανότητα αν είναι τυχαία ή αν ικανοποιεί την υπόθεση DDH. Ο $\mathcal{B}$ εσωτερικά θα χρησιμοποιεί τον $\mathcal{A}$ λειτουργώντας ως $\mathcal{C}$ στο παιχνίδι CPA:

- Είσοδος: $g^\alpha \pmod p, g^\beta \pmod p, g^c \pmod p$
- Με βάση τα παραπάνω κατασκευάζουμε την είσοδο του παιχνιδιού CPA, θέτοντας το δημόσιο κλειδί $y = g^\alpha \pmod p$

- Όταν ο $\mathcal{A}$ εξάγει δύο μηνύματα διαλέγουμε ένα τυχαίο $b \in \{0, 1\}$, και 'κρυπτογραφούμε' το M_b με τυχαιότητα το δεύτερο στοιχείο της τριάδας. Δηλαδή στέλνουμε: $(G = g^{\beta} \pmod p), M_b \cdot g^c \pmod p)$

- Ο $\mathcal{A}$ επιστρέφει την τιμή του b*.

- Αν μάντεψε σωστά τότε επιστρέφουμε 1, αλλιώς 0.

Στην περίπτωση που η είσοδος είναι τριάδα DH, τότε $g^c = (g^{\alpha})^{\beta} = y^{\beta} \pmod p$ δηλαδή ο $\mathcal{A}$ θα λάβει ένα έγκυρο κρυπτοκείμενο ElGamal. Κατά συνέπεια η πιθανότητα να μαντέψει σωστά είναι τουλάχιστον: $1/2 +$ non-negl(η). Σε διαφορετική περίπτωση, ο $\mathcal{A}$ θα πρέπει να μαντέψει τυχαία, οπότε η πιθανότητα να μαντέψει σωστά είναι $\frac{1}{2}$. Άρα ο $\mathcal{B}$ έχει πιθανότητα επιτυχίας τουλάχιστον $textnon - negl(\eta)$, που σημαίνει ότι μπορεί να ξεχωρίσει μία DH τριάδα από μία τυχαία με μη αμελητέα πιθανότητα. □

Από την άλλη το ElGamal δεν διαθέτει την ιδιότητα IND-CCA. Αυτό φαίνεται ως εξής: έστω ότι ο $\mathcal{A}$ μπορεί να αποκρυπτογραφήσει μηνύματα επιλογής του, εκτός ενός κρυπτοκειμένου $c = (G, M) = (g^r, m_b h^r)$, το οποίο θέλει να αποκρυπτογραφήσει. Αυτό μπορεί να επιτευχθεί ως εξής:

- Κατασκευή $c' = (G', M') = (Gg^{r'}, Mah^{r'}) = (g^{r+r'}, m_b a h^{r+r'})$, όπου a επιλέγεται από τον $\mathcal{A}$

- Η αποκρυπτογράφηση $\frac{M'}{G'^x}$ δίνει το am_b και κατά συνέπεια το m_b

- Αν $m_b = m_0$ επιστρέφει $b^* = 0$ αλλιώς επιστρέφει $b^* = 1$

Το εκθετικό ElGamal Μία παραλλαγή του κρυπτοσυστήματος ElGamal υπολογίζει την κρυπτογράφηση g^m αντί για την κρυπτογράφηση του m. Δηλαδή, το κρυπτοκείμενο είναι $c = (G, M) = g^r, g^m \cdot y^r$. Αυτό σημαίνει ότι κατά η αποκρυπτογράφηση θα επιστρέψει το στοιχείο g^m. Για να υπολογίσουμε το m θα πρέπει να επιλύσουμε ένα πρόβλημα διακριτού λογαρίθμου.

Παρά τη δυσκολία αυτή, το εκθετικό ElGamal χρησιμοποιείται αρκετά στη βιβλιογραφία σε περίπτωση που θέλουμε ένα κρυπτοσύστημα με προσθετικές ομομορφικές ιδιότητες (βλ. 9.5), αλλά και σε περίπτωση που ο διακριτός λογάριθμος μπορεί να βρεθεί σχετικά γρήγορα (λόγω για παράδειγμα σχετικά λίγων πιθανών τιμών) όπως στην ενότητα (βλ. 11.1.2).

6.6 Λοιπά κρυπτοσυστήματα Δημοσίου Κλειδιού

6.6.1 Το κρυπτοσύστημα Rabin

Το κρυπτοσύστημα Rabin, είναι επίσης ένα κρυπτοσύστημα δημοσίου κλειδιού, και η κρυπτανάλυση του (σε αντίθεση με το RSA) είναι εξίσου δύσκολη με την παραγοντοποίηση (ισοδύναμη με το πρόβλημα παραγοντοποίησης). Δηλαδή το κρυπτοσύστημα Rabin είναι υπολογιστικά απρόσιτο (intractable), αν το $N = pq$ δεν μπορεί να παραγοντοποιηθεί με αποδοτικό τρόπο.Η κρυπτογράφηση E για το κρυπτοσύστημα αυτό δεν είναι συνάρτηση 1-1 και επομένως η αποκρυπτογράφηση είναι *διφορούμενη (ambiguous)*.

Περιγραφή

- **Δημιουργία Κλειδιών**: Ακολουθείται μία διαδικασία παρόμοια με το RSA ώστε να επιλέγουν δύο διαφορετικοί περιττοί πρώτοι αριθμοί p, q, οι οποίοι συνιστούν το ιδιωτικό κλειδί της κρυπτογράφησης. Το γινόμενό τους $N = p \cdot q$ δημοσιοποιείται. Επίσης δημοσιοποιείται και ένα τυχαίο $b \in \mathbb{Z}_n$.

- **Κρυπτογράφηση** Το μήνυμα m κωδικοποιείται στο $\mathbb{Z}_n$ και η συνάρτηση κρυπτογράφησης είναι:

$$Enc_{N,b}(m) = (m \cdot (m + b)) \bmod N$$

- **Αποκρυπτογράφηση** Η συνάρτηση **αποκρυπτογράφησης** $Dec_{N,b}$ παρέχει για κάθε κρυπτοκείμενο c, μια λύση u (από τις συνολικά τέσσερις) της δευτεροβάθμιας εξίσωσης $m \cdot (m + b) = c \pmod N$. Αποδεικνύεται ότι η ισοτιμία $x \cdot (x+b) = y \pmod N$ έχει μια λύση αν και μόνο αν η η ισοτιμία $x'^2 = y + b^2/4 \pmod N$ έχει λύση (14). Έτσι μπορούμε να γράψουμε:

$$D_{N,b}(y) = \sqrt{\frac{b^2}{4} + y} - \frac{b}{2}$$

(προσοχή: η τετραγωνική ρίζα είναι modulo N και δεν υπολογίζεται εύκολα...).

Η αποκρυπτογράφηση είναι εύκολη αν οι πρώτοι αριθμοί p, q είναι γνωστοί. Πράγματι, για ένα κρυπτογραφημένο μήνυμα c (τέτοιο ώστε τα p, q να μην διαιρούν το c) μπορούμε να υπολογίσουμε τις ρίζες r, s των ισοτιμιών $m \cdot (m + b) = c \pmod p$ και $m \cdot (m + b) = y \pmod q$ αντίστοιχα. Χρησιμοποιώντας τον ευκλείδειο αλγόριθμο υπολογίζουμε τους ακεραίους k, l

τέτοιους ώστε $k \cdot p + l \cdot q = 1$. Τότε η $lqr + kps$ είναι μια λύση της ισοτιμίας $m \cdot (m + b) = c \pmod N$.

Ουσιαστικά, η αποκρυπτογράφηση του Rabin συνίσταται στην εύρεση ριζών της $m'^2 = c' \pmod N$, όπου $c' = c + \frac{b^2}{4}$. Επομένως, αν τα p και q είναι γνωστά, αρκεί να βρούμε τις ρίζες των $m'^2 = c' \pmod p$ και $m'^2 = c' \pmod q$.

Παρατηρήσεις : Ισχύουν τα εξής:

- Αν $p = 3 \pmod 4$, τότε υπάρχει ένας απλός τύπος για τον υπολογισμό των τετραγωνικών ριζών modulo p. Υποθέτουμε ότι το R είναι τετραγωνικό υπόλοιπο και $p = 3 \pmod 4$. Τότε έχουμε:

$$(\pm R^{(p+1)/4})^2 = R^{(p+1)/2} = R^{(p-1)/2}C = R \pmod p$$

  Εδώ χρησιμοποιούμε το κριτήριο του Euler σύμφωνα με το οποίο αν το R είναι τετραγωνικό υπόλοιπο modulo p, τότε $R^{(p-1)/2} = 1 \pmod p$. Επομένως οι δύο τετραγωνικές ρίζες του R modulo p είναι $\pm R^{(p+1)/4} \pmod p$. Παρομοίως οι δύο τετραγωνικές ρίζες του R modulo q είναι $\pm R^{(q+1)/4} \pmod q$.

- Είναι ενδιαφέρον ότι για $p = 1 \pmod 4$ δεν υπάρχει γνωστός (αποδοτικός) ντετερμινιστικός αλγόριθμος για τον υπολογισμό των τετραγωνικών υπολοίπων modulo p. Υπάρχει όμως αποδοτικός πιθανοτικός αλγόριθμος (τύπου Las Vegas).

Παρακάτω αποδεικνύουμε ότι αν η παραγοντοποίηση του N δεν είναι γνωστή, τότε η αποκρυπτογράφηση του Rabin είναι τουλάχιστον εξίσου δύσκολη. Προσέξτε ότι, όπως φαίνεται παραπάνω, η αποκρυπτογράφηση του Rabin είναι ισοδύναμη με την εύρεση λύσης της ισοτιμίας $x^2 = y \pmod N$.

Θεώρημα 6.17 (Θεώρημα Παραγοντοποίησης του Rabin)**.** *Έστω N το γινόμενο δυο περιττών πρώτων αριθμών. Τότε οι επόμενες προτάσεις είναι ισοδύναμες:*

1. *Υπάρχει ένας αποδοτικός αλγόριθμος A τέτοιος ώστε για κάθε $y \in QR_N$ (δηλαδή για όλα τα τετραγωνικά υπόλοιπα (quadratic residues) modulo N), $A(N, y)$ είναι μια τυχαία λύση της ισοτιμίας $x^2 = y \pmod N$.*

2. *Υπάρχει αποδοτικός αλγόριθμος (τύπου Las Vegas) για την παραγοντοποίηση του N.*

Απόδειξη. Αρκεί να δείξουμε ότι $(1) \Rightarrow (2)$
Επιλέγουμε τυχαία έναν ακέραιο a τέτοιο ώστε $(a, N) = 1$ και έστω $y = a^2 \pmod N$. Αν $u = A(N, y)$, τότε και το a και το u είναι λύσεις της ισοτιμίας $x^2 = y \pmod N$. Αν ο $u \notin \{a, N - a\}$, τότε οι $(N, u \pm a)$ είναι οι παράγοντες p, q του N. Αν όμως $u \in \{a, N - a\}$ τότε διαλέγουμε ένα άλλο a και επαναλαμβάνουμε τη διαδικασία. Αφού το $u \notin \{a, N - a\}$ με πιθανότητα $1/2$, μετά από k προσπάθειες με πιθανότητα $1 - (\frac{1}{2})^k$ θα καταφέρουμε να παραγοντοποιήσουμε το N. □

6.6.2 Το κρυπτοσύστημα Paillier

Το κρυπτοσύστημα Paillier προτάθηκε στο [9]. Είναι ένα ενδιαφέρον κρυπτοσύστημα με την έννοια ότι το μήνυμα που κρυπτογραφείται βρίσκεται στον εκθέτη, όπως στο εκθετικό ElGamal (6.5.2) χωρίς όμως να υπάρχει η ανάγκη για υπολογισμό διακριτών λογαρίθμων. Φυσικά δεν ήταν το πρώτο με αυτή την ιδιότητα, οπότε έχει αξία η αναφορά στα κρυπτοσυστήματα που οδήγησαν σε αυτό.

Goldwasser-Micali Ο πρόγονος του Paillier παρουσιάστηκε στο [6] ως το πρώτο κρυπτοσύστημα με αποδείξιμη ασφάλεια. Μπορεί να θεωρηθεί ως κατασκευή *a proof of concept*, για τις έννοιες της πιθανοτικής κρυπτογράφησης και σημασιολογικής ασφάλειας, καθώς οι λειτουργίες γίνονται για κάθε bit ξεχωριστά.

- **Δημιουργία Κλειδιών**
  - Τυχαία και ανεξάρτητη επιλογή δύο μεγάλων πρώτων p, q
  - Υπολογισμός $n = p \cdot q$
  - Επιλογή $y \in \mathbb{Z}_n^*$ ώστε $gcd(y, n) = 1$. Επιπλέον δεν πρέπει να υπάρχει $x \in \mathbb{Z}_n^*$ ώστε $y = x^2 \pmod n$, δηλαδή το y δεν είναι *τετραγωνικό υπόλοιπο*.
  - Το δημόσιο κλειδί είναι N, y
  - Το ιδιωτικό κλειδί είναι p, q
- **Κρυπτογράφηση**
  - Για ένα μήνυμα $m = m_1 \cdots m_k$ επιλέγουμε τυχαίο για κάθε m_i τυχαίο r_i με $gcd(r_i, N) = 1$. Η κρυπτογράφηση παράγεται ανά bit ως:

$$c_i = \begin{cases} yr_i^2 \pmod n, & m_i = 1 \\ r_i^2 \pmod n, & m_i = 0 \end{cases}$$

ή πιο συμπυκνωμένα: $c_i = y^{m_i} r_i^2 \pmod{n}$

- **Αποκρυπτογράφηση**
  - Για το κρυπτοκείμενο $c = c_1 \cdots c_k$ η αποκρυπτογράφηση γίνεται ανα bit πάλι ως:

$$m_i = \begin{cases} 1, & c_i \notin QR_N \\ 0, & c_i \in QR_N \end{cases}$$

    Αυτή η λειτουργία μπορεί να γίνει εύκολα καθώς είναι γνωστή η παραγοντοποίηση του n.

Η ασφάλεια του [6] βασίζεται στο πρόβλημα των τετραγωνικών υπολοίπων, δηλαδή ότι είναι δύσκολο να διακρίνει κανείς αν κάποιο $x \in \mathbb{Z}_n^*$ είναι τετραγωνικό υπόλοιπα χωρίς την παραγοντοποίηση του n.

Κρυποσύστημα Benaloh Το επόμενο βήμα που οδήγησε στο Paillier δόθηκε στο [1], όπου γενικεύεται το [6] χρησιμοποιώντας r υπόλοιπα αντί για τετραγωνικά. Ένας αριθμός y είναι ένα r υπόλοιπο $\pmod{n}$ αν υπάρχει x ώστε $y = x^r \pmod{n}$. Όπως και πριν αν είναι γνωστή η παραγοντοποίηση του n, τότε υπάρχει πολυωνυμικός αλγόριθμος για την επίλυση του προβλήματος. Οι λεπτομέρειες του κρυπτοσυστήματος Benaloh δίνονται παρακάτω:

- **Δημιουργία Κλειδιών**
  - Επιλογή πρώτου r
  - Τυχαία επιλογή πρώτων p, q ώστε $r \mid (p-1)$ και $r \not| (q-1)$
  - Υπολογισμός $n = p \cdot q$
  - Επιλογή $y \in \mathbb{Z}_n^*$ με $gcd(y, n) = 1$ χωρίς να υπάρχει $x \in \mathbb{Z}_n^*$ ώστε $y = x^r \pmod{n}$.
  - Το δημόσιο κλειδί είναι N, y, r
  - Το ιδιωτικό κλειδί είναι p, q
- **Κρυπτογράφηση**
  - Για κάποιο μήνυμα m δημιουργούμε το $c = y^m x^r \pmod{n}$ όπου το x είναι τυχαίο.
- **Αποκρυπτογράφηση**

- Αφού η παραγοντοποίηση του n είναι γνωστή, τότε ο παραλήπτης μπορεί να υπολογίσει το $\phi(n) = (p-1)(q-1)$
- Στην συνέχεια υψώνει υπολογίζει :

$$u = Enc(m)^{\frac{\phi(n)}{r}} = (y^m x^r)^{\frac{\phi(n)}{r}} = y^{m\frac{\phi(n)}{r}} x^{\phi(n)} = y^{m\frac{\phi(n)}{r}}$$

- Αν $u = 1$ τότε το μήνυμα είναι το $m = 0$
- Αν $u \neq 1$ τότε ο παραλήπτης $\forall t \in \{0, \cdots, r-1\}$ ελέγχει εάν

$$uEnc(t) = Enc(m)Enc(t) = Enc(m+t) = Enc(0) = 1$$

. Όταν ένα τέτοιο t βρεθεί, έχουμε την αποκρυπτογράφηση $m = -t \pmod{n}$.

- Φυσικά αυτή η εξαντλητική μέθοδος δεν μπορεί να δουλέψει ικανοποιητική για μεγάλα r ακόμα και αν χρησιμοποιηθούν βελτιωμένοι αλγόριθμοι (βλ.4.8).

Κρυποσύστημα Paillier Ας ασχοληθούμε τώρα με το [9] και την γενίκευση του ([3], [2]), που συνήθως αναφέρεται ως κρυπτοσύστημα Damgård-Jurik.

Τα τρία συστατικά του κρυπτοσυστήματος Paillier είναι:

- **Δημιουργία Κλειδιών**
  - Τυχαία επιλογή δύο μεγάλων πρώτων p, q ώστε $gcd(pq, (p-1)(q-1)) = 1$. Μπορεί εύκολα να αποδειχθεί ότι αυτή η ιδιότητα ικανοποιείται εύκολα αν οι p, q έχουν το ίδιο μήκος
  - Υπολογισμός $n = pq$
  - Υπολογισμός $\lambda = lcm(p-1, q-1) = \frac{(p-1)(q-1)}{gcd(p-1,q-1)}$ (Συνάρτηση του Carmichael). Ο υπολογισμός της είναι εύκολος αν γνωρίζουμε τα p, q ενώ διαθέτει τις πολύ σημαντικές ιδιότητες:
    * $\forall x \in \mathbb{Z}^*_{n^2} : x^{\lambda(n)} = 1 \pmod{n}$
    * $\forall x \in \mathbb{Z}^*_{n^2} : x^{n\lambda(n)} = 1 \pmod{n^2}$ καθώς $n\lambda(n) = \lambda(n^2)$
  - Επιλογή γεννήτορα $g \in \mathbb{Z}^*_{n^2}$ με τάξη πολλαπλάσιο του n.
  - Υπολογισμός αντιστρόφου $\mu = L(g^\lambda \pmod{n^2})^{-1} \pmod{n}$ όπου $L(x) = \frac{x-1}{n}$. Η παραπάνω συνάρτηση είναι πολύ σημαντική στο κρυπτοσύστημα του Paillier. Οι λειτουργίες της είναι:
    * Η $L()$ λαμβάνει στοιχεία ισοδύναμα με $1 \pmod{n}$
    * Στην συνέχεια *'λύνει'* το πρόβλημα διακριτού λογαρίθμου για αυτά και *'αποκρυπτογραφεί'*

    * Το αντίστροφο που υπολογίζει υπάρχει πάντα αν ο g είναι έγκυρος γεννήτορας
  - Το δημόσιο κλειδί είναι (n, g) ενώ το ιδιωτικό (λ, μ)

- **Κρυπτογράφηση:** $\mathbb{Z}_n \times \mathbb{Z}_n^* \to \mathbb{Z}_{n^2}^*$
  - Κωδικοποίηση m στο $\mathbb{Z}_n$
  - Επιλογή τυχαίου $r \in \mathbb{Z}_n^*$
  - Επιστροφή $c = Enc(m, r) = g^m r^n \pmod{n^2}$

- **Αποκρυπτογράφηση:** $\mathbb{Z}_{n^2}^* \to \mathbb{Z}_n$
  - Έστω ένα κρυπτοκείμενο $c \in \mathbb{Z}_{n^2}^*$
  - Υπολογισμός $m = L(c^\lambda \, mod \, n^2)\mu \, (mod \, n) = \frac{L(c^\lambda \, mod \, n^2)}{L(g^\lambda \, mod \, n^2)}(mod n)$

Η ασφάλεια του κρυπτοσυστήματος Paillier βασίζεται στο πρόβλημα Σύνθετων Υπολοίπων (Composite Residuosity) το οποίο είναι:
indexΠρόβλημα!Σύνθετων Υπολοίπων

Ορισμός 6.18 (CRP)**.** Δίνεται $n = pq$ και $z \in \mathbb{Z}_{n^2}^*$ και z είναι $n-residue$ module n^2. Υπάρχει $y \in \mathbb{Z}_{n^2}^*$ ώστε $z = y^n \pmod{n^2}$.

Μέχρι τώρα δεν έχει βρεθεί αποδοτικός αλγόριθμος για το CRP.

Το κρυπτοσύστημα Damgård-Jurik Στο [3],[2], δίνεται μία γενίκευση και μια απλοποίηση του Paillier που μπορούν να χρησιμοποιηθούν σε πρακτικές εφαρμογές. Πρώτα από όλα $\forall s \geq 1$ μπορούμε να ορίσουμε ένα κρυπτοσύστημα $\mathcal{CS}_s$:

Γενίκευση

- **Δημιουργία Κλειδιών:**
  - Επιλογή αποδεκτών p, q με μήκος η bits και υπολογισμός του $n = pq$
  - Επιλογή τυχαίου j με $gcd(j, n) = 1$ και τυχαίο $x \in \mathbb{Z}_{n^s}$ για υπολογισμό $g = (1 + n)^j x \pmod{n^{s+1}}$
  - Υπολογισμός $\lambda = lcm(p - 1, q - 1)$
  - Υπολογισμός d ώστε
    * $d = 1 \pmod{n} \in \mathbb{Z}_{n^s}^*$
    * $d = 0 \pmod{\lambda}$

  - Το δημόσιο κλειδί είναι (n, g) ενώ το ιδιωτικό είναι d

- **Κρυπτογράφηση** $\mathbb{Z}_{n^s} \times \mathbb{Z}_n^* \to \mathbb{Z}_{n^{s+1}}^*$
  - Κωδικοποίηση m στο $\mathbb{Z}_{n^s}$
  - Επιλογή τυχαίου $r \in \mathbb{Z}_n^*$
  - Υπολογισμός $c = Enc(m, r) = g^m r^{n^s} \pmod{n^{s+1}}$
- **Αποκρυπτογράφηση**
  - Δίνεται κρυπτοκείμενο $c \in \mathbb{Z}_{n^{s+1}}^*$
  - Υπολογισμός $c^d = (1+n)^{mjd} \pmod{n^s}$
  - Εξαγωγή mjd.Υπάρχει συγκεκριμένος αλγόριθμος γι' αυτό που παρουσιάζεται παρακάτω (figure 6.4)
  - Υπολογισμός $g^d = (1+n)^{jd} \pmod{n^s}$
  - Εξαγωγή jd
  - $m = \frac{mjd}{jd}$

Είναι προφανές ότι για $s = 1$, λαμβάνουμε το αρχικό κρυπτοσύστημα Paillier. Έτσι προκύπτουν και οι ιδιότητες ασφαλείας.

Θεώρημα 6.19. *$\forall s$ $\mathcal{CS}_s$ είναι μονής κατεύθυνσης αν το κρυπτοσύστημα Paillier ($\mathcal{CS}_1$) είναι μονόδρομο και και σημασιολογικά ασφαλής αν ισχύει η DCRA.*

Απλοποίηση Εδώ παρατηρούμε ότι το $g = (1+n)$ είναι έγκυρος γεννήτωρ. Έτσι μπορούμε να ορίσουμε το παρακάτω κρυπτοσύστημα:

- **Δημιουργία κλειδιών** Όπως και πριν έχουμε:
  - Το δημόσιο κλειδί είναι $n = pq$
  - Το ιδιωτικό κλειδί είναι $\lambda = lcm(p-1, q-1)$
- Το s μπορεί να επιλεγεί οποιαδήποτε στιγμή πριν την κρυπτογράφηση αρκεί $m < n^s$
- **Encryption** $\mathbb{Z}_{n^s} \times \mathbb{Z}_n^* \to \mathbb{Z}_{n^{s+1}}^*$
  - Επιλογή κατάλληλου s ώστε το μήνυμα m να μπορεί να κωδικοποιηθεί στο $\mathbb{Z}_{n^s}$
  - Επιλογή τυχαίου $r \in \mathbb{Z}_n^*$

– Επιστροφή $c = Enc(m, r) = (1+n)^m r^{n^s} \pmod{n^{s+1}}$

- **Decryption** $\mathbb{Z}^*_{n^{s+1}} \to \mathbb{Z}_{n^s}$

  – Το κρυπτοκείμενο είναι το $c \in \mathbb{Z}^*_{n^{s+1}}$. Μπορούμε να ανακαλύψουμε το s από το μήκος του c

  – Υπολογισμός $c^\lambda = (1+n)^{m\lambda \pmod{n^s}} r^{\lambda n^s} = (1+n)^{m\lambda} \pmod{n^{s+1}}$ καθώς $r^{\lambda n^s} = r^{\lambda n^{s+1} = 1 \pmod{n^{s+1}}}$

  – Εξαγωγή του $m\lambda$ από το $(1+n)^{m\lambda} \pmod{n^{s+1}}$. Για να το πετύχουμε αυτό θα πρέπει να χρησιμοποιήσουμε την συνάρτηση $L() \pmod{n^s}$, λαμβάνοντας υπόψιν ότι κάποιοι όροι θα είναι $0 \pmod{n^s}$

$$L((1+n)^{m\lambda} \pmod{n^{s+1}}) =$$
$$\frac{1-(1+n)^{m\lambda} \pmod{n^{s+1}}}{n} \pmod{n^s} =$$
$$m\lambda + \binom{m\lambda}{2} n + \cdots + \binom{m\lambda}{s} n^{s-1} \pmod{n^s}$$

Και εδώ δεν μπορούμε να εξάγουμε το $m\lambda$ τόσο εύκολα όσο στην περίπτωση του n^2. Κατά συνέπεια θα το εξάγουμε βήμα, βήμα. Πρώτα θα εξάγουμε την τιμή του $m\lambda \pmod{n}$, μετά του $m\lambda \pmod{n^2}$. Αυτή η διαδικασία μπορεί να παρομοιαστεί με την συσσώρευση των ψηφίων ενός αριθμού από LSB στο MSB. Για απλότητα θα αναφερθούμε στην τιμή που εξάγεται ως i. Έτσι αν έχουμε υπολογίσει το i_{j-1} τότε $i_j = i_{j-1} + kn^{j-1}$ όπου $k \in \{0, \cdots n-1\}$. Δηλαδή πηγαίνουμε στο επόμενο αριθμό προσθέτοντας ένα ψηφίο στα αριστερά:

$$L((1+n)^i \pmod{n^{j+1}}) =$$
$$i_j + \binom{i_j}{2} n + \cdots + \binom{i_j}{j} n^{j-1} \pmod{n^j} =$$
$$i_{j-1} + kn^{j-1} + \binom{i_{j-1}}{2} n + \cdots + \binom{i_{j-1}}{j} \pmod{n^j}$$

Το κρίσιμο σημείο είναι η αντικατάσταση $\binom{i_j}{t} n^{t-1} = \binom{i_{j-1}}{t} n^{t-1}$. Αυτό ισχύει γιατί:

$$\begin{aligned}
\binom{i_j}{t} n^{t-1} &= \\
n^{t-1} \prod_{x=1}^{t} \frac{i_j - (t-x)}{x} &= \\
\frac{n^t}{n} \prod_{x=1}^{t} \frac{i_j - (t-x)}{x} &= \\
\frac{1}{n} \prod_{x=1}^{t} \frac{n(i_j - (t-x))}{x} &= \\
\frac{1}{n} \prod_{x=1}^{t} \frac{n(i_{j-1} + kn^{j-1} - (t-x))}{x} &= \\
\frac{1}{n} \prod_{x=1}^{t} \frac{(ni_{j-1} + kn^j - n(t-x))}{x} &= \\
\frac{1}{n} \prod_{x=1}^{t} \frac{(ni_{j-1} - n(t-x))}{x} &= \\
\frac{n^t}{n} \prod_{x=1}^{t} \frac{(i_{j-1} - (t-x))}{x} &= \\
n^{t-1} \binom{i_{j-1}}{t} \pmod{n^j}
\end{aligned}$$

Κατά συνέπεια:

$L((1+n)^i \pmod{n^{j+1}}) = kn^{j-1} + i_{j-1} + \binom{i_{j-1}}{2}n + \cdots + \binom{i_{j-1}}{j} \pmod{n^j}$

που σημαίνει ότι:

$kn^{j-1} = L((1+n)^i \pmod{n^{j+1}}) - (\binom{i_{j-1}}{2}n + \cdots + \binom{i_{j-1}}{j}) \pmod{n^j}$

Αν αντικαταστήσουμε το kn^{j-1} στην $i_j = i_{j-1} + kn^{j-1}$ έχουμε:

$i_j = L((1+n)^i \pmod{n^{j+1}}) - (\binom{i_{j-1}}{2}n + \cdots + \binom{i_{j-1}}{j}n^{j-1}) \pmod{n^j}$

Έτσι προκύπτει ο παρακάτω αλγόριθμος:

function extract:
  $i := 0$
  for $j := 1$ **to** $s+1$ **do**
  begin
    $n_j := n^j$
    $t_1 := (x \bmod (n_j \cdot n) - 1) \operatorname{div} n$
    $t_2 := i$
    $sum := 0$
    for k:=2 **to** $j+1$ **do**
      $sum := sum + \binom{t_2}{k} \cdot n^{k-1} \bmod n_j$
    $t_1 = (t_1 - sum) \bmod n_j$
    $i = t_1$
  end
  return i

Σχήμα 6.4: Εξαγωγή Μηνύματος στην αποκρυπτογράφηση Damgård-Jurik

– Μετά την εξαγωγή λαμβάνουμε το m από το $m\lambda$ πολλαπλασιάζοντας με $\lambda - 1 \pmod{n^{s+1}}$

Απόδειξη Ορθότητας Για να αποδείξουμε ότι το κρυπτοσύστημα Paillier λειτουργεί σωστά πρέπει να αποδείξουμε το παρακάτω θεώρημα:

Θεώρημα 6.20. *Για* $c = Enc(m,r) = (1+n)^m r^n \pmod{n^2}$ *η αποκρυπτογράφηση* $\frac{L(c^\lambda \pmod{n^2})}{L(g^\lambda \pmod{n^2})} \pmod{n}$ *δίνει* m.

Απόδειξη. Η απόδειξη θα γίνει στην απλοποιημένη έκδοση.Απλοποιούμε τον συμβολισμό χρησιμοποιώντας $[c]_{n+1}$ για να αναπαραστήσουμε το μήνυμα που που αντιστοιχεί στο κρυπτοκείμενο c με $g = n+1$. Δηλαδή: $w = Enc_{n+1}([w]_{n+1}, r)$. Το κύριο βήμα είναι η απόδειξη του παρακάτω λήμματος:

Λήμμα 6.21. $\forall\, w \in \mathbb{Z}^*_{n^2} : L(w^\lambda \pmod{n^2}) = \lambda[w]_{n+1} \pmod{n}$

Απόδειξη. Αρχικά αποδεικνύουμε ότι:

Λήμμα 6.22. $\forall x \in \mathbb{Z}_n : (1+n)^x = 1 + nx \pmod{n^2}$

Απόδειξη. Οι πιο πολλοί όροι του διώνυμου είναι 0 στο $\mathbb{Z}_{n^2}$:

$$\begin{aligned}(1+n)^x \pmod{n^2} = \\ 1 + \tbinom{x}{1}n + \tbinom{x}{2}n^2 + \cdots + \tbinom{x}{x}n^x \pmod{n^2} = \\ 1 + xn \pmod{n^2}\end{aligned}$$

□

Στην συνέχεια αποδεικνύουμε ότι:

Λήμμα 6.23. $\forall c \in \mathbb{Z}^*_{n^2}$, *και κατάλληλους γεννήτορες* g_1, g_2 *:* $\quad [c]_{g_1} = [c]_{g_2}[g_2]_{g_1}$

Απόδειξη. Έστω $y = [g_2]_{g_1}$. Αυτό σημαίνει: $g_2 = g_1^y b^n$ όπου $b \in_R \mathbb{Z}^*_n$
Έστω $z = [c]_{g_2}$. Αυτό σημαίνει: $c = g_2^z d^n$ όπου $d \in_R \mathbb{Z}^*_n$.
Τώρα $c = g_2^z d^n = (g_1^y b^n)^z d^n = g_1^{zy}(b^z d)^n$ που σημαίνει ότι $yz = [c]_{g_1}$
Με αλλαγή γραφής έχουμε: $[c]_{g_1} = [c]_{g_2}[g_2]_{g_1}$ □

Για το κύριο λήμμα: $\forall\, w \in \mathbb{Z}^*_{n^2} : L(w^\lambda \, mod \, n^2) = \lambda[w]_{n+1} b mod n$
Αφού το $n+1$ είναι έγκυρος γεννήτορας, $\forall\, w \in \mathbb{Z}^*_{n^2}$: εξ ορισμού:

$$w = Enc_{n+1}([w]_{n+1}, r) = (n+1)^{[w]_{n+1}} r^n \pmod{n^2}$$

Η αποκρυπτογράφηση δίνει::

$$w^\lambda = (n+1)^{\lambda[w]_{n+1}} r^{n\lambda} \pmod{n^2}$$

Οι όροι του διωνύμου εξαφανίζονται $\pmod{n^2}$ και παίρνουμε:

$$w^\lambda = (1 + \lambda[w]_{n+1} n) r^{n\lambda} \pmod{n^2}$$

Αφού $n\lambda(n) = \lambda(n^2)$:

$$w^\lambda = (1 + \lambda[w]_{n+1} n) r^{\lambda(n^2)} \pmod{n^2}$$

που σημαίνει:

$$w^\lambda = (1 + \lambda[w]_{n+1} n)$$

Άρα: $L(w^\lambda) = \frac{w^\lambda - 1}{n} = \lambda[w]_{n+1}$

□

Έτσι κατά την αποκρυπτογράφηση:

$$\frac{L(c^\lambda \bmod n^2)}{L(g^\lambda \bmod n^2)} = \frac{\lambda[c]_{n+1}}{\lambda[g]_{n+1}} = \frac{[c]_{n+1}}{[g]_{n+1}} = \frac{[c]_g[g]_{n+1}}{[g]_{n+1}} = [c]_g = m$$

□

Οι ιδιότητες ασφάλειας προκύπτουν από το αρχικό κρυπτοσύστημα.

6.7 Ασκήσεις

1. Δείξτε ότι η συνθήκη $\texttt{P} \neq \texttt{NP}$ είναι αναγκαία για την ύπαρξη συναρτήσεων μονής κατεύθυνσης.

2. Δείξτε ότι στο κρυπτοσύστημα RSA ισχύει $\texttt{Decrypt}(\texttt{Encrypt}(x)) = x$ για κάθε $x \in \mathbb{Z}_n$.

3. Αποδείξτε ότι αν στο κρυπτοσύστημα RSA αντικαταστήσουμε τη συνάρτηση $\phi(n)$ με τη $\lambda(n) = \frac{(p-1)(q-1)}{gcd(p-1,q-1)}$, τότε έχουμε πάλι ένα καλά ορισμένο κρυπτοσύστημα.

4. Σταθερό σημείο ενός κρυπτοσυστήματος ονομάζουμε ένα μήνυμα που το κρυπτοκείμενό του είναι το ίδιο το μήνυμα, δηλαδή $\texttt{Encrypt}(m) = m$. Στην περίπτωση του RSA, αν το δημόσιο κλειδί είναι το (N, e), τότε για ένα σταθερό σημείο ισχύει $m^e \equiv m \pmod N$. Αποδείξτε ότι το πλήθος των σταθερών σημείων στο RSA είναι $[gcd(e-1, p-1)+1][gcd(e-1, q-1)+1]$.(Υπόδειξη: Δείξτε ότι $m^e \equiv m \pmod N$ ανν $m^e \equiv m \pmod p$ και $m^e \equiv m \pmod q$.)

5. Δείξτε πώς σε ένα κρυπτοσύστημα RSA μπορούμε να παραγοντοποιήσουμε το γινόμενο n αν διαρρεύσει ο ιδιωτικός εκθέτης d και έχουμε μικρό δημόσιο εκθέτη, π.χ. $e = 3$ (χωρίς να χρησιμοποιήσετε τον πιθανοτικό αλγόριθμο παραγοντοποίησης).

6. Έστω σύστημα RSA με δημόσιο κλειδί $(n = 481, e = 5)$. Ας υποθέσουμε ότι ένας κατάσκοπος καταφέρνει να βρει το ιδιωτικό κλειδί $d = 173$. Δείξτε τι ακριβώς θα κάνει για να παραγοντοποιήσει το n (χρησιμοποιήστε τον πιθανοτικό αλγόριθμο παραγοντοποίησης με αρχική τιμή $w = 7$).

7. Οι υπάλληλοι μιας εταιρείας κρυπτογραφούν τα email που ανταλλάσσουν με χρήση RSA και για ευκολία χρησιμοποιούν όλοι το ίδιο γινόμενο n με διαφορετικούς δημόσιους εκθέτες, πρώτους μεταξύ τους. Για παράδειγμα η Αλίκη και ο Βασίλης, χρησιμοποιούν e_A, e_B, με $gcd(e_A, e_B) = 1$. Δείξτε ότι αν ο διευθυντής στείλει στην Αλίκη και τον Βασίλη το ίδιο μήνυμα m (π.χ. μια κοινή υπενθύμιση) κρυπτογραφημένο με το σύστημα αυτό, τότε η Κάκια που έχει πρόσβαση στα email της εταιρείας μπορεί να ανακτήσει το μήνυμα m.

8. Ο διευθυντής μιας εταιρείας χρειάζεται να παίρνει συχνά κρυπτογραφημένα μηνύματα από τους υπαλλήλους του. Για τον σκοπό αυτό χρησιμοποιεί RSA, δίνοντας σε όλους το δημόσιο κλειδί του $\langle n, e\rangle$ όπου $n = pq$ με p, q πρώτους. Φυσικά κρατάει κρυφούς τους πρώτους p, q.

   Για ευκολία, δίνει επιπλέον στη γραμματέα του μία συσκευή με την οποία θα μπορούν οι υπάλληλοι που δεν διαθέτουν το πρόγραμμα κρυπτογράφησης να κρυπτογραφούν τα μηνύματά τους.

   Η συσκευή υποτίθεται ότι λειτουργεί ως εξής για είσοδο m:

   - Υπολογίζει $c_p = m^e \bmod p$,
   - Υπολογίζει $c_q = m^e \bmod q$, και
   - Συνδυάζει τις λύσεις με CRT ώστε να δώσει ως έξοδο τη μοναδική τιμή $c \in \mathbb{Z}_n$ τ.ω. $c = m^e \pmod n$.

   Λόγω όμως εργοστασιακού λάθους, στο δεύτερο βήμα η συσκευή υπολογίζει $c'_q = m^e + 1 \bmod q$ και δίνει στην έξοδο $c' \in \mathbb{Z}_n$, τ.ώ. $c' = c_p \pmod p$ και $c' = c'_q \pmod q$.

   Όπως είναι φυσικό, ο διευθυντής σύντομα διαπιστώνει (με ποιον τρόπο;) ότι κάτι δεν πάει καλά, και ζητάει από την γραμματέα του να στείλει την συσκευή για επισκευή. Η γραμματέας όμως, που έχει παρακολουθήσει προσεκτικά ένα σεμινάριο κρυπτογραφίας, κατορθώνει πριν στείλει την συσκευή στο service να βρει το ιδιωτικό κλειδί του διευθυντή. Πώς το κατάφερε αυτό;

9. Έστω ότι η Αλίκη θέλει να στείλει το ίδιο μήνυμα m στον Βασίλη και στον Γιάννη, με το κρυπτοσύστημα Rabin. Από απροσεξία, τα δημόσια κλειδιά τους $(n_B, b), (n_C, b)$ χρησιμοποιούν το ίδιο b. Δείξτε ότι αν ο Ευάγγελος υποκλέψει τα κρυπτογραφημένα μηνύματα μπορεί να βρεί το m χωρίς να βρει τα ιδιωτικά κλειδιά. Θα υπήρχε ο ίδιος κίνδυνος αν είχαν χρησιμοποιήσει διαφορετικά b;

10. Η Άννα δουλεύει στην ΕΥΠ. Από τον κανονισμό της υπηρεσίας, κάθε email που της στέλνεται από συναδέλφους της πρέπει να έχει το μήνυμα συνημμένο, κρυπτογραφημένο με το σύστημα ElGamal. Ο σύζυγός της Ορέστης

υποπτεύεται ότι κάποια μηνύματα που της έρχονται από τον Βησσαρίωνα δεν είναι αυστηρώς επαγγελματικά, και έχει πολύ μεγάλη περιέργεια να τα διαβάσει. Έχει παρατηρήσει ότι όταν η Άννα λαμβάνει μηνύματα, τα κατεβάζει στον υπολογιστή της, τα αποκρυπτογραφεί, και αν δεν βγάζουν νόημα τα θεωρεί λανθασμένα και τα αφήνει σε έναν φάκελο αποκρυπτογραφημένα για να ειδοποιήσει τον αποστολέα σχετικά (αλλιώς τα σβήνει αμέσως). Πώς μπορεί ο Ορέστης να εκμεταλλευτεί την απροσεξία της Άννας για να διαβάσει τα μηνύματα που της στέλνει ο Βησσαρίων, αν αποκτήσει προσωρινή πρόσβαση στο email της (π.χ. αν μια μέρα το ξεχάσει ανοιχτό); Φυσικά, δεν γνωρίζει το ιδιωτικό της κλειδί, αλλά γνωρίζει το δημόσιό της κλειδί.

11. Ας υποθέσουμε ότι υπάρχει μία έμπιστη αρχή η οποία υπολογίζει για λογαριασμό καποιων χρηστών τις ψηφιακές υπογραφές τους. Δηλαδή αν κάποιος χρήστης θέλει να υπολογίσει την (RSA) ψηφιακή υπογραφή του για το κείμενο m, τότε στέλνει στην έμπιστη αρχή το κείμενο m. Η έμπιστη αρχή, που διαθέτει τον εκθέτη αποκρυπτογραφησης d και τα p, q του χρήστη αυτού, υπολογίζει την ψηφιακή υπογραφή ως εξής: υπολογίζει πρώτα τα $S_1 = m^d \bmod p$ και $S_2 = m^d \bmod q$ και μετά με χρήση του κινέζικου θεωρήματος υπολογίζει την υπογραφή S. Αν υποθέσουμε ότι η έμπιστη αρχή έκανε λάθος στον υπολογισμό του S_1 και υπολόγισε $\tilde{S}_1 \neq S_1$ – και για υπογραφή $\tilde{S} \neq S$ – δείξτε ότι τότε μπορεί κάποιος που γνωρίζει τα $m, \tilde{S}$ και το δημόσιο κλειδί (N, e) να παραγοντοποιήσει αποδοτικά το N. Πώς μπορεί η έμπιστη αρχή να αποφύγει αυτόν τον κίνδυνο;
Υπόδειξη: εξετάστε την ισοτιμία του $\tilde{S}^e$ modulo p και modulo q.

12. Υλοποιήστε ένα κρυπτοσύστημα RSA. Δηλαδή φτιάξτε ένα πρόγραμμα με τις εξής λειτουργίες (υπορουτίνες): (α) κατασκευής δημόσιου και ιδιωτικού κλειδιού (β) κρυπτογράφησης και (γ) αποκρυπτογράφησης.

    Υποδείξεις: Θα χρειαστεί να φτιάξετε ρουτίνες για πράξεις με μεγάλους αριθμούς, όπου τα ψηφία κάθε αριθμού που θα δίνετε θα αποθηκεύονται σε έναν πίνακα και θα πρέπει να κατασκευάσετε αλγόριθμους για πρόσθεση, αφαίρεση, πολλαπλασιασμό και ακέραια διαίρεση (πηλίκο και υπόλοιπο). Για την ύψωση σε δύναμη να χρησιμοποιήσετε επαναλαμβανόμενο τετραγωνισμό.

    Για την κατασκευή του κλειδιού θα χρειαστείτε τον Επεκταμένο Αλγόριθμο του Ευκλείδη (για την εύρεση του e και του d), καθώς επίσης και ένα πρόγραμμα για primality testing, είτε των Solovay-Strassen, είτε των Miller-Rabin (για να ελέγχετε αν p, q είναι πρώτοι).

    Εάν θέλετε μπορείτε να χρησιμοποιήσετε δυαδικούς αριθμούς.

13. Θεωρήστε το κρυπτοσύστημα Rabin στη μορφή $\texttt{Encrypt}(x) = x^2 \bmod n$.

Έχει την ιδιότητα IND-CPA; Την ιδιότητα IND-CCA2; Αιτιολογήστε τις απαντήσεις σας.

14. Υλοποιήστε το κρυπτοσύστημα Rabin. Φροντίστε ώστε η αποκρυπτογράφηση να μην είναι διφορούμενη.

15. Αποδείξτε ότι η ισοτιμία $x \cdot (x + b) = y \pmod N$ έχει μια λύση αν και μόνο αν η η ισοτιμία $x'^2 = y + b^2/4 \pmod N$ έχει λύση.

16. Να εξετάσετε αν τα κρυπτοσυστήματα Benaloh, Paillier και Damgård Jurik διαθέτουν τις ιδιότητες IND-CPA, IND-CCA, IND-CCA2.

6.8 Ηλεκτρονικό Υλικό

Το κρυπτοσύστημα δημοσίου κλειδιού RSA, λόγω της σημασίας του, έχει αποτελέσει αντικείμενο παρουσιάσεων, διαδραστικών και όχι, υλοποίησεων εκπαιδευτικών και εμπορικών. Μία αξιόλογη διαδραστική παρουσίαση έχει δοθεί από την ομάδα CrypTool και μπορεί να βρεθεί στην ιστοσελίδα τους. Επίσης έχει αποτελέσει και αντικείμενο εκπαιδευτικών βίντεο, με στόχο την προσέγγιση ενός λιγότερου εξειδικευμένου κενού. Ένα χαρακτηριστικό παράδειγμα δίνεται από την εκπαιδευτική σειρά Art of the Problem, στο Gambling with Secrets. Σε ότι αφορά τις υλοποιήσεις ο παγκόσμιος ιστός βρίθει από δεκάδες παραλλάγες. Για απλή εξάσκηση υπάρχουν παραδείγματα σε Javascript. Για πειρασματισμό με τον αλγόριθμο η Invent with Python. Φυσικά υπάρχουν και έτοιμες βιβλιοθήκες για ενσωμάτωση σε εμπορικά προγράμματα όπως η Crypto++, η pyCrypto κά. Υπάρχει υλοποίηση ακόμα και σε λογιστικά φύλλα με επεξηγηματικό βίντεο.

Συγκεντρωτικά, το ενδεικτικό υλικό παρουσιάζεται εδώ:

- Διαδραστικές Παρουσιάσεις - Video
  - CrypTool Team, The functionality of the RSA Cipher, ιστοσελίδα
  - Art of the Problem, Gambling with Secrets: (RSA Encryption)
  - Daniele Cordano, RSA spreadsheet
- Διαδραστικές Υλοποιήσεις
  - Public-Key Encryption by RSA Algorithm
  - JavaScript RSA Cryptography Demo
  - Daniele Cordano, RSA spreadsheet
  - El Gamal Demo

  - Υλοποίηση Damgård - Jurik

- Κώδικας
  - Crypto++ μια βιβλιοθήκη κρυπτογραφίας σε C++
  - pyCrypto μια βιβλιοθήκη κρυπτογραφίας σε Python
  - Python πηγαίος κώδικας RSA

Βιβλιογραφία

[1] Josh D. Cohen and Michael J. Fischer. A robust and verifiable cryptographically secure election scheme (extended abstract). In *FOCS*, pages 372–382, 1985.

[2] Ivan Damgård, Mads Jurik, and Jesper Buus Nielsen. A generalization of paillier's public-key system with applications to electronic voting. *P Y A RYAN*, page 3, 2003.

[3] Ivan Damgård and Mats Jurik. A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In *Proceedings of the 4th International Workshop on Practice and Theory in Public Key Cryptography: Public Key Cryptography*, PKC '01, pages 119–136, London, UK, UK, 2001. Springer-Verlag.

[4] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Trans. Inf. Theor.*, 22(6):644–654, November 1976.

[5] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *CRYPTO*, pages 10–18, 1984.

[6] Shafi Goldwasser and Silvio Micali. Probabilistic encryption & how to play mental poker keeping secret all partial information. In *STOC '82: Proceedings of the fourteenth annual ACM symposium on Theory of computing*, pages 365–377, New York, NY, USA, 1982. ACM Press.

[7] Shafi Goldwasser, Silvio Micali, and Po Tong. Why and how to establish a private code on a public network (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982*, pages 134–144, 1982.

[8] Alexander May. Computing the rsa secret key is deterministic polynomial time equivalent to factoring. In *Advances in Cryptology–CRYPTO 2004*, pages 213–219. Springer, 2004.

[9] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Proceedings of the 17th international conference on Theory and application of cryptographic techniques*, EUROCRYPT'99, pages 223–238, Berlin, Heidelberg, 1999. Springer-Verlag.

[10] R.L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21:120–126, 1978.

[11] Burt Kaliski Ronald L. Rivest. Rsa problem, 2003.

[12] Adi Shamir. A polynomial time algorithm for breaking the basic merkle-hellman cryptosystem. In *Advances in Cryptology*, pages 279–288. Springer, 1983.

[13] Michael J Wiener. Cryptanalysis of short rsa secret exponents. *Information Theory, IEEE Transactions on*, 36(3):553–558, 1990.

Κεφάλαιο 7

Ψηφιακές Υπογραφές

7.1 Εισαγωγή

Στο κεφάλαιο αυτό θα ασχοληθούμε με τα *Σχήματα Υπογραφών* ή *Σχήματα Ψηφιακών Υπογραφών* (Digital Signature Schemes) όπως αλλιώς ονομάζονται. Θα μιλήσουμε για την αναγκαιότητα αλλά και το ρόλο τους και θα σταθούμε περισσότερο σε κάποια ευρέως διαδεδομένα *Σχήματα Υπογραφών*.

Ας γίνουμε όμως πιο συγκεκριμένοι. Η υπογραφή είναι ένας από τους βασικούς μηχανισμούς παροχής εγκυρότητας και εφαρμοσιμότητας σε μία συναλλαγή. Σε γενικό επίπεδο, παρέχει στοιχεία για την αυθεντικοποίηση ενός κειμένου και την αποδοχή του από τον συγγραφέα του. Επιπλέον δίνει το νόημα της τέλεσης (ceremony) σε μία πράξη, επισείοντας την προσοχή των συμμετεχόντων στο ότι η συναλλαγή ή συμφωνία τους κ.ο.κ. είναι δεσμευτική και μπορεί να έχει νομικές συνέπειες.

Η χειρόγραφη υπογραφή βασίζεται στην δημιουργία ενός χαρακτηριστικού σημαδιού χρησιμοποιώντας κάποιο είδος γραφής, το οποίο χαρακτηρίζει τον υπογράφοντα και τοποθετείται στο τέλος ενός κειμένου. Οι χημικές ιδιότητες του μελανιού και του χαρτιού διασφαλίζουν την φυσική σύνδεση της υπογραφής με το κείμενο. Η φυσική αυτή σύνδεση έχει ως συνέπεια την λογική συσχέτιση, έτσι ώστε η υπογραφή να είναι ένδειξη για [5]:

- Την αποδοχή του κειμένου από τον υπογράφοντα και την συμφωνία του με αυτό.
- Το γεγονός ότι ο υπογράφων έλαβε γνώση του κειμένου και αναλαμβάνει την ευθύνη για το περιεχόμενο του.
- Την ταυτότητα του υπογράφοντα ως συντάκτη του κειμένου. Δηλαδή η υπογραφή ταυτοποιεί τον υπογράφοντα.

- Την πραγματοποίηση μιας συναλλαγής.

Οι ιδιόχειρες υπογραφές μπορούν να αμφισβητηθούν από τους υποτιθέμενους υπογράφοντες. Οι λεπτομέρειες ποικίλουν ανάλογα με το συγκεκριμένο νομικό πλαίσιο, σε γενικές γραμμές πάντως οι παρακάτω ισχυρισμοί είναι καθ' όλα θεμιτοί σε μία δικαστική αίθουσα:

- Μία ιδιόχειρη υπογραφή δεν γίνεται αποδεκτή, καθώς είναι προϊόν πλαστογραφίας.
- Μία ιδιόχειρη υπογραφή που δεν είναι προϊόν πλαστογραφίας, δεν γίνεται αποδεκτή καθώς:
  - Ο υπογράφων εξαπατήθηκε.
  - Ασκήθηκε ψυχολογική και άλλου είδους πίεση, στον υπογράφονται προκειμένου να υπογράψει.

Είναι προφανές, ότι οι υπογραφές με την παραδοσιακή τους μορφή και υλοποίηση δεν μπορούν να εφαρμοστούν στις ηλεκτρονικές συναλλαγές. Κατά συνέπεια απαιτείται ένας νέος τύπος υπογραφής, μία ψηφιακή υπογραφή δηλαδή, η οποία θα μπορεί να προσαρτηθεί σε και να συνδεθεί με ηλεκτρονικά δεδομένα με τον ίδιο τρόπο, που μία ιδιόχειρη υπογραφή αφορά χειρόγραφα κείμενα. Το πολύ ενδιαφέρον με την ψηφιακή υπογραφή είναι ότι δεν αντικαθιστά απλά την ιδιόχειρη, αλλά τη βελτιώνει. Επιπλέον καθιστά δυνατή τη δημιουργία καινούριων ειδών υπογραφής.

Μία σύντομη σύγκριση της "χειρόγραφης" με την ψηφιακή υπογραφή κάνει εμφανείς τις παρακάτω διαφορές:

- Η χειρόγραφη επισυνάπτεται φυσικά σε ένα μήνυμα έτσι που κάθε γνήσιο αντίγραφό του την περιέχει, ενώ η ψηφιακή είναι δυνατό να αφαιρεθεί από το αρχικό μήνυμα. Για να αντιμετωπιστεί το πρόβλημα αυτό είναι απαραίτητο ο *αλγόριθμος υπογραφής* να "συνδέει" με κάποιο τρόπο το μήνυμα με την υπογραφή. Ένας τρόπος για να γίνει αυτό είναι να κρυπτογραφήσουμε πρώτα το υπογεγραμμένο μήνυμα και έπειτα να το στείλουμε σ' εκείνον που θέλουμε.[1]

[1] Εδώ θέλει προσοχή αφού πρέπει η διαδικασία να γίνει με τη σειρά Υπογραφή → κρυπτογράφηση διότι σε αντίθετη περίπτωση, αν ο $\mathcal{A}$ καταφέρει να κλέψει το υπογεγραμμένο μήνυμα του αποστολέα προς τον παραλήπτη, μπορεί να αφαιρέσει την υπογραφή και να προσθέσει τη δική του. Έτσι ο $\mathcal{A}$ θα μπορεί να υποδύεται τον αποστολέα στις υπόλοιπες επικοινωνίες του με τον παραλήπτη

- Από την άλλη μεριά η χρήση μίας *ασφαλούς* ψηφιακής υπογραφής είναι πολύ βολική, αφού η *επαλήθευσή* της *(verification)*, γίνεται μ' έναν δημόσιο (public) *αλγόριθμο επαλήθευσης*, σε αντίθεση με τη περίπτωση της "χειρόγραφης" που μόνο ο γραφολόγος μπορεί να την επιβεβαιώσει με ανάλογη ασφάλεια.

Στην επόμενη ενότητα θα ορίσουμε πιο αυστηρά το *σχήμα ψηφιακής υπογραφής* και θα δώσουμε κάποια βασικά χαρακτηριστικά του.

7.2 Γενικός ορισμός

Ξεκινάμε με κάποιους βασικούς ορισμούς και συμβολισμούς που θα μας ακολουθήσουν σ' όλο το υπόλοιπο κεφάλαιο

Ορισμός 7.1. Ένα *Σχήμα Ψηφιακής Υπογραφής* είναι μια εξάδα $(\mathtt{M}, \mathtt{S}, \mathtt{K}, gen, sig_k, ver_k)$ όπου:

1. $\mathtt{M}$: Ο χώρος όλων των πιθανών μηνυμάτων

2. $\mathtt{S}$: Ο χώρος όλων των πιθανών υπογραφών

3. $\mathtt{K}$: Ο χώρος όλων των πιθανών κλειδιών που μπορεί να χρησιμοποιηθούν για την υπογραφή

4. $sig_K(m)$: Ένας μετασχηματισμός από το $\mathtt{M}$ στο $\mathtt{S}$ που χρησιμοποιείται για να δημιουργήσουμε την ψηφιακή υπογραφή. Είναι γνωστός μόνο στον υπογράφοντα και ονομάζεται και *συνάρτηση υπογραφής* (Signing Function).

5. $ver_K(m, s)$: Ένας μετασχηματισμός από το $\mathtt{M} \times \mathtt{S}$ στο σύνολο $\{1, 0\}$ που χρησιμοποιείται για να επαληθεύσει ότι η υπογραφή s έχει πράγματι προκύψει από την εφαρμογή του sig_K στο m. Ήτοι:

$$ver_K(m, s) = \begin{cases} 1, & sig_K(m) = s \\ 0, & sig_K(m) \neq s \end{cases}$$

   Η ver_K ονομάζεται και *συνάρτηση επαλήθευσης* (Verification Function).

Ασφάλεια Η πιο σημαντική ιδιότητα κάθε σχήματος ψηφιακών υπογραφών είναι η μη δυνατότητα πλαστογράφησης (unforgeability). Αυτό με απλά λόγια, σημαίνει να είναι *υπολογιστικά απρόσιτο (intractable)* για κάποιον άλλον εκτός από τον υπογράφοντα να υπολογίσει το $s \in \mathtt{S}$ για κάποιο $m \in \mathtt{M}$, έτσι που $ver_K(m, s) = 1$ (τουλάχιστον για όσο χρόνο η υπογραφή θα χρειάζεται να είναι έγκυρη). [2]

Τυπικά η μη δυνατότητα πλαστογράφησης ορίζεται ως εξής:

Ορισμός 7.2. Έστω ένας PPT αντίπαλος $\mathcal{A}$ ο οποίος έχει πρόσβαση σε ένα μαντείο υπογραφής $sig_K()$. Αν μετά από l αλληλεπιδράσεις με αυτό ο $\mathcal{A}$ έχει αμελητέο πλεονέκτημα στο να δημιουργήσει $l+1$ έγκυρα ζεύγη μηνυμάτων και υπογραφών τότε λέμε ότι το σχήμα υπογραφής είναι ασφαλές ενάντια στην επίθεση επιλεγμένων μηνυμάτων *(*chosen message attacks). Αν απαιτούμε τα παραπάνω μηνύματα να είναι διαφορετικά μεταξύ τους τότε μιλάμε για ισχυρή δυνατότητα μη πλαστογράφησης (strong unforgeability).

Κάθε *Σχήμα ψηφιακής υπογραφής* πρέπει να έχει επιπλέον τις παρακάτω βασικές ιδιότητες:

1. να ισχύει: $ver_K(m, s) = 1 \Leftrightarrow sig_K(m) = s, \forall m \in \mathtt{M}\ , s \in \mathtt{S}$
2. να είναι υπολογιστικά "εύκολο" για κάποιον να παράξει την υπογραφή του, αλλά και για οποιονδήποτε να επαληθεύσει τη γνησιότητά της.
3. ακριβώς επειδή η Ψηφιακή Υπογραφή δεν αποτελεί τμήμα του κειμένου στο οποίο επισυνάπτεται, συχνά είναι σκόπιμο αυτό να περιέχει πληροφορίες όπως ημερομηνία και ώρα ώστε να αποφεύγεται η επαναχρησιμοποίησή του. (Σκεφτείτε τι θα συνέβαινε αν ο A έδινε στον B μία "ηλεκτρονική επιταγή" ψηφιακά υπογεγραμμένη που να μην περιείχε ημερομηνία και ώρα...)

Κλείνοντας την ενότητα αυτή, είναι σκόπιμο να διακρίνουμε το σύνολο των ψηφιακών υπογραφών σε δύο μεγάλες κατηγορίες

1. Σχήματα ψηφιακής υπογραφής *με παράρτημα* (Digital Signatures Schemes with appendix). Εδώ ανήκουν τα σχήματα στα οποία το αρχικό μήνυμα είναι απαραίτητο για την πιστοποίηση γνησιότητας της αντίστοιχης υπογραφής, όπως είναι το Digital Signature Standard (DSS) (7.5) και το ElGamal (7.4)).
2. Σχήματα Ψηφιακής υπογραφής *με ικανότητα ανάκτησης του μηνύματος* (Digital Signatures Schemes with message recovery, στα οποία το αρχικό μήνυμα μπορεί να παραχθεί από την ίδια την υπογραφή, όπως το RSA (7.3).

[2] Κανείς ακόμα δεν έχει αποδείξει τυπικά (μαθηματικά) ότι ένα τέτοιο *σχήμα ψηφιακής υπογραφής* υπάρχει. Εντούτοις υπάρχουν κάποιοι πολύ καλοί υποψήφιοι, που προέρχονται από μετατροπές σε *Κρυπτοσυστήματα Δημοσίου Κλειδιού*, σε κάποιους απ' τους οποίους θα αναφερθούμε στη συνέχεια.

7.3 Σχήμα υπογραφής RSA

Ένα από τα πιο γνωστά σχήματα υπογραφής, είναι το RSA. Το σχήμα αυτό οφείλει την ευρεία χρήση του, όχι τόσο στην αποδοτικότητά του, όσο στο γεγονός ότι δεν είναι παρά μία εφαρμογή του κρυπτοσυστήματος RSA με *αντιστροφή* του ρόλου των κλειδιών (δημόσιο - ιδιωτικό).

Πιο συγκεκριμένα, ας συμβολίσουμε με $\texttt{Encrypt}_K(m)$ τη συνάρτηση κρυπτογράφησης του RSA για ένα απλό κείμενο m και ένα κλειδί K, και με $\texttt{Decrypt}_K(c)$ την αντίστοιχη συνάρτηση αποκρυπτογράφησης για το κρυπτοκείμενο c.

Προφανώς (από τον ορισμό του κρυπτοσυστήματος) ισχύει:

$\texttt{Decrypt}_K(\texttt{Encrypt}_K(m)) = m$.

Εκείνο που δεν είναι προφανές, αλλά μπορεί εύκολα να αποδειχθεί είναι ότι αν ο χώρος των κρυπτοκειμένων είναι ο ίδιος με το χώρο των απλών κειμένων τότε ισχύει και $\texttt{Encrypt}_K(\texttt{Decrypt}_K(m)) = m$ (βλ. Ασκ 1).

Η τελευταία παρατήρηση μας οδηγεί να ορίσουμε ένα σχήμα υπογραφής όπως φαίνεται παρακάτω (σχήμα 7.1).

Σημαντικό είναι εδώ να τονιστεί ότι το σχήμα ψηφιακής υπογραφής που περιγράφεται στο σχήμα 7.1 δεν εγγυάται ότι είναι δύσκολο να πλαστογραφηθεί η οποιαδήποτε υπογραφή. Αντίθετα, είναι πολύ εύκολο για κάποιον που ξέρει το δημόσιο κλειδί ενός άλλου χρήστη A να πλαστογραφήσει την υπογραφή του σε ένα τυχαίο μήνυμα m με τον ακόλουθο τρόπο (no message attack):

Επιλέγει μια τυχαία υπογραφή $s \in \texttt{S}$. Υπολογίζει το $\texttt{Encrypt}(s) = m$, και έτσι έχουμε μια 'καλή' (αλλά πλαστογραφημένη) υπογραφή s για το μήνυμα m καθώς $\texttt{Decrypt}(m) = s$. Το m φυσικά είναι μια τυχαία ακολουθία, και δεν μπορεί να είναι ένα μήνυμα σε μορφή κειμένου, παρόλα αυτά, εξακολουθεί να είναι ένα πρόβλημα, μιας και πάρα πολλές φορές τα μηνύματα m δεν είναι απαραιτήτως κείμενα που βγάζουν νόημα (για παράδειγμα, το m μπορεί να είναι ένα κλειδί για ένα συμμετρικό κρυπτοσύστημα). Η επίθεση αυτή ονομάζεται *επίθεση χωρίς μήνυμα - no message attack*.

Επιπλέον, επειδή το RSA είναι εύπλαστο (malleable), αν s_1, s_2 είναι δύο έγκυρες υπογραφές του ίδιου χρήστη, μπορεί εύκολα να εξαχθεί η υπογραφή $s = s_1 s_2 \bmod n$ η οποία αντιστοιχεί στο μήνυμα $m = m_1 m_2 \bmod n$ όπως είναι εύκολο να δει κανείς. Η επίθεση αυτή ονομάζεται *επίθεση επιλεγμένου μηνύματος - chosen message attack*.

Για τη λύση των παραπάνω προβλημάτων έχουν υπάρξει πολλές προτάσεις. Μία αφορά τη χρήση της *Συνάρτησης Πλεονάζουσας Πληροφορίας*.

Ορισμός 7.3. Η Συνάρτηση Πλεονάζουσας Πληροφορίας (redundancy function) είναι μία δημόσια γνωστή αντιστρέψιμη προβολή από το χώρο $\texttt{M}$ των απλών κει-

Υποθέτουμε ότι ο A θέλει να στείλει στον B το μήνυμα m υπογεγραμμένο ψηφιακά με το RSA

1. **Δημιουργία κλειδιών** Η διαδικασία παραγωγής κλειδιού είναι ίδια με εκείνη του RSA οπότε ο αναγνώστης παραπέμπεται στο αντίστοιχο κεφάλαιο 6.4.1. Για λόγους πληρότητας απλά υπενθυμίζεται ότι το κλειδί K είναι μία πεντάδα

   $$K = ((n, e), (p, q, d)) : n = pq, p, q : \text{πρώτοι}\ , ed \equiv 1 \pmod{\phi(n)}.$$

   Οι τιμές n, e είναι το δημόσιο, ενώ οι p, q, d το ιδιωτικό κλειδί

2. **Δημιουργία υπογραφής**
   Ο A υπολογίζει το $s = (sig_K(m) =)\texttt{Decrypt}_K(m) = m^d \pmod{n}$. Το s είναι η ψηφιακή υπογραφή του το οποίο και στέλνει στον B

3. **Επαλήθευση υπογραφής**
   Ο B χρησιμοποιεί το δημόσιο κλειδί του A για να επαληθεύσει την υπογραφή s ανακτώντας παράλληλα το αρχικό μήνυμα: $m_1 = \texttt{Encrypt}_K(s) = s^e \pmod{n}$,

4. Ισχύει ότι $ver_K(m_1, s) = 1$ καθώς $s^e = m^{de} = m$ δηλαδή το m_1 είναι το αρχικό μήνυμα ήτοι m.

Σχήμα 7.1: Το Σχήμα υπογραφής RSA

μένων σε έναν υπόχωρό του με συγκεκριμένες ιδιότητες.

Ένα παράδειγμα τέτοιας προβολής είναι η μετατροπή ενός δυαδικού κειμένου σε τέτοια μορφή ώστε ανάμεσα σε κάθε 8 bit να υπάρχει η λέξη 10101. Εφαρμόζοντας ο A στο μήνυμά του μια τέτοια συνάρτηση, καθιστά σχεδόν αδύνατη την πλαστογράφηση της υπογραφής του (εκτός φυσικά αν μπορέσει ο πλαστογράφος να λύσει το πρόβλημα της παραγοντοποίησης). Πράγματι είναι πρακτικά απίθανο μία τυχαία επιλεγμένη συμβολοακολουθία από το χώρο των υπογραφών να δώσει ένα μήνυμα που να έχει τις ιδιότητες που προσδίδει σε τυχαίο μήνυμα η *Συνάρτησης Πλεονάζουσας Πληροφορίας* του «αυθεντικού» αποστολέα.

Μία άλλη πρόταση για ενίσχυση της ασφάλειας του σχήματος υπογραφής RSA είναι η χρήση μίας συνάρτησης σύνοψης $\mathcal{H}$ σαν αυτές που ua μελετήσουμε στην

ενότητα 8.4.3.

7.4 Σχήμα υπογραφής ElGamal

Στην ενότητα αυτή θα δούμε τον ουσιαστικό ρόλο του κρυπτοσυστήματος ElGamal 6.5.2 στο χώρο των ψηφιακών υπογραφών (σχήμα 7.2)
Η ορθότητα του συστήματος προκύπτει από τις παρακάτω προτάσεις:

Λήμμα 7.4. *Αν p πρώτος και g πρωταρχικό στοιχείο (γεννήτορας) του $\mathbb{Z}_p^*$, τότε για κάθε $x, y \in \mathbb{Z}$*

$$\alpha^x \equiv \alpha^y \pmod{p} \Leftrightarrow x \equiv y \pmod{p-1}$$

Η απόδειξη είναι η άσκηση 2.

Πρόταση 7.5. *Η συνθήκη επαλήθευσης της υπογραφής επιστρέφει «Αληθής» αν τα (γ, δ) έχουν προκύψει σαν υπογραφή του A στο μήνυμα m, με το σχήμα υπογραφής ElGamal.*

Απόδειξη. Πράγματι είναι προφανές ότι από τις σχέσεις δημιουργίας υπογραφής έχουμε:

$$y^\gamma \gamma^\delta \equiv g^{x\gamma} g^{k\delta} \pmod{p} \tag{7.3}$$

Αρκεί λοιπόν να δούμε ότι $x\gamma + k\delta \equiv m \pmod{p-1}$ αφού τότε

$$g^{x\gamma} g^{k\delta} = g^{x\gamma + k\delta} \equiv g^m \pmod{p}$$

Αυτό προκύπτει άμεσα από το Λήμμα 7.4 □

Παρατήρηση 16. Από το σχήμα 7.2 φαίνεται ότι το σχήμα υπογραφής ElGamal είναι πιθανοτικό, καθώς υπάρχουν πολλές έγκυρες υπογραφές για ένα μήνυμα m(αφού η sig_K εξαρτάται και από το τυχαίο k). Εντούτοις η συνάρτηση επαλήθευσης δέχεται οποιαδήποτε υπογραφή έχει προκύψει από αυτό το σχήμα σαν έγκυρη.

Λόγω της ευρείας χρήσης του σχήματος υπογραφής ElGamal είναι σκόπιμο σ' αυτό το σημείο να σταθούμε λίγο σε θέματα που αφορούν την ασφάλειά του. Αρχικά παρατηρούμε ότι, σε αντίθεση με το RSA δεν είναι εύκολο να πλαστογραφήσει κανείς την υπογραφή του A σ' ένα επιλεγμένο ή τυχαίο μήνυμα m.
Με πρώτη ματιά βλέπουμε τρία πιθανά σενάρια στα οποία ο $\mathcal{A}$ θα μπορούσε να πλαστογραφήσει την υπογραφή του A χωρίς να γνωρίζει το ιδιωτικό του κλειδί.

- **Δημιουργία κλειδιών**

  1. Επιλέγουμε έναν πρώτο p ώστε το DLOG να είναι υπολογιστικά απρόσιτο στο $\mathbb{Z}_p^*$ και έναν τυχαίο $g \in \mathbb{Z}_p^*$
  2. Επιλέγουμε έναν τυχαίο $x, 0 \leq x \leq p-2$ και υπολογίζουμε το $y = g^x \pmod{p}$
  3. Το δημόσιο κλειδί είναι η τριάδα (p, g, y) ενώ το ιδιωτικό κλειδί είναι το x.

- **Δημιουργία υπογραφής**

  1. Ο A θέλει να υπογράψει ένα μήνυμα m και επιλέγει έναν τυχαίο ακέραιο $k \in \mathbb{Z}_{p-1}^*$.
  2. Ο A υπολογίζει τα

$$\gamma = g^k \bmod p \tag{7.1}$$
$$\delta = (m - x\gamma)k^{-1} \bmod (p-1) \tag{7.2}$$

  3. Η ψηφιακή υπογραφή του A για το μήνυμα m για το (τυχαία επιλεγμένο) k είναι η $sig(m,k) = (\gamma, \delta)$. [3]
  4. ο A στέλνει στον B την τριάδα (m, γ, δ), ήτοι το αρχικό του κείμενο με την ψηφιακή του υπογραφή. [4]

- **Επαλήθευση υπογραφής** Ο B επαληθεύει:

$$ver(m,\gamma,\delta) = \begin{cases} 1, & y^\gamma \gamma^\delta \equiv g^m \pmod{p} \\ 0, & y^\gamma \gamma^\delta \neq g^m \pmod{p} \end{cases}$$

Σχήμα 7.2: Το Σχήμα υπογραφής ElGamal

1. Ο $\mathcal{A}$ επιλέγει μία τιμή για το γ και το m και προσπαθεί να βρει μία τιμή για το δ που να ικανοποιεί τη σχέση **??**. Στην περίπτωση αυτή, θα πρέπει να λύσει το πρόβλημα διακριτού λογαρίθμου: $\delta = \log_\gamma g^m y^{-\gamma}$

2. Ο $\mathcal{A}$ επιλέγει τα δ, m και προσπαθεί να υπολογίσει αντίστοιχο γ. Στην περίπτωση αυτή ο $\mathcal{A}$ θα πρέπει τότε να βρει το γ στην παρακάτω ισοτιμία:

$$y^\gamma \gamma^\delta \equiv g^m \pmod{p} \tag{7.4}$$

   Η επίλυση της 2 είναι ένα πρόβλημα για το οποίο δεν είναι γνωστή καμία εφικτή λύση και δε φαίνεται να μπορεί να αναχθεί σε κάποιο από τα γνωστά προβλήματα της Κρυπτολογίας (όπως είναι το *DLOG*). Παραμένει εντούτοις ανοικτό το ενδεχόμενο να υπάρχει τρόπος να επιλεγεί το m μόνο και να υπολογιστούν ταυτόχρονα γ και δ που να ικανοποιούν τη σχέση 2.

3. Αν τέλος ο $\mathcal{A}$ επιλέξει τα γ και δ και προσπαθήσει να υπολογίσει το m, τότε βρίσκεται και πάλι αντιμέτωπος με ένα στιγμιότυπο του *DLOG*.

Παρατηρούμε δηλαδή ότι ο $\mathcal{A}$ δεν μπορεί χρησιμοποιώντας αυτήν την προσέγγιση να πλαστογραφήσει την υπογραφή του A σε τυχαίο μήνυμα m.
Υπάρχει παρ' όλα αυτά τρόπος με τον οποίο ο $\mathcal{A}$ μπορεί να παράγει πλαστή υπογραφή για τυχαίο μήνυμα m, επιλέγοντας τα δ, γ και m ταυτόχρονα:
Αρχικά επιλέγει ακεραίους i και j, τέτοιους που $0 \leq i, j \leq p-2$ και $gcd(j, p-1) = 1$. Στη συνέχεια υπολογίζει τα:

$$\begin{aligned} \gamma &= g^i y^j \bmod p \\ \delta &= -\gamma j^{-1} \bmod (p-1) \\ m &= -\gamma i j^{-1} \bmod (p-1) \end{aligned}$$

Το j^{-1} υπολογίζεται modulo $(p-1)$, γι' αυτό και αρχικά απαιτούμε $(j, p-1) = 1$.

Ισχυρισμός: *το* (γ, δ) *είναι έγκυρη υπογραφή για το ElGamal.*
Απόδειξη του Ισχυρισμού: Πράγματι παρατηρούμε ότι ισχύουν οι ισοδυναμίες:

$$\begin{aligned} y^\gamma \gamma^\delta &\equiv y^\gamma (g^i y^j)^{-\gamma j^{-1}} \pmod{p} \\ &\equiv y^\gamma g^{-ij^{-1}\gamma} y^{-\gamma} \pmod{p} \\ &\equiv g^{-ij^{-1}\gamma} \pmod{p} \\ &\equiv g^m \pmod{p} \end{aligned}$$

Παρόλο πάντως που η μέθοδος αυτή δουλεύει για τυχαίο συνδυασμό των m, δ, γ δε φαίνεται ικανή να συμβάλλει στην παραγωγή υπογραφής για επιλεγμένο από τον $\mathcal{A}$ μήνυμα.

Τέλος αναφέρουμε δύο περιπτώσεις εσφαλμένης χρήσης του σχήματος υπογραφής ElGamal, στις οποίες μπορεί να μειωθεί η ασφάλειά του.

- Το τυχαία επιλεγμένο k πρέπει να κρατείται κρυφό αφού η γνώση του δίνει στον $\mathcal{A}$ τη δυνατότητα να υπολογίσει εύκολα το ιδιωτικό κλειδί x (Άσκηση: 3)
- Επιπλέον η επανάληψη της χρήσης του ίδιου k καθιστά για τον $\mathcal{A}$ εφικτό τον υπολογισμό του και επομένως και τον υπολογισμό του x. (Άσκηση: 4)

7.5 Πρότυπο Ψηφιακής Υπογραφής

Τον Αύγουστο του 1991 το Αμερικάνικο *Εθνικό Ινστιτούτο Προτύπων και Τεχνολογίας* (NIST) πρότεινε ένα *Πρότυπο Ψηφιακής υπογραφής* (Digital Signature Standard - DSS) βασισμένο στο Σύστημα υπογραφής ElGamal, το οποίο υιοθετήθηκε το Δεκέμβριο του 1994. Το DSS αποτελεί μία παραλλαγή του ElGamal που προσπαθεί να μειώσει το μέγεθος της ψηφιακής υπογραφής που παράγεται (ας μην ξεχνάμε ότι μία άμεση εφαρμογή των ψηφιακών υπογραφών συναντάται στις «έξυπνες κάρτες» (smart cards), όπου το μέγεθος της μνήμης είναι περιορισμένο).Στα σχήματα 7.3 και 7.4 φαίνεται η διαδικασία δημιουργίας κλειδιού και η ανταλλαγή μηνύματος υπογεγραμμένου με το DSS.

---

1. ο A επιλέγει έναν πρώτο q μεγέθους 160-bit και στη συνέχεια βρίσκει πρώτο p μεγέθους n-bit όπου το $n = 64r$,$r = 8, 9, 10, \ldots, 16$, τέτοιον που $q|(p-1)$
2. υπολογίζει ένα g που να είναι q-στη ρίζα της μονάδας $modulo\ p$, δηλαδή $g^q \equiv 1 \pmod{p}$. Ένας τρόπος γι' αυτό δίνεται στο λήμμα 7.6
3. επιλέγει έναν ακέραιο x που θα είναι το ιδιωτικό του κλειδί
4. τέλος υπολογίζει το $y \equiv g^x \pmod{p}$.

Το δημόσιο κλειδί του A είναι το (p, q, g, y).

Σχήμα 7.3: Κατασκευή κλειδιού για το DSS

Λήμμα 7.6. *Αν p πρώτος και $q : q|(p-1)$ και αν g_0: πρωταρχικό στοιχείο του $\mathbb{Z}_p^*$, τότε το*

$$g = {g_0}^{(p-1)/q}$$

Υποθέτουμε ότι ο A θέλει να στείλει στον B το μήνυμα m υπογεγραμμένο ψηφιακά με το DSS χρησιμοποιώντας το κλειδί $K = (p, q, g, x, y)$

1. **Δημιουργία υπογραφής**

   (Î±Í΄) Ο A επιλέγει έναν τυχαίο ακέραιο $k,\ 1 \leq k \leq (q-1)$.

   (Î²Í΄) Ο A υπολογίζει τα $\gamma = (g^k \bmod p) \bmod q$ και $\delta = (m + x\gamma)k^{-1} \bmod q$.

   (Î³Í΄) Η ψηφιακή υπογραφή του A για το μήνυμα m για το (τυχαία επιλεγμένο) k είναι η $sig_K(m, k) = (\gamma, \delta)$.

   (Î΄Í΄) ο A στέλνει στον B την τριάδα (m, γ, δ), δηλαδή το αρχικό του κείμενο με την ψηφιακή του υπογραφή.

2. **Επαλήθευση υπογραφής**

   (Î±Í΄) Ο B υπολογίζει τις τιμές: $e_1 = m\delta^{-1} \bmod q$ και $e_2 = \gamma\delta^{-1} \bmod q$.

   (Î²Í΄) Στη συνέχει ο B υπολογίζει την τιμή της συνάρτησης: $ver_K(m, \gamma, \delta) = \begin{cases} 1, & (g^{e_1} y^{e_2} \bmod p) \bmod q = \gamma \\ 0, & (g^{e_1} y^{e_2} \bmod p) \bmod q \neq \gamma \end{cases}$
   και πιστοποιεί ότι το μήνυμα m προέρχεται πράγματι από τον A αν και μόνον αν $ver_K(m, \gamma, \delta) = 1$

Σχήμα 7.4: Το Σχήμα υπογραφής DSS

είναι q-στη ρίζα της μονάδας modulo p.

Απόδειξη:

$$g^q \equiv (g_0^{(p-1/q)})^q \equiv g_0^{p-1} \equiv 1 \pmod{p}$$

αφού το g_0 είναι πρωταρχικό στοιχείο του $\mathbb{Z}_p^*$.

Παράδειγμα 11. *Έστω ότι ο A θέλει να στείλει στον B το μήνυμα $m = 3152$ (σε κάποια κωδικοποίηση), υπογράφοντάς το με το σχήμα υπογραφής DSS.*

- ***Κατασκευή Κλειδιού***
  Ο A επιλέγει τον $q = 107$ και $p = 86 \times 107 + 1 = 9203$. Το $g_0 = 2$ είναι πρωταρχικό στοιχείο του $\mathbb{Z}^*_{9203}$, άρα σύμφωνα με το λήμμα 7.6 μπορώ να πάρω:

$$\begin{aligned} g &= g_0^{(p-1)/q} \bmod p \\ &= 2^{(9202)/107} \bmod 9203 \\ &= 2^{86} \bmod 9203 \\ &= 645 \end{aligned}$$

  Σαν ιδιωτικό κλειδί ο A επιλέγει το $x = 111$ και υπολογίζει το

$$y = g^x \bmod p = 645^{111} \bmod 9203 = 1336\ .$$

  Το δημόσιο κλειδί του A είναι το: $(p, q, g, y) = (9203, 107, 645, 1336)$.

- ***Παραγωγή υπογραφής***
  Ο A επιλέγει τυχαία τον $k = 456$, υπολογίζει τα

$$\begin{aligned} \gamma = (g^k \bmod p) \bmod q &= (645^{456} \bmod 9203) \bmod 107 = 96 \\ k^{-1} \bmod (q) &= 456^{-1} \bmod 107 = 65 \\ \delta = (m + x\gamma)k^{-1} \bmod q &= (3152 + 111 \cdot 96) \cdot 65 \bmod 107 \\ &= 4 \end{aligned}$$

  Στη συνέχεια στέλνει στον B την υπογραφή του μαζί με το μήνυμα m δηλαδή το $(m, \gamma, \delta) = (3152, 96, 4)$.

  Επαλήθευση υπογραφής

  Ο B λαμβάνει το μήνυμα από τον A και υπολογίζει τα $\delta^{-1} \bmod q = 4^{-1} \bmod 107 = 27$,

$$\begin{aligned} e_1 = m \cdot \delta^{-1} \bmod q &\qquad e_2 = \gamma \cdot \delta^{-1} \bmod q \\ = 3152 \cdot 27 \pmod{107} &\qquad = 96 \cdot 27 \pmod{107} \quad \textit{και} \\ = 39\ , &\qquad = 24 \\ (g^{e_1}\beta^{e_2} \bmod p) \bmod q &= (645^{39} \cdot 1336^{24} \bmod 9203) \bmod 107 \\ &= 96 (= \gamma) \end{aligned}$$

οπότε καταλήγει στο συμπέρασμα ότι η υπογραφή είναι πράγματι του A.

Παρατηρήσεις στο DSS

1. Στη σχέση 1Î²Í´ έχουμε μια διαφορά με το ElGamal (το $-$ γίνεται $+$) που αποτελεί και το λόγο για την αλλαγή της συνάρτησης επικύρωσης γνησιότητας.

2. Το γεγονός ότι όλοι οι υπολογισμοί γίνονται $modulo\ q$, κάνει το μέγεθος της υπογραφής πολύ μικρότερο από την αντίστοιχη για το ElGamal. Για παράδειγμα ας θεωρήσουμε ότι ο p είναι ένας πρώτος μεγέθους 768 bit[5]. Τότε το ElGamal θα παράγει μία υπογραφή που το μέγεθός της θα είναι 1536 bit (αφού τα γ και δ υπολογίζονται $modulo 768$). Το DSS από την άλλη θα παράγει μία υπογραφή μεγέθους 320 bit.

3. Παρατηρώντας το DSS βλέπουμε ότι όλοι οι μετασχηματισμοί γίνονται μέσα σε μία υποομάδα του $\mathbb{Z}_p^*$ μεγέθους 2^{160}. Η ασφάλεια του, στηρίζεται στην εικασία ότι η επίλυση του DLOG είναι «πολύ δύσκολη» σε μια τέτοια υποομάδα του $\mathbb{Z}_p^*$.

4. Σημαντικό είναι ακόμη ο A να αποφύγει το ενδεχόμενο $\delta \equiv 0 \pmod{q}$. Αυτό γιατί στην περίπτωση αυτή δεν υπάρχει το $\delta^{-1} \bmod q$. Αν κατά τη διαδικασία της παραγωγής της υπογραφής, το δ υπολογιστεί ίσο με $0 \bmod q$, τότε ο A θα πρέπει να επαναλάβει τη διαδικασία επιλέγοντας νέο τυχαίο k.

5. Τέλος αναφέρουμε ότι η υπογραφή ενός μηνύματος με το DSS είναι εν γένει γρηγορότερη διαδικασία από την επαλήθευσή της. Το γεγονός αυτό είχε (μεταξύ άλλων ...)[6] προκαλέσει πολλές αντιδράσεις στην υιοθέτηση του προτύπου από τον NIST.[7] Η απάντηση ήταν ότι δεν έχει σημασία ποια διαδικασία είναι γρηγορότερη, αρκεί να μπορούν και οι δύο να γίνουν σε «ικανοποιητικά» μικρό χρόνο ...

[5] Για να είναι υπολογιστικά απρόσιτο.

[6] Γενικά υπάρχει μία δυσπιστία απέναντι στα κρυπτογραφικά πρότυπα που υιοθετούνται από τον NIST, που αφορά το κατά πόσο αυτά είναι ασφαλή όταν βρεθούν αντιμέτωπα με μία επίθεση από την NSA.

[7] Εκείνοι που αντιδρούσαν υποστήριζαν ότι έπρεπε να συμβαίνει το αντίστροφο, αφού συνήθως η υπογραφή ενός εγγράφου χρειάζεται να παραχθεί μία φορά, ενώ η επικύρωση του γνησίου της μπορεί να χρειαστεί να γίνει σε περισσότερες από μία περιπτώσεις.

7.6 Σχήματα υπογραφών με επιπρόσθετη λειτουργικότητα

Στην ενότητα αυτή θα συζητήσουμε κάποια *σχήματα υπογραφών*, που παρουσιάζουν συγκεκριμένες ιδιότητες, ώστε να εξυπηρετούν κάποιες πρόσθετες ανάγκες των ατόμων που τις χρησιμοποιούν.

7.6.1 Υπογραφές μιας χρήσης (One-time signatures)

Με τον όρο *Υπογραφές μιας χρήσης* εννοούμε *σχήματα υπογραφών* στα οποία κάθε υπογραφή (κάθε κλειδί) μπορεί να χρησιμοποιηθεί για να υπογράψει ένα μόνο μήνυμα, αλλιώς η υπογραφή μπορεί να πλαστογραφηθεί. Η πλειοψηφία αυτών των σχημάτων έχουν το πλεονέκτημα ότι τόσο η παραγωγή όσο και η επικύρωση της γνησιότητας της υπογραφής υλοποιούνται με πολύ αποδοτικούς αλγορίθμους, και επομένως προτιμούνται σε συσκευές με μικρή υπολογιστική ισχύ (π.χ. chipcards).

Χαρακτηριστικό όλων των *σχημάτων υπογραφών μιας χρήσης* είναι η χρήση κάποιας συνάρτησης μονής κατεύθυνσης (one-way function).[8]. Παρακάτω θα περιγράψουμε ένα τέτοιο σχήμα το *Σχήμα υπογραφής Lamport*, το οποίο φέρει αρκετές εννοιολογικές ομοιότητες με τον κώδικα Vernam 1.2.2.

Έστω ότι ο A θέλει να κατασκευάσει ένα κλειδί για να υπογράψει ένα μήνυμα m που είναι μία δυαδική ακολουθία μήκους k-bit, όπου $k \in \mathbb{N}$. Έστω ακόμη ότι $\mathtt{S} = Y$ και $f : Y \to Z$ είναι μία συνάρτηση μονής κατεύθυνσης δημοσίως γνωστή.

1. ο A επιλέγει $2k$ τιμές από το σύνολο Y, τα $y_{i,j}, 1 \leq i \leq k,\ j = 0, 1$. Κατασκευάζει έτσι το ιδιωτικό του κλειδί που είναι ο $k \times 2$ πίνακας $(y_{i,j})$.
2. υπολογίζει τα $z_{i,j} \in Z$ τέτοιοι ώστε $z_{i,j} = f(y_{i,j}), 1 \leq i \leq k,\ j = 0, 1$. Κατασκευάζει έτσι το δημόσιο κλειδί του που είναι ο $k \times 2$ πίνακας $(z_{i,j})$.

Σχήμα 7.5: Παραγωγή κλειδιού για το Lamport

Παράδειγμα 12. *Έστω ότι η* $f(x) = 3^x \bmod 7879$. *Ο* A *επιλέγει τυχαίους αριθμούς*

[8] Μία τέτοια συνάρτηση είναι για παράδειγμα η $f(x) = g^x \bmod p$ για p πρώτο και g πρωταρχικό στοιχείου του $\mathbb{Z}_p^*$.

Υποθέτουμε ότι ο Α θέλει να στείλει στον Β το μήνυμα

$$m = (x_1, x_2, \ldots, x_k), \text{ όπου } x_i \in \{0, 1\}, i = 1, 2, \ldots, k \ ,$$

υπογεγραμμένο ψηφιακά με το Lamport χρησιμοποιώντας το κλειδί K που δημιουργήθηκε με τη διαδικασία του σχήματος 7.5

1. **Δημιουργία υπογραφής**
   Η ψηφιακή υπογραφή του A για το μήνυμα m με χρήση του κλειδιού K είναι η $sig_K(x_1, x_2, \ldots, x_k) = (y_{1,x_1}, y_{2,x_2}, \ldots, y_{k,x_k}) = \vec{s}$. Ο A στέλνει στον B το μήνυμα $(m, \vec{s})$.

2. **Επαλήθευση υπογραφής**
   Ο B υπολογίζει την τιμή της συνάρτησης:

$$ver_K(x_1, x_2, \ldots, x_k, c_1, c_2, \ldots, c_k) =$$
$$\begin{cases} 1, & f(c_i) = z_{i,x_i}, i = 1, 2, \ldots, k \\ 0, & f(c_j) \neq z_{j,x_j} \end{cases} \text{, όπου } \vec{s} = (c_1, c_2, \ldots, c_k)$$

   και πιστοποιεί ότι το μήνυμα m προέρχεται πράγματι από τον A αν και μόνον αν $ver_K(m, \vec{s}) = 1$

Σχήμα 7.6: Το Σχήμα υπογραφής Lamport

και κατασκευάζει τον πίνακα:

$$(y_{i,j}) = \begin{pmatrix} y_{1,0} = 5831 & y_{1,1} = 735 \\ y_{2,0} = 803 & y_{2,1} = 2467 \\ y_{3,0} = 4285 & y_{3,1} = 6449 \end{pmatrix}$$

Στη συνέχεια υπολογίζει τις εικόνες των $y_{i,j}$ πάνω από την f και κατασκευάζει τον πίνακα:

$$(z_{i,j}) = \begin{pmatrix} z_{1,0} = 2009 & z_{1,1} = 3810 \\ z_{2,0} = 4672 & z_{2,1} = 4721 \\ z_{3,0} = 268 & z_{3,1} = 5731 \end{pmatrix},$$

τον οποίο και δημοσιεύει.
Έστω τώρα ότι ο Α θέλει να στείλει στον Β το μήνυμα $m = (1, 1, 0)$ υπογεγραμμένο. Η υπογραφή του θα είναι:

$$\begin{aligned} sig_K(1,1,0) &= (y_{1,1}, y_{2,1}, y_{3,0}) \\ &= (735, 2467, 4285) \end{aligned}$$

Για να επαληθεύσει την υπογραφή ο B υπολογίζει τα

$$\begin{aligned} f(735) &= 3^{735} \bmod 7879 \\ &= 3810 (= z_{1,1}), \\ f(2467) &= 3^{2467} \bmod 7879 \\ &= 4721 (= z_{2,1}), \\ f(4285) &= 3^{4285} \bmod 7879 \\ &= 268 (= z_{3,0}) \end{aligned}$$

και καταλήγει στο συμπέρασμα ότι η υπογραφή είναι πράγματι γνήσια.

Όσον αφορά την ασφάλεια του σχήματος Lamport, αυτή προφανώς οφείλεται στην αδυναμία του $\mathcal{A}$ να αντιστρέψει τη συνάρτηση f. Είναι εντούτοις πολύ σημαντικό ο ίδιος πίνακας $(y_{i,j})$ να μη χρησιμοποιηθεί για την υπογραφή περισσοτέρων του ενός μηνυμάτων. Σε αντίθετη περίπτωση ο $\mathcal{A}$ θα μπορεί να κατασκευάσει την υπογραφή του A και σε άλλα μηνύματα.

Παράδειγμα 13. *Ας υποθέσουμε ότι ο A χρησιμοποιεί τον ίδιο πίνακα $(y_{i,j})$, για να υπογράψει τα μηνύματα:*

$$m_1 = (0,1,1) \text{ και } m_2 = (1,0,1).$$

οι υπογραφές θα είναι οι:

$$sig_K(m_1) = (y_{1,0}, y_{2,1}, y_{3,1}) \text{ και } sig_K(m_2) = (y_{1,1}, y_{2,0}, y_{3,1})\,.$$

αντίστοιχα. Τότε ο $\mathcal{A}$ μπορεί εύκολα να παράγει υπογραφές για τα μηνύματα:

$$m_3 = (1,1,1) \text{ και } m_4 = (0,0,1)\,,$$

που θα είναι οι :

$$(y_{1,1}, y_{2,1}, y_{3,1}) \text{ και } (y_{1,0}, y_{2,0}, y_{3,1})\,.$$

7.6.2 Τυφλές υπογραφές (Blind signatures)

Τα σχήματα τυφλών υπογραφών εξυπηρετούν γενικά ανάγκες ηλεκτρονικής επικοινωνίας που η μία πλευρά επιθυμεί ανωνυμία απέναντι στην άλλη. Πρόκειται πρακτικά περισσότερο για ένα *πρωτόκολλο επικοινωνίας* μεταξύ τους με την παραπάνω ιδιότητα. Σε αυτό μία οντότητα θέλει να λάβει μία υπογραφή σε κάποιο

μήνυμα, χωρίς όμως ο υπογράφων να λάβει γνώση του τι υπογράφει. Εφαρμογές τέτοιων πρωτοκόλλων συναντούμε στο ηλεκτρονικό χρήμα 11.3, στις ηλεκτρονικές ψηφοφορίες 11.1.4 και σε πολλές άλλες περιπτώσεις.

Η ιδέα των απλών υπογραφών είναι αρκετά απλή και μπορεί να περιγραφεί με μία απλή αναλογία, ενός πελάτη που θέλει να αγοράσει κάποια αγαθά από έναν έμπορο μέσω τράπεζας, χωρίς όμως η τράπεζα να μπορεί να παρακολουθήσει τη συναλλαγή [2].

- Ο καταναλωτής τοποθετεί ένα κενό φύλλο χαρτιού σε ένα φάκελο που περιέχει ένα κομμάτι καρμπόν.
- Αποστέλλει τον φάκελο στην τράπεζα ζητώντας από το λογαριασμό του, το χρηματικό ποσό που αξίζει το αγαθό.
- Η τράπεζα επαληθεύει ότι το συγκεκριμένο ποσό υπάρχει στο λογαριασμό και σε θετική περίπτωση υπογράφει το φάκελο. Λόγω του καρμπόν η συναλλαγή μεταφέρεται στο κενό χαρτί.
- Ο καταναλωτής επαληθεύει την υπογραφή της τράπεζας και ανακτά το περιεχόμενο του φακέλου με το οποίο πληρώνει τον έμπορο.
- Ο έμπορος λαμβάνει το υπογεγραμμένο φύλλο και το παρουσιάζει στην τράπεζα.
- Η τράπεζα επαληθεύει την υπογραφή και μεταφέρει στον έμπορο το συμφωνημένο ποσό.

Η αναλογία οδηγεί στον παρακάτω τυπικό ορισμό για τα σχήματα τυφλής υπογραφής [4]:

Ορισμός 7.7. Ένα σχήμα τυφλών υπογραφών είναι ένα σύνολο από 4 αλγόριθμους $(Blind, Sign, Verify, Unblind)$ ώστε:

1. Το μήνυμα m προς υπογραφή 'τυφλώνεται' με χρήση του αλγόριθμου τύφλωσης και κάποια τυχαιότητας, δίνοντας $b = Blind(m, r)$
2. Στη συνέχεια εφαρμόζεται ο αλγόριθμος υπογραφής $S_b = Sign(b)$ με τρόπο ώστε η υπογραφή να μεταφερθεί στο m. Η S_b είναι η *τυφλή υπογραφή*.
3. Ο κάτοχος του μηνύματος εφαρμόζει την συνάρτηση αποτύφλωσης, αποκτώντας μια κανονική υπογραφή στο αρχικό μήνυμα $S_m = Unblind(S_b)$
4. Οποιοσδήποτε μπορεί να την επαληθεύσει εκτελώντας την συνάρτηση $Verify(S_m)$

Το πρώτο σχήμα τυφλής υπογραφής (7.7) παρουσιάστηκε από τον David Chaum στο [1] και βασίζεται στο RSA.

Υποθέτουμε ότι ο A θέλει από τον B να υπογράψει τυφλά *ένα μήνυμα* m.
Υποθέτουμε επιπλέον ότι το δημόσιο κλειδί του B *για το RSA είναι το* (n, e) *και το αντίστοιχο ιδιωτικό του το* (p, q, d) *(βλέπε 7.3).*

1. (Τύφλωση) Ο A επιλέγει έναν τυχαίο ακέραιο r τέτοιον ώστε $0 \leq r \leq n-1$ και $gcd(n, k) = 1$ και υπολογίζει το $b = Blind(m, r) = r^e H(m) \pmod{n}$
2. (Υπογραφή) Ο B υπολογίζει το $S_b = Sign(b) = r^{ed \pmod{\phi(n)}} H(M)^d \pmod{n} = rH(m)^d$ και το στέλνει στον A.
3. (Αποτύφλωση) $S_m = Unblind(S_b) = S_b \frac{1}{r} = H(m)^d \pmod{n}$
4. Οποιοσδήποτε μπορεί να επαληθεύσει την υπογραφή ως: $Verify(S_m) =$ if ${S_m}^e = H(m)$ then $True$ else $False$

Σχήμα 7.7: Το Σχήμα υπογραφής του Chaum

7.6.3 Αδιαμφισβήτητες υπογραφές (Undeniable signatures)

Βασικό χαρακτηριστικό των *αδιαμφισβήτητων σχημάτων υπογραφής* (Undeniable Signature Schemes) είναι ότι η επικύρωση γνησιότητας της υπογραφής του A δεν μπορεί να γίνει χωρίς τη συνεργασία του ίδιου. Έτσι ο A μπορεί να γνωρίζει για το έγγραφο που υπογράφει τις περιπτώσεις στις οποίες χρειάστηκε η επικύρωση της υπογραφής του. Μπορεί επομένως να αποφύγει την επαναχρησιμοποίηση του εγγράφου του από φορείς που δεν επιθυμεί. Ας δούμε όμως μερικά παραδείγματα χρήσης ενός τέτοιου σχήματος υπογραφής.

Παράδειγμα 14. *Υποθέτουμε ότι ο A είναι ένας πελάτης της τράπεζας B και θέλει να πάρει πρόσβαση σε μία υψηλής ασφάλειας περιοχή της τράπεζας (π.χ. το θησαυροφυλάκιο). Η τράπεζα (B) ζητάει από τον A να υπογράψει ένα κείμενο με ημερομηνία και ώρα για να του δώσει την απαιτούμενη πρόσβαση. Αν ο A υπογράψει χρησιμοποιώντας αδιαμφισβήτητη υπογραφή τότε ο B δε θα μπορεί να αποδείξει ότι η υπογραφή αυτή ανήκει στον A, παρά μόνο με τη συνεργασία του ίδιου του A*

Παράδειγμα 15. *Υποθέτουμε τώρα ότι μία μεγάλη εταιρία λογισμικού A δημιουργεί ένα νέο πακέτο λογισμικού. Η A υπογράφει το πακέτο και το στέλνει στην εταιρία B. Η B τώρα αντιγράφει το πακέτο και το μεταπωλεί σε έναν πελάτη της C. Ο C δεν μπορεί να επαληθεύσει τη γνησιότητα του πακέτου χωρίς τη συνεργασία της A. Φυσικά ο B θα μπορούσε να επαναϋπογράψει το πακέτο και να το στείλει στον C σαν δικό του, αλλά τότε αφενός το πακέτο θα έχανε το αγοραστικό πλεονέκτημα της προέλευσης από τη γνωστή εταιρία A, αφετέρου θα ήταν εύκολο να αποδειχθεί η απάτη αυτή.*

Στα σχήματα 7.8 και 7.9 φαίνεται η δημιουργία κλειδιού και η διαδικασία υπογραφής – επαλήθευσης για το *αδιαμφισβήτητο σχήμα υπογραφής Chaum-van Antwerpen*.

Λήμμα 7.8. *Αν p, q: πρώτοι τέτοιοι που $p = 2q + 1$ τότε το σύνολο G που δημιουργείται από τα τετραγωνικά υπόλοιπα $modulo\ p$ των στοιχείων του $\mathbb{Z}_p^*$ αποτελεί πολλαπλασιαστική υποομάδα του $\mathbb{Z}_p^*$ τάξης q.*

Παρατήρηση 17. Από το λήμμα 7.6 μπορούμε επίσης να υπολογίσουμε μία τέτοια υποομάδα του $\mathbb{Z}_p^*$ που θα αποτελείται απο τις μέχρι τάξης q δυνάμεις του $g = {g_0}^{(p-1)/q}$, όπου g_0 πρωταρχικό στοιχείο του $\mathbb{Z}_p^*$.

Πρόταση 7.9. *Η συνθήκη επαλήθευσης επιστρέφει «Αληθής» αν τα (m, d) έχουν προκύψει με τη διαδικασία που περιγράφεται στο σχήμα 7.9.*

Απόδειξη: Πράγματι επειδή

$$y \equiv g^x \pmod{p}$$

Έστω ότι ο A θέλει να κατασκευάσει ένα κλειδί για να υπογράψει ένα μήνυμα m με το αδιαμφισβήτητο σχήμα υπογραφής Chaum-van Antwerpen. Τα βήματα που ακολουθεί είναι:

1. Επιλέγει p, q: πρώτους τέτοιους που $p = 2q + 1$
2. Κατασκευάζει την τάξης q πολλαπλασιαστική υποομάδα του $\mathbb{Z}_p^*$ G, και υπολογίζει ένα πρωταρχικό στοιχείο της g.
3. Επιλέγει x τέτοιον που $1 \leq x \leq p-1$ και υπολογίζει το $y \equiv g^x \pmod{p}$.

Το δημόσιο κλειδί του A είναι το (p, q, g, y) και το ιδιωτικό του το x.

Σχήμα 7.8: Παραγωγή κλειδιού για το αδιαμφισβήτητο σχήμα υπογραφής Chaum-van Antwerpen

έχουμε ότι

$$y^{x^{-1}} \equiv g \pmod{p} \tag{7.5}$$

όμοια και

$$s \equiv m^x \pmod{p}$$

συνεπάγεται ότι

$$s^{x^{-1}} \equiv m \pmod{p} \tag{7.6}$$

όμως

$$\begin{aligned} d &\equiv c^{x^{-1}} \pmod{p} \\ &\equiv s^{e_1 x^{-1}} y^{e_2 x^{-1}} \pmod{p}. \end{aligned}$$

Από τις σχέσεις (7.6.3), (7.6.3), (7.6.3) έχουμε ότι

$$d \equiv m^{e_1} g^{e_2} \pmod{p}$$

Δίνουμε παρακάτω ένα θεώρημα χωρίς απόδειξη που αφορά την ασφάλεια του *σχήματος υπογραφής Chaum-van Antwerpen*.

Θεώρημα 7.10. *Αν $s' \not\equiv m^x \pmod{p}$ τότε η πιθανότητα να δεχθεί ο B το s' ως έγκυρη υπογραφή για το μήνυμα m, χρησιμοποιώντας τη διαδικασία του σχήματος 7.9 είναι $1/q$*

Υποθέτουμε ότι ο A θέλει να στείλει στον B το μήνυμα m υπογεγραμμένο ψηφιακά με το αδιαμφισβήτητο σχήμα υπογραφής Chaum-van Antwerpen χρησιμοποιώντας το κλειδί $K = (p, q, g, y, x)$

1. **Δημιουργία υπογραφής**

   - Ο A υπολογίζει το $(sig_K(m, k) =)s = m^x \bmod p$ που είναι η ψηφιακή υπογραφή του A για το μήνυμα m.
   - στέλνει στον B το (m, s)

2. **Επαλήθευση υπογραφής**

   - Ο B επιλέγει τυχαίους $e_1, e_2 \in \mathbb{Z}_q^*$ και υπολογίζει το $c = s^{e_1} y^{e_2} \bmod p$ και το στέλνει στον A
   - Ο A υπολογίζει το $d = c^{x^{-1} \bmod q} \bmod p$ και το στέλνει στον B
   - Ο B υπολογίζει την τιμή της συνάρτησης:

$$ver_K(m, d) = \begin{cases} 1, & d \equiv m^{e_1} y^{e_2} \pmod p \\ 0, & d \not\equiv m^{e_1} y^{e_2} \end{cases}$$

   και πιστοποιεί ότι το μήνυμα m προέρχεται πράγματι από τον A αν και μόνον αν $ver_K(m, d) =$Αληθής

Σχήμα 7.9: Το αδιαμφισβήτητο σχήμα υπογραφής Chaum-van Antwerpen

Παρατήρηση 18. Αυτό σημαίνει ότι ο C μπορεί με πιθανότητα $1/q$ να κατασκευάσει μία πλαστή υπογραφή για το m που ο B θα τη δεχθεί ως έγκυρη.

Μέχρι εδώ όμως δεν έχει φανεί πουθενά πώς δικαιολογείται ο τίτλος *Αδιαμφισβήτητη Υπογραφή*. Η ονομασία αυτή οφείλεται σε ένα *Πρωτόκολλο Αποκήρυξης (Disavowal Protocol)* που συνοδεύει αυτά τα σχήματα υπογραφών. Με χρήση αυτού του πρωτοκόλλου είναι πολύ δύσκολο για τον A να αρνηθεί ότι μία υπογραφή που έχει κατασκευάσει είναι πράγματι δική του. Ας δούμε όμως πρώτα πώς θα μπορούσε να γίνει αυτή η άρνηση από πλευράς του A

Υποθέτουμε ότι ο A ισχυρίζεται ότι η υπογραφή s του μηνύματος m είναι πλαστή.
Η διαδικασία αποτελείται από τα παρακάτω βήματα:

1. Ο B επιλέγει τυχαία $e_1, e_2 \in \mathbb{Z}_q^*$. Στη συνέχεια υπολογίζει το $c = s^{e_1} y^{e_2} \bmod p$ και το στέλνει στον A
2. Ο A υπολογίζει το $d = c^{x^{-1} \bmod q} \bmod p$ και το στέλνει στον B
3. Ο B επαληθεύει ότι $d \not\equiv m^{e_1} g^{e_2} \pmod p$. Στη συνέχεια επιλέγει τυχαία $f_1, f_2 \in \mathbb{Z}_q^*$ και υπολογίζει το $C = s^{f_1} \beta^{f_2} \bmod p$ το οποίο και στέλνει στον A
4. Ο A υπολογίζει το $D = C^{x^{-1} \bmod q} \bmod p$ και το στέλνει στον B
5. Ο B επαληθεύει ότι $D \not\equiv m^{f_1} g^{f_2} \pmod p$ και συμπεραίνει ότι το s είναι προϊόν πλαστογραφίας αν και μόνο αν

$$(dg^{-e_2})^{f_1} \equiv (Dg^{-f_2})^{e_1} \pmod p \tag{7.7}$$

Σχήμα 7.10: Πρωτόκολλο Αποκήρυξης

Ο A μπορεί:

1. να αρνηθεί εξ αρχής να συμμετάσχει στη διαδικασία επαλήθευσης της υπογραφής.
2. να δώσει πλαστά δεδομένα κατά τη διαδικασία επαλήθευσης
3. να ισχυριστεί ότι η υπογραφή είναι πλαστή παρόλο που η συνάρτηση επαλήθευσης επιστρέφει 1

Στην περίπτωση (1) η κίνηση αυτή του A θα θεωρηθεί άμεσα ύποπτη οπότε η υπογραφή δε θα ληφθεί υπόψιν σε πιθανό δικαστήριο. Για να αντιμετωπιστούν οι περιπτώσεις (2) και (3) χρειάζεται ένα *Πρωτόκολλο Αποκήρυξης* όπως αυτό που περιγράφεται στο σχήμα 7.10

Παρατήρηση 19. Αποδεικνύεται ότι αν $s \not\equiv m^a \pmod p$, και οι A και B ακολου-

θήσουν το *Πρωτόκολλο Αποκήρυξης* τότε

$$(dg^{-e_2})^{f_1} \equiv (Dg^{-f_2})^{e_1} \pmod{p}$$

(η απόδειξη αφήνεται ως άσκηση)

Παρατήρηση 20. Αν $s \equiv m^x \pmod{p}$ και ο B ακολουθήσει το *Πρωτόκολλο Αποκήρυξης* τότε αν ο A δώσει ψευδή d και D τότε η πιθανότητα να ισχύει η σχέση 5 (δηλαδή να κατασκευάσει ο A δεδομένα τέτοια που να οδηγούν σε συμπέρασμα ότι έχει γίνει πλαστογραφία) είναι $1/q$.

7.6.4 Οι υπογραφές Fail-Stop

Το σχήμα υπογραφής Fail-Stop παρέχει επιπλέον ασφάλεια για την περίπτωση όπου κάποιος πολύ ισχυρός αντίπαλος θα μπορούσε να πλαστογραφήσει μια υπογραφή. Σε μια τέτοια περίπτωση, ο υπογράφων A είναι σε θέση να αποδείξει με πολύ μεγάλη πιθανότητα ότι η υπογραφή είναι πλαστογραφημένη.

Ένα σχήμα υπογραφής Fail-Stop προτάθηκε από τους van Heyst και Pedersen το 1992. Το *σχήμα υπογραφής Fail-Stop των van Heyst και Pedersen* είναι ένα σχήμα μιας χρήσης, και φαίνεται στο σχήμα 7.11. Η βασική διαφορά του με τα σχήματα που περιγράψαμε μέχρι τώρα είναι η ύπαρξη μιας *Έμπιστης Αρχής* (συχνά αναφερόμαστε σ' αυτήν σαν *Έμπιστη Τρίτη Οντότητα (Trusted Third Party (TTP))*).

Μπορούν να αποδειχθούν τα εξής:

Αν ο αντίπαλος ξέρει τα $sig_K(m) = s$ και $m' \neq m$, μπορεί να υπολογίσει το $sig_K(m')$ με πιθανότητα $1/q$. Με άλλα λόγια,δεδομένης μιας έγκυρης υπογραφής s για ένα μήνυμα m, υπάρχουν q διαφορετικά κλειδιά που θα μπορούσαν να έχουν υπογράψει το μήνυμα m με s (αυτά μπορούν να έχουν χρησιμοποιηθεί από τον νόμιμο χρήστη). Για κάθε άλλο μήνυμα όμως, $m' \neq m$ αυτά τα q διαφορετικά κλειδιά θα δώσουν διαφορετικές υπογραφές.

Έτσι, αν δίνεται μια υπογραφή s για ένα κείμενο x, ο αντίπαλος δεν μπορεί να υπολογίσει την υπογραφή s' για ένα άλλο μήνυμα m'. Από την άλλη μεριά, είναι δυνατόν ο αντίπαλος να υπολογίσει μια πλαστογραφημένη υπογραφή $s'' \neq sig_K(m')$ η οποία να μπορεί παρόλα αυτά να επικυρωθεί. Στην περίπτωση αυτή όμως, το άτομο του οποίου η υπογραφή πλαστογραφήθηκε, μπορεί με πιθανότητα $1 - 1/q$ να αποδείξει την πλαστογράφηση. Η απόδειξη της πλαστογράφησης είναι η τιμή $x = \log_g y$, που είναι γνωστή μόνο στην κεντρική αρχή.

7.6.5 Ομαδικές υπογραφές (Group signatures

Τα σχήματα ομαδικών υπογραφών προτάθηκαν από τον Chaum και τον Heyst στο [3].

Σε αυτές μία ομάδα οντοτήτων θέλει να δημιουργήσει ένα σχήμα υπογραφών με τις εξής ιδιότητες:

- Μόνο τα μέλη της ομάδας μπορούν να υπογράφουν μηνύματα.
- Ο παραλήπτης της υπογραφής μπορεί να επαληθεύσει την εγκυρότητα της υπογραφής χωρίς όμως να μπορεί να διακρίνει το μέλος της ομάδας που υπέγραψε (anonymity)
- Σε περίπτωση αντιδικίας μπορεί να ανακαλυφθεί το μέλος που υπέγραψε, με την βοήθεια ενός ειδικού μέλους του αρχηγού.(traceability)

Στη βιβλιογραφία έχουν προταθεί αρκετά σχήματα ομαδικών υπογραφών. Στην ενότητα αυτή θα περιγράψουμε το [6], το οποίο περιγράφει πώς μπορεί να δημιουργηθεί ένα σχήμα ομαδικών υπογραφών από ένα απλό κρυπτοσύστημα δημοσίου κλειδιού.

Απαιτεί την ύπαρξη μια έμπιστης αρχής η οποία εκτός από τις παραμέτρους των κρυπτοσυστημάτων, παράγει το ζεύγος κλειδιών της ομάδας (x_G, y_G) το οποίο μοιράζεται σε όλα τα μέλη της ομάδας. Ο αρχηγός παράγει το δικό του ζεύγος (x_z, y_z). Υποθέτουμε ότι τα μέλη της ομάδας έχουν ταυτότητες $ID_1, \cdots ID_n$.

Για την υπογραφή ενός μηνύματος m το κάθε μέλος κρυπτογραφεί την ταυτότητα του με το δημόσιο κλειδί του αρχηγού, παράγοντας το $c = E_{y_z}(ID_U)$. Στη συνέχεια προσαρτά το c στο μήνυμα και παράγει το $m' = \mathcal{H}(m, c)$ όπου $\mathcal{H}$ μια συνάρτηση σύνοψης. Στη συνέχεια παράγει την υπογραφή $\sigma = sign_{x_G}(m')$ Η τελική υπογραφή είναι η τετράδα $(m, c, \sigma, proof)$ όπου $proof$ είναι μία απόδειξη ότι το c κρυπτογραφεί σωστά μία από τις ταυτότητες χωρίς φυσικά να αποκαλύπτει ποια (βλ. 10).

Για την επαλήθευση της υπογραφής εκτελείται ο αλγόριθμος $verify$ του σχήματος υπογραφών και ελέγχεται η απόδειξη $proof$. Φυσικά μόνο ο αρχηγός της ομάδας μπορεί να αποκρυπτογραφήσει την ταυτότητα και να βρει ποιο μέλος υπέγραψε, ικανοποιώντας έτσι την ιδιότητα του traceability.

7.7 Ασκήσεις

1. Να αποδείξετε στο RSA ότι αν ο χώρος των κρυπτοκειμένων είναι ο ίδιος με το χώρο των απλών κειμένων τότε ισχύει και $\texttt{Encrypt}_K(\texttt{Decrypt}_K(m)) = m$.

2. Να αποδείξετε το λήμμα 7.4.

3. Να αποδείξετε ότι στο σχήμα υπογραφής ElGamal ο παράγοντας τυχαιότητας k πρέπει να κρατείται κρυφό αφού η γνώση του δίνει στον $\mathcal{A}$ τη δυνατότητα να υπολογίσει εύκολα το ιδιωτικό κλειδί x.

4. Να αποδείξετε ότι στο σχήμα υπογραφής ElGamal η επανάληψη της χρήσης του ίδιου k καθιστά για τον $\mathcal{A}$ εφικτό τον υπολογισμό του και επομένως και τον υπολογισμό του x.

5. Θεωρήστε την παραλλαγή του σχήματος υπογραφής El Gamal όπου η μόνη διαφορά βρίσκεται στον υπολογισμό του δ:

$$\delta = (m - k\gamma)\alpha^{-1} \pmod{p-1}$$

(i) Περιγράψτε τη συνάρτηση επαλήθευσης της υπογραφής (γ, δ) για το μήνυμα m.

(ii) Υπάρχει κάποιο υπολογιστικό πλεονέκτημα του τροποποιημένου σχήματος έναντι του αρχικού;

(iii) Συγκρίνετε σύντομα την ασφάλεια του αρχικού και του τροποποιημένου σχήματος.

6. Ας υποθέσουμε ότι υπάρχει μία έμπιστη αρχή η οποία υπολογίζει για λογαριασμό καποιων χρηστών τις ψηφιακές υπογραφές τους. Δηλαδή αν κάποιος χρήστης θέλει να υπολογίσει την (RSA) ψηφιακή υπογραφή του για το κείμενο m, τότε στέλνει στην έμπιστη αρχή το κείμενο m. Η έμπιστη αρχή, που διαθέτει τον εκθέτη αποκρυπτογραφησης d και τα p, q του χρήστη αυτού, υπολογίζει την ψηφιακή υπογραφή ως εξής: υπολογίζει πρώτα τα $S_1 = m^d \pmod{p}$ και $S_2 = m^d \pmod{q}$ και μετά με χρήση του κινέζικου θεωρήματος υπολογίζει την υπογραφή S. Αν υποθέσουμε ότι η έμπιστη αρχή έκανε λάθος στον υπολογισμό του S_1 και υπολόγισε $\tilde{S}_1 \neq S_1$ – και για υπογραφή $\tilde{S} \neq S$ – δείξτε ότι τότε μπορεί κάποιος που γνωρίζει τα $m, \tilde{S}$ και το δημόσιο κλειδί (N, e) να παραγοντοποιήσει αποδοτικά το N.Πώς μπορεί η έμπιστη αρχή να αποφύγει αυτόν τον κίνδυνο;

Υπόδειξη: εξετάστε την ισοτιμία του $\tilde{S}^e$ modulo p και modulo q.

7. Οι χρήστες ενός δικτύου χρησιμοποιούν το κρυπτοσύστημα RSA. Κάθε χρήστης U_i διαθέτει ένα δημόσιο κλειδί n_i, e_i και ένα ιδιωτικό κλειδί d_i. Οι χρήστες χρησιμοποιούν το σύστημα τόσο για κρυπτογράφηση όσο και για υπογραφή. Δηλαδή, κάθε χρήστης U_i χρησιμοποιεί το ιδιωτικό του κλειδί d_i (και το n_i) για να υπογράψει ένα μήνυμα, και το δημόσιο κλειδί του U_k, δηλ. το (e_k, n_k) για να κρυπτογραφήσει ένα μήνυμα και να το στείλει στον U_k.

(α) Δείξτε ότι αυτό μπορεί να είναι επικίνδυνο στο εξής σενάριο: αν η διευθύντρια U_i υπογράφει ο,τιδήποτε της δίνει ο έμπιστος γραμματέας της, τότε ο γραμματέας μπορεί να αποκρυπτογραφήσει κάθε μήνυμα που στέλνει ο χρήστης U_k στη διευθύντρια.

(β) Έστω ότι η διευθύντρια είναι κάπως καχύποπτη, και αρνείται να δώσει το υπογεγραμμένο μήνυμα στον γραμματέα, αν αυτό μοιάζει πολύ με ένα κανονικό κείμενο. Πώς ο γραμματέας μπορεί να παρακάμψει αυτό το "πρόβλημα";

Βιβλιογραφία

[1] David Chaum. Blind signatures for untraceable payments. In D. Chaum, R.L. Rivest, and A.T. Sherman, editors, *Advances in Cryptology Proceedings of Crypto 82*, pages 199–203, 1983.

[2] David Chaum. Security without identification: transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10):1030–1044, October 1985.

[3] David Chaum and Eugene Heyst. Group Signatures. In DonaldW. Davies, editor, *Advances in Cryptology — EUROCRYPT '91*, volume 547 of *Lecture Notes in Computer Science*, pages 257–265. Springer Berlin Heidelberg, 1991.

[4] Aggelos Kiayias. Cryptography primitives and protocols, 2015. Διαθέσιμο στο http://crypto.di.uoa.gr/class/Kryptographia/Semeioseis_files/Cryptograph_Primitives_and_Protocols.pdf.

[5] Christopher Barth Kuner and Anja Miedbrodt. Written signature requirements and electronic authentication: A comparative perspective. *The EDI Law Review*, 6(2-3):143–154, 1999.

[6] Holger Petersen. How to Convert any Digital Signature Scheme into a Group Signature Scheme. In *Security Protocols Workshop*, pages 177–190, 1997.

Σχήμα υπογραφής van Heyst–Pedersen

Έστω $p = 2q + 1$ πρώτος αριθμός, τέτοιος ώστε ο q να είναι πρώτος αριθμός και το πρόβλημα του διακριτού λογαρίθμου στο $\mathbb{Z}_p^*$ είναι απρόσιτο (intractable). Έστω $g \in \mathbb{Z}_p^*$ πρωταρχικό στοιχείο του $\mathbb{Z}_q^*$. Έστω $x \in \mathbb{Z}_p^*$. Ορίζουμε το $y = g^x \bmod p$. Οι τιμές p, q, g, y και x έχουν επιλεγεί από μία TTP. Τα p, q, g, y είναι γνωστά δημοσίως. Η τιμή του x κρατείται μυστική. Υποθέτουμε ότι η A θέλει να στείλει το μήνυμα $m (\in \mathbb{Z}_q^*)$ στον B.

1. **Δημιουργία υπογραφής**

   (Î±Í΄) Η A επιλέγει $a_1, a_2, b_1, b_2 \in \mathbb{Z}_p^*$ *(αυτό είναι το ιδιωτικό του κλειδί)*.

   (Î²Í΄) Η A υπολογίζει τα: $\gamma_1 = g^{a_1} y^{a_2} \bmod p$, $\gamma_2 = g^{b_1} y^{b_2} \bmod p$. Το δημόσιο κλειδί του A είναι το $K = (\gamma_1, \gamma_2)$

   (Î³Í΄) Η A υπολογίζει τα $s_1 = a_1 + m b_1 \bmod q$, $s_2 = a_2 + m b_2 \bmod q$

   (Î΄Í΄) Η υπογραφή της A για το μήνυμα m είναι: $(sig_K(m) =) s = (s_1, s_2)$. Και ο A στέλνει στον B το (m, s)

2. **Επαλήθευση υπογραφής**
   Ο B υπολογίζει την τιμή της συνάρτησης:

$$ver_K(m, s) = \begin{cases} 1, & \gamma_1 \gamma_2^m = g^{s_1} y^{s_2} \bmod p \\ 0, & \gamma_1 \gamma_2^m \neq g^{s_1} y^{s_2} \bmod p \end{cases}$$

   και πιστοποιεί ότι το μήνυμα m προέρχεται πράγματι από την A αν και μόνον αν $ver_K(m, s) = 1$

3. **Απόδειξη της πλαστογράφησης**
   Η A δέχεται την πιθανή πλαστή υπογραφή (s_1', s_2') για το μήνυμα m'. Αν $(s_1, s_2) = (s_1', s_2')$ τότε επιστρέφει στο πρώτο βήμα αλλιώς υπολογίζει την τιμή $g = (s_1' - s_1)(s_2 - s_2')^{-1} \bmod q$. Όμως ο g υποτίθεται ότι είναι γνωστός μόνο στην TTP. Η εύρεση του από την A αποδεικνύει ότι η υπογραφή (s_1', s_2') είναι πλαστή.

Σχήμα 7.11: Σχήμα υπογραφής Fail-Stop των van Heyst και Pedersen

Κεφάλαιο 8

Συναρτήσεις Σύνοψης

8.1 Εισαγωγή

Οι *Κρυπτογραφικές Συναρτήσεις Σύνοψης (ή Κατακερματισμού)* (σμβ. ΣΣ) παίζουν σημαντικό και θεμελιακό ρόλο στη σύγχρονη κρυπτογραφία. Όπως και οι ΣΣ που χρησιμοποιούνται σε άλλα πεδία εφαρμογών σε υπολογιστές, απεικονίζουν στοιχεία ενός συνόλου με πολλά στοιχεία σε κάποιο άλλο σύνολο με λιγότερα. Έτσι οι συναρτήσεις αυτές είναι της μορφής $\mathcal{H} : X \to Y \quad |X| > |Y|$, όπου δεν αποκλείεται $|X| = \infty$ ενώ το Y είναι κάποιο πεπερασμένο σύνολο. Είναι σαφές λοιπόν, ότι θα υπάρχουν κάποια στοιχεία που οι εικόνες τους μέσω της συνάρτησης θα ταυτίζονται. Την εικόνα ενός στοιχείου μέσω μιας ΣΣ θα ονομάζουμε και *αποτύπωμα*.

Στις περισσότερες περιπτώσεις στην κρυπτογραφία οι ΣΣ χρησιμοποιούνται για να παράγουν μια συντμημένη μορφή κάποιων δεδομένων. Η συντμημένη αυτή μορφή ή αποτύπωμα μπορεί να χρησιμοποιηθεί σε συστήματα ηλεκτρονικών υπογραφών ή για επαλήθευση δεδομένων. Σε αυτές τις περιπτώσεις τα δεδομένα αποστέλλονται μαζί με το αποτύπωμα επεξεργασμένο ανάλογα με την περίπτωση, έτσι ώστε ο παραλήπτης να μπορεί να ελέγξει την ταυτότητα του αποστολέα ή την πιστότητα των δεδομένων. Ο παραλήπτης παράγει εκ νέου το αποτύπωμα και το συγκρίνει με αυτό που έχει παραλάβει, αν είναι ίδια τότε ο έλεγχος επιτυγχάνει.

Η ύπαρξη στοιχείων που έχουν το ίδιο αποτύπωμα -αναπόφευκτη βέβαια- δημιουργεί κάποιες αμφιβολίες για το πόσο εμπιστοσύνη μπορεί να έχει ο παραλήπτης στον έλεγχο αυτόν. Η χρήση κάποιας 1-1 συνάρτησης με κατάλληλες ιδιότητες θα έλυνε το πρόβλημα, αλλά τότε το αποτύπωμα θα ήταν σε μήκος περίπου όσο και το μήνυμα. Αφού λοιπόν δεν είναι συμφέρον να χρησιμοποιηθεί κάποια 1-1 συνάρτηση, αυτό που ενδιαφέρει στην κατασκευή μιας ΣΣ είναι να μην είναι εύκολο σε κάποιον να βρίσκει στοιχεία με το ίδιο αποτύπωμα είτε δοκιμάζοντας

πολλά στοιχεία, είτε χρησιμοποιώντας κάποιο αλγόριθμο ο οποίος θα στηρίζεται σε κάποια κατασκευαστική ιδιαιτερότητα της εκάστοτε ΣΣ. Διότι αν κάποιος μπορεί να βρει κάποια τέτοια στοιχεία τότε θα μπορεί να εξαπατήσει κάποιο σύστημα που χρησιμοποιεί ΣΣ.

8.1.1 Ορισμοί

Ορισμός 8.1. *Συνάρτηση Σύνοψης (σμβ. ΣΣ)* θα ονομάζουμε μια συνάρτηση που έχει τις ακόλουθες ιδιότητες:

- Συμπίεση. Η τιμή $\mathcal{H}(x)$ έχει συγκεκριμένο μήκος για οποιαδήποτε είσοδο x.
- Ευκολία Υπολογισμού. Ο υπολογισμός της τιμής $\mathcal{H}(x)$ για κάποιο x να γίνεται "εύκολα". Δηλαδή υπάρχει ντετερμινιστικός αλγόριθμος Α πολυωνυμικού χρόνου, έτσι ώστε για κάθε x να ισχύει $\mathcal{H}(x) = A(\mathcal{H}, x)$.

Από αυτόν τον ορισμό είναι φανερό ότι μια ΣΣ είναι είναι μια συνάρτηση (συνήθως) όχι 1-1, που αντιστοιχεί πολλά διαφορετικά στοιχεία στην ίδια τιμή. Όταν υπάρχει ένα ζευγάρι τιμών x_1, x_2 οι οποίες για την ΣΣ $\mathcal{H}$, ισχύει $\mathcal{H}(x_1) = \mathcal{H}(x_2)$ τότε λέμε ότι έχουμε μια *σύγκρουση* για την $\mathcal{H}$.
Αν $\mathcal{H}$ είναι ΣΣ τότε η σχέση

$$x \sim x' \longleftrightarrow \mathcal{H}(x) = \mathcal{H}(x')$$

είναι σχέση ισοδυναμίας στο πεδίο ορισμού της $\mathcal{H}$. Την κλάση ισοδυναμίας του x θα συμβολίζουμε με $[x] = \mathcal{H}^{-1}(\mathcal{H}(x)) = \{x' \mid \mathcal{H}(x') = \mathcal{H}(x)\}$.

Ορισμός 8.2. Επιθυμητές ιδιότητες μιας ΣΣ $\mathcal{H}$ στην κρυπτογραφία

- Αντίσταση Πρώτου Ορίσματος. Είναι υπολογιστικά δύσκολο[1] για ένα δεδομένο στοιχείο y στο πεδίο τιμών της $\mathcal{H}$ να βρεθεί τιμή x έτσι ώστε να ισχύει $\mathcal{H}(x) = y$.
- Αντίσταση Δεύτερου Ορίσματος. Είναι υπολογιστικά δύσκολο για ένα δεδομένο στοιχείο x στο πεδίο ορισμού της $\mathcal{H}$ να βρεθεί άλλο στοιχείο x' έτσι ώστε $x \neq x'$ και $\mathcal{H}(x) = \mathcal{H}(x')$.

[1] Την 'υπολογιστική δυσκολία" την οποία δεν την ορίζουμε εδώ με σαφήνεια, συνήθως εκφράζουμε με την μη ύπαρξη αλγόριθμου πολυωνυμικού χρόνου είτε ντετερμινιστικού είτε πιθανοτικού. Συνήθως επιτρέπουμε ένα πολύ μικρό ποσοστό επιτυχίας στον αλγόριθμο (Δηλαδή για ένα πολύ μικρό μέρος των εισόδων ο αλγόριθμος δύναται να δίνει αποτέλεσμα σε πολυωνυμικό χρόνο).

- Δυσκολία Εύρεσης Συγκρούσεων. Είναι υπολογιστικά δύσκολο να βρεθούν δύο διαφορετικές τιμές του πεδίου ορισμού της $\mathcal{H}$, x, x' έτσι ώστε $\mathcal{H}(x) = \mathcal{H}(x')$.

- Ελευθερία Συσχετισμού. Είναι υπολογιστικά δύσκολο να βρεθούν δύο διαφορετικές τιμές του πεδίου ορισμού της $\mathcal{H}$, x, x' έτσι ώστε η απόσταση Hamming[2] $d(\mathcal{H}(x), \mathcal{H}(x'))$ να είναι είναι πιο μικρή από την αναμενόμενη απόσταση Hamming δύο εικόνων με τυχαία επιλογή των x, x'.

Ορισμός 8.3. *Συνάρτηση Σύνοψης Μιας Κατεύθυνσης (συμβ. ΣΣΜΚ)* ονομάζουμε μια ΣΣ $\mathcal{H}$ που έχει αντίσταση πρώτου και δεύτερου ορίσματος.

Πρέπει να παρατηρηθεί ότι μια ΣΣΜΚ διαφέρει από μια συνάρτηση μιας κατεύθυνσης, από την απαίτηση της αντίστασης δεύτερου ορίσματος. Βέβαια πολλές συναρτήσεις μιας κατεύθυνσης που χρησιμοποιούνται δεν έχουν περιορισμένο πεδίο τιμών και έτσι πολλές φορές είναι 1-1, οπότε και η απαίτηση της αντίστασης δεύτερου ορίσματος ικανοποιείται τετριμμένα.

Ορισμός 8.4. *Συνάρτηση Σύνοψης Χωρίς Συγκρούσεις (σμβ. ΣΣΧΣ)* ονομάζουμε μια ΣΣ $\mathcal{H}$ που έχει την ιδιότητα της δυσκολίας εύρεσης συγκρούσεων.

8.1.2 Παραδείγματα Συναρτήσεων Σύνοψης

1. Η συνάρτηση $f(x) = (x^2 - 1) \bmod p$ μπορεί να δίνει φαινομενικά τυχαίες τιμές στο διάστημα από 0 έως $p - 1$ αλλά δεν είναι μιας κατεύθυνσης αφού η εύρεση τετραγωνικών ριζών της μονάδας στο $\mathbb{Z}_p$ είναι κάτι που γίνεται σε πολυωνυμικό χρόνο.

2. Η συνάρτηση $g(x) = x^2 \bmod n$ όπου το $n = pq$ με p, q αρκετά μεγάλους πρώτους αριθμούς, δίνει και αυτή "τυχαίες" τιμές σε ένα περιορισμένο διάστημα από 0 έως $n - 1$. Η εύρεση μιας προ-εικόνας έχοντας σαν δεδομένο ένα y στο πεδίο τιμών της συνάρτησης, είναι ισοδύναμο με την εύρεση τετραγωνικής ρίζας στο δακτύλιο $\mathbb{Z}_n$ κάτι που αποδεικνύεται ότι είναι ισοδύναμο (υπολογιστικά) με την παραγοντοποίηση του n. Άρα η συνάρτηση έχει αντίσταση πρώτου ορίσματος (αν βέβαια η παραγοντοποίηση δε γίνεται σε πολυωνυμικό χρόνο). Όμως αφού ισχύει ότι $g(x) = g(-x)$ η συνάρτηση g δεν έχει αντίσταση δεύτερου ορίσματος ούτε και είναι ελεύθερη συγκρούσεων.

[2] Η απόσταση Hamming είναι ο αριθμός των bits που διαφέρουν στη δυαδική γραφή των x, y. Δηλ. $d(x, y) = |\{i : x_i \neq y_i\}|$

3. Η συνάρτηση $\mathcal{H} : \{0, \ldots, q-1\}^2 \to \mathbb{Z}_p - \{0\}$ με $\mathcal{H}(x_1, x_2) = \alpha^{x_1}\beta^{x_2} \bmod p$ όπου p πρώτος ώστε $q = \frac{p-1}{2}$ με q πρώτο επίσης, και α, β γεννήτορες του $\mathbb{Z}_p$. $\mathcal{H}$ συνάρτηση αυτή έχει αντίσταση πρώτου ορίσματος αφού ο υπολογισμός μιας προ-εικόνας θα απαιτούσε υπολογισμό διακριτού λογαρίθμου, κάτι που θεωρείται δύσκολο. Επίσης αποδεικνύεται ότι αν κάποιος έχει κατασκευάσει κάποιον αλγόριθμο που να βρίσκει συγκρούσεις για την $\mathcal{H}$, τότε θα μπορεί να τον χρησιμοποιήσει για να βρίσκει διακριτούς λογαρίθμους. Έτσι με την υπόθεση ότι το πρόβλημα διακριτού λογαρίθμου δε λύνεται σε πολυωνυμικό χρόνο η $\mathcal{H}$ είναι ΣΣΧΣ. Η συνάρτηση αυτή αναφέρεται και ως ΣΣ Chaum-Van Heijst-Pfitzman από τα ονόματα αυτών που την πρότειναν.

8.2 Βασικές Προτάσεις

Πρόταση 8.5. *Αν για μια ΣΣ $\mathcal{H} : X \to Y$ με πεπερασμένα X, Y με $|X| \geq k|Y|$, $k > 0$, υπάρχει αλγόριθμος Α που την αντιστρέφει με πιθανότητα επιτυχίας δ τότε υπάρχει αλγόριθμος Β που βρίσκει συγκρούσεις με πιθανότητα επιτυχίας $\delta(1 - \frac{1}{k})$.*

Απόδειξη. Εστω ο αλγόριθμος Β:

1. Επιλογή ενός τυχαίου $x \in X$.
2. Υπολογισμός του $y = \mathcal{H}(x)$
3. Εκτέλεση αλγορίθμου Α με είσοδο το y και έξοδο το x', σε περίπτωση αποτυχίας διακοπή
4. Αν $x \neq x'$ τότε επέστρεψε x, x' αλλιώς διακοπή

Είναι φανερό πως ο αλγόριθμος Β υπολογίζει μια σύγκρουση για τη συνάρτηση $\mathcal{H}$ χρησιμοποιώντας τον αλγόριθμο Α. Θα υπολογίσουμε την πιθανότητα επιτυχίας του. Στο βήμα 3 με πιθανότητα δ ο αριθμός των δυνατών εξόδων του Α είναι $|[x]|$ και ο αλγόριθμος Β θα βρει κάποια σύγκρουση εκτός αν ο Α δώσει σαν έξοδο το αρχικό x. Στις υπόλοιπες $|[x]| - 1$ περιπτώσεις ο Β επιτυγχάνει. Έτσι για ένα συγκεκριμένο $x \in X$ η πιθανότητα επιτυχίας θα είναι $\delta\frac{|[x]|-1}{|[x]|}$. Παίρνοντας το μέσο όρο σε όλα τα x έχουμε:

P(ο Β επιστρέφει μια σύγκρουση)=

$$= \frac{1}{|X|}\sum_{x \in X} \delta\frac{|[x]| - 1}{|[x]|} =$$

(Συμβολίζουμε με C το σύνολο των κλάσεων ισοδυναμίας $[x]$)

$$= \frac{1}{|X|}\delta\sum_{c \in C}\sum_{x \in c}\frac{|c| - 1}{|c|} = \frac{1}{|X|}\delta\sum_{c \in C}(|c| - 1) = \frac{1}{|X|}\delta(\sum_{c \in C}|c| - \sum_{c \in C} 1)$$

Ο αριθμός των κλάσεων ισοδυναμίας είναι $|Y|$ αφού σε κάθε $[x]$ αντιστοιχεί ένα συγκεκριμένο y και επίσης αφού οι κλάσεις $[x]$ είναι ξένα ανά δύο σύνολα και $\bigcup_x [x] = X$ έχουμε ότι $\sum_{c\in C} |c| = |X|$. Έτσι,
P(ο Β επιστρέφει μια σύγκρουση)=

$$= \frac{1}{|X|}\delta(|X| - |Y|) \geq \delta\frac{|X| - |X|/k}{|X|} = \delta(1 - \frac{1}{k})$$

□

Παρατήρηση Αν η πιθανότητα επιτυχίας του αλγόριθμου Α είναι $\delta := \delta(y)$ τότε η πρόταση δεν ισχύει πάντα. Π.χ. αν η g είναι μια ΣΣΧΣ τότε η

$$\mathcal{H}(x) = \begin{cases} 1||x, & |x| = n \\ 0||g(x), & |x| \neq n \end{cases}$$

είναι ΣΣΧΣ παρόλο που υπάρχει ένας προφανής αλγόριθμος αντιστροφής κάποιων εικόνων y (αυτών που έχουν πρώτο bit το 1). Σε αυτήν την περίπτωση έχουμε ότι $\delta(y) = 0$ για όλα τα y που η συνάρτηση αντιστρέφεται, έτσι ο αλγόριθμος Β της απόδειξης δεν έχει καμία πιθανότητα επιτυχίας. Βέβαια το παραπάνω παράδειγμα καθώς και άλλα παρόμοια είναι κάπως παθολογικά και έτσι (με κάποιες επιφυλάξεις) μπορούμε να διατυπώσουμε την παρακάτω πρόταση:

Πρόταση 8.6. *Μια ΣΣΧΣ $\mathcal{H}$ έχει αντίσταση πρώτου και δεύτερου ορίσματος.*

Απόδειξη.

1. Έστω η ΣΣΧΣ $\mathcal{H}$, και ένα στοιχείο x στο πεδίο ορισμού της. Αν ήταν υπολογιστικά εφικτό να βρεθεί ένα διαφορετικό στοιχείο x' ώστε $\mathcal{H}(x) = \mathcal{H}(x')$ τότε θα είχε βρεθεί μια σύγκρουση κάτι που αντιβαίνει στον ορισμό της συνάρτησης.

2. Υποθέτουμε ότι η $\mathcal{H}$ δεν έχει αντίσταση πρώτου ορίσματος. Από την υπόθεση μας έπεται ότι υπάρχει κάποιος αλγόριθμος Α ο οποίος για κάθε y δίνει ένα x έτσι ώστε $\mathcal{H}(x) = y$ με πιθανότητα δ. Κατά την προηγούμενη πρόταση υπάρχει αλγόριθμος που βρίσκει συγκρούσεις για την $\mathcal{H}$ άρα η $\mathcal{H}$ δεν είναι ελεύθερη συγκρούσεων.

□

8.2.1 Εύρεση Συγκρούσεων - Επίθεση Τετραγωνικής Ρίζας

Όπως έχει γίνει σαφές πάντα υπάρχουν συγκρούσεις για μια ΣΣ, το ερώτημα είναι κατά πόσο μπορούν να βρεθούν. Αν $\mathcal{H} : X \to Y$ είναι μια ΣΣ με $|X| = m$ και $|Y| = n$ τότε υπάρχουν τουλάχιστον $m - n$ συγκρούσεις. Όσο περίπλοκα

κι αν είναι ορισμένη μια ΣΣ δεν παύει να είναι μια απεικόνιση σε ένα πεπερασμένο σύνολο τιμών και εξαιτίας αυτού, κάποιος θα μπορούσε να προσπαθήσει να δοκιμάσει τυχαία κάποιες τιμές, ελπίζοντας να βρει κάποια σύγκρουση, χρησιμοποιώντας τον αλγόριθμο:

Επίθεση Τετραγωνικής Ρίζας

1. Τυχαία επιλογή $x_1, x_2, \ldots, x_k$
2. $y_i = \mathcal{H}(x_i) \quad i = 1, \ldots, k$
3. Ταξινόμηση των y_i
4. Έλεγχος αν υπάρχει $i \in 1, \ldots, k-1$ με $y_i = y_{i+1}$ αν ναι Επιστροφή των αντίστοιχων x_i αλλιώς Αποτυχία

Αυτός ο απλοϊκός αλγόριθμος μπορεί να χρησιμοποιηθεί εναντίον οποιασδήποτε ΣΣ και οι πιθανότητες που έχει να επιτύχει θα καθορίσουν τα κάτω όρια στα πεδία τιμών των ΣΣ.

Πρόταση 8.7. *Χρησιμοποιώντας την αλγόριθμο της Επίθεσης Τετραγωνικής Ρίζας* [3] *η πιθανότητα να βρεθεί μια σύγκρουση μετά από έλεγχο περίπου* $\sqrt{n}$ *εικόνων είναι* 0.5.

Απόδειξη. Θα υπολογίσουμε την πιθανότητα όλα τα y_i που προκύπτουν στο βήμα 2 να είναι διαφορετικά. Όλες οι δυνατές k-άδες χωρίς συγκρούσεις (επαναλήψεις) είναι $n(n-1)\ldots(n-k+1)$. Οι δυνατές k-άδες με συγκρούσεις είναι σε πλήθος n^k. Έτσι η πιθανότητα μια k-άδα να μην έχει συγκρούσεις θα είναι

$$\frac{n(n-1)\ldots(n-k+1)}{n^k} = \prod_{i=1}^{k-1}(1-\frac{i}{n})$$

Υποθέτοντας ότι $k << n$ και από το ότι $1-x \simeq e^{-x}$ για μικρά x έχουμε ότι:

$$\prod_{i=1}^{k-1}(1-\frac{i}{n}) \simeq \prod_{i=1}^{k-1} e^{-\frac{i}{n}} = e^{-\frac{\sum_{i=1}^{k-1} i}{n}} = e^{-\frac{k(k-1)}{2n}}$$

Έτσι η πιθανότητα να βρεθεί μια σύγκρουση είναι $p \simeq 1 - e^{-\frac{k(k-1)}{2n}}$, λύνοντας ως προς k έχουμε:

$$1-p \simeq e^{\frac{k(k-1)}{2n}} \Rightarrow -\frac{k(k-1)}{2n} \simeq \ln(1-p) \Rightarrow k^2 - k2n\ln\frac{1}{1-p}$$

[3] Square Root Attack, αναφέρεται και ως Επίθεση των Γενεθλίων, Birthday Attack.

Τελικά $k \simeq 1 + \sqrt{2n \ln \frac{1}{1-p}}$. Για $p = \frac{1}{2}$ έχουμε $k \cong 1 + 1.17\sqrt{n}$. Δηλαδή θέτοντας στον αλγόριθμο Τετραγωνικής Ρίζας το k να είναι λίγο παραπάνω από $\sqrt{n}$ η πιθανότητα να βρεθεί μια σύγκρουση είναι 0.5. □

Μέχρι στιγμής εξετάσαμε μερικές γενικές ιδιότητες των ΣΣ και προϋποθέσεις για τη χρήση τους στην κρυπτογραφία. Το παράδειγμα της ΣΣ Chaum-Van Heijst-Pfitzman καθώς και άλλα ανάλογα έχουν καλό θεωρητικό υπόβαθρο, καθώς στηρίζονται σε γνωστά δύσκολα προβλήματα αλλά συνήθως υστερούν σε ταχύτητα.

8.3 Η μέθοδος Merkle για επέκταση Συναρτήσεων Σύνοψης

Στα παραδείγματα που είδαμε μέχρι στιγμής το πεδίο ορισμού των ΣΣ ήταν πεπερασμένο. Αυτό βέβαια είναι ένας σοβαρός περιορισμός όταν έχουμε πολλά δεδομένα.

Έστω ότι έχουμε μια συνάρτηση $f : X \to Y$ με $|X| = n + k$ και $|Y| = n$. Θα κατασκευάσουμε μια νέα συνάρτηση $\mathcal{H} : \Sigma^* \to Y$ (δηλαδή οποιουδήποτε μήκους δεδομένα είναι στο πεδίο ορισμού της συνάρτησης) ως εξής:

Μέθοδος Merkle

1. Εστω x δεδομένα των οποίων θέλουμε να υπολογίσουμε την εικόνα μέσω της $\mathcal{H}$ (θα αναφέρονται και σαν είσοδος) με μήκος $|x| = b$. Χωρίζουμε την είσοδο x σε t μέρη $x_1, x_2, \ldots, x_t$ έτσι ώστε $|x_i| = k$ παραθέτοντας στο τελευταίο μέρος όσα "0" bits χρειάζονται.

2. Προσθέτουμε ένα ακόμα μέρος x_{t+1} το οποίο θα κρατά το αρχικό μήκος b (είτε υποθέτοντας ότι $b < 2^k$ είτε τοποθετώντας στο x_{t+1} τα k λιγότερο σημαντικά bits του b).

   Ένας εναλλακτικός τρόπος να γίνουν τα βήματα 1 και 2 είναι να παραθέσουμε στο x μια σειρά bit της μορφής 10^* ώστε το $|x|$ να γίνει πολλαπλάσιο του k. Αυτός ο τρόπος υστερεί στο ότι σε κάποιες περιπτώσεις είναι δυνατόν να προσθέσουμε ένα ακόμη "αχρείαστο" μέρος στο x, αλλά γενικά προτιμάται αφού είναι ευκολότερο να αποφασισθεί πιο είναι το αρχικό μήνυμα.

3. Υπολογίζουμε την αναδρομική συνάρτηση

$$H_0 = IV, |IV| = n$$

$$H_i = f(H_{i-1}||x_i) \quad i = 1, \ldots, t+1$$

Όπου IV είναι κάποια αρχική καθορισμένη τιμή που είναι πάντα η ίδια για όλους τους υπολογισμούς της $\mathcal{H}$.

4. Το αποτύπωμα του x μέσω της $\mathcal{H}$ θα είναι η τιμή $\mathcal{H}(x) := H_{t+1}$.

Πρόταση 8.8. *Αν f είναι μια ΣΣΧΣ τότε και η $\mathcal{H}$ όπως προκύπτει από τη μέθοδο του Merkle είναι ΣΣΧΣ.*

8.4 Εφαρμογές στην κρυπτογραφία

8.4.1 Σχήματα Δέσμευσης

Μία πολύ σημαντική εφαρμογή των συναρτήσεων σύνοψης, η οποία περικλείει κάποιες από τις παραπάνω είναι η χρήση τους για την *δημιουργία σχημάτων δέσμευσης (commitment schemes)*. Με αυτά θα ασχοληθούμε αναλυτικά στην ενότητα 9.2, καθώς υπάρχουν πολλοί διαφορετικοί τρόποι υλοποίησης.

Περιληπτικά εδώ αξίζει να αναφέρουμε ότι οι οντότητες στα σχήματα δέσμευσης έχουν δύο ρόλους. Η πρώτη (Alice) έχει υπολογίσει μία τιμή (κάποιο σημαντικό αποτέλεσμα) και θέλει να αποδείξει στη δεύτερη (Bob) αυτό το γεγονός, χωρίς βέβαια να αποκαλύψει το αποτέλεσμα, αλλά και χωρίς να υπάρχει δυνατότητά εξαπάτησης.

Ένας τρόπος να υλοποιηθεί το παραπάνω πρωτόκολλο είναι μέσω συναρτήσεων σύνοψης. Για τον σκοπό αυτό η Alice δημιουργεί μία σύνοψη του αποτελέσματος την οποία και δημοσιοποιεί. Αν η συνάρτηση σύνοψης έχει την ιδιότητα της Αντίστασης Πρώτου Ορίσματος, που προαναφέραμε ο Bob δεν μπορεί να μάθει το αποτέλεσμα.

Όταν αργότερα υπολογιστεί η συγκεκριμένη τιμή και από άλλους ή μπορεί να δημοσιοποιηθεί, τότε το αποτέλεσμα αυτό δίνεται εκ νέου στην συνάρτηση σύνοψης. Αν τα δύο αποτυπώματα ταιριάζουν τότε η Alice υπολόγισε σωστά το αποτέλεσμα πρώτη, κάτι το οποίο ενισχύεται και από την ιδιότητα της Αντίστασης Δευτέρου Ορίσματος.

Κάτι που πρέπει να προσεχθεί κατά την κατασκευή σχημάτων δέσμευσης από συναρτήσεις σύνοψης είναι το γεγονός ότι σε περίπτωση που το σύνολο τιμών όπου ανήκει το αποτέλεσμα της Alice είναι πεπερασμένο, ο Bob μπορεί να προσπαθήσει εξαντλητικά να υπολογίζει τα αποτυπώματα όλων των μελών του μέσω της συνάρτησης σύνοψης και να τα συγκρίνει με αυτό που του έδωσε η Alice, παρακάμπτοντας την Αντίσταση Πρώτου Ορίσματος της συνάρτησης σύνοψης. Για τον σκοπό αυτό σε τέτοιες περιπτώσεις πρέπει να γίνεται χρήση τυχαιότητας (nonce) από την Alice, η οποία θα πρέπει να αποκαλύπτεται τελικά μαζί με το αποτέλεσμα.

Από τα σχήματα δέσμευσης προκύπτουν οι παρακάτω επιπλέον εφαρμογές σε πιο πρακτικό επίπεδο βέβαια:

- Σε συνδυασμό με κάποιο άλλο κρυπτογραφικό αλγόριθμο που κρυπτογραφεί μηνύματά, δημιουργείται ένα σύστημα ηλεκτρονικών υπογραφών. Παραδείγματα αυτής της μορφής είναι η MD5 που χρησιμοποιείται με το RSA στο πακέτο PGP, η SHA-1 που χρησιμοποιείται στο DSS (Digital Signature Standard).

- Μια παρόμοια χρήση τους, για έλεγχο πιστότητας δεδομένων, γίνεται διαμορφώνοντας κατάλληλες ΣΣ ώστε να δέχονται σαν παράμετρο εκτός από την είσοδο (το μήνυμα) και ένα κλειδί.

- Μια συνηθισμένη χρήση είναι για έλεγχο λαθών σε δεδομένα που μεταβιβάζονται μέσα σε δίκτυα.

8.4.2 Padded RSA και OAEP

Στις ενότητες 6.4.2 και 6.4.3 είδαμε τα προβλήματα ασφάλειας και κάποιες επιθέσεις που μπορούν να πραγματοποιηθούν στο παραδοσιακό RSA. Για την αντιμετώπιση τους προτάθηκαν διάφορες παραλλαγές.

Η πρώτη ονομάστηκε *padded RSA* και όπως προκύπτει από το όνομα, προσθέτει κάποια ψηφία τυχαιοποίησης στο μήνυμα, με αποτέλεσμα σε κάθε μήνυμα να αντιστοιχούν πολλές κρυπτογραφήσεις. Συγκεκριμένα, υποθέτουμε ότι το μήνυμα m έχει μήκος l bits. αν (e, n) είναι το δημόσιο κλειδί και (d, p, q) το ιδιωτικό έχουμε:

- Πριν την κρυπτογράφηση δημιουργείται το μήνυμα: $\bar{m} = r||m$, όπου r είναι μια τυχαία συμβολοσειρά από $|n| - l - 1$ bits.

- Η κρυπτογράφηση γίνεται κανονικά ως: $\bar{c} = \bar{m}^e \bmod n$

- Η αποκρυπτογράφηση γίνεται κανονικά και δίνει $\bar{c}^d = \bar{m} \pmod{n}$

- Από το $\bar{m}$ κρατάμε μόνο τα l bits χαμηλότερης τάξης.

Με τον τρόπο αυτό και με κάποιες υποθέσεις σχετικά με το l και το $|n|$ το σχήμα μπορεί να αποδειχθεί ότι διαθέτει ασφάλεια CPA. Η τεχνική αυτή χρησιμοποιείται στο πρότυπο κρυπτογράφησης του Διαδικτύου PKCS1 vl.5. Παρά την ευρεία χρήση αυτής της τεχνικής έχουν υπάρξει αρκετές πετυχημένες επιθέσεις.

Ένα πιο αποτελεσματικό σχήμα κρυπτογράφησης είναι το Optimal Asymmetric Encryption Padding (OAEP) το οποίο προτάθηκε στο [3] και χρησιμοποιεί συναρτήσεις σύνοψης. Συγκεκριμένα, υποθέτουμε και πάλι ότι το μήνυμα m έχει μήκος l bits. Αν (e, n) είναι το δημόσιο κλειδί και (d, p, q) το ιδιωτικό έχουμε:

- Επιλέγουμε τα ψηφία τυχαιοποίησης $r \in \{0,1\}^{2l}$
- Στη συνέχεια υπολογίζουμε το $m' = m||0^l$
- Χρησιμοποιώντας μια συνάρτηση σύνοψης $\mathcal{G}$ παράγουμε το μήνυμα $m_1 = \mathcal{G} \oplus m'$
- Τέλος υπολογίζουμε το $\bar{m} = m_1||(r \oplus \mathcal{H}(m_1))$
- Η κρυπτογράφηση γίνεται ως: $\bar{c} = \bar{m}^e \bmod n$ και η αποκρυπτογράφηση ως $\bar{c}^d \bmod n = \bar{m}$
- Για την ανάκτηση του αρχικού μηνύματος θεωρούμε ότι $\bar{m} = m_a||m_b$ μεγέθους $2l$ το κάθε ένα.
- Υπολογίζουμε το $r = \mathcal{H}(m_a) \oplus m_b$
- Υπολογίζουμε το $m' = m_a \oplus \mathcal{G}(r)$
- Αν τα l bits χαμηλότερης τάξης του m' είναι 0, τότε το μήνυμα είναι τα l bits υψηλότερης τάξης. Σε διαφορετική περίπτωση η αποκρυπτογράφηση έχει αποτύχει.

Η συγκεκριμένη παραλλαγή έχει αποδειχθεί ασφαλής εναντίον επιθέσεων CCA χρησιμοποιώντας τη μεθοδολογία του τυχαίου μαντείου 8.6.

Τέλος, έχουν υπάρξει και αντίστοιχες προτάσεις για την προσθήκη ασφάλειας κατά CCA στο κρυπτοσύστημα ElGamal, τις οποίες θα εξετάσουμε στην ενότητα 10.6.2.

8.4.3 Ψηφιακές Υπογραφές

Στην ενότητα 7.3 είδαμε πόσο εύκολη είναι η πλαστογράφηση μιας ψηφιακής υπογραφής RSA. Ένας τρόπος αντιμετώπισης αντιμετώπισης της επίθεσης αυτής είναι η χρήση συναρτήσεων σύνοψης στη δημιουργία των υπογραφών.

Στο συγκεκριμένο σχήμα υποθέτουμε ότι το δημόσιο κλειδί περιλαμβάνει και μία συνάρτηση σύνοψης $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_n^*$. Το σύστημα πλέον έχει ως εξής:

- **Δημιουργία υπογραφής**

  - Πρώτα υπολογίζεται η σύνοψη του μηνύματος $\mathcal{H}(m)$
  - Η υπογραφή εφαρμόζεται σε αυτή: $s = \mathcal{H}(m)^d \bmod n$

- **Επαλήθευση υπογραφής**
  - Ο παραλήπτης υπολογίζει με τη σειρά του τη σύνοψη.
  - Για την επαλήθευση ελέγχει αν $s^e =_? \mathcal{H}(m)^d \bmod n$

Το παραπάνω σχήμα ονομάζεται *RSA-FDH (Full Domain Hash)* ή πιο απλά RSA με συναρτήσεις σύνοψης. Μπορούμε να θεωρήσουμε στο παραπάνω σχήμα πως ο χρήστης δεσμεύεται πως γνωρίζει το μήνυμα που υπογράφει μέσω της συνάρτησης σύνοψης. Για να είναι ασφαλές το RSA-FDH πρέπει να έχει κάποιες από τις ιδιότητες στις οποίες αναφερθήκαμε νωρίτερα:

- Η $\mathcal{H}$ πρέπει να διαθέτει αντίσταση πρώτου ορίσματος. Αν ισχύει αυτό δεν μπορούν πραγματοποιηθούν οι επιθέσεις χωρίς μήνυμα καθώς και η επίθεση επιλεγμένου μηνύματος λόγω του εύπλαστου του RSA της ενότητας 7.3.
- Η συνάρτηση σύνοψης να διαθέτει δυσκολία εύρεσης συγκρούσεων. Πράγματι, αν κάποιος μπορεί να βρει μηνύματα m_1, m_2 ώστε $\mathcal{H}m_1 = \mathcal{H}m_2$ μπορεί να πλαστογραφήσει μία υπογραφή σε κάποιο από τα δύο.

Στην ενότητα 8.6.1 θα αποδείξουμε την ασφάλεια του συγκεκριμένου συστήματος σε ένα μοντέλο με κάποιες επιπλέον υποθέσεις όμως.

8.4.4 Χρονοσήμανση

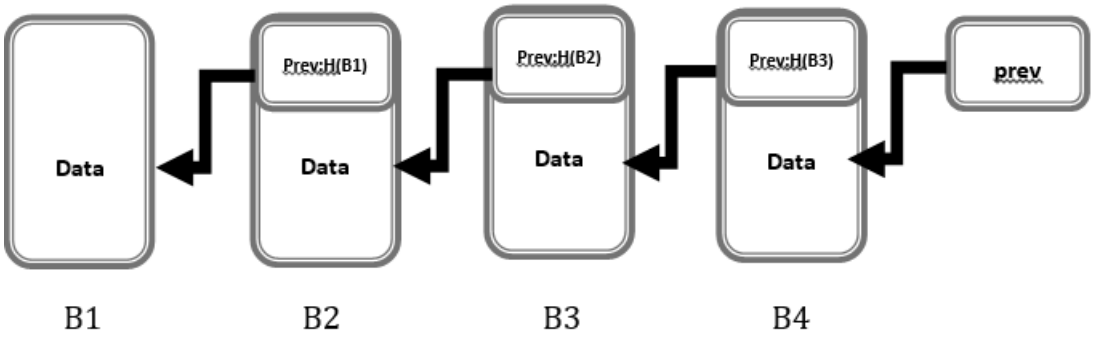
Μία άλλη χρήσιμη εφαρμογή των συναρτήσεων σύνοψης αφορά την *χρονοσήμανση δεδομένων (timestamping)*, η οποία προτάθηκε από τους Haber, Stornetta και Bayer στα [5] και [1]. Στον ψηφιακό κόσμο είναι πάρα πολύ δύσκολο να υπάρχει αμοιβαία συμφωνία για το πότε δημιουργήθηκε ή τροποποιήθηκε. Μία λύση σε αυτό θα μπορούσε να δοθεί με την μορφή μίας *έμπιστης τρίτης οντότητας (trusted third party - TTP)* η οποία θα διατηρεί μία έννοια του χρόνου την οποία θα προσαρτά σε όλα τα δεδομένα που της στέλνονται. Η λύση αυτή έχει όλα τα προβλήματα που έχουν οι έμπιστες αρχές. Για παράδειγμα, η TTP μπορεί να συνεργαστεί με κάποιον κακόβουλο χρήστη και να παρακάμψει την σωστή χρονική σειρά. Επιπλέον ενδεχομένως μπορεί να παραβιάσει την εμπιστευτικότητα των δεδομένων, κάτι βέβαια που μπορεί να αποφευχθεί με την χρήση ενός σχήματος δέσμευσης.

Συγκεκριμένα τώρα ότι ο A θέλει να φτιάξει μια πειστική χρονοσφραγίδα σε ένα μήνυμα x, και δεχόμαστε ότι οι $f, \mathcal{H}$ είναι δημοσίως γνωστές συναρτήσεις σύνοψης.

Πρώτα, ο A πρέπει να αποκτήσει κάποια "τρέχουσα", δημοσίως διαθέσιμη πληροφορία η οποία δεν θα μπορούσε να είχε προβλεφθεί πριν να συμβεί.
Συμβολίζουμε την πληροφορία αυτή ως C. Κατόπιν:

- Υπολογίζουμε τα $z = \mathcal{H}(x)$ και $z' = f(z||C)$ [4]
- Υπολογίζουμε το $y = sig_K(z')$.
- Τέλος, ο A πρέπει να δημοσιεύσει το (z, C, y). (στην εφημερίδα της επόμενης ημέρας για παράδειγμα)

Αφού η υπογραφή του A περιέχει την πληροφορία του C, δεν θα μπορούσε να έχει δημιουργηθεί πριν από τη στιγμή όπου αποκτήθηκε το C. Επίσης, με τη δημοσίευση των y, ξέρουμε ότι η υπογραφή δεν μπορούσε να έχει δημιουργηθεί μετά από το χρόνο αυτό.
Η δημόσια πληροφορία C μπορεί να είναι η σειρά με την οποία καταφθάνουν οι αιτήσεις χρονοσήμανσης. Δηλαδή, αν μία χρονοσφραγίδα μπορεί να αναφερθεί με κάποιο τρόπο στις προηγούμενες αιτήσεις, τότε είναι αυταπόδεικτο ότι είναι μεταγενέστερη. Επιπλέον δεν μπορεί να εκδοθεί μεταγενέστερη χρονοσφραγίδα, αφού δεν μπορούν να προβλεφτούν και έτσι δεν θα είναι διαθέσιμες οι ενδιάμεσες αιτήσεις.
Η δημιουργία μιας τέτοιας *αλυσίδας* είναι δυνατή με την έννοια της *σύνοψης - δείκτη (hash - pointer)*. Ένας τέτοιος δείκτης περιέχει δύο πράγματα: την τοποθεσία στην οποία βρίσκονται κάποια δεδομένα και την σύνοψη των δεδομένων αυτών. Με αυτούς του δείκτες μπορούμε να δημιουργήσουμε αλυσίδες δεδομένων όπως φαίνονται στο παρακάτω σχήμα:



Σχήμα 8.1: Αλυσίδα συναλλαγών

[4] με το σύμβολο || συμβολίζουμε την παράθεση (concatenation)

Κάθε κόμβος της αλυσίδας αποτελείται από δύο τμήματα. Πρώτα από όλα από τα ίδια τα δεδομένα και από μία σύνοψη δείκτη στο προηγούμενο block. Με τον τρόπο αυτό σε περίπτωση που υπάρχει οποιαδήποτε τροποποίηση σε κάποιο block, αμέσως ο δείκτης που υπάρχει στο επόμενο block γίνεται άκυρος καθώς η σύνοψη περιέχει το προηγούμενα δεδομένα. Έτσι οι αλυσίδες αυτές μας διαβεβαιώνουν για την σειρά με την οποία δημιουργήθηκαν τα μπλοκ καθώς και ότι δεν έχουν αλλαχθεί από τότε που μπήκαν στην αλυσίδα. Επιπλέον αντί να ανταλλάσσεται ολόκληρη η αλυσίδα, αρκεί να μεταβιβάζεται ο δείκτης στο πιο πρόσφατο block.

Ένα μειονέκτημα των χρονοσφραγίδων, όπως τις περιγράψαμε παραπάνω είναι ότι ενώ καταργούν την έμπιστη αρχή για τη διατήρηση του χρόνου την χρησιμοποιούν για την ύπαρξη μιας έμπιστης αρχής για την παροχή της δημόσιας πληροφορίας C.

8.5 Δέντρα Πιστοποίησης Γνησιότητας Merkle

Τα δέντρα πιστοποίησης γνησιότητας Merkle είναι μία σημαντική τεχνική, με πολλές εφαρμογές (βλ. και Bitcoin, 11.4).

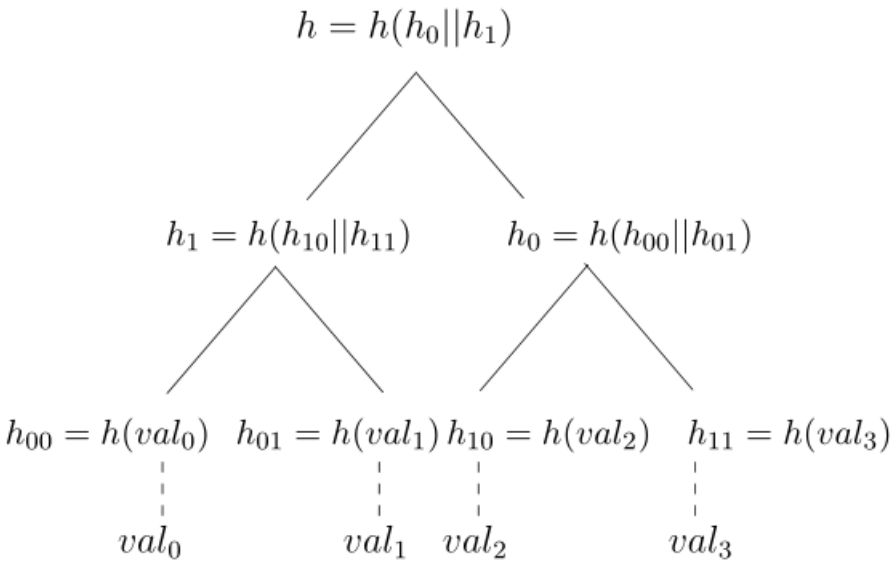
Η ιδέα είναι πολλές τιμές να επικυρώνονται ταυτόχρονα, ως εξής: Υποθέτουμε ότι διαθέτουμε μια ασφαλή *ΣΣΧΣ*, έστω $\mathcal{H}$. Τοποθετούμε τα αποτυπώματα όλων των αρχικών τιμών στα φύλλα ενός δυαδικού δένδρου με κατάλληλο ύψος. Σε κάθε εσωτερικό κόμβο μπαίνει το αποτύπωμα της συνένωσης των κόμβων-παιδιών, και αυτό γίνεται αναδρομικά έως ότου φτάσουμε στη ρίζα, η οποία ουσιαστικά επικυρώνει *ταυτόχρονα* όλες τις δοσμένες τιμές.

Όποτε ο χρήστης a_i θέλει να αποδείξει ότι η τιμή του val_i είναι έγκυρη, δεν έχει παρά να δώσει το αποτύπωμα (hash) των αδελφών όλων των κόμβων που βρίσκονται στο μονοπάτι του δέντρου από τη ρίζα στο φύλλο που περιέχει το $\mathcal{H}(val_i)$. Με τις τιμές αυτές, μπορεί οποιοσδήποτε να ανακατασκευάσει τη ρίζα και να επαληθεύσει έτσι την τιμή που δηλώνεται.

Έτσι, για την επικύρωση n τιμών, αρκούν $2 \log n$ αποτυπώματα (hash values). Ένα παράδειγμα δέντρου Merkle φαίνεται στο Σχ. 8.2: για να πιστοποιήσει κανείς την τιμή, π.χ., val_3 αρκεί να δώσει τα h_{10} και h_0 - από αυτά, μαζί με το $h_{11} = \mathcal{H}(val_3)$ μπορεί να υπολογίσει κανείς μια τιμή που θα πρέπει να είναι ίση με $\mathcal{H}$.

8.6 Το μοντέλο του τυχαίου μαντείου

Οι συναρτήσεις σύνοψης, όπως ίσως έγινε κατανοητό, παίζουν ένα σημαντικό ρόλο στη σύγχρονη κρυπτογραφία. Ο ρόλος αυτός όμως έχει διαφορετικές δια-



Σχήμα 8.2: Παράδειγμα Merkle authentication tree, για πιστοποίηση 4 τιμών.

στάσεις στην θεωρία και στην πράξη. Συγκεκριμένα, *στη θεωρία* θα θέλαμε οι συναρτήσεις σύνοψεις να συμπεριφέρονται ως *τυχαίες* συναρτήσεις. Μία τυχαία συνάρτηση η οποία αντιστοιχεί εισόδους μεγέθους n σε εξόδους μεγέθους $\lambda(n)$ μπορεί να κατασκευαστεί μέσω ενός 'πίνακα τιμών', δηλαδή ενός πίνακα ο οποίος αντιστοιχίζει κάθε μία από τις 2^n πιθανές εισόδους σε μια από τις $2^{\lambda(n)}$ πιθανές εξόδους. [5]. Άρα θα έχει τουλάχιστον 2^n γραμμές, καθώς η είσοδος είναι σαφώς μεγαλύτερη λόγω και της ύπαρξης συγκρούσεων. Αυτό σημαίνει, ότι για να αποτιμηθεί ένας τέτοιος πίνακας, χρειάζεται εκθετικό χρόνο (πόσο μάλλον για να αποθηκευθεί). Αυτό το γεγονός καθιστά αδύνατη την πρακτική κατασκευή και χρήση τέτοιων συναρτήσεων. Αυτό που γίνεται, όπως είδαμε, είναι να χρησιμοποιούνται άλλες τεχνικές, οι οποίες ουσιαστικά 'συμπιέζουν' τον πίνακα τιμών της συνάρτησης. Η συμπίεση αυτή, σύμφωνα με τη Θεωρία Πληροφορίας περιορίζει την έννοια της τυχαιότητας. Συνοπτικά λοιπόν, η διάσταση στην οποία αναφερθήκαμε, είναι ότι θεωρητικά χρειαζόμαστε τις συναρτήσεις σύνοψης να συμπεριφέρονται ως τυχαίες συναρτήσεις, αλλά πρακτικά δεν μπορούμε να τις κατασκευάσουμε.

Η διάσταση αυτή έχει μεγάλη σημασία στις αποδείξεις ασφάλειας ενός κρυπτογραφικού πρωτοκόλλου που χρησιμοποιεί τέτοιες συναρτήσεις. Αν υποθέταμε, για

[5] Υπάρχουν $2^{\lambda(n)2^n}$ τέτοιες συναρτήσεις.

χάρη της απόδειξης, ότι οι συναρτήσεις σύνοψης είναι πραγματικά τυχαίες, τότε θα έπρεπε ο αντίπαλος που τις χρησιμοποιεί να μπορέσει να εκτελεστεί για εκθετικό χρόνο, μόνο και μόνο για να τις αποτιμήσει. Σε αυτή την περίπτωση όλες οι υποθέσεις ασφαλείας που κάναμε δεν έχουν κανένα νόημα, καθώς μπορεί ο αντίπαλος μπορεί να δοκιμάσει όλες τις πιθανές τιμές των κλειδιών ή των κρυπτοκειμένων. Επιπλέον από την άλλη αν δεν δεχτούμε την τυχαιότητα, θα πρέπει να ασχοληθούμε και με τις λεπτομέρειες κατασκευής της τυχαίας συνάρτησης, ώστε να δούμε πώς η συμπίεση που απορρέει από αυτή, επηρεάζει το πρωτόκολλο, κάτι που θα απομακρύνει την απόδειξη από τον στόχο της.

Για τους παραπάνω λόγους, στις κρυπτογραφικές αποδείξεις οι συναρτήσεις σύνοψης χρησιμοποιούνται ως *μαντεία* που επιστρέφουν τυχαίες απαντήσεις. Το μαντείο είναι ένα 'μαύρο κουτί' το οποίο δέχεται ερωτήσεις και παράγει απαντήσεις για τη συγκεκριμένη είσοδο. Η χρήση τους είναι συχνή στην Θεωρία Υπολογιστικής Πολυπλοκότητας. Σε μια κρυπτογραφική απόδειξη χρησιμοποιούνται από οποιαδήποτε οντότητα θέλει να αποτιμήσει μία συνάρτηση σύνοψης, στέλνοντας την είσοδο μέσω ενός ασφαλούς καναλιού στο μαντείο και λαμβάνοντας ως απάντηση την έξοδο της συνάρτησης. Η αλληλεπίδραση με το μαντείο (είσοδος-έξοδος) θεωρούμε ότι γίνεται μέσω ασφαλούς καναλιού, καθώς το ανάλογο που μοντελοποιεί, αφορά την αποτίμηση της συνάρτησης από τον ίδιο τον χρήστη. Για τον ίδιο λόγο στην απόδειξη δεν θεωρείται καν γνωστό ότι κάποιος *εξωτερικός* χρήστης ρώτησε το μαντείο. Φυσικά η τυχαία συνάρτηση πρέπει να διατηρεί όλες τις ιδιότητες που περιμένουμε. Για παράδειγμα, αν μιλάμε για συναρτήσεις σύνοψης, κάθε φορά που στέλνεται στο μαντείο η ίδια είσοδος, πρέπει να εξάγει την ίδια απάντηση. Πρέπει να υπάρχουν συγκρούσεις, λόγω της αρχής του Περιστερώνα, αλλά να είναι δύσκολο να βρεθούν. Εσωτερικά το μαντείο θεωρούμε ότι διατηρεί τον πίνακα που προαναφέραμε (χωρίς να μπορεί να τον μάθει οποιοσδήποτε αντίπαλος), ο οποίο είναι αρχικά άδειος. Κάθε φορά που λαμβάνει ένα ερώτημα για τον υπολογισμό της συνάρτησης για μια είσοδο, ελέγχει τον πίνακα. Αν βρει την είσοδο τότε επιστρέφει την ήδη υπολογισμένη έξοδο. Αλλιώς παράγει μία τυχαία τιμή την καταχωρεί στον πίνακα για μελλοντικές ερωτήσεις και την επιστρέφει στην οντότητα που τη ζήτησε. Ο παραπάνω τρόπος 'οκνηρής αποτίμησης (lazy evaluation)' επιτρέπει την χρήση της μεθοδολογίας αυτής και για συναρτήσεις με άπειρο πεδίο ορισμού.

Μία απόδειξη ασφάλειας που χρησιμοποιεί τις τυχαίες συναρτήσεις με τον τρόπο που περιγράψαμε παραπάνω, παρέχει ασφάλεια στο *Μοντέλο του Τυχαίου Μαντείου - Random Oracle Mode* σε αντίθεση με τις αποδείξεις ασφάλειας που λειτουργούν στο κανονικό μοντέλο - *Standard Model*. Οι πρώτες αποδείξεις που χρησιμοποιούν το τυχαίο μαντείο με το τρόπο που παρουσιάσαμε παραπάνω, δόθηκαν από τους Bellare και Rogaway στο [2]. Από τότε δεκάδες συστήματα έχουν αποδειχθεί ασφαλή στο συγκεκριμένο μοντέλο. Παρά το γεγονός αυτό, έχουν υπάρξει

και αρκετές επικρίσεις για το συγκεκριμένο μοντέλο με την έννοια του οτι η συνάρτηση η οποία θα 'υλοποιήσει' το τυχαίο μαντείο μπορεί να έχει διαφορετική συμπεριφορά από αυτή της απόδειξης. Επιπλέον ο Canetti στο [4] κατασκεύασε τεχνικά πρωτόκολλα τα οποία δεν είναι ασφαλή με οποιονδήποτε τρόπο επιλεγεί το τυχαίο μαντείο, παρά της ύπαρξης της σχετικής απόδειξης ασφάλειας. Παρ' όλα αυτά η ουσία της μεθοδολογίας του τυχαίου μαντείου είναι ότι δείχνει ότι κάποιο πρωτόκολλο δεν έχει *εγγενή προβλήματα ασφαλείας* και όχι ότι οποιαδήποτε πραγματική υλοποίηση (με συγκεκριμένη συνάρτηση στη θέση του) θα έχει τις ιδιότητες που απορρέουν από την απόδειξη.

8.6.1 Ασφάλεια του RSA-FDH

Για να κατανοήσουμε καλύτερα την λειτουργία του μοντέλου του τυχαίου μαντείου, θα αποδείξουμε την ασφάλεια των υπογραφών RSA με συναρτήσεις σύνοψης που είδαμε νωρίτερα.

Συγκεκριμένα, ακολουθώντας τη μεθοδολογία της 1.4.4, θα αξιοποιήσουμε έναν αντίπαλο $\mathcal{A}$ ο οποίος χρησιμοποιώντας ένα τυχαίο μαντείο μπορεί να πλαστογραφήσει μία υπογραφή RSA-FDH, για να κατασκευάσουμε έναν αντίπαλο $\mathcal{B}$ ο οποίος μπορεί να λύσει το πρόβλημα RSA (6.4).

Θα ασχοληθούμε με τις εξής περιπτώσεις:

Επίθεση χωρίς μήνυμα Η είσοδος του $\mathcal{B}$ είναι το δημόσιο κλειδί του RSA (e, n) και το στοιχείο $y \in \mathbb{Z}_n^*$ για το οποίο πρέπει να υπολογίσει την τιμή $x = y^{\frac{1}{e}}$.

Ο $\mathcal{A}$ έχει ως είσοδο μόνο το δημόσιο κλειδί (e, n) και παράγει ως εξόδο την πλαστογράφηση (m, s) με $s = \mathcal{H}(m)^{\frac{1}{e}} \bmod n$ για κάποιο μήνυμα m. Αυτό σημαίνει ότι κάποια στιγμή το μαντείο του απάντησε με το $\mathcal{H}(m)$ στο μήνυμα m που του είχε αποστείλει. Όμως το μαντείο χρησιμοποιείται εσωτερικά από τον $\mathcal{A}$ και ο $\mathcal{B}$ δεν μπορεί να ξέρει ποια από τις q απαντήσεις $\{y_i\}_{i=1}^{q}$ στα ισάριθμα ερωτήματα (πολυωνυμικού πλήθους) που είχε κάνει, αντιστοιχεί στο $\mathcal{H}(m)$. Βέβαια αφού ο $\mathcal{A}$ εξάγει το (m, s) θα υπήρξε i ώστε $s = y_i^{1/e} \bmod n$. Άρα ο $\mathcal{B}$ εξάγει απλά την υπογραφή s. Αν ο $\mathcal{A}$ έχει πιθανότητα επιτυχίας λ για την πλαστογράφηση τότε η πιθανότητα επιτυχίας του $\mathcal{B}$ θα είναι λ/q. Έτσι αν το λ είναι αμελητέο το ίδιο θα ισχύει και το $\frac{\lambda}{q}$.

Επίθεση με επιλεγμένες υπογραφές Στην περίπτωση αυτή ο $\mathcal{A}$, προκειμένου να επιτύχει στην πλαστογράφηση μπορεί να ζητήσει - εκτός από τις συνόψεις - και υπογραφές για μηνύματα της επιλογής του.Στόχος του $\mathcal{A}$ είναι να εξάγει μια έγκυρη υπογραφή s για κάποιο μήνυμα m για το οποίο *δεν* έχει ζητήσει υπογραφή. Αυτες οι υπογραφές πρέπει να δωθούν από τον $\mathcal{B}$, ο οποίος όμως δεν μπορεί να τις

δημιουργήσει καθώς δεν γνωρίζει το ιδιωτικό κλειδί d. Το πρόβλημα αυτό μπορεί να παρακαμφθεί αν αντί για την απάντηση $\mathcal{H}(m)$ σε κάποιο μήνυμα, το μαντείο αποστείλει κατευθείαν την τιμή $s^e \bmod n$, οπότε και η υπογραφή θα επαληθεύεται τετριμμένα. Για να συμβεί αυτό, όμως, το μαντείο πρέπει να γνωρίζει εκ των προτέρων την τιμή s.

Με βάση αυτή την παρατήρηση μπορούμε να κατασκευάσουμε τον αντίπαλο $\mathcal{B}$ ως εξής:

- Δέχεται ως είσοδο και πάλι την τριάδα $(e, n), y$
- Δημιουργεί έναν, αρχικά άδειο, πίνακα $\mathcal{T}$ ο οποίος θα περιέχει τριάδες έγκυρων υπογραφών, δηλ. τριάδες (m_i, y_i, s_i) για τις οποίες θα ισχύει: $s_i^e = y_i = \mathcal{H}(m_i)$.
- Δίνει το δημόσιο κλειδί στον $\mathcal{A}$.
- Όταν ο $\mathcal{A}$ ζητήσει τη σύνοψη του μηνύματος m_i, τότε ο $\mathcal{B}$:
  - Διαλέγει τυχαίο $s_i \in \mathbb{Z}_n^*$
  - Υπολογίζει την τιμή $y_i = s_i^e \bmod n$
  - Εισάγει την τριάδια m_i, y_i, s_i στον $\mathcal{T}$
  - Απαντάει θέτωντας $\mathcal{H}(m_i) \leftarrow y_i$
- Όταν ο $\mathcal{A}$ ζητήσει την υπογραφή του μηνύματος $\mathcal{H}(m_i)$, ο $\mathcal{B}$ αναζητά στον $\mathcal{T}$ την εγγραφή y_i και επιστρέφει το αντίστοιχο s_i, παράγοντας λόγω της κατασκευής μια έγκυρη υπογραφή.

Υποθέτουμε ότι για να εξάγει την πλαστογράφηση (m, s) ο $\mathcal{A}$ θα πρέπει να έχει ζητήσει προηγούμενως από το τυχαίο μαντείο τη σύνοψη του μηνύματος m, χωρίς όμως να έχει προχωρήσει στην υπογραφή.

Η επίθεση τώρα λειτουργεί ως εξής. Ο $\mathcal{B}$ επιλέγει μία από τις q ερωτήσεις του $\mathcal{A}$ στο μαντείο και απαντάει με y, 'ελπίζοντας' ότι δεν θα κάνει την ερώτηση στο μαντείο για την τιμή αυτή, αλλά θα εξάγει την πλαστογράφηση (m, s) ώστε $s^e = y \bmod n$. Σε αυτή την περίπτωση εξάγει ως απάντηση την υπογραφή s. Η πιθανότητα να συμβεί αυτό είναι e/q όπου e η πιθανότητα επιτυχούς πλαστογράφησης από τον $\mathcal{A}$.

8.7 Ασκήσεις

1. Έστω $\mathcal{H}$ συνάρτηση σύνοψης, η οποία συμπιέζει ακολουθίες μήκους $2n$ σε ακολουθίες μήκους n και έχει την ιδιότητα δυσκολίας εύρεσης συγκρούσεων (collision free). Θέλουμε να φτιάξουμε μία συνάρτηση σύνοψης που

να συμπιέζει ακολουθίες μήκους $4n$ σε ακολουθίες μήκους n, η οποία να έχει επίσης την ιδιότητα δυσκολίας εύρεσης συγκρούσεων. Έχουμε τις εξής υποψήφιες:

- $\mathcal{H}_1(x_1, x_2, x_3, x_4) = \mathcal{H}(x_1, x_2) \oplus \mathcal{H}(x_3, x_4)$
- $\mathcal{H}_2(x) = \mathcal{H}(\mathcal{H}(\mathcal{H}(x_1, x_2), x_3), x_4)$

Θεωρήστε ότι για κάθε i, $|x_i| = n$. Για κάθε μία από τις παραπάνω συναρτήσεις εξετάστε αν έχει την ιδιότητα δυσκολίας εύρεσης συγκρούσεων ή όχι. Αποδείξτε τον ισχυρισμό σας.

2. Έστω h συνάρτηση σύνοψης, η οποία συμπιέζει ακολουθίες μήκους $2n$ σε ακολουθίες μήκους n και έχει την ιδιότητα της *δυσκολίας εύρεσης συγκρούσεων* (collision free). Θέλουμε να φτιάξουμε μία συνάρτηση σύνοψης που να συμπιέζει ακολουθίες μήκους $4n$ σε ακολουθίες μήκους n, η οποία να έχει επίσης την ιδιότητα της δυσκολίας εύρεσης συγκρούσεων. Έχουμε τις εξής υποψήφιες:

   - $h_1(x) = h((x_1 \oplus x_2)||(x_3 \oplus x_4))$
   - $h_2(x) = h(x_1||x_2) \oplus h(x_3||x_4)$

   (με "$\oplus$" συμβολίζουμε το XOR, με "||" την παράθεση και θεωρούμε $|x_i| = n$)

   Για κάθε μία από τις παραπάνω h_i, δείξτε εάν έχει την ιδιότητα της δυσκολίας εύρεσης συγκρούσεων ή όχι. Για να δείξετε ότι την έχει, αποδείξτε ότι αν μπορούσαμε να βρούμε συγκρούσεις για την h_i, τότε θα μπορούσαμε να βρούμε συγκρούσεις και για την h. Για να δείξετε ότι δεν την έχει, δώστε έναν τρόπο εύρεσης συγκρούσεων για την h_i.

3. Να τροποποιήσετε την απόδειξη ασφάλειας της ενότητας 8.6.1 ώστε η πιθανότητα επίλυσης του προβλήματος RSA, να είναι ακριβώς ίδια με την πιθανότητα επιτυχούς πλαστογράφησης. (*Υπόδειξη:* Εκμεταλλευτείτε το γεγονός ότι μπορείτε να καθορίσετε τις απαντήσεις του τυχαίου μαντείου).

4. Έστω μια συνάρτηση κρυπτογράφησης $E(k, m)$, όπου k είναι το κλειδί και m το αρχικό κείμενο, τέτοια ώστε $k, m,$ και $E(k, m)$ έχουν όλα τον ίδιο αριθμό bits n. Θέλουμε να την χρησιμοποιήσουμε για να φτιάξουμε συνάρτηση σύνοψης που να συμπιέζει ακολουθίες $2n$ bits σε ακολουθία n bits.

   - Δείξτε ότι η χρήση της E αυτούσιας δεν είναι καλή ιδέα. Συγκεκριμένα, δείξτε ότι η συνάρτηση

   $$h_1(x, x') = E(x, x')$$

δεν είναι ελεύθερη συγκρούσεων. (Υπόδειξη: θεωρήστε ότι η συνάρτηση αποκρυπτογράφησης $D(k, c)$ είναι επίσης γνωστή και σκεφτείτε με τι είναι ίσο το $E(k, D(k, c))$.)

- Εξετάστε αν η παρακάτω συνάρτηση είναι ελεύθερη συγκρούσεων:

$$h_2(x, x') = E(x', x) \oplus x'$$

5. Έστω g, h στοιχεία της πολλαπλασιαστικής ομάδας Z_p^* έτσι ώστε $\langle g \rangle = \langle h \rangle$ και το μέγεθος της $\langle g \rangle$ είναι q όπου q γνωστός πρώτος αριθμός. Έστω η οικογένεια συναρτήσεων $H_{g,h} : \{0, 1, ..., q-1\} \times \{0, 1, ..., q-1\} \to \{0, ..., p-1\}$ που ορίζεται ως εξής: $H_{g,h}(x, y) = g^x \cdot h^y \bmod p$.

   - Δείξτε ότι η συνάρτηση $H_{g,g}$ είναι μια συνάρτηση σύνοψης με αντίσταση πρώτου ορίσματος αλλά χωρίς αντίσταση δεύτερου ορίσματος.
   - Δείξτε ότι ένας αλγόριθμος που βρίσκει συγκρούσεις για οποιαδήποτε συνάρτηση της οικογένειας $H_{g,h}$ μπορεί να χρησιμοποιηθεί για να λύσει το πρόβλημα του διακριτού λογαρίθμου στην υποομάδα $\langle g \rangle$.
   - Σχεδιάστε έναν αλγόριθμο για την εύρεση του διακριτού λογαρίθμου στην υποομάδα $\langle g \rangle$ που να είναι πιο αποδοτικός από τον brute-force αλγόριθμο. Υπόδειξη: αξιοποιήστε το παράδοξο των γενεθλίων.

Βιβλιογραφία

[1] Dave Bayer, Stuart Haber, and W. Scott Stornetta. Improving the efficiency and reliability of digital time-stamping. In *Sequences II: Methods in Communication, Security and Computer Science*, pages 329–334. Springer-Verlag, 1993.

[2] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM conference on Computer and communications security*, pages 62–73. ACM, 1993.

[3] Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption. In *Advances in Cryptology—EUROCRYPT'94*, pages 92–111. Springer, 1995.

[4] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *J. ACM*, 51(4):557–594, July 2004.

[5] Stuart Haber and W. Scott Stornetta. How to time-stamp a digital document. *Journal of Cryptology*, 3:99–111, 1991.

Κεφάλαιο 9

Κρυπτογραφικά πρωτόκολλα και τεχνικές

Στην ενότητα αυτή θα εξετάσουμε πρακτικά θέματα που προκύπτουν από την χρήση των δομικών στοιχείων που περιγράψαμε στα προηγούμενα κεφάλαια. Επίσης θα αναφερθούμε σε πρωτόκολλα 'ανώτερου επιπέδου' που κάνουν χρήση των συγκεκριμένων τεχνικών για να επιλύσουν προβλήματα εμπιστοσύνης που δημιουργούνται κατά την αλληλεπίδραση οντοτήτων με αντικρουόμενα συμφέροντα.

9.1 Υποδομή Δημοσίου Κλειδιού

Οι χρήστες ενός κρυπτοσυστήματος δημοσίου κλειδιού πρέπει να είναι σίγουροι ότι τα δημόσια κλειδιά με τα οποία κρυπτογραφούν μηνύματα ή επαληθεύουν υπογραφές αντιστοιχούν όντως στους χρήστες με τους οποίους θέλουν να επικοινωνήσουν. Για παράδειγμα, ένας ενεργός αντίπαλος θα μπορούσε να αντικαταστήσει το δημόσιο κλειδί ενός χρήστη με το δικό του, με αποτέλεσμα να μπορεί να παρεμβάλλεται στις επικοινωνίες και να αποκρυπτογραφεί όλα τα μηνύματα.

Η λύση του παραπάνω προβλήματος είναι απαραίτητη για την χρήση της κρυπτογραφίας δημοσίου κλειδιού σε ευρεία κλίμακα. Παρ' όλα αυτά δεν έχει προταθεί ακόμα κάποια τεχνική η οποία είναι αποτελεσματική τόσο θεωρητικά όσο και πρακτικά. Αυτό που έχει συμβεί στην πραγματικότητα είναι η μετάθεση του προβλήματος.

Συγκεκριμένα, για να λειτουργήσει ένα κρυπτοσύστημα δημοσίου κλειδιού απαιτείται η ύπαρξη μίας *έμπιστης τρίτης οντότητες - trusted third party (TTP)* η οποία

ονομάζεται *αρχή πιστοποίησης - certification authority (CA)* [1]. Ο ρόλος της είναι να *πιστοποιεί* την αντιστοίχιση δημοσίου κλειδιού - χρήστη, ότι δηλαδή το δημόσιο κλειδί αντιστοιχεί όντως στον χρήση ο οποίος ισχυρίζεται κάτι τέτοιο. Ένα τέτοιο πιστοποιητικό θα πρέπει να περιέχει πληροφορία ταυτοποίησης του χρήστη μαζί με το δημόσιο κλειδί του. Για λόγους αυθεντικοποίησης και ακεραιότητας όμως, τα δεδομένα αυτά θα πρέπει να περιέχουν μία ψηφιακή υπογραφή από την αρχή πιστοποίησης. Η επαλήθευση της θα γίνει χρησιμοποιώντας το δημόσιο κλειδί της.

Από την παραπάνω περιγραφεί προκύπτει το εξής ερώτημα: Ποιος εγγυάται την σχέση δημοσίου κλειδιού - ταυτότητας για την αρχή πιστοποίησης; Με άλλα λόγια, το πρόβλημα μετατέθηκε από τους απλούς χρήστες στις αρχές πιστοποίησης. Στο σημείο αυτό υπάρχουν δύο πιθανές λύσεις: Είτε να μεταθέσουμε την πιστοποίηση των αρχών σε ένα ανώτερο επίπεδο μέτα-πιστοποίησης (μόνο για CAs) ή οι αρχές να υπογράφουν μόνες τους τα πιστοποιητικά που τις αφορούν. Και οι δύο παραπάνω λύσεις εφαρμόζονται στην πράξη.

Το όλο 'οικοσύστημα' χρηστών, αρχών πιστοποίησης που προκύπτει από την παραπάνω περιγραφή ονομάζεται *Υποδομή Δημοσίου Κλειδιού* και προτάθηκε για πρώτη φορά από τον Kohnfelder στο [1]. Στην πράξη βέβαια τα ψηφιακά πιστοποιητικά, έχουν εξελιχθεί από την πρόταση αυτή και έχουν αρκετά πολύπλοκη δομή. Είναι προφανές ότι για την χρήση τους από όσο το δυνατόν περισσότερες εφαρμογές, είναι απαραίτητη η προτυποποίηση των περιεχομένων τους. Η βασική προτυποποίηση στον τομέα αυτόν προέρχεται από την ITU, η οποία έχει δημοσιεύσει το βασικό πρότυπο για ψηφιακά πιστοποιητικά, το X.509 [2]. Αξίζει εδώ να σημειώσουμε ότι το συγκεκριμένο πρότυπο στην αρχική του μορφή δεν προορίζονταν για την χρήση που προαναφέραμε. Αποτελούσε τμήμα του προτύπου X.500 του για την πρόσβαση σε κατανεμημένες υπηρεσίες ενός παγκόσμιου καταλόγου, του οποίου την λειτουργία θα είχαν οι μεγαλύτεροι τηλεπικοινωνιακοί οργανισμοί. Η αυθεντικοποίηση των οντοτήτων για την πρόσβαση στον συγκεκριμένο κατάλογο βασίζονταν σε ασύμμετρα κρυπτοσυστήματα. Τα πιστοποιητικά X.509 συσχέτιζαν ένα δημόσιο κλειδί σε μία οντότητα του καταλόγου. Με βάση την παραπάνω συσχέτιση γινόταν ο έλεγχος πρόσβασης.

Από την παραπάνω περιγραφή προκύπτει ότι αρχές πιστοποίησης λειτουργούν ως ενδιάμεσοι στην εκτέλεση μιας ηλεκτρονικής συναλλαγής. Επιπλέουν μπορεί να παρέχουν και επιπλέον υπηρεσίες οι οποίες απαιτούνται για την πρακτική τους λειτουργία. Για παράδειγμα κάποιος χρήστης μπορεί να θεωρεί ότι το ιδιωτικό του κλειδί έχει παραβιαστεί και θέλει να αλλάξει. Πρέπει με κάποιο τρόπο να ενημερωθούν οι υπόλοιποι χρήστες ότι το τρέχον πιστοποιητικό δεν ισχύει (έχει ανακληθεί δηλαδή) και πρέπει να εκδοθεί καινούριο.

[1] Νομικά χρησιμοποιείται ο όρος πάροχος υπηρεσιών πιστοποίησης

Συγκεντρωτικά οι πιο σημαντικές υπηρεσίες των αρχών πιστοποίησης είναι:

- Υπηρεσίες Διάδοσης / Ανάκτησης Πιστοποιητικών (Directory / Dissemination Services). Με τις υπηρεσίες αυτές, τα πιστοποιητικά διατίθενται στις οντότητες που θέλουν να επαληθεύσουν ηλεκτρονικές υπογραφές. Η διάδοση αυτή γίνεται μέσω μιας υπηρεσίες καταλόγου (directory service), στην οποία τοποθετούνται τα ψηφιακά πιστοποιητικά και οι ενδιαφερόμενες οντότητες τα ανακτούν.
- Υπηρεσίες Εγγραφής / Ταυτοποίησης Οντοτήτων (Registration Services). Οι συγκεκριμένες υπηρεσίες έχουν ως στόχο την λειτουργία ενός τμήματος καταχώρησης στοιχείων των οντοτήτων που θέλουν να αποκτήσουν ένα πιστοποιητικό και επαλήθευσης της ταυτότητας τους.
- Υπηρεσίες Διαχείρισης Ανάκλησης Πιστοποιητικών (Revocation Management Services): Οι συγκεκριμένες υπηρεσίες είναι υπεύθυνες για την παραλαβή και εξέταση των αιτήσεων για ανάκληση πιστοποιητικών.
- Υπηρεσίες ενημέρωσης για την κατάσταση των πιστοποιητικών (Revocation Status Service). Οι συγκεκριμένες υπηρεσίες είναι υπεύθυνες για την δημοσίευση των αποτελεσμάτων των ενεργειών της προηγούμενης υπηρεσίας. Η υλοποίηση της συγκεκριμένης υπηρεσίας μπορεί να γίνεται σε πραγματικό χρόνο ή σε τακτά χρονικά διαστήματα.
- Προαιρετικά, υπηρεσίες δημιουργίας κρυπτογραφικών κλειδιών (Private Key Generation Service) και υπηρεσίες παροχής ασφαλών συσκευών δημιουργίας υπογραφών (Subscriber Device Provision Service). Εδώ η αρχή πιστοποίησης, είτε παρέχει στον συνδρομητή το ιδιωτικό του κλειδί, είτε αρχικοποιεί την ασφαλή διάταξη δημιουργίας υπογραφής, η οποία θα χρησιμοποιείται για την δημιουργία των υπογραφών.
- Προαιρετικά υπηρεσίες χρονοσήμανσης και αρχειοθέτησης.

Από την παραπάνω περιγραφή βλέπουμε ότι αν και η ιδέα των ψηφιακών πιστοποιητικών είναι θεωρητικά απλή, η πρακτική υλοποίηση της μέσω υποδομών δημοσίου κλειδιού είναι αρκετά πολύπλοκη. Μία εναλλακτική προσέγγιση στο παραπάνω πρόβλημα αποτελεί η κρυπτογράφηση με βάση την ταυτότητα η οποία θα αναλυθεί στην ενότητα 12.3.4.

9.2 Σχήματα Δέσμευσης

Τα σχήματα δέσμευσης είναι εργαλεία τα οποία επιτρέπουν σε μία οντότητα να επιλέξει μία τιμή χωρίς να την αποκαλύψει (ιδιότητα *απόκρυψης*) και χωρίς να

μπορεί να την αλλάξει (ιδιότητα *δέσμευσης*).

Η χρησιμότητα τέτοιων σχημάτων γίνεται άμεσα χρήσιμη με το παρακάτω κλασικό παράδειγμα της 'τηλεφωνικής' ρίψης νομίσματος, που πρωτοχρησιμοποιήθηκε από τον Manuel Blum.

Η Alice (ο *αποστολέας*) και ο Bob (ο *παραλήπτης*) βρίσκονται σε διαφορετικές τοποθεσίες και θέλουν να ρίξουν ένα νόμισμα. Συμφωνούν ότι η Alice θα διαλέξει κορώνα ή γράμματα ενώ ο Bob θα πραγματοποιήσει τη ρίψη και θα ανακοινώσει το αποτέλεσμα. Ένας κακόβουλος Bob δεν χρειάζεται να κάνει οτιδήποτε - απλά ανακοινώνει ότι νίκησε και η Alice δεν μπορεί να κάνει τίποτα. Εναλλακτικά μία κακόβουλη Alice μπορεί να απαιτήσει να μην γνωστοποιήσει την επιλογή της, ώστε ο Bob να μην έχει καμία πληροφορία στη διάθεση του. Έτσι τώρα θα κερδίσει αυτή ανεξάρτητα από το αποτέλεσμα της ρίψης του Bob.

Ένα σχήμα δέσμευσης μπορεί να επιλύσει το παραπάνω πρόβλημα χρησιμοποιώντας τις ιδιότητες της απόκρυψης και της δέσμευσης:

- Η Alice δεσμεύεται σε κορώνα ή γράμματα και στέλνει τη δέσμευση στον Bob. Λόγω της ιδιότητας της απόκρυψης ο Bob δεν μπορεί να μάθει τίποτα από την τιμή που παρέλαβε.
- Ο Bob πραγματοποιεί την ρίψη και ανακοινώνει το αποτέλεσμα.
- Η Alice αποκαλύπτει την επιλογή της. Λόγω της ιδιότητας της δέσμευσης η τιμή που αποκαλύπτεται θα είναι η αρχική.
- Ελέγχουν το αποτέλεσμα και αποδέχονται τον νικητή.

Η παραπάνω αλληλεπίδραση προϋποθέτει δύο φάσεις σε ένα σχήμα δέσμευσης: την φάση *δέσμευσης* και τη φάση *ανοίγματος* Επιπλέον πρέπει να πληρούνται οι εξής ιδιότητες:

- Η φάση της δέσμευσης πρέπει να είναι υλοποιήσιμη.
- Η φάση του ανοίγματος είναι μονοσήμαντη, δηλαδή μία δέσμευση σε ένα bit δεν μπορεί να ανοίξει τόσο ως 0 όσο και ως 1.
- Οι δεσμεύσεις δεν αποκαλύπτουν ή διαρρέουν τίποτα.

Παρακάτω παρουσιάζονται κάποια σχήματα κρυπτογραφικά σχήματα δέσμευσης, εκτός από αυτά που είδαμε στην ενότητα 8.4.1.

Pedersen Commitment

Ένα σχήμα δέσμευσης το οποίο είναι υπολογιστικά δεσμευτικό και παρέχει πληροφοριοφοριοθεωρητική απόκρυψη προτάθηκε από τον Pedersen βασίζεται στο πρόβλημα Διακριτού Λογαρίθμου [3]. Λειτουργεί ως εξής:

- Ο παραλήπτης επιλέγει δύο πρώτους p, q και έναν γεννήτορα g από μια υποομάδα τάξης q της $\mathbb{Z}_q^*$.
- Επιλέγει μια τυχαία τιμή $r \in \mathbb{Z}_q$ και υπολογίζει το $h = g^r$
- Ο αποστολέας λαμβάνει τις τιμές (p, q, g, h).
- Για να δεσμευτεί σε μία τιμή $M \in \mathbb{Z}_q$ επιλέγει $R \in \mathbb{Z}_q$ και υπολογίζει $C = g^R h^M$

Η επαλήθευση της δέσμευσης γίνεται ως:

- Ο αποστολέας αποστέλλει τα M, R στον παραλήπτη.
- Αυτός επαληθεύει αν ισχύει $g^R h^M = C$ και σε θετική περίπτωση δέχεται

Θεώρημα 9.1. *Το σχήμα δέσμευσης του Pedersen ικανοποιεί τις ιδιότητες απόκρυψης και δέσμευσης αν το πρόβλημα του διακριτού λογαρίθμου είναι δύσκολο*

Απόδειξη. Για την ιδιότητα της δέσμευσης υποθέτουμε αρχικά ότι δεν ισχύει. Έτσι ο αποστολέας μπορεί να αλλάξει το μήνυμα M σε M' χωρίς να επηρεαστεί η δέσμευση. Αυτό σημαίνει ότι:

$$C = g^R h^M$$
$$C = g^{R'} h^{M'}$$

Έτσι:

$$\begin{aligned} g^R h^M = g^{R'} h^{M'} &\Rightarrow \\ g^{R-R'} = hM - M' &\Rightarrow \\ g^{(R-R')k} = h \quad \text{όπου:} \quad k(M - M') = 1 \pmod q & \end{aligned}$$

Αυτό όμως δεν είναι μπορεί να συμβαίνει καθώς το k μπορεί να βρεθεί γρήγορα χρησιμοποιώντας τον Εκτεταμένο Αλγόριθμο του Ευκλείδη.

Για την ιδιότητα της απόκρυψης, θα αποδείξουμε ότι ισχύει χωρίς προϋποθέσεις καθώς οποιαδήποτε τιμή $M^* \in \mathbb{Z}_q$ έχει ίδια πιθανότητα να είναι αυτή στην οποία έγινε η δέσμευση. Αν ο παραλήπτης έχει απεριόριστες υπολογιστές δυνατότητα τότε με δεδομένο το C, μπορεί να υπολογίσει τα M, R. Όμως:

$$\begin{aligned} \forall M^* \in \mathbb{Z}_q \quad \exists R^* \quad \text{ώστε:} \\ C = g^{R^*} h^{M^*} \\ C = g^{R+Mr} = g^{R^*+M^*r} \Rightarrow \\ R + Mr = R^* + M^*r \Rightarrow \\ R^* = (M - M^*)r + R \pmod q \end{aligned}$$

□

9.3 Διανομή Απορρήτων (Secret Sharing)

Ένα σχήμα *διανομής (ή κατανομής) απορρήτων* παρέχει ένα τρόπο για την επίλυση του εξής προβλήματος:

Έστω ότι έχουμε l πρόσωπα (παίκτες) P_i. Κάθε πρόσωπο P_i έχει στην κατοχή του κάποια πληροφορία p_i η οποία είναι άγνωστη σε όλους τους άλλους P_j, όπου $j \neq i$. Θέλουμε να δώσουμε ένα σχήμα με το οποίο κάποια πληροφορία s μπορεί εύκολα να υπολογιστεί από κάθε t-άδα εκ των p_i αλλά η γνώση $t-1$ εκ των p_i δεν αρκεί για να υπολογιστεί το s. Ο κάτοχος της πληροφορίας αυτή ονομάζεται διανομέας (dealer).

Ο επόμενος ορισμός, περιγράφει την ιδέα ενός σχήματος κατωφλίου (threshold scheme):

Ορισμός 9.2. Έστω t, l θετικοί ακέραιοι, $t \leq l$. Το σύνολο $P = \{p_1, \ldots, p_l\}$ λέγεται (t, l)-**σχήμα κατωφλιού** αν ισχύουν τα παρακάτω:

- Το s, μπορεί να υπολογιστεί εύκολα από οποιαδήποτε t στοιχεία του συνόλου A.
- Η γνώση οποιωνδήποτε $t-1$ στοιχείων του A δεν αρκεί για να υπολογιστεί το c.

9.3.1 Διανομή Απορρήτων Shamir

Η πρώτη μέθοδος υλοποίησης ενός τέτοιου σχήματος προτάθηκε από τον Shamir στο [4]. Επιτρέπει τον διαμοιρασμό ενός τέτοιο μυστικού στοιχείου s μεταξύ l παικτών ώστε οποιοδήποτε υποσύνολο από t να μπορεί να το ανακτήσει, αλλά όχι οποιοδήποτε υποσύνολο από $t-1$ παίκτες. Για να το πετύχει αυτό χρησιμοποιεί την ιδέα της *πολυωνυμικής παρεμβολής*, με τη μέθοδο των συντελεστών Lagrange:

- Ένα πολυώνυμο $\mathcal{P}$ βαθμού $t-1$ μπορεί να ανακατασκευαστεί από t σημεία $\{(x_i, y_i)\}_{i=1}^{t}$

- $P(x) = \sum_{i=1}^{t} y_i \prod_{j=1, j\neq i}^{t} \frac{x-x_j}{x_i-x_j}$

Κατά συνέπεια:

- Ο διανομέας διαλέγει ένα τυχαίο πολυώνυμο $\mathcal{P}$ βαθμού $t-1$ ώστε $\mathcal{P}(0) = s$
- Παράγει και διανέμει l ζεύγη $(x_i, \mathcal{P}(x_i)), \quad x_i \neq 0$
- Σύμφωνα με την πολυωνυμική παρεμβολή t παίκτες μπορούν να συνεννοηθούν (και να ανακτήσουν το s), όχι όμως $t-1$.

Για τη διανομή ενός ακέραιου μυστικό το παραπάνω σχήμα μπορεί να εφαρμοστεί στο $\mathbb{Z}_p$ όπου p ένας πρώτος μεγαλύτερος από το μυστικό. Έτσι όλοι οι συντελεστές επιλέγονται τυχαία από το $\mathbb{Z}_p$ και όλες οι αριθμητικές πράξεις γίνονται $(\mathrm{mod}\ p)$.

Το παραπάνω σχήμα υποθέτει ότι όλοι οι παίκτες είναι *τίμιοι*. Κάτι τέτοιο όμως μπορεί να μην ισχύει πάντοτε [5]:

- Ο διανομέας μπορεί να δώσει λανθασμένα μερίδια σε όλους ή σε τμήμα των παικτών. Αυτό θα έχει ως συνέπεια να μην μπορεί να ανακατασκευαστεί το μυστικό. Για να αποφευχθεί κάτι τέτοιο πρέπει να δοθεί δυνατότητα στους παίκτες να επαληθεύσουν την ορθότητα των μεριδίων τους.
- Κάποιος παίκτης μπορεί να μην παρέχει τα σωστά μερίδια κατά τη διάρκεια της ανακατασκευής. Πρέπει λοιπόν όλοι να μπορούν να ελέγξουν ότι είναι αυτά που δόθηκαν αρχικά από τον διανομέα..

Για να αντιμετωπιστούν οι παραπάνω απειλές προτάθηκαν κάποιες παραλλαγές του [4].

Επαληθεύσιμη Διανομή Απορρήτων [6]

Ο διανομέας εισάγει επιπλέον πληροφορία ώστε να είναι δυνατή η επαλήθευση της ορθότητας των μεριδίων. Πιο συγκεκριμένα υπολογίζει και διανέμει δεσμεύσεις στους συντελεστές του πολυωνύμου. Αν δηλαδή $\mathcal{P}(x) = \sum_{i=0}^{t} a_i x^i$, ο διανομέας μοιράζει τις δεσμεύσεις $\{c_i = g^{a_i}\}_{i=0}^{t}$, όπου g είναι ο γεννήτορας μιας ομάδας με δύσκολο πρόβλημα διακριτού λογαρίθμου.

Για να επαληθεύσει κανείς ότι το μερίδιο (x_i, y_i) είναι έγκυρο, ότι δηλαδή ισχύει $y_i = \mathcal{P}(x_i)$ ελέγχει ότι:

$$g^{y_i} = \prod_{j=0}^{t} c_j^{x_i^j}$$

το οποίο πρέπει να είναι έγκυρο καθότι:

$$\prod_{j=0}^{t} c_j^{x_i^j} = \prod_{j=0}^{t} g^{a_j x_i^j} = g^{\sum_{j=0}^{t} a_j x_i^j} = g^{P(x_i)}$$

Public Verifiable Secret Sharing

Για να αντιμετωπίσουμε την περίπτωση όπου ένας παίκτης δεν παρουσιάζει τα σωστά μερίδα κατά την φάση της ανακατασκευής μπορεί να χρησιμοποιηθεί το παρακάτω πρωτόκολλο, το οποίο προτάθηκε στο [5] και αποτελείται από τις φάσεις της διανομής και της ανακατασκευής:

- **Διανομή**
  - Υποθέτουμε πάλι ότι το πολυώνυμο είναι: $\mathcal{P}(x) = \sum_{i=0}^{t} a_i x^i$
  - Κάθε χρήστης έχει ένα μυστικό κλειδί sk_i και ένα δημόσιο κλειδί $pk_i = G^{sk_i}$ όπου G είναι ένας γεννήτορας μιας ομάδας με δύσκολο πρόβλημα DLOG. Εκτός του G, θα χρησιμοποιήσουμε και έναν άλλο γεννήτορα g.
  - Ο διανομέας δεσμεύεται στους συντελεστές $\{c_i = g^{a_i}\}_{i=0}^{t}$
  - Στη συνέχεια κρυπτογραφεί τα μερίδια χρησιμοποιώντας το δημόσιο κλειδί κάθε συμμετέχοντα $\{Y_i = pk_i^{p(i)}\}_{i=1}^{n}$
  - Ο διανομέας χρησιμοποιεί το πρωτόκολλο των 10.3.2 για να δείξει ότι τα μερίδια είναι έγκυρα. Συγκεκριμένα αποδεικνύει ότι: $\{log_g \prod_{j=0}^{t} c_j^{x_i^j} = log_{pk_i} Y_i = p(i)\}_{i=1}^{n}$.
- **Ανακατασκευή**
  - Η αποκρυπρογράφηση των μεριδίων γίνεται με το ιδιωτικά κλειδιά $Y_i^{\frac{1}{sk_i}} = pk_i^{p(i)\frac{1}{sk_i}} = G^{sk_i p(i)\frac{1}{sk_i}} = G^{p(i)}$
  - Το μυστικό ανακατασκευάζεται χρησιμοποιώντας τους συντελεστές Lagrange.

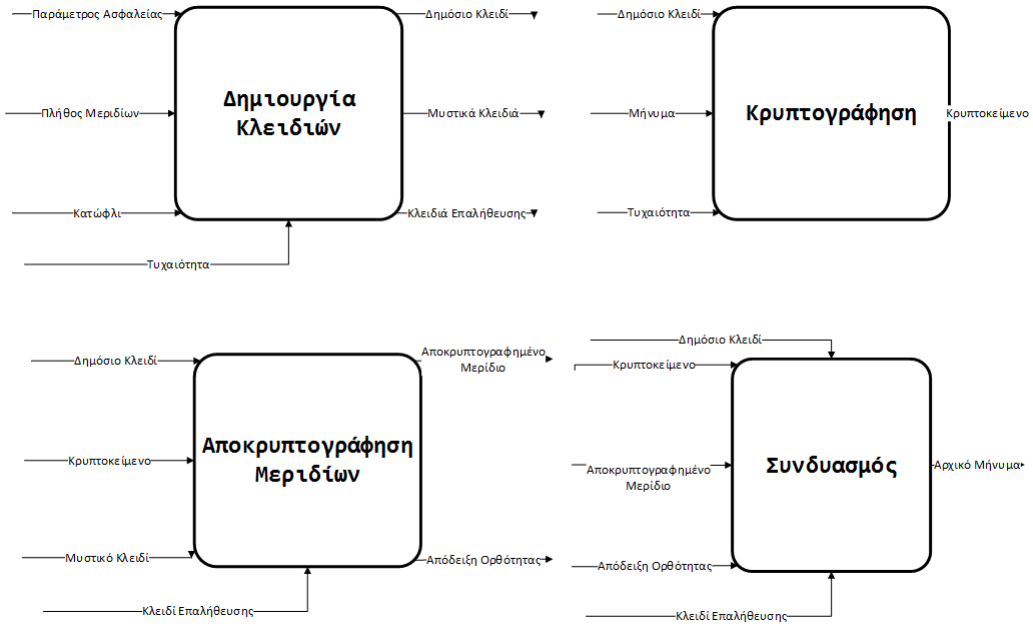
9.3.2 Κρυπτοσυστήμα Κατωφλίου - Threshold Cryptosystems

Στα κρυπτοσυστήματα δημοσίου κλειδιού ο κάτοχος του ιδιωτικού κλειδιού μπορεί να αποκρυπτογραφεί (αλλά και να υπογράφει). Αυτό του δίνει αρκετή ισχύ που σε πολλές περιπτώσεις δεν είναι επιθυμητό να συγκεντρώνεται μόνο σε μία οντότητα. Για να περιορίσουμε την ισχύ αυτή μπορούμε να θεωρήσουμε το ιδιωτικό κλειδί ως ένα μυστικό προς διαμοιρασμό σε οντότητες με αντικρουόμενα συμφέροντα. Δημιουργούμε έτσι μία νέα κατηγορία κρυπτοσυστημάτων που ονομάζονται *κρυπτοσυστήματα κατωφλίου* (threshold). Σε αυτά η λειτουργία της κρυπτογράφησης λειτουργεί κανονικά, χρησιμοποιώντας ένα δημόσιο κλειδί. Το αντίστοιχο ιδιωτικό όμως έχει διαμοιραστεί σε κάποιες οντότητες, ένα υποσύνολο των οποίων είναι απαραίτητο για την αποκρυπτογράφηση.

Ένα κρυπτοσύστημα κατωφλίου αποτελείται από τις εξής φάσεις:

- Δημιουργία Κλειδιών
- Κρυπτογράφηση
- Αποκρυπτογράφηση Μεριδίων
- Συνδυασμός

οι οποίες φαίνονται καλύτερα στο παρακάτω σχήμα:



Σχήμα 9.1: Στοιχεία (επαληθεύσιμου) κρυπτοσυστήματος κατωφλιού

Threshold El Gamal

Για να γίνουν καλύτερα κατανοητά τα παραπάνω θα περιγράψουμε το κρυπτοσύστημα που προκύπτει από τον συνδυασμό του κρυπτοσυστήματος ElGamal με το σχήμα διανομής του Shamir, όπου το κλειδί μοιράζεται σε l οντότητες από τις οποίες πρέπει να συνεργαστούν t για την αποκρυπτογράφηση ενός μηνύματος. Φυσικά κατά τη διάρκεια της αποκρυπτογράφησης δεν αναδημιουργείται το ιδιωτικό κλειδί κάτι που θα έβαζε σε κίνδυνο την ασφάλεια του συστήματος. Πιο συγκεκριμένα:

- **Δημιουργία Κλειδιών** Ένας *έμπιστος* διανομέας χωρίζει το κλειδί $x \in_R \mathbb{Z}_q$ σε l μερίδια $\{x_i = P(i)\}_{i=1}^{l}$ με το σχήμα του Shamir και τα διανέμει στους παίκτες. Επίσης υπολογίζει και δημοσιοποιεί το δημόσιο κλειδί $y = g^x$

- **Κρυπτογράφηση** Λειτουργεί κανονικά και παράγει κρυπτοκείμενα της μορφής $c = (G, M)$

- **Αποκρυπτογράφηση Μεριδίων** Κάθε παίκτης εξάγει το πρώτο μέρος του κρυπτοκειμένου και υπολογίζει το $c_i = G^{x_i} \pmod p$ το οποίο και δημοσιοποιεί.

- **Combination** Συνδυάζονται t αποκρυπτογραφημένα μερίδα τα οποία πολλαπλασιάζονται δίνοντας $\hat{c} = \prod_{i=1}^{t} c_i^{\lambda_i} \pmod p$ όπου $\lambda_i = j(j-i)^{-1} \pmod q$ είναι οι συντελεστές Lagrange. Έτσι προκύπτει το $\hat{c} = G^{P(0)} = G^x$

- **Decryption** Η αποκρυπτογράφηση γίνεται κανονικά δίνοντας $M/\hat{c}$

Αν δεν επιθυμούμε την ύπαρξη έμπιστου διανομέα μπορούμε να χρησιμοποιήσουμε την κατασκευή του [7] ως εξής:

- **Δημιουργία Κλειδιών**
  - Κάθε παίκτης i διαλέγει $x_i \in_R \mathbb{Z}_q$ ως το μερίδιο του.
  - Επιπλέον διαλέγει το πολυώνυμο $f_i(z) = \sum_{j=0}^{t-1} f_{ij} z^j$ όπου $f_{i0} = x_i$ και $f_{ij} \in_R \mathbb{Z}_q$.
  - Για κάθε συντελεστεί δημοσιοποιεί το $F_{ij} = g^{f_{ij}}$.
  - Κάθε παίκτης i στέλνει το $s_i j = f_i(j)$ στον συμμετέχοντα j.
  - Όλοι επαληθεύουν την ορθότητα των μεριδίων τους ελέγχοντας αν $g^{s_{ij}} = \prod_{l=0}^{t-1} F_{jl}^{il}$
  - Κάθε παίκτης i δεσμεύεται στα μερίδια δίνοντας $y_i = g^{x_i}$

- **Κρυπτογράφηση** Προχωράει κανονικά όπως στο ElGamal.

- **Αποκρυπτογράφηση Μεριδίων** και **Συνδυασμό** Για την αποκρυπτογράφηση του (G, M) κάθε παίκτης υπολογίζει το $w_i = G^{x_i}$. Το μήνυμα δίνεται από την σχέση $\frac{M}{\prod_{i \in \Lambda} w_j^{\lambda_i}}$. Φυσικά πρέπει να επαληθευτεί ότι $x_i = log_G w_i = log_g y_i$ κάτι που μπορεί να γίνει με το πρωτόκολλο Chaum - Pedersen 10.3.2.

9.4 Ασφαλής Υπολογισμός Συνάρτησης - Μη-Συνειδητή Μεταφορά

Στην ενότητα αυτή θα ασχοληθούμε με ένα βασικό κρυπτογραφικό πρόβλημα τον *Ασφαλή Υπολογισμό Συνάρτησης* Secure Function Evaluation (SFE) και έναν τρόπο υλοποίησης του, τη *Μη-Συνειδητή Μεταφορά* (η οποία είναι γνωστή κυρίως με τον αντίστοιχο αγγλικό όρο Oblivious Transfer (OT)).

Ορισμός 9.3. Ασφαλής Υπολογισμός Συνάρτησης m οντότητες θέλουν να υπολογίσουν από κοινού τη συνάρτηση $f(x_1, x_2, \cdots, x_n)$. Κάθε οντότητα m_i συνεισφέρει στον υπολογισμό τη δική της είσοδο x_i. Ο υπολογισμός πρέπει να θεωρείται ασφαλής δηλαδή να μην διαρρέει καμία επιπλέον πληροφορία εκτός από το αποτέλεσμα.

Στον παραπάνω ορισμό πρέπει να λάβουμε υπόψιν μας επιπλέον περιορισμούς, όπως για παράδειγμα την υπολογιστική πολυπλοκότητα, το πλήθος των bits που είναι απαραίτητο για την επικοινωνία και τον συγχρονισμό των οντοτήτων ώστε να εκτελέσουν τον υπολογισμό της συνάρτησης. Σε μία γενίκευση του παραπάνω προβλήματος κάθε οντότητα πρέπει να υπολογίσει τη δική της συνάρτηση, ζητώντας είσοδο από τις υπόλοιπες. Γενικά είναι αποδεκτός οποιοσδήποτε υπολογισμός με οποιεσδήποτε ανταλλαγές μηνυμάτων, οπότε προκύπτει μία ευρύτερη περιοχή, ο *Ασφαλής υπολογισμός Πολλών-Μερών* (Secure Multi Party Computation (MPC)), με την οποία θα ασχοληθούμε σε επόμενη ενότητα.

Αξίζει να τονίσουμε πως οι παραπάνω υπολογισμοί, όπως και πολλά άλλα στη σύγχρονη κρυπτογραφία, θα μπορούσαν να υλοποιηθούν με χρήση μιας έμπιστης τρίτης οντότητας. Φυσικά κάτι τέτοιο δεν είναι αποδεκτό καθώς η εμπιστοσύνη πρέπει να παρέχεται από το κρυπτογραφικό πρωτόκολλο που ακολουθείται.

9.4.1 Το πρόβλημα των εκατομμυριούχων

Ένα γλαφυρό παράδειγμα της αναγκαιότητας για ασφαλή υπολογισμό συνάρτησης τέθηκε από τον Andrew Yao [8] το 1982 ως εξής:

Οι εκατομμυριούχοι (από την συνεχή χρήση του ονόματός τους) Alice και Bob, θέλουν να μάθουν ποιος από τους δύο είναι πλουσιότερος, χωρίς όμως να αποκαλύψουν ακριβώς την έκταση της περιουσίας τους. Το πρόβλημα αυτό αντιστοιχεί στον υπολογισμό της συνάρτησης δύο εισόδων ($m = 2$):

$$f(x_a, x_b) = a \text{ if } x_a > x_b \text{ else } b$$

Στη διατύπωση του προβλήματος είναι δημόσια γνωστό το εύρος που κυμαίνονται οι περιουσίες από 1 έως κάποια τιμή n. Αν εκμεταλλευτούμε το γεγονός αυτό τότε

μπορεί να προκύψει μία λύση, η οποία περιγράφηκε αρχικά και από τον ίδιο τον Yao στο [8].

- Ο Bob δημιουργεί n ταυτόσημα κουτιά.
- Επιλέγει έναν αριθμό, τον οποίο τοποθετεί στο κουτί b.
- Τα υπόλοιπα κουτιά τα γεμίζει με ένα τυχαίο αριθμό, ο οποίος δεν έχει κάποια ιδιαίτερη σημασία και θεωρείται αναλώσιμος.
- Όλα τα κουτιά αποστέλλονται στην Alice.
- Η Alice με τη σειρά της τα ανοίγει.
- Αφήνει το περιεχόμενο στα πρώτα a από αυτά αμετάβλητο, ενώ αυξάνει κατά 1 τα υπόλοιπα $n - a$.
- Όλα τα κουτιά αποστέλλονται στον Bob.
- Αν ο αριθμός του Bob (αυτός που βρίσκεται στο κουτί b) έχει αλλάξει, τότε αυτός είναι πλουσιότερος, ενώ αν έχει παραμείνει ο ίδιος τότε η Alice είναι πλουσιότερη.

Παρατηρούμε ότι το παραπάνω πρωτόκολλο, αν και λύνει το πρόβλημα, έχει κάποια προβλήματα. Για παράδειγμα ο Bob μπορεί να μάθει την περιουσία της Alice, αν κρατήσει αρχείο των αριθμών που έχει τοποθετήσει σε κάθε κουτί. Επιπλέον με την παραπάνω περιγραφή μόνο ο Bob μαθαίνει ποιος είναι πλουσιότερος. Για να μάθει και η Alice πρέπει το πρωτόκολλο να εκτελεστεί με αντιστροφή των ρόλων. Σε αυτή την περίπτωση ο Bob μπορεί να κλέψει αλλάζοντας την τιμή του b και θέτοντας την ίση με το n, έτσι ώστε να φαίνεται πάντα πλουσιότερος από την Alice.

9.4.2 Ανταλλαγή Μυστικών

Ο Rabin στο [9] προσπάθησε να επιλύσει κάποια από τα παραπάνω προβλήματα βάζοντας αυστηρότερες προϋποθέσεις στο πρωτόκολλο. Συγκεκριμένα, γενίκευσε το πρόβλημα βάζοντας την Alice και τον Bob να ανταλλάξουν τα μυστικά τους s_a, s_b, με τους εξής κανόνες:

- Δεν μπορεί να χρησιμοποιηθεί έμπιστη τρίτη οντότητα
- Κανένας δεν πρέπει να μπορεί να κλέψει ή να τερματίσει το πρωτόκολλο, αφού μάθει το μυστικό του άλλου

- Η ανταλλαγή πρέπει να είναι ταυτόχρονη

Η πρώτη παρατήρηση σε ένα τέτοιο σύστημα είναι ότι οποιοδήποτε τετριμμένο πρωτόκολλο για το πρόβλημα αυτό είναι προβληματικό: Έστω ότι τα μυστικά μπορούν να υπολογιστούν από την ανταλλαγή:

- $s_a = f(a_1, a_2, \cdots, a_n)$
- $s_b = g(b_1, b_2, \cdots, b_n)$

Πάντοτε υπάρχει κάποιο k ώστε το s_a να μπορεί να υπολογιστεί από τα $a_1, a_2, \cdots, a_k$ αλλά το s_b να μην μπορεί να υπολογιστεί από $b_1, b_2, \cdots, b_{k-1}$. Με απλά λόγια, το πρωτόκολλο μπορεί να τερματιστεί νωρίτερα από έναν από τους δύο συμμετέχοντες, μόλις αυτός πάρει αυτό που θέλει.
Για να επιλύσει το πρόβλημα αυτό ο Rabin χρησιμοποίησε την έννοια της Μη-Συνειδητής Μεταφοράς. Σε αυτήν, με πολύ απλά λόγια, ο αποστολέας ενός μηνύματος *δεν μπορεί να μάθει αν ο παραλήπτης το έλαβε*. Για την υλοποίηση τής χρησιμοποιήθηκε το πρόβλημα των τετραγωνικών υπολοίπων 2.4. Συγκεκριμένα ο αποστολέας θέλει να στείλει την παραγοντοποίηση ενός αριθμού στον παραλήπτη ο οποίος θα την λάβει με πιθανότητα $1/2$.

- Η Alice και ο Bob διαθέτουν τα κλειδιά k_a, k_b για κάποιο σχήμα δέσμευσης.
- Η Alice διαλέγει έναν RSA modulus $n_a = p_a \dot{q}_a$ τον οποίο και στέλνει στον Bob χωρίς φυσικά να αποκαλύψει τους πρώτους παράγοντες.
- Ο Bob διαλέγει κάποιο $x \leq n_a$ και υπολογίζει το $c = x^2 \pmod{n}_a$.
- Στη συνέχεια αποστέλλει το c και μία δέσμευση στο x $E_{k_b}(x)$.
- Η Alice εφόσον διαθέτει την παραγοντοποίηση του n_a μπορεί να υπολογίζει μία τετραγωνική ρίζα του c χρησιμοποιώντας το κινέζικο θεώρημα των υπολοίπων (βλ. 2.3.2).
- Η ρίζα αυτή x_1 αποστέλλεται στον Bob.
- Στην συνέχεια ο Bob ελέγχει αν το $x_1 = x \pmod{n}_a$ ή αν $x_1 = -x \pmod{n}_a$. Σε αυτή την περίπτωση δεν μπορεί να μάθει την παραγοντοποίηση του n_a. Στην αντίθετη περίπτωση όμως κάτι τέτοιο είναι δυνατό χρησιμοποιώντας τον αλγόριθμο του Ευκλείδη.

Χρησιμοποιώντας το παραπάνω πρωτόκολλο ως συστατικό, η ανταλλαγή μυστικών μπορεί να επιτευχθεί ως εξής:

Σχήμα 9.2: Μη-Συνειδητή Μεταφορά

- Η Alice χρησιμοποιεί μη συνειδητή μεταφορά για την παραγοντοποίηση ενός αριθμού n_a
- Ο Bob χρησιμοποιεί μη συνειδητή μεταφορά για την παραγοντοποίηση ενός αριθμού n_b
- Και οι δύο συμμετέχοντες θέτουν μια τιμή $v_a = 0$ και $v_b = 0$ αν και μόνο ν έλαβαν την παραγοντοποίηση των n_b, n_a αντίστοιχα.
- Υπολογίζουν αντίστοιχα τις τιμές $e_a = v_a \oplus s_a$ και $e_b = v_b \oplus s_b$ τις οποίες και ανταλλάσσουν.
- Στη συνέχεια ενσωματώνουν τα μυστικά s_a, s_b σε μηνύματα m_a, m_b τα οποία και κρυπτογραφούν χρησιμοποιώντας το RSA με κλειδιά τα n_a, n_b.
- Αν $v_a = 0$ τότε η Alice μπορεί να αποκρυπτογραφήσει το m_b και κατά συνέπεια $e_a = s_a$ οπότε ο Bob έχει λάβει το μυστικό. Αντίστοιχα και για τον Bob.

9.4.3 Μη-Συνειδητή Μεταφορά (Oblivious Transfer)

Οι Even, Goldreich, Lempel στο [10] έδωσαν έναν αυστηρό ορισμό της Μη-Συνειδητής Μεταφοράς έχοντας ως πρότυπο την έννοια της μετάδοσης με θόρυβο.

Ορισμός 9.4. Ένα πρωτόκολλο Μη-Συνειδητής Μεταφοράς $OT(S, R, M)$ είναι ένα πρωτόκολλο με το οποίο ένας αποστολέας S μεταφέρει σε έναν παραλήπτη R το μήνυμα M έτσι ώστε:

- ο R λαμβάνει το M με πιθανότητα $\frac{1}{2}$. Αν δεν λάβει το μήνυμα δεν μαθαίνει καμία επιπλέον πληροφορία.
- Η εκ των υστέρων (*a posteriori*) πιθανότητα για τον S για το συμβάν ο R να λάβει το M είναι $\frac{1}{2}$.

Σχήμα 9.3: 1-από-2 Μη-Συνειδητή Μεταφορά

Σχήμα 9.4: OT από OT_1^2

- Οποιαδήποτε απόπειρα παράκαμψης του πρωτοκόλλου είναι άμεσα ανιχνεύσιμη.

Επιπλέον όρισαν μία παραλλαγή την 1-από-2 Μη-Συνειδητή Μεταφορά $OT_1^2(S, R, M_1, M_2)$ ως το πρωτόκολλο στο οποίο ο R επιλέγει μεταξύ δύο μηνυμάτων για μεταφορά με πιθανότητα $1/2$ και ο S το μεταφέρει χωρίς ασφαλώς να γνωρίζει ποιο μετέφερε. Μπορούμε να προσομοιώσουμε την τυχαία επιλογή χρησιμοποιώντας ένα bit.

Μια γενικευμένη παραλλαγή μπορεί να οριστεί ως 1-από-n Μη-Συνειδητή Μεταφορά $OT_1^n(S, R, M_1, \cdots, M_n)$ όπου ο R επιλέγει μεταξύ n μηνυμάτων να λάβει το i. Φυσικά ο S δεν το μαθαίνει, ενώ ο R δεν μαθαίνει τα $M_j, j \neq i$.

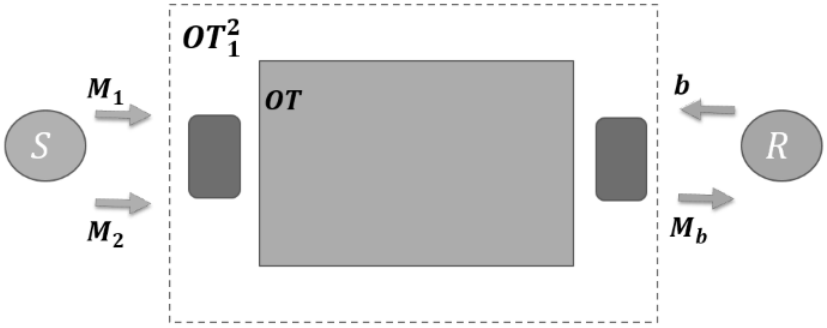
Επίσης μπορεί να οριστεί η k-από-n Μη-Συνειδητή Μεταφορά με δύο παραλλαγές. Στην πρώτη από αυτές ο R λαμβάνει ταυτόχρονα k μηνύματα, ενώ στη δεύτερη η μεταφορά γίνεται σειριακά, με δυνατότητα τροποποίησης επιλογών με βάση τις ήδη ληφθείσες (adaptive k-από-n).

Θεώρημα 9.5. $OT_1^2 \Leftrightarrow OT$

Απόδειξη. Οι Even, Goldreich, Lempel στο [10] απέδειξαν ότι $OT_1^2 \Rightarrow OT$ ενώ αργότερα ο Crepeau [11] απέδειξε ότι $OT \Rightarrow OT_1^2$.

Λήμμα 9.6. $OT_1^2 \Rightarrow OT$

Απόδειξη. Ο S θέλει να αποστείλει ένα μήνυμα M με πιθανότητα $\frac{1}{2}$ στον R. Θέτει ως είσοδο το M στην μηχανή OT η οποία δημιουργεί ένα επιπλέον μήνυμα M' (τυχαίο) για να το χρησιμοποιήσει εσωτερικά στην δεδομένη OT_1^2 μηχανή, η οποία έχει δύο εισόδους όπως φαίνεται άλλωστε και από το παρακάτω σχήμα.



Σχήμα 9.5: OT_1^2 από OT

Επιπλέον δημιουργεί δύο τυχαία bits ακόμα, το b για προσομοίωση του άλλου άκρου και το b' για την ρύθμιση της εισόδου στην OT_1^2. Συγκεκριμένα: Αν $b' = 0$ τότε εισάγει στην OT_1^2 το ζεύγος $(M_0, M_1) = (M, M')$ ενώ αν $b' = 1$ εισάγει το ζεύγος $(M_0, M_1) = (M', M)$.

Σε κάθε περίπτωση η OT εξάγει στον παραλήπτη το M_b που έχει εξάγει η OT_1^2. Έτσι $b = b'$ τότε θα εξαχθεί το M ενώ σε διαφορετική περίπτωση το M' που θα εκληφθεί ως θόρυβος.

Έτσι ο R μαθαίνει το M με πιθανότητα $1/2$ και προσομοιώνεται πλήρως η μηχανή OT.

□

Λήμμα 9.7. $OT \Rightarrow OT_1^2$

Απόδειξη. Για την αντίστροφη κατεύθυνση, δηλαδή για τη δημιουργία μιας μηχανής OT_1^2 χρησιμοποιώντας OT, ο Crepeau [11] αρχικά όρισε μία μηχανή OT_p η οποία μεταφέρει κάποιο μήνυμα με πιθανότητα p. Είναι φανερό ότι η μηχανή αυτή αποτελεί γενίκευση της μηχανής OT.

Στη συνέχεια απέδειξε ότι $OT_p \Rightarrow OT_1^2$. Ο S στέλνει κανονικά τα μηνύματα M_0, M_1 και ο R εισάγει το bit b. Εσωτερικά λοιπόν κατασκευάζεται ένα διάνυσμα από bit $\vec{s}$, τα οποία στέλνονται *διαδοχικά* στη μηχανή OT_p.

Ανάλογα με το b σε κάποιες από τις θέσεις του $\vec{s}$ η μεταφορά θα επιτύχει (στο σύνολο δεικτών I_b) ενώ σε κάποιες θα αποτύχει (στο σύνολο δεικτών I_{1-b}). Με βάση την πιθανοτική ανάλυση της OT_p με πολύ μεγάλη πιθανότητα, το σύνολο I_b μπορεί να βρεθεί, ενώ το σύνολο I_{1-b} θα περιέχει τουλάχιστον μία τιμή όπου απέτυχε η μεταφορά OT.

Τελικά αποστέλλεται στη μηχανή OT το $M_{\oplus_{i \in I_b}} s_i, \in \{0, 1\}$. Η μία από τις παραπάνω XOR θα αποτύχει και έτσι ακριβώς ένα από τα M_0, M_1 θα μεταφερθεί. Δηλαδή έχουμε OT_1^2. □

□

9.4.4 Πρακτική κατασκευή

Σε πρακτικό επίπεδο, για να κατασκευάσουμε ένα σύστημα OT_1^2 μπορούμε να χρησιμοποιήσουμε ένα κρυπτοσύστημα δημοσίου κλειδιού που έχει την ιδιότητα $\mathtt{M} = \mathtt{C}$. Για να υλοποιηθεί η μη συνειδητή μεταφορά των M_0, M_1 ο R επιλέγει τυχαία δύο συμβολοσειρές x_0, x_1. Τώρα για να πάρει το M_0 προχωρά στα εξής βήματα:

- Στέλνει στον S το $(Enc(x_0), x_1)$
- Ο R αποκρυπτογραφεί, παράγοντας το $(x_0, Dec(x_1))$. Η αποκρυπτογράφηση μπορεί να γίνει και στις δύο περιπτώσεις λόγω της ιδιότητας του κρυπτοσυστήματος, αλλά φυσικά μόνο στην περίπτωση του x_0 έχει νόημα.
- Τελικά ο S αποστέλλει το $(M_0 \oplus x_0, M_1 \oplus Dec(x_1))$
- Τελικά ο R ανακτά το M_0 με XOR του πρώτου συστατικού: $M_0 \oplus x_0 \oplus x_0$

9.4.5 Ασφαλής υπολογισμός συνάρτησης

Οι έννοιες της Μη-Συνειδητής Μεταφοράς και του Ασφαλούς Υπολογισμού Συνάρτησης συνδυάστηκαν από τον Yao [12], ο οποίος απέδειξε ότι χρησιμοποιώντας την πρώτη ως συστατικό στοιχείο μπορούμε να κατασκευάσουμε ένα κύκλωμα C που υπολογίζει ασφαλώς, ως προς έναν παθητικό αντίπαλο, κάποια συνάρτηση f. Οι συμμετέχοντες παρέχουν στο C τις εισόδους και μαθαίνουν το αποτέλεσμα χωρίς να αποκαλυφθεί οποιαδήποτε ενδιάμεση τιμή.

Η βασική ιδέα του Yao ήταν η χρήση της Μη-Συνειδητής Μεταφοράς για την κατασκευή *αλλοιωμένων* πινάκων τιμών για τις λογικές πύλες που συνιστούν ένα κύκλωμα. Για παράδειγμα για την κατασκευή της πύλης OR με αυτό τον τρόπο ο S παρέχει ένα bit s και ο R θα παρέχει ένα bit r οπότε θα υπολογιστεί $x = s \ \mathtt{OR} \ r$, όπως φαίνεται στον πίνακα 9.6:

Για την αλλοίωση ο S αρχικά θα επιλέξει δύο τυχαίες μεταθέσεις $v_S, v_R : \{0,1\} \to \{0,1\}$ και θα την εφαρμόσει στον πίνακα. Στη συνέχεια θα διαλέξει 4 ζεύγη από συναρτήσεις κρυπτογράφησης και αποκρυπτογράφησης:

$$(E_0^s, D_0^s), (E_1^s, D_1^s), (E_0^r, D_0^r), (E_1^r, D_1^r)$$

Στην συνέχεια εφαρμόζει τις συναρτήσεις κρυπτογράφησης στο αποτέλεσμα. Προκύπτει ο πίνακας 9.7 ο οποίο αποστέλλεται στον R μαζί με τη v_r.

s	r	s OR r
0	0	0
0	1	1
1	0	1
1	1	1

Σχήμα 9.6: Αρχικός πίνακας υπολογισμού OR

s	r	s OR r
$v_s(0)$	$v_r(0)$	$E^S_{v_s(0)}(E^R_{v_R(0)}(0))$
$v_s(0)$	$v_r(1)$	$E^S_{v_s(0)}(E^R_{v_R(1)}(1))$
$v_s(1)$	$v_r(0)$	$E^S_{v_s(1)}(E^R_{v_R(0)}(1))$
$v_s(1)$	$v_r(1)$	$E^S_{v_s(1)}(E^R_{v_R(1)}(1))$

Σχήμα 9.7: Αλλοιωμένος πίνακας υπολογισμού OR

Στη συνέχεια ο S υπολογίζει το δικό του τμήμα του υπολογισμού δηλαδή το $v_S(s)$ και στέλνει στον R το ζεύγος $(v_S(s), D_{v_S}(s)^s)$.

Ο R τώρα υπολογίζει το $v_R(r)$. Για να αποκρυπτογραφήσει χρειάζεται την συνάρτηση $D_(v_R(r))$ την οποία όμως κατέχει ο S και φυσικά πρέπει να του στείλει χωρίς όμως να αποκαλυφθεί το $v_R(r)$. Για τον σκοπό αυτό χρησιμοποιείται η Μη-Συνειδητή Μεταφορά $OT^2_1(S, R, D^R_0, D^R_1)$. Τελικά ο S μπορεί να υπολογίσει το αποτέλεσμα $D^R_{v_R(r)}(D^S_{v_S(s)}(E^S_{v_S(s)}(E^R_{v_R(r)}(x))))$.

Δηλαδή με τη μετάθεση των γραμμών του πίνακα αλήθειας προκύπτει μία τυχαία μετάθεση του αποτελέσματος της πύλης. Κατά συνέπεια τόσο το αποτέλεσμα όσο και οι είσοδοι μπορούν να θεωρηθούν τυχαία κλειδιά. Χρειάζονται 6 ανά πύλη όπως φαίνεται στον πίνακα 9.8.

s	r	s OR r	Computation
k^S_0	k^R_0	k^{OR}_0	$E_{k^S_0}(E_{k^R_0}(k^{OR}_0))$
k^S_0	k^R_1	k^{OR}_1	$E_{k^S_0}(E_{k^R_1}(k^{OR}_1))$
k^S_1	k^R_0	k^{OR}_1	$E_{k^S_1}(E_{k^R_0}(k^{OR}_1))$
k^S_1	k^R_1	k^{OR}_1	$E_{k^S_1}(E_{k^R_1}(k^{OR}_1))$

Σχήμα 9.8: Τελικός πίνακας υπολογισμού OR

Για να φτιαχτεί το κύκλωμα αλλοιώνονται με τον τρόπο που περιγράψαμε παραπάνω όλες οι πύλες και συνδυάζονται τροφοδοτώντας τις εξόδους κάθε μιας στις επόμενες. Οι ενδιάμεσες έξοδοι παραμένουν κρυπτογραφημένες ενώ μόνο οι τελικές έξοδοι αποκρυπτογραφούνται από τον R.

9.5 Ομομορφική Κρυπτογραφία

Η ομομορφική κρυπτογραφία αφορά την εκτέλεση υπολογισμών σε κρυπτογραφημένα δεδομένα, επιτρέποντας τον συνδυασμό κρυπτοκειμένων με τέτοιο τρόπο, ώστε το κρυπτοκείμενο που προκύπτει, να αντιστοιχεί στον υπολογισμό κάποιας (άλλης) συνάρτησης στο μηνύματα. Έτσι το αποτέλεσμα μπορεί να ανακτηθεί με μία απλή αποκρυπτογράφηση. Φυσικά όλα τα κρυπτοκείμενα πρέπει να έχουν κρυπτογραφηθεί με το ίδιο δημόσιο κλειδί, για να μπορέσει να έχει νόημα ο συνδυασμός τους. Αυτό φαίνεται καλύτερα από την παρακάτω σχέση:

$$\texttt{Encrypt}(m_1) \otimes \texttt{Encrypt}(m_2) = \texttt{Encrypt}(m_1 \oplus m_2)$$

Όλα τα κρυπτοσυστήματα που έχουμε δει μέχρι τώρα έχουν ομομορφικές ιδιότητες, οι οποίες έγιναν αντιληπτές από πολύ νωρίς καθώς σχετίζονται με το ευμετάβλητο των κρυπτοκειμένων. Συγκεκριμένα:

- Το κρυπτοσύστημα RSA (6.4) και το ElGamal (6.5.2) είναι πολλαπλασιαστικά ομομορφικά καθώς:
  - Στο RSA:

$$\texttt{Encrypt}_{(e,n)}(m_1) \cdot \texttt{Encrypt}_{(e,n)}(m_2) = m_1^e \bmod n \cdot m_2 \bmod n = (m_1 \cdot m_2)^e \bmod n = \texttt{Encrypt}_{(e,n)}(m_1 \cdot m_2)$$

  - Στο ElGamal αν πολλαπλασιάσουμε τα κρυπτοκείμενα κατά μέλη:

$$\begin{aligned} \texttt{Encrypt}_{(g,y)}(m_1) \cdot \texttt{Encrypt}_{(g,y)}(m_2) = \\ (g_1^r, m_1 \cdot y^{r_1}) \cdot ((g_2^r, m_2 \cdot y^{r_2}) = \\ (g^{r_1+r_2}), (m_1 \cdot m_2) \cdot y^{r_1+r_2}) = \texttt{Encrypt}_{(g,y)}(m_1 \cdot m_2) \end{aligned}$$

- Το εκθετικό ElGamal κρυπτοσύστημα (6.5.2), καθώς και τα κρυπτοσυστήματα Goldwasser Micali (6.6.2), Benaloh (6.6.2), Paillier (6.6.2)(καθώς και όλες οι παραλλαγές του) είναι προσθετικά ομομορφικά, καθώς:

$$\texttt{Encrypt}(m_1) \cdot \texttt{Encrypt}(m_2) = \texttt{Encrypt}(m_1 + m_2)$$

Όπως, ίσως παρατηρεί κανείς, στο ElGamal ο ομομορφισμός έχει μία ενδιαφέρουσα ιδιότητα η οποία ονομάζεται *επανακρυπτογράφηση - reencryption* και επιτρέπει την *αλλαγή* της τυχαιότητας που χρησιμοποιήθηκε κατά την κρυπτογράφηση, και κατ' επέκταση της μορφής του κρυπτοκειμένου, χωρίς να είναι απαραίτητη η γνώση του ιδιωτικού κλειδιού. Συγκεκριμένα, αν θέσουμε ως δεύτερο κρυπτοκείμενο, μία κρυπτογράφηση του $m_2 = 1$ έχουμε:

$$\texttt{Encrypt}(m1, r1) \cdot \texttt{Encrypt}(1, r2) = \texttt{Encrypt}(m_1, r_1 + r_2)$$

Φυσικά το ίδιο συμβαίνει στην περίπτωση του εκθετικού ElGamal για το μήνυμα $m_2 = 0$.

Άλλες ενδιαφέρουσες ομομορφικές ιδιότητες [13] προκύπτουν από το κρυπτοσύστημα Damgård-Jurik (6.6.2), όπου ένα κρυπτοκείμενο $c \in \mathbb{C}_{\mathcal{CS}_s}$ είναι και στοιχείο του $\mathbb{Z}^*_{s+1}$. Αυτό σημαίνει ότι ανήκει στο $MSG_{\mathcal{CS}_{s+t}}$, όπου $t \geq 1$ και έχει τις εξής συνέπειες, αν συμβολίσουμε με $\texttt{Encrypt}_s$ την κρυπτογράφηση στο $\mathcal{CS}_s$ και με $\texttt{Encrypt}_t$ την κρυπτογράφηση στο $\mathcal{CS}_{s+t}$:

- $\texttt{Encrypt}_t(1)^{\texttt{Encrypt}_s(m)} = \texttt{Encrypt}_t(m)$
- $\texttt{Encrypt}_t(0)^{\texttt{Encrypt}_s(m)} = \texttt{Encrypt}_t(0)$
- $\texttt{Encrypt}_t(0)\texttt{Encrypt}_t(\texttt{Encrypt}_s(m)) = \texttt{Encrypt}_t(\texttt{Encrypt}_s(m))$
- $\texttt{Encrypt}_t(\texttt{Encrypt}_s(0, r_0))^{\texttt{Encrypt}_s(m, r_m)} = \texttt{Encrypt}_t(\texttt{Encrypt}_s(0, r_0)\texttt{Encrypt}_s(m, r_m)) = \texttt{Encrypt}_t(\texttt{Encrypt}_s(m, r_0 r_m))$

Οι παραπάνω ιδιότητες επιτρέπουν την μεταφορά κρυπτοκειμένων από το $\mathcal{CS}_s$ στο $\mathcal{CS}_{s+t}$ χωρίς την αλλαγή κλειδιών, αφού όπως είδαμε, η επιλογή του s μπορεί να γίνει μετά την δημιουργία κλειδιών. Η απόδειξη τους είναι η άσκηση 4.

9.6 Ασφαλής Υπολογισμός Πολλών Συμμετεχόντων

Πολλές από τις έννοιες που είδαμε στην τρέχουσα ενότητα, αλλά και άλλες που θα συναντήσουμε στα επόμενα κεφάλαια, μπορούν να συγκεντρωθούν σε ένα ενιαίο - γενικότερο πλαίσιο. Το πλαίσιο αυτό ονομάζεται MPC και ενσωματώνει πέρα από κρυπτογραφικές τεχνικές, αλγόριθμους κατανεμημένου υπολογισμού, θεωρία πληροφορίας και πολυπλοκότητας και άλλες περιοχές της επιστήμης υπολογιστών.

Ο MPC ξεκινάει από τον παρακάτω ορισμός ο οποίος θυμίζει πολύ τον ορισμό που δώσαμε στην ενότητα 9.4.5 για την ασφαλή αποτίμηση συνάρτησης:

Ορισμός 9.8. Ασφαλής Υπολογισμός Πολλών Συμμετεχόντων
m οντότητες θέλουν να υπολογίσουν από κοινού τη συνάρτηση $f(x_1, \cdots, x_n) = (y_1, \cdots, y_n)$, όπου κάθε οντότητα m_i συνεισφέρει στον υπολογισμό τις δικές της εισόδους x_i.

Η διαφορά των δύο ορισμών έγκειται στο γεγονός ότι ο MPC είναι διαρκής, με την έννοια ότι μπορεί σε περισσότερες από μία στιγμές να παρέχονται είσοδοι και να εξάγονται αποτελέσματα.
Η ασφάλεια ορίζεται με τον ίδιο τρόπο:

- Ο υπολογισμός πρέπει να χαρακτηρίζεται από *ορθότητα*
- Κάθε συμμετέχων μαθαίνει μόνο τις εξόδους που προορίζονται για τον ίδιο, ενώ οι είσοδοι παραμένουν ιδιωτικοί, εκτός από ό,τι διαρρέει από τις εξόδους του πρωτοκόλλου.

Τα ερευνητικά ερωτήματα που αφορούν τη συγκεκριμένη περιοχή αφορούν τα αν μπορεί να υπάρξει πρωτόκολλο που να υλοποιεί ένα συγκεκριμένο είδος MPC και την αποδοτικότητα του, η οποία μπορεί να εκφραστεί με πολλά μέτρα, όπως για παράδειγμα:

- Υπολογιστική Πολυπλοκότητα
- Πολυπλοκότητα Επικοινωνίας (σε γύρους ή σε μηνύματα που ανταλλάσσονται)
- Τυχαιότητα που χρειάζεται

Ο αντίπαλος Η δυσκολία στο παραπάνω εγχείρημα είναι ότι οι παραπάνω προϋποθέσεις πρέπει να ισχύουν ακόμα και όταν κάποιοι συμμετέχοντες *δεν* ακολουθούν επακριβώς και αποκλειστικά το πρωτόκολλο. Για λόγους απλότητας στα πρωτόκολλα MPC δεν ασχολούμαστε με κάθε κακόβουλο παίκτη ξεχωριστά, αλλά θεωρούμε ότι υπάρχει ένας αντίπαλος $\mathcal{A}$ ο οποίος έχει διαφθείρει τους συμμετέχοντες, οι οποίοι πλέον ακολουθούν αποκλειστικά τις εντολές του και λειτουργούν ως υποχείριά τού.

Ανάλογα με τη συμπεριφορά του, ο $\mathcal{A}$ μπορεί να είναι:

- *Παθητικός (Passive ή Semi-Honest* Ακολουθεί τα βήματα του πρωτοκόλλου, πραγματοποιώντας όμως και επιπλέον δικούς του 'εσωτερικούς' υπολογισμούς. Φυσικά έχει πρόσβαση σε όλη την εσωτερική πληροφορία των παικτών που ελέγχει. Στόχος του κυρίως αποτελεί η παραβίαση της ιδιωτικότητας των υπολοίπων.

- *Ενεργοί (Active)* Μπορούν να πραγματοποιήσουν αυθαίρετες δικές τους ενέργειες για να πετύχουν τους στόχους τους.

Ανάλογα με το πότε εκδηλώνεται η επίθεση, δηλαδή ο έλεγχος των παικτών, έχουμε:

- *Στατικό (Static)* αντίπαλο, όπου η επίθεση γίνεται πριν την έναρξη του πρωτοκόλλου.
- *Δυναμικό (Dynamic)* αντίπαλο, όπου η επίθεση εκδηλώνεται κατά την διάρκεια του πρωτοκόλλου.
- *Κινητό (Mobile)* αντίπαλο, όπου θεωρούμε ότι ο $\mathcal{A}$ μπορεί να αλλάζει υποχείρια, αρκεί να διατηρείται κάποιο σταθερό χαρακτηριστικό - για παράδειγμα ο συνολικός αριθμός να είναι κάτω από κάποιο όριο.

Το σύνολο της συμμετοχής του αντίπαλου καθορίζεται από την *επιθετική δομή - adversarial structure*, δηλαδή τα διάφορα υποσύνολα των παικτών που μπορεί να διαφθείρει, αλλά και τα διάφορα σενάρια-στρατηγικές επίθεσης. Στην πιο απλή μορφή της, η επιθετική δομή, προσδιορίζει απλά το πλήθος t των παικτών που μπορεί να ελέγξει ο αντίπαλος. Πιο σύνθετες επιθετικές δομές, καθορίζουν τις σχέσεις των υποσυνόλων που ελέγχει ο $\mathcal{A}$. Σε αυτή την εντύπωση μιλάμε για *γενικό αντίπαλο - general adversary*.

Ανάλογα με τις δυνατότητες του αντιπάλου ορίζουμε δύο είδη αντιπάλων των πρωτοκόλλων MPC:

- Με απεριόριστη ισχύ οπότε θέλουμε *Πληροφοριοθεωρητική (information theoretic)* ασφάλεια.
- Με περιορισμένη ισχύ όπου ο αντίπαλος θεωρείται ότι δεν μπορεί να λύσει κάποιο πρόβλημα, σε σχέση με το οποίο ορίζουμε την ασφάλεια του πρωτοκόλλου. Εδώ θα έχουμε *υπολογιστική* ασφάλεια.

9.6.1 Μοντέλο Ασφάλειας

Ο ορισμός και η απόδειξη της ασφάλειας ενός πρωτοκόλλου MPC πρέπει να είναι αρκετά γενικός ώστε να καλύψει όλες τις περιπτώσεις στις οποίες θα βρει εφαρμογές το πρωτόκολλο. Δεν έχει νόημα λοιπόν η αναζήτηση συγκεκριμένων και χρήση συγκεκριμένων ιδιοτήτων. Αντίθετα χρησιμοποιείται και εδώ το πλαίσιο της προσομοίωσης και σύγκρισης.

Θεωρούμε, λοιπόν, έναν ιδεατό κόσμο, στον οποίο η λειτουργία του πρωτοκόλλου $\mathcal{F}$ υλοποιείται από μία έμπιστη αρχή (TTP). Η $\mathcal{F}$ περιγράφει το στόχο του πρωτοκόλλου - για παράδειγμα να υπολογίζει με ασφάλεια μια συνάρτηση. Συγκεκριμένα υποθέτουμε ότι στον ιδεατό κόσμο, οι είσοδοι και οι έξοδοι μεταφέρονται με

ασφάλεια στην TTP, η οποία καθώς είναι ιδανική υλοποιεί την $\mathcal{F}$ με ορθότητα και ασφάλεια, παρά τις πιθανές επιθέσεις (που συμβαίνουν και στους δύο κόσμους).

Ο ιδεατός κόσμος, συγκρίνεται με την υλοποίηση της λειτουργικότητας από το πρωτόκολλο $\pi^{\mathcal{F}}$, όπου περιγράφεται το MPC πρωτόκολλο. Αν οι δύο υλοποιήσεις είναι μη διαχωρίσιμες, δηλαδή αν οποιαδήποτε επίθεση που πετυχαίνει ενάντια στο πραγματικό πρωτόκολλο, πετυχαίνει και στο ιδανικό, τότε το πρωτόκολλο υλοποιείται με ασφάλεια. Ο κριτής της διαχωρισιμότητας είναι μία οντότητα η οποία ονομάζεται περιβάλλον.

Παράδειγμα 16. *Η λειτουργικότητα του ασφαλή υπολογισμού μιας συνάρτησης f, $\mathcal{F}_{SFE}^{f}$ είναι πολύ εύκολο να υλοποιηθεί στον ιδεατό κόσμο από την TTP ως εξής:*

- *Αναμονή για την ασφαλή μετάδοση της εισόδου x_i από κάθε συμμετέχοντα*
- *Εσωτερικός υπολογισμός της συνάρτησης $(y_1, \cdots, y_n) = f(x_1, \cdots, x_n)$*
- *Ασφαλής μετάδοση των εξόδων y_i στους συμμετέχοντες*

Αν ένας στατικός, παθητικός αντίπαλος αποφασίσει να ελέγξει ένα σύνολο συμμετεχόντων C, στον ιδεατό κόσμο το μόνο που θα μάθει είναι η είσοδος και έξοδος στο C:

$$IDEAL_C(\mathcal{F}_{SFE}^{f}(x_1, \cdots, x_n) = \{x_i, y_i\}_{i \in C}$$

Στον πραγματικό κόσμο το πρωτόκολλο π_{SFE}^{f} υλοποιείται επιπλέον με ανταλλαγή μηνυμάτων μεταξύ των συμμετεχόντων καθώς και τυχαιότητας. Ο αντίπαλος για το ίδιο σύνολο υποχείριων εκτός από την είσοδο και έξοδο των παραπάνω μαθαίνει και αυτά, δηλ:

$$REAL_C(\pi_{SFE}^{f}(x_1, \cdots, x_n) = \{x_i, y_i, r_i, msg_i\}_{i \in C}$$

Το π_{SFE}^{f} είναι ασφαλές αν η πραγματική πληροφορία του $REAL_C$ δεν είναι περισσότερη από την ιδεατή πληροφορία του $IDEAL_C$.

9.6.2 Πρωτόκολλα MPC

Στην ενότητα αυτή θα παρουσιάσουμε κάποια σημαντικά αποτελέσματα σχετικά με τα όρια του MPC, τόσο σε γενικές όσο και σε ειδικότερες περιπτώσεις. Αρχικά δίνουμε τα αποτελέσματα που αφορούν αντιπάλους όπου η επιθετική δομή είναι απλά ένα κατώφλι (threshold) στο πλήθος των υποχείριων:

- Οποιαδήποτε υπολογίσιμη συνάρτηση μπορεί να υπολογιστεί με πληροφοριοθεωρητική ασφάλεια, ενάντια σε στατικό και παθητικό αντίπαλο αν και μόνο αν αυτός δεν ελέγχει λιγότερους από $n/2$ συμμετέχοντες.

- Αν ο αντίπαλος είναι ενεργός και δυναμικό αντίπαλο για να έχουμε πληροφοριοθεωρητική ασφάλεια, πρέπει να είναι περισσότερο αδύναμος σε ό,τι αφορά τα υποχείρια. Συγκεκριμένα αρκεί και πρέπει να ελέγχει λιγότερους από $n/3$ συμμετέχοντες.

Σε περίπτωση που θέλουμε υπολογιστική ασφάλεια, αποδεικνύεται ότι μπορεί να επιτευχθεί για οποιοδήποτε $t < n$.

Για γενικό αντίπαλο, τα αντίστοιχα αποτελέσματα γίνονται ως εξής:

- Οποιαδήποτε υπολογίσιμη συνάρτηση μπορεί να υπολογιστεί με πληροφοριοθεωρητική ασφάλεια, ενάντια σε στατικό και παθητικό αντίπαλο αν και μόνο αν δεν υπάρχουν 2 υποσύνολα συμμετεχόντων που να καλύπτουν το πλήρες σύνολα.

- Οποιαδήποτε υπολογίσιμη συνάρτηση μπορεί να υπολογιστεί με πληροφοριοθεωρητική ασφάλεια, ενάντια σε δυναμικό και ενεργό αντίπαλο αν και μόνο αν δεν υπάρχουν 3 υποσύνολα συμμετεχόντων που να καλύπτουν το πλήρες σύνολα.

9.6.3 Σύνθεση Πρωτοκόλλων

Όλα τα παραπάνω ισχύουν στην περίπτωση που το πρωτόκολλο υλοποιεί μία λειτουργία. Τα σύγχρονα κατανεμημένα συστήματα συνδυάζουν πολλά πρωτόκολλα για να υλοποιήσουν τον σκοπό τους. Προκύπτει λοιπόν το εξής ερώτημα: Μπορούμε με κάποιο τρόπο να συνδυάσουμε τις εγγυήσεις ασφάλειας που παρέχουν οι αποδείξεις των μεμονωμένων πρωτοκόλλων, ώστε να λάβουμε εγγυήσεις ασφάλειας για το τελικό πρωτόκολλο.

Στην ενότητα αυτή θα παρουσιάσουμε κάποια στοιχεία από την προσέγγιση του Ran Canetti [14] στο παραπάνω ερώτημα, ορίζοντας το Πλαίσιο Καθολικής Συνθεσιμότητας Universal Composability Framework (UC) . Σε αυτό θα ορίσουμε τις κρυπτογραφικές ενέργειες που θέλουμε να υλοποιήσουμε ως ιδανικές λειτουργίες $\mathcal{F}$, οι οποίες υλοποιούνται από έμπιστες τρίτες οντότητες. Κάθε $\mathcal{F}$ υλοποιείται από κάποιο πρωτόκολλο π το οποίο εκτελούν οι συμμετέχουσες οντότητες (κάποιες από τις το τηρούν 'ευλαβικά', ενώ κάποιες άλλες είναι εχθρικές και το ακολουθούν αυθαίρετα). Ο κριτής είναι το περιβάλλον $\mathcal{Z}$, το οποίο θα κρίνει αν το πρωτόκολλο εκτελέστηκε με επιτυχία ή όχι.

Η ουσία του πλαισιου UC, είναι τα διάφορα θεωρήματα σύνθεσης, τα οποία επιτρέπουν την ασφαλή κατασκευή πολύπλοκων από απλούστερα.

Θεώρημα 9.9. *Θεώρημα Σύνθεσης Έστω ένα πρωτόκολλο π το οποίο χρησιμοποιεί σαν υπορουτίνα ένα άλλο πρωτόκολλο ϕ το οποίο υλοποιείται με ασφάλεια από ένα*

πρωτοκόλλο ρ. Αν αντικαταστήσουμε κάθε κλήση του π προς το ϕ με κλήσεις στο ρ, προκύπτει ένα πρωτόκολλο $\pi^{\rho|\phi}$ το οποίο είναι αδιαχώριστο από το π.

Το παραπάνω μας δίνει τη δυνατότητα της αρθρωτής κατασκευής πρωτοκόλλων. Για να κατασκευάσουμε ένα πρωτόκολλο Π, κατασκευάσουμε υποπρωτόκολλα $\Pi_1, \Pi_2, \cdots, \Pi_n$ τα οποία και αποδεικνύουμε ασφαλή. Στην συνέχεια αποδεικνύουμε το πρωτόκολλο Π ασφαλές,με τις κλήσεις Π_i να θεωρούνται ιδανικές. Το θεώρημα σύνθεσης, αποδεικνύει ότι και το Π είναι ασφαλές.

9.7 Ασκήσεις

1. Θεωρήστε το παρακάτω σχήμα δέσμευσης:

   - Δημόσιες τιμές: ένας μεγάλος πρώτος p, και δύο στοιχεία g, h στο $\mathbb{Z}_p^*$ τάξης q με q πρώτο.

   - Δέσμευση: η Αλίκη δεσμεύεται σε μια τιμή $x \in [0, q-1]$ επιλέγοντας τυχαίο $r \in [0, q-1]$ και υπολογίζοντας $b = g^x h^r \pmod p$, το οποίο στέλνει στον Βασίλη.

   - Αποκάλυψη: για να αποκαλύψει αργότερα την τιμή x, με τρόπο που να μην επιδέχεται αλλοίωσή της, η Αλίκη στέλνει στον Βασίλη το ζεύγος (x, r). Ο Βασίλης επαληθεύει ότι $b = g^x h^r \pmod p$.

   Αποδείξτε ότι το σχήμα αυτό είναι ασφαλές και δεσμευτικό. Για την ασφάλεια, δείξτε ότι ο Βασίλης δεν μπορεί να εξαγάγει από το b καμία πληροφορία για το x, δηλαδή ότι θα μπορούσε να είναι οποιαδήποτε τιμή στο $[0, q-1]$. Για να δείξετε ότι είναι δεσμευτικό το σχήμα, αποδείξτε ότι αν μπορούσε να βρει η Αλίκη ένα ζεύγος (x', r') τέτοιο ώστε $b = g^{x'} h^{r'} \pmod p$, με $x \neq x'$, τότε θα μπορούσε να υπολογίσει τον διακριτό λογάριθμο του h ως προς g (modulo p).

2. Θεωρήστε το εξής απλό σχήμα για διαμοιρασμό απορρήτου (secret sharing): η 'μάνα' επιλέγει $t-1$ μυστικές τιμές από το $\mathbb{Z}_m$, έστω $y_1, \ldots, y_{t-1}$, και τις δίνει στους 'παίκτες' $P_1, \ldots, P_{t-1}$. Στον παίκτη P_t στέλνει την τιμή

   $$y_t = K - \sum_{1}^{t-1} y_i \bmod m$$

   όπου K είναι η απόρρητη πληροφορία (π.χ. ένα κλειδί).

   Εξετάστε αν το σύνολο $\{y_1, \ldots, y_t\}$ είναι ένα (t, t)-σχήμα κατωφλίου.

3. Στο 9.4.5 ο S χρησιμοποιεί τη μη συνειδητή μεταφορά $OT_1^2(S, R, D_0^R, D_1^R)$. Δεν θα ήταν πιο απλό να αποστείλει τόσο το D_0^R όσο και το D_1^R Περιγράψτε τι προβλήματα συναντά αυτή η εναλλακτική λύση.

4. Να αποδείξετε τις ομομορφικές ιδιότητες του κρυπτοσυστήματος (9.5) Damgård-Jurik.

5. Να σχεδιάσετε ένα πρωτόκολλο το οποίο θα επιτρέπει σε n συμμετέχοντες με είσοδο x_i να υπολογίσουν το άθροισμα των εισόδων τους $\sum_{i=1}^{n} x_i$. Κάθε συμμετέχων θα πρέπει να μάθει μόνο το άθροισμα και καμία άλλη είσοδο. Το πρωτόκολλο θα πρέπει να είναι ασφαλές ενάντια σε αντίπαλο που ελέγχει ένα το πολύ παίκτη.

6. Οι χρήστες $A_1, \ldots, A_n$ ενός δικτύου θέλουν να φτιάξουν ένα κοινό κλειδί με τη βοήθεια μιας έμπιστης αρχής T. Όλοι θα πρέπει να είναι σε θέση να υπολογίσουν αυτό το κλειδί, αλλά για κάποιον υποκλοπέα θα πρέπει να είναι δύσκολο να το υπολογίσει.

   Για να πετύχουν το στόχο τους χρησιμοποιούν την εξής παραλλαγή του Diffie - Hellman: Έχουν για δημόσιο κλειδί έναν πρώτο αριθμό p και ένα στοιχείο $g \in Z_p$ τάξης q με q πρώτο και $q \mid (p-1)$. Η αρχή T διαλέγει έναν κρυφό τυχαίο αριθμό $t \in [1, \ldots, q-1]$ και υπολογίζει το $K = g^t \pmod p$. Κάθε χρήστης A_i διαλέγει έναν κρυφό τυχαίο αριθμό $a_i \in [1, \ldots, q-1]$ και υπολογίζει το $x_i = g^{a_i} \pmod p$. Μετά ο A_i στέλνει το x_i στην T, που του απαντάει στέλνοντάς του το $z_i = x_i^t$.

   (α) Περιγράψτε τι πρέπει να κάνει ο χρήστης A_i για να υπολογίσει το K.

   (β) Δείξτε ότι το πρωτόκολλο είναι ασφαλές κάτω από την υπόθεση Diffie-Hellman. Δηλαδή, δείξτε ότι ένας αλγόριθμος που με είσοδο x_i, z_i μπορεί να υπολογίσει το K, μπορεί να χρησιμοποιηθεί για την επίλυση του Decisional Diffie-Hellman προβλήματος.

9.8 Ηλεκτρονικό Υλικό

- Yuval Ishai, Secure Multiparty Computation I

- 5ο Χειμερινό Σχολείο για Ασφαλή Υπολογισμό Πολλών Συμμετεχόντων. Διαφάνειες και Βιντεοσκοπημένες Διαλέξεις.

Βιβλιογραφία

[1] Loren M Kohnfelder. *Towards a practical public-key cryptosystem*. PhD thesis, Massachusetts Institute of Technology, 1978.

[2] INTERNATIONAL TELECOMMUNICATION UNION ITU. The Directory and Authentication Framework. Series X: Data Communication Networks: Directory 8, International Telecommunication Union, SuÃÃ§a, Genebra, nov 1988. URL . Reedition of CCITT Recommendation X.509 published in the Blue Book, Fascicle VIII.8 (1988).

[3] Aggelos Kiayias. Cryptography primitives and protocols, 2015. Διαθέσιμο στο http://crypto.di.uoa.gr/class/Kryptographia/Semeioseis_files/Cryptograph_Primitives_and_Protocols.pdf.

[4] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, November 1979. ISSN 0001-0782. doi: 10.1145/359168.359176. URL http://doi.acm.org/10.1145/359168.359176.

[5] Berry Schoenmakers. A simple publicly verifiable secret sharing scheme and its application to electronic voting. In *In CRYPTO*, pages 148–164. Springer-Verlag, 1999.

[6] Paul Feldman. A practical scheme for non-interactive verifiable secret sharing. In *Proceedings of the 28th Annual Symposium on Foundations of Computer Science*, SFCS '87, pages 427–438, Washington, DC, USA, 1987. IEEE Computer Society. ISBN 0-8186-0807-2. doi: 10.1109/SFCS.1987.4. URL http://dx.doi.org/10.1109/SFCS.1987.4.

[7] Torben Pryds Pedersen. A threshold cryptosystem without a trusted party. In *Proceedings of the 10th annual international conference on Theory and application of cryptographic techniques*, EUROCRYPT'91, pages 522–526, Berlin, Heidelberg, 1991. Springer-Verlag. ISBN 3-540-54620-0. URL http://dl.acm.org/citation.cfm?id=1754868.1754929.

[8] Andrew C. Yao. Protocols for Secure Computations. In *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, SFCS '82, pages 160–164, Washington, DC, USA, 1982. IEEE Computer Society. doi: 10.1109/SFCS.1982.88. URL http://dx.doi.org/10.1109/SFCS.1982.88.

[9] Michael O. Rabin. How To Exchange Secrets with Oblivious Transfer. URL https://eprint.iacr.org/2005/187.pdf.

[10] Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. *Communications of the ACM*, 28(6):637–647, 1985. URL http://dl.acm.org/citation.cfm?id=3818.

[11] Claude Crepeau. Equivalence Between Two Flavours of Oblivious Transfers. In *A Conference on the Theory and Applications of Cryptographic Techniques on Advances in Cryptology*, CRYPTO '87, pages 350–354, London, UK, UK, 1988. Springer-Verlag. ISBN 3-540-18796-0. URL http://dl.acm.org/citation.cfm?id=646752.704744.

[12] Andrew Yao. How to generate and exchange secrets. In *Foundations of Computer Science, 1986., 27th Annual Symposium on*, pages 162–167. IEEE, 1986. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4568207.

[13] Ben Adida and Douglas Wikstrom. How to shuffle in public. In *Theory of Cryptography*, pages 555–574. Springer, 2007.

[14] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*, pages 136–145. IEEE, 2001.

Κεφάλαιο 10

Αποδείξεις Μηδενικής Γνώσης

10.1 Εισαγωγή

Οι αποδείξεις μηδενικής γνώσης (Zero Knowledge Proofs) προτάθηκαν στη δεκαετία του 1980 από τους Shaffi Goldwasser, Silvio Micali και Charles Rackoff [7] και αποτέλεσαν μία έννοια η οποία έδωσε πολλές εφαρμογές, τόσο θεωρητικές όσο και πρακτικές, λόγος για τον οποίο οι δύο πρώτοι συγγραφείς βραβεύθηκαν με το βραβείο Turing (2013). Προτάθηκαν ως μία παραλλαγή των διαλογικών συστημάτων αποδείξεων (interactive proof systems), στα οποία ένας υπολογισμός υλοποιείται με ανταλλαγή μηνυμάτων μεταξύ μιας οντότητας η οποία ονομάζεται Prover (αποδείκτης σμβ. με $\mathcal{P}$) και μιας οντότητας η οποία ονομάζεται Verifier (επαληθευτής σμβ.$\mathcal{V}$). Τυπικά, ο $\mathcal{P}$ θέλει να πείσει τον $\mathcal{V}$ ότι μία συμβολοσειρά (witness) ανήκει σε μία γλώσσα, ή ισοδύναμα ότι μία πρόταση είναι αληθής. Ο $\mathcal{P}$ και ο $\mathcal{V}$ νοούνται ως πιθανοτικές μηχανές Turing. Συνήθως ο $\mathcal{P}$ έχει απεριόριστη υπολογιστική ισχύ, ενώ ο $\mathcal{V}$ περιορίζεται σε πιθανοτικούς υπολογισμούς πολυωνυμικής πολυπλοκότητα (PPT), όπως είδαμε και στο 3.7.

Σε ένα οποιοδήποτε σύστημα αποδείξεων είναι επιθυμητές οι εξής δύο ιδιότητες:

- **Πληρότητα** Όλες οι αληθείς προτάσεις μπορούν να αποδειχθούν. Σε ένα αλληλεπιδραστικό σύστημα επομένως, ένας 'τίμιος' $\mathcal{P}$, (που όντως κατέχει μία συμβολοσειρά που ανήκει σε μία γλώσσα) πείθει ένα τίμιο $\mathcal{V}$ (που δηλαδή ακολουθεί ακριβώς το πρωτόκολλο) με πολύ μεγάλη πιθανότητα.
- **Ορθότητα** Οι ψευδείς προτάσεις δεν μπορούν να αποδειχθούν. Δηλ. ένας κακόβουλος $\mathcal{P}$, που προσπαθεί να αποδείξει μία ψευδή πρόταση, δεν μπορεί να πείσει ένα τίμιο $\mathcal{V}$, παρά με αμελητέα πιθανότητα.

Οι Goldwasser, Micali και Rackoff ασχολήθηκαν με το πόσο πληροφορία διαρρέει ένα τέτοιο σύστημα. Με άλλα λόγια, διερεύνησαν τι επιπλέον μαθαίνει ο $\mathcal{V}$ πέρα

από το γεγονός ότι ο ισχυρισμός του $\mathcal{P}$ είναι έγκυρος. Έτσι προσέθεσαν μία τρίτη ιδιότητα στα διαλογικά συστήματα

- **Μηδενική Γνώση** Ο $\mathcal{V}$ δεν μαθαίνει τίποτε παραπάνω από το γεγονός ότι ο ισχυρισμός του $\mathcal{P}$ είναι αληθής.

Κομβικό ρόλο στην απόδειξη του ότι ένα διαλογικό σύστημα έχει την ιδιότητα της μηδενικής γνώσης διαδραματίζει ο Simulator ($\mathcal{S}$), ο οποίος προσομοιώνει τον $\mathcal{P}$, χωρίς όμως να έχει πρόσβαση στον witness. Η συνεισφορά του είναι η εξής: Ο $\mathcal{V}$ αλληλεπιδρά με τον $\mathcal{S}$. Κάποια στιγμή ο $\mathcal{V}$ θα φέρει τον $\mathcal{S}$ στην 'δύσκολη θέση' να μην μπορεί να απαντήσει ένα ερώτημα, καθώς δεν έχει πρόσβαση στον witness. Σε αυτή την περίπτωση επαναφέρουμε την ταινία του $\mathcal{V}$ σε μία κατάσταση πριν την δυσάρεστη ερώτηση (rewind) και τρέχουμε το πρωτόκολλο από εκείνο το σημείο και μετά. Αν τελικά ο $\mathcal{V}$ (με συνεχή rewinds) αποδεχθεί την απόδειξη του $\mathcal{S}$, το πρωτόκολλο κατέχει την ιδιότητα της μηδενικής γνώσης, καθώς ο $\mathcal{V}$ δεν μπορεί να ξεχωρίσει έναν $\mathcal{P}$ που γνωρίζει τον witness και έναν $\mathcal{S}$ που υποκρίνεται. Ο $\mathcal{V}$ δηλαδή δεν εξάγει καμία επιπλέον πληροφορία από το πρωτόκολλο (αφού στην δεύτερη περίπτωση δεν υπάρχει πληροφορία να εξαχθεί).

10.1.1 Τυπικός Ορισμός και Παραλλαγές

Μετά την διαισθητική περιγραφή των ιδιοτήτων των αλληλεπιδραστικών συστημάτων αποδείξεων θα προχωρήσουμε στον αυστηρό ορισμό τους [1].

Ορισμός 10.1. Έστω μία NP γλώσσα $\mathcal{L}$ και $\mathcal{M}$ μία πολυωνυμική μηχανή Turing τέτοια ώστε: $x \in \mathcal{L} \Leftrightarrow \exists w \in \{0,1\}^{p(|x|)} : M(x,w) = 1$, όπου p είναι ένα πολυώνυμο.
Μία απόδειξη μηδενικής γνώσης για την $\mathcal{L}$ είναι δύο πιθανοτικές μηχανές Turing πολυωνυμικού χρόνου (PPT) $\mathcal{P}$, $\mathcal{V}$ για τις οποίες ισχύουν οι παρακάτω τρεις ιδιότητες:

- **Πληρότητα:**. Αν $x \in \mathcal{L}$ και w ένας μάρτυρας γι' αυτό (δηλ. $M(x,w) = 1$) τότε $Pr[out_{\mathcal{V}} < \mathcal{P}(x,w), \mathcal{V}(x) > (x)] = 1 \geq \frac{2}{3}$, όπου:
  - $\mathcal{P}(x,w), \mathcal{V}(x)$ είναι η αλληλεπίδραση μεταξύ των $\mathcal{P}$, $\mathcal{V}$ με κοινή (δημόσια είσοδο) το x και ιδιωτική είσοδο του $\mathcal{P}$ το w.
  - $out_{\mathcal{V}}$ είναι η έξοδος του $\mathcal{V}$ στο τέλος του πρωτοκόλλου.
- **Ορθότητα:** Αν $x \notin \mathcal{L}$ τότε $\forall(\mathcal{P}^*, w)\ Pr[out_{\mathcal{V}} < \mathcal{P}^*(x,w), \mathcal{V}(x) > (x)] = 1 \leq \frac{2}{3}$. Ο $\mathcal{P}^*$ δεν χρειάζεται να είναι PPT.

- **(Τέλεια) Μηδενική Γνώση:** Για κάθε PPT $\mathcal{V}^*$ υπάρχει μία PPT $\mathcal{S}$ τέτοια ώστε $\forall x \in \mathcal{L}$ με αντίστοιχα w για τα οποία $M(x,w) = 1$ να ισχύει ότι οι τυχαίες μεταβλητές $out_{\mathcal{V}*} < \mathcal{P}(x,w), \mathcal{V}*(x) > (x)$ και $S(x)$ να ακολουθούν ακριβώς την ίδια κατανομή.

Παρατηρήσεις Στον παραπάνω ορισμό φαίνεται ότι οι αποδείξεις μηδενικής γνώσης έχουν σχέση με την κλάση πολυπλοκότητας NP. Η σχέση αυτή θα γίνει εμφανής με τα παραδείγματα που ακολουθούν.

Για να είναι σαφής η διάκριση x, w θα αναφέρουμε ένα παράδειγμα: Στο πρόβλημα του διακριτού λογαρίθμου, αν $b = g^a$ το b παίζει το ρόλο του x ενώ το a είναι ο μάρτυρας w. Η έννοια της τέλειας μηδενικής γνώσης μπορεί να χαλαρώσει με τις εξής παραλλαγές:

- **Statistical Zero Knowledge** Οι κατανομές $out_{\mathcal{V}*} < \mathcal{P}(x,w), \mathcal{V}*(x) > (x)$ και $S(x)$ έχουν αμελητέα στατιστική απόσταση.

- **Computational Zero Knowledge** Οι κατανομές $out_{\mathcal{V}*} < \mathcal{P}(x,w), \mathcal{V}*(x) > (x)$ και $S(x)$ δεν μπορούν να διαχωριστούν από κάποιον αντίπαλο με πολυωνυμική υπολογιστή ισχύ.

- **Honest Verifier Zero Knowledge** Ο $\mathcal{V}$ είναι τίμιος, ακολουθεί το πρωτόκολλο και τα μηνύματα του προέρχονται από την ομοιόμορφη κατανομή.

10.2 Αποδείξεις μηδενικής γνώσης και πολυπλοκότητα

Οι αποδείξεις μηδενικής γνώσης έχουν τις ρίζες τους στη Θεωρία Πολυπλοκότητας. Προτού λοιπόν ασχοληθούμε με τις κρυπτογραφικές εφαρμογές τους θα δώσουμε δύο παραδείγματα στα οποία φαίνεται η δύναμη τους σαν υπολογιστικό μοντέλο [6].

10.2.1 Ισομορφισμός Γραφημάτων

Δύο γραφήματα G_1, G_2 λέγονται ισόμορφα αν έχουν το ίδιο πλήθος κορυφών και υπάρχει μετάθεση, δηλαδή συνάρτηση 1-1 και επί, μεταξύ των κόμβων τους τέτοια ώστε δύο κόμβοι του ενός να συνδέονται με ακμή αν και μόνο αν οι αντίστοιχοι κόμβοι του άλλου συνδέονται με ακμή. Ισοδύναμα, υπάρχει μετονομασία των κόμβων του ενός γράφου τέτοια ώστε οι γράφοι να ταυτίζονται. Το πρόβλημα του

ισομορφισμού των γραφημάτων ανήκει στην κλάση NP, αλλά δεν είναι γνωστό εάν είναι NP-complete ή όχι.

Υποθέτουμε ότι τόσο ο $\mathcal{P}$ όσο και ο $\mathcal{V}$ γνωρίζουν τα γραφήματα G_1, G_2, δηλ. τα τελευταία αποτελούν κοινή είσοδο του πρωτοκόλλου. Επιπλέον ο $\mathcal{P}$ γνωρίζει και τον μεταξύ τους ισομορφισμό $\phi : G_1 \to G_2$ (ιδιωτική είσοδος του $\mathcal{V}$ ή ο witness που προαναφέραμε). Χρησιμοποιώντας ένα πρωτόκολλο μηδενικής γνώσης μπορεί να αποδείξει ότι γνωρίζει τον ισομορφισμό χωρίς να τον αποκαλύψει.

1. Ο $\mathcal{P}$ επιλέγει τυχαία έναν από τους G_1, G_2, έστω G_i. Με κάποια μετάθεση ψ των κορυφών του G_i, ο P παράγει το γράφημα $H = \psi(G_i)$, που είναι ισόμορφο με το G_i. Επειδή ο $\mathcal{P}$ ξέρει τον ισομορφισμό ψ μεταξύ των H, G_i, ξέρει και τον ισομορφισμό $\psi\dot{\phi}$ μεταξύ των H, G_{3-i}. Οποιοσδήποτε άλλος δυσκολεύεται εξίσου να βρει έναν ισομορφισμό μεταξύ των H, G_1 ή μεταξύ των H, G_2, όσο και να βρει έναν ισομορφισμό μεταξύ των αρχικών G_1, G_2.

2. Ο $\mathcal{P}$ **δεσμεύεται** στον ψ, στέλνοντας το H στον $\mathcal{V}$.

3. Ο $\mathcal{V}$ επιλέγει τυχαία ένα γράφημα από τα G_1, G_2, έστω G_j και στέλνει την επιλογή του, ως **πρόκληση** στον $\mathcal{P}$ ζητώντας του να αποδείξει ότι οι H και G_j είναι ισόμορφοι. Ζητάει δηλαδή μια μετάθεση του G_j που να παράγει τον H.

4. Ο P **απαντά**, κάνοντας τα εξής:

   (Î±Í΄) αν $G_i = G_j$, στέλνει στον $\mathcal{V}$ την μετάθεση ψ.

   (Î²Í΄) αν $G_i \neq G_j$ τότε:

      i. αν τα G_1, G_2 είναι ισόμορφα (οπότε $\exists\rho : G_i = \rho(G_j)$), στέλνει στον $\mathcal{V}$ την μετάθεση $\psi\dot{\rho}$.

      ii. αν τα G_1 και G_2 δεν είναι ισόμορφα (δηλ. ο $\mathcal{P}$ δεν είναι τίμιος) τότε δεν μπορεί να βρει κατάλληλη μετάθεση και στέλνει μια οποιαδήποτε τυχαία μετάθεση.

5. Αν ο $\mathcal{V}$ λάβει μια σωστή μετάθεση συνεχίζει (επανάληψη των βημάτων 1-5), αλλιώς σταματάει απορρίπτοντας (θεωρεί δηλαδή ότι τα γραφήματα δεν είναι ισόμορφα).

Αν ο $\mathcal{V}$ δεν έχει απορρίψει μετά από k επαναλήψεις των βημάτων 1-5 τότε αποδέχεται (θεωρεί ότι τα γραφήματα είναι ισόμορφα).

Το παραπάνω πρωτόκολλο πληρεί τις ιδιότητες της μηδενικής γνώσης που προαναφέραμε. Καταρχήν είναι πλήρες γιατί, αν υπάρχει ισομορφισμός μεταξύ των G_1

και G_2, τότε ο $\mathcal{P}$ θα πείσει τον $\mathcal{V}$ με πιθανότητα 1 (ο $\mathcal{V}$ δεν απορρίπτει ποτέ). Σχετικά με την ορθότητα αν δεν υπάρχει ο ισομορφισμός, τότε ο $\mathcal{P}$ έχει πιθανότητα 1/2 σε κάθε βήμα να εξαπατήσει τον $\mathcal{V}$ (αυτό θα συμβεί μόνο αν $G_i = G_j$). Μετά από k επαναλήψεις η πιθανότητα αυτή γίνεται $1/(2^k)$. Σχετικά με τη μηδενική γνώση, ο $\mathcal{V}$ δεν παίρνει καμία επιπλέον πληροφορία όσον αφορά τον ισομορφισμό μεταξύ των G_1 και G_2. Αυτό συμβαίνει επειδή κατά την αλληλεπίδραση με τον $\mathcal{S}$ το πρώτο βήμα του θα είναι ακριβώς το ίδιο με τον $\mathcal{P}$, δηλαδή θα φτιάξει κάθε φορά ένα καινούριο τυχαίο γράφημα ισόμορφο με κάποιον από τους G_1 και G_2. Η πιθανότητα επιλογής είτε του G_1 είτε G_2 είναι ακριβώς 1/2. Σε αυτή την φάση λοιπόν ο $\mathcal{V}$ δεν μπορεί να τους διαχωρίσει. Έτσι η πιθανότητα εξαπάτησης σε k επαναλήψεις παραμένει $1/(2^k)$. Άρα ο αναμενόμενος χρόνος εκτέλεσης είναι πολυωνυμικός καθώς προκύπτει από την σχέση $T_{\mathcal{V}} \sum_{k=1}^{\infty} 1/(2^k) = T_{\mathcal{V}}$ όπου $T_{\mathcal{V}}$ ο χρόνος εκτέλεσης του $\mathcal{V}$ ο οποίος είναι πολυωνυμικός.

10.2.2 3-Χρωματισμός

Ένα πρωτόκολλο μηδενικής γνώσης για ένα NP-Complete πρόβλημα, θα σήμαινε ότι όλα τα NP προβλήματα έχουν πρωτόκολλα μηδενικής γνώσης. Στο [6] οι Micali, Goldreich, Wigderson έδωσαν ένα τέτοιο πρωτόκολλο για το NP-Complete πρόβλημα του 3-Χρωματισμού.

Σε αυτό ο $\mathcal{P}$ γνωρίζει ένα χρωματισμό c για ένα γράφημα $G = (V, E)$ τέτοιο ώστε $c : V \to \{1, 2, 3\}$ και $c(v_1) \neq c(v_2) \Leftrightarrow (v_1, v_2) \in E$. Θέλει να αποδείξει τη γνώση αυτή στον $\mathcal{V}$ χωρίς να αποκαλύψει το c.

- Ο $\mathcal{P}$ επιλέγει μια τυχαία μετάθεση π του $\{1, 2, 3\}$. Από αυτήν προκύπτει ένας εναλλακτικός 3 - χρωματισμός $\pi.c$ του G. Στην συνέχεια χρησιμοποιεί ένα σχήμα δέσμευσης για το $\pi.c$, δηλαδή υπολογίζει τις τιμές $commit((\pi.c)(v_i), r_i), \forall v_i \in V$ και τις στέλνει στον $\mathcal{V}$.
- Ο $\mathcal{V}$ επιλέγει μία τυχαία ακμή $(v_i, v_j) \in E$ και την στέλνει στον $\mathcal{P}$.
- Ο $\mathcal{P}$ αποδεσμεύει τις τιμές $\pi.c(v_i), \pi.c(v_j)$ και τις στέλνει στο στον $\mathcal{V}$.
- Ο $\mathcal{V}$ ελέγχει αν $\pi.c(v_i) \neq \pi.c(v_j)$

Είναι προφανές ότι το παραπάνω πρωτόκολλο είναι πλήρες. Σχετικά με την ορθότητα, παρατηρούμε ότι αν ο $\mathcal{P}$ δεν διαθέτει έναν έγκυρο 3-χρωματισμό, τότε ο $\mathcal{V}$ θα διαλέξει μία ακμή με ίδια χρώματα κορυφών με πιθανότητα $1/|E|$. Με επανάληψη του πρωτοκόλλου μπορούμε να κάνουμε την πιθανότητα να τον ξεγελάσει ο prover $1 - \frac{1}{|E|}$ εξαιρετικά μικρή. Σχετικά με τη μηδενική γνώση, έστω ο $\mathcal{S}$ που δεν διαθέτει έναν έγκυρο χρωματισμό. Σε περίπτωση που ο $\mathcal{V}$ θα διαλέγει μία ακμή με

ίδια χρώματα κορυφών, τότε γίνεται rewind σε μια προηγούμενη κατάσταση και ο $\mathcal{S}$ επιλέγει μια νέα τυχαία μετάθεση π' την οποία χρησιμοποιεί στη νέα εκτέλεση. Μπορεί να αποδειχθεί ότι το πρωτόκολλο με τον $\mathcal{S}$ δεν έχει αναμενόμενο χρόνο εκτέλεσης διαφορετικής τάξης μεγέθους από ότι με τον $\mathcal{P}$, ο $\mathcal{V}$ δεν καταλαβαίνει διαφορά. Άρα το πρωτόκολλο έχει την ιδιότητα της μηδενικής γνώσης. Μία διαδραστική επίδειξη του παραπάνω πρωτοκόλλου μπορεί να βρεθεί στο [9].

10.3 Σ-Πρωτόκολλα

Ένα πρωτόκολλο 3 γύρων με honest verifier ονομάζεται και Σ-Πρωτόκολλο. Οι 3 γύροι συνήθως είναι:

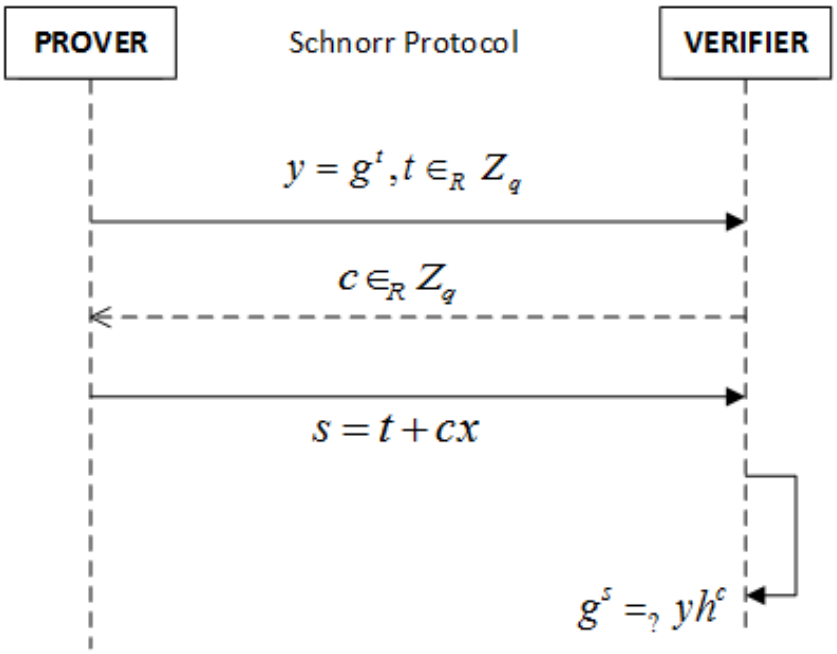
1. **Commit** Ο $\mathcal{P}$ δεσμεύεται σε μία τιμή.
2. **Challenge** Ο $\mathcal{V}$ διαλέγει μία τυχαία πρόκληση. Εφόσον είναι τίμιος θεωρούμε ότι η πιθανότητα επιλογής πρόκλησης είναι ομοιόμορφα κατανεμημένη.
3. **Response** Ο $\mathcal{P}$ απαντάει χρησιμοποιώντας τη δέσμευση, το μυστικό και την τυχαία τιμή.

Το ιδιαίτερο χαρακτηριστικό αυτών των πρωτοκόλλων είναι ότι είναι ιδιαίτερα εύκολο να αποδειχθεί μια ειδική μορφή ορθότητας η οποία ονομάζεται special soundness, στην οποία αν το πρωτόκολλο εκτελεστεί δύο φορές με την ίδια δέσμευση, μπορούμε να εξάγουμε τον witness. Μπορεί να αποδειχθεί ότι η ειδική ορθότητα αυτή ισοδυναμεί με επιλογή πρόκλησης από τον $\mathcal{V}$ με ομοιόμορφη κατανομή.

10.3.1 Το πρωτόκολλο του Schnorr

Το πρωτόκολλο θεωρεί ότι τόσο ο $\mathcal{P}$ όσο και ο $\mathcal{V}$ γνωρίζουν ένα γεννήτορα g μιας ομάδας τάξης q. Ο $\mathcal{P}$ έχει ένα witness x ώστε $h = g^x$ και θέλει να το αποδείξει χωρίς να αποκαλύψει x. Η διαδικασία προχωράει ως εξής:

- **Commit ($\mathcal{P} \to \mathcal{V}$):** Τυχαία επιλογή $t \in_R \mathbb{Z}_q$, και υπολογισμός $y = g^t$. Το y αποστέλλεται στον $\mathcal{V}$.
- **Challenge ($\mathcal{V} \to \mathcal{P}$):** Ο $\mathcal{V}$ διαλέγει τυχαία $c \in_R \mathbb{Z}_q$ και το στέλνει στον $\mathcal{P}$.
- **Response ($\mathcal{P} \to \mathcal{V}$):** Ο $\mathcal{P}$ υπολογίζει το $s = t + cx \pmod q$ και το στέλνει στον $\mathcal{V}$



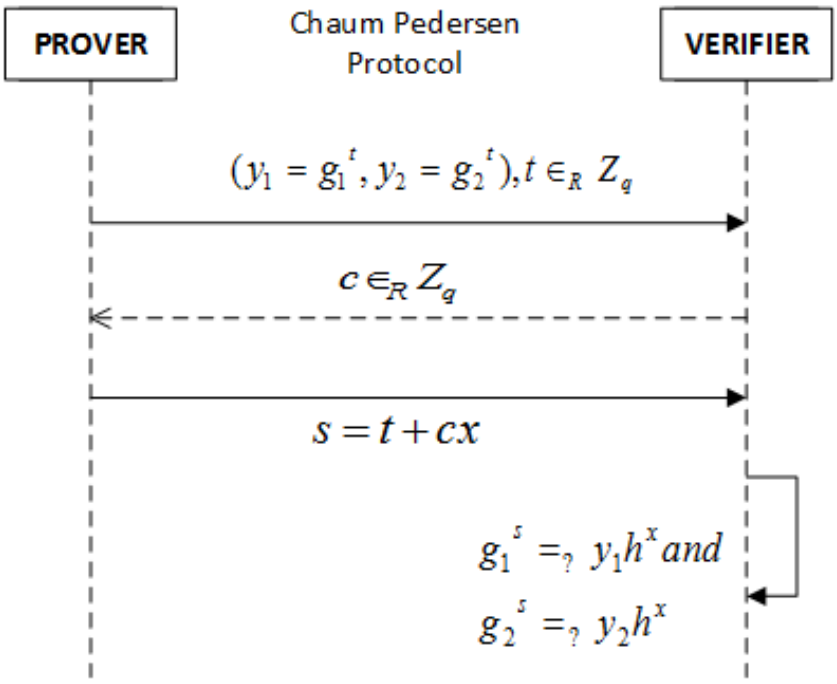
Σχήμα 10.1: Πρωτόκολλο Schnorr

- Ο $\mathcal{V}$ αποδέχεται αν $g^s = yh^c$

Η πληρότητα του πρωτοκόλλου μπορεί να αποδειχθεί με απλή αντικατάσταση. Πράγματι: $g^s = g^{t+cx} = g^t g^{cx} = yh^c$ που είναι και τιμές γνωστές στον $\mathcal{V}$. Για την ορθότητα του πρωτοκόλλου, παρατηρούμε ότι ένας $\mathcal{S}$ που δεν γνωρίζει το x μπορεί να το εκτελέσει με επιτυχία με πιθανότητα 1/q, η οποία είναι αμελητέα, ως εξής:

- Αρχικά διαλέγει $c' \in_R \mathbb{Z}_q$
- Στην συνέχεια επιλέγει $t \in_R \mathbb{Z}_q$ και δεσμεύεται στο $y = g^t h^{-c'}$
- Αν ο $\mathcal{V}$ επιλέξει $c = c'$ τότε ο ο $\mathcal{P}$ θέτει $s = t$. Παρατηρούμε ότι ο θα δεχτεί αφού $yh^{c'} = g^t h^{-c'} h^{c'} = g^t = g^s$, αλλά αυτό μπορεί να συμβεί με πιθανότητα ακριβώς 1/q.

Για τον ίδιο λόγο το πρωτόκολλο δεν μπορεί να αποδειχθεί η ιδιότητα της μηδενικής γνώσης. Για να επιτύχει ένας $\mathcal{S}$ πρέπει να σταματήσει και να επανεκκινήσει τον $\mathcal{V}$ κατά μέσο όρο q φορές, δηλ. εκθετικό αριθμό φορών.



Σχήμα 10.2: Πρωτόκολλο Chaum Pedersen

10.3.2 Το πρωτόκολλο Chaum Pedersen

Για να δείξουμε ότι δύο διακριτοί λογάριθμοι είναι ίδιοι μπορούμε να χρησιμοποιήσουμε την παρακάτω παραλλαγή του πρωτοκόλλου του Schnorr, η οποία οφείλεται στους Chaum και Pedersen [2].

Οι κοινοί είσοδοι του πρωτοκόλλου είναι γεννήτορες g_1, g_2 μιας ομάδας τάξης q. Ο $\mathcal{P}$ θέλει να αποδείξει ότι γνωρίζει x τέτοιο ώστε $h_1 = g_1^x$ $h_2 = g_2^x$ χωρίς φυσικά να το αποκαλύψει. Ακολουθεί τους τρεις παρακάτω γύρους:

- **Commit:** Ο $\mathcal{P}$ διαλέγει $t \in_R \mathbb{Z}_q$, και υπολογίζει $y_1 = g_1^t, y_2 = g_2^t$ τα οποία και στέλνει στον $\mathcal{V}$
- **Challenge:** Ο $\mathcal{V}$ διαλέγει $c \in_R \mathbb{Z}_q$
- **Response:** Ο $\mathcal{P}$ υπολογίζει $s = t + cx \pmod q$ και το στέλνει στον $\mathcal{V}$
- Ο $\mathcal{V}$ δέχεται αν $g_1^s = y_1 h_1^c$ και $g_2^s = y_2 h_2^c$

Η απόδειξη των ιδιοτήτων του πρωτοκόλλου μπορεί να γίνει με τρόπο ανάλογο με το πρωτόκολλο του Schnorr.

Το πρωτόκολλο των Chaum-Pedersen μπορεί να χρησιμοποιηθεί για να δείξει ότι ένα ζεύγος c_1, c_2 είναι κρυπτογράφηση ElGamal ενός μηνύματος m. καθώς $(c_1, c_2) = (g^r, mh^r)$ που σημαίνει ότι $log_g c_1 = log_h(\frac{c_2}{m})$.

10.4 Witness Hiding and Witness Indistinguishable Protocols

Τα πρωτόκολλα των Schnorr και Chaum Pedersen που είδαμε νωρίτερα έχουν την ιδιότητα της μηδενικής γνώσης όταν ο $\mathcal{V}$ είναι τίμιος. Μία άλλη παραλλαγή η οποία παρουσιάζει ενδιαφέρον είναι τα πρωτόκολλα που δεν επιτρέπουν σε οποιονδήποτε επαληθευτή (είτε είναι τίμιος είτε όχι) να μάθει τον witness που έχει στη διάθεση του ο $\mathcal{P}$. Τέτοια πρωτόκολλα ονομάζονται **witness hiding**, όταν ο $\mathcal{V}$ δεν μπορεί να μάθει ολόκληρο τον μάρτυρα. Μία παραλλαγή τους είναι η **witness indistinguishability** κατά την οποία δεν διαρρέεται πληροφορία ικανή να ξεχωρίσει δύο ισοπίθανους μάρτυρες. Είναι φανερό πως ένα πρωτόκολλο μηδενικής γνώσης είναι και **witness hiding**, ενώ το αντίστροφο δεν ισχύει καθώς μπορεί να διαρρέονται κάποια μεμονωμένα τμήματα του μάρτυρα (bits).

Ένα τέτοιο πρωτόκολλο δόθηκε από τον Okamoto [8]. Έστω μία ομάδα G τάξης q, και τυχαία $g_1, g_2 \in G$. Έστω ότι $(x_1, x_2) \in zn_q$ ο witness και $h = g_1^{x_1} g_2^{x_2}$ η δημόσια είσοδος του πρωτοκόλλου.

- **Commit ($\mathcal{P} \rightarrow \mathcal{V}$):** Τυχαία επιλογή $t_1, t_2 \in_R \mathbb{Z}_q$, και υπολογισμός $y = g_1^{t_1} g_2^{t_2}$. Το y αποστέλλεται στον $\mathcal{V}$.
- **Challenge ($\mathcal{V} \rightarrow \mathcal{P}$):** Ο $\mathcal{V}$ διαλέγει τυχαία $c \in_R \mathbb{Z}_q$ και το στέλνει στον $\mathcal{P}$.
- **Response ($\mathcal{P} \rightarrow \mathcal{V}$):** Ο $\mathcal{P}$ υπολογίζει το $s_1 = t_1 + cx_1 \pmod q$ και $s_2 = t_2 + cx_2 \pmod q$ και στέλνει το ζεύγος (s_1, s_2) στον $\mathcal{V}$
- Ο $\mathcal{V}$ αποδέχεται αν $g_1^{s_1} g_2^{s_2} = yh^c$

Μπορεί με τρόπο ανάλογο με το πρωτόκολλο του Schnorr να αποδειχθεί ότι το παραπάνω πρωτόκολλο είναι Honest Verifier Zero Knowledge. Σχετικά με την ιδιότητα της απόκρυψης μάρτυρα, τώρα παρατηρούμε ότι υπάρχουν ακριβώς q μάρτυρες (x_1, x_2) οι οποίοι επαληθεύουν τη σχέση $h = g_1^{x_1} g_2^{x_2}$.

Αρχικά θα αποδείξουμε ότι το παραπάνω πρωτόκολλο είναι witness indistinguishable. Έστω (y, c, s_1, s_2) τα μηνύματα που ανταλλάσσουν οι $\mathcal{P}$, $\mathcal{V}$ με χρήση του μάρτυρα (x_1, x_2) από τον $\mathcal{P}$ και τυχαίων τιμών (t_1, t_2) στο πρώτο βήμα. Για έναν διαφορετικό μάρτυρα (x_1', x_2') υπάρχει μοναδικό ζεύγος διαφορετικών τιμών (t_1', t_2')

οι οποίες όμως δίνουν την ίδια συζήτηση. Αυτές είναι οι $t'_{1,2} = t_{1,2} + c(x_{1,2} - x'_{1,2})$. Πράγματι ισχύει:

$$y' = g_1^{t'_1} g_2^{t'_2} = g_1^{t_1 + cx_1 - cx'_1} g_2^{t_2 + cx_2 - cx'_2} = g_1^{t_1} g_2^{t_2} h^c / h^c = y$$

και

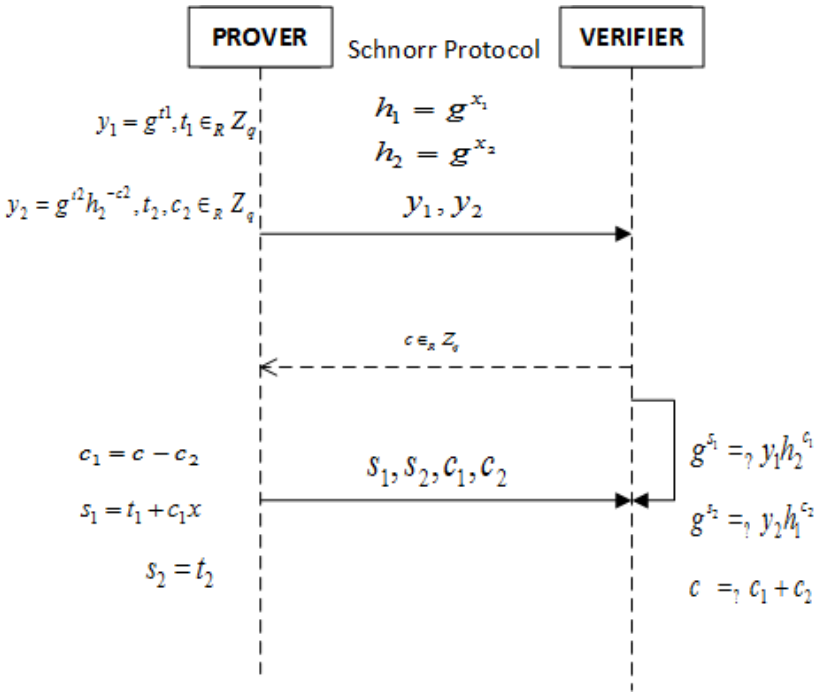
$$s'_{1,2} = t'_{1,2} + cx'_{1,2} = t_{1,2} + c(x_{1,2} - x'_{1,2}) + cx'_{1,2} = t_{1,2} + cx_{1,2} = s_{1,2}$$

Άρα οποιαδήποτε συζήτηση δεν μπορεί να οδηγήσει σε διάκρισή μεταξύ δύο μαρτύρων, αυτών που πραγματικά χρησιμοποιήθηκε. Έστω τώρα ότι ένας $\mathcal{V}$ καταφέρνει να εξάγει ένα μάρτυρα (x'_1, x'_2) μετά από ένα πολυωνυμικό αριθμό αλληλεπιδράσεων με τον $\mathcal{P}$. Τότε όμως θα σημαίνει ότι $h = g_1^{x_1} g_2^{x_2} = g_1^{x'_1} g_2^{x'_2}$ ή ισοδύναμα: $g_1^{x_1 - x'1} = g_2^{x'_2 - x_2}$ δηλαδή: $log_{g_1} g_2 = \frac{x_1 - x'1}{x'_2 - x_2}$ πράγμα που σημαίνει ότι μπορεί να λύσει το πρόβλημα του διακριτού λογαρίθμου για δύο τυχαία στοιχεία του G.

Ένας γενικότερος τρόπος κατασκευής witness indistinguishable πρωτοκόλλων από ένα πρωτόκολλο HVZK δόθηκε στο [3]. Η μέθοδος επιστρατεύει τον $\mathcal{S}$ και ένα σύστημα διαμοιρασμού μυστικού. Μια απλουστευμένη περιγραφή είναι η παρακάτω:

- Έστω $W = \{w_1, ..., w_n\}$ οι εναλλακτικοί μάρτυρες.
- Για αυτόν που κατέχει ο $\mathcal{P}$ ακολουθεί το πρωτόκολλο.
- Για τους υπόλοιπους ο $\mathcal{P}$ καλεί τον $\mathcal{S}$ ο οποίος υπολογίζει τις δεσμεύσεις που θα έκαναν τον $\mathcal{V}$ να δεχθεί σε μία προσομοιωμένη συζήτηση.
- Όλες οι δεσμεύσεις αποστέλλονται στον $\mathcal{V}$.
- Ο τελευταίος απαντάει με μία τυχαία πρόκληση.
- Ο $\mathcal{P}$ ερμηνεύει την πρόκληση ως ένα μυστικό που πρέπει να χωριστεί.
- Κάθε μερίδιο θα αντιστοιχεί σε μία απάντηση του $\mathcal{P}$ στο τρίτο βήμα του πρωτοκόλλου.
- Ο $\mathcal{V}$ αποδέχεται αν όλες τις απαντήσεις που έλαβε στο τελευταίο βήμα είναι έγκυρες.

Στο παρακάτω σχήμα φαίνεται η προσαρμογή της παραπάνω τεχνικής στο πρωτόκολλο του Schnorr:



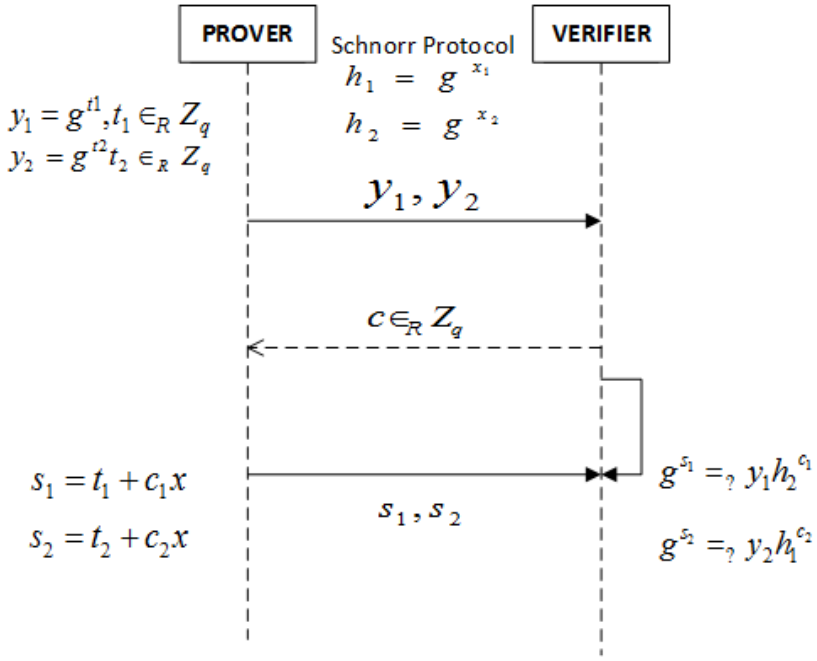
Σχήμα 10.3: Witness Hidding Schnorr

10.4.1 Σύνθεση Σ-Πρωτοκόλλων

Το παραπάνω σχήμα μπορεί να θεωρηθεί και μία *σύνθεση OR* δύο πρωτοκόλλων Schnorr στα οποία ο $\mathcal{P}$ γνωρίζει είτε τον ένα διακριτό λογάριθμο (οπότε για τον άλλο επικαλείται τον $\mathcal{S}$) ή και τους δύο. Μία τέτοια σύνθεση μπορεί να οριστεί μεταξύ δύο οποιονδήποτε Σ-Πρωτοκόλλων. Μπορεί να αποδειχθεί ότι το πρωτόκολλο που προκύπτει διατηρεί και αυτό τις ιδιότητες των Σ-Πρωτοκόλλων, δηλαδή Πληρότητα, Ειδική Ορθότητα και Μηδενική Γνώση για Τίμιο Επαληθευτή. Επιπλέον υπάρχουν και άλλα είδη σύνθεσης εκτός από την OR. Για παράδειγμα υπάρχει η **σύνθεση EQ** όπου ο $\mathcal{P}$ διαθέτει δύο witnesses για δύο σχέσεις και εκτός από το γεγονός αυτό πρέπει να αποδείξει ότι αυτή ταυτίζονται. Ένα τέτοιο παράδειγμα είναι το πρωτόκολλο Chaum - Pedersen 10.3.2. Ως ένα τελικό παράδειγμα στο παρακάτω σχήμα φαίνεται η σύνθεση AND δύο πρωτοκόλλων Schnorr (στην ουσία τρέχουμε παράλληλα τα δύο πρωτόκολλα με κοινή όμως πρόκληση.

10.5 Μη διαλογικές αποδείξεις

Τα πρωτόκολλα που περιγράφονται παραπάνω απαιτούν την ενεργό συμμετοχή του $\mathcal{V}$ και κατά συνέπεια είναι διαλογικά. Οι Fiat και Shamir πρότειναν στο [5]



Σχήμα 10.4: Συνθεση AND Schnorr

μία παραλλαγή στην οποία ο $\mathcal{P}$ μπορεί να παράγει την απόδειξη μόνος του, και η συζήτηση να μπορεί να επαληθευθεί από οποιονδήποτε. Η βασική ιδέα του μετασχηματισμού αυτού είναι η αντικατάσταση της τυχαίας πρόκλησης του $\mathcal{V}$ από το αποτέλεσμα μιας ψευδοτυχαίας συνάρτησης, στην οποία δίνεται ως είσοδος η δέσμευση (τουλάχιστον). Συνήθως το ρόλο της ψευδοτυχαίας συνάρτησης παίρνει μια κατάλληλα επιλεγμένη συνάρτηση σύνοψης.

Για παράδειγμα, η μη διαλογική έκδοση του πρωτοκόλλου του Schnorr είναι:

- Τυχαία επιλογή $t \in_R \mathbb{Z}_q$, και υπολογισμός $y = g^t$.
- Υπολογισμός $c = \mathcal{H}(y)$ όπου $\mathcal{H}$ είναι μια συνάρτηση σύνοψης που δίνει τιμές στο znq
- Υπολογισμός $s = t + cx \pmod{q}$
- Η 'συζήτηση' που δημοσιοποιείται είναι η τριάδα (h, c, s)
- Η επαλήθευση μπορεί να γίνει από οποιονδήποτε ελέγχοντας αν $c = \mathcal{H}(g^s h^{-c})$

Πρακτικά ο μετασχηματισμός των Fiat και Shamir μετατρέπει ένα σύστημα απόδειξης σε σύστημα υπογραφής. Συνήθως για να εξαρτάται η απόδειξη από τον $\mathcal{P}$ προστίθεται στο δεύτερο βήμα ως είσοδο στην $\mathcal{H}$ κάποια πληροφορία ταυτότητας.

10.6 Εφαρμογές

10.6.1 Σχήματα ταυτοποίησης (Identification Schemes)

Η πιο σημαντική εφαρμογή των πρωτοκόλλων μηδενικής γνώσης αφορά την διαδικασία ταυτοποίησης και αυθεντικοποίησης που προηγείται της χρήσης οποιασδήποτε υπηρεσίας. Ο συνηθισμένος τρόπος είναι με την χρήση συνθηματικών. Ο πάροχος της υπηρεσίας διατηρεί συνήθως την σύνοψη του συνθηματικού κάθε χρήστη. Κάθε φορά που ο χρήστης θέλει να συνδεθεί στην υπηρεσία, το συνθηματικό δίνεται στη συνάρτηση σύνοψης και το αποτέλεσμα συγκρίνεται με το αποθηκευμένο. Μπορεί μεν το πρωτόκολλο αυτό να μην επιτρέπει την αποθήκευση του συνθηματικού στην αρχική του μορφή, όμως ο εξυπηρετητής το μαθαίνει έστω και προσωρινά. Η διαδικασία αυτή θα μπορούσε να αντικατασταθεί με μία απόδειξη μηδενικής γνώσης που να δείχνει ότι ο κάθε πελάτης κατέχει το συνθηματικό.

10.6.2 Non-Malleable Cryptography – Το κρυπτοσύστημα Cramer-Shoup

Μία ακόμη σημαντική εφαρμογή που έχουν βρει οι αποδείξεις μηδενικής γνώσης αφορά στην κατασκευή ασφαλών κατά CCA κρυπτοσυστημάτων 1.4.4. Η βασική ιδέα είναι ότι στο κρυπτοκείμενο επισυνάπτεται και μία απόδειξη μηδενικής γνώσης για το μήνυμα που κρυπτογραφείται. Οποιαδήποτε αλλαγή στο κρυπτοκείμενο θα καταστήσει την απόδειξη και κατά συνέπεια την αποκρυπτογράφηση άκυρη. Ένα από τα πιο σημαντικά τέτοια κρυπτοσυστήματα είναι αυτό των Cramer και Shoup [4], το οποίο βασίζεται στο ElGamal. Έτσι διαλέγουμε μία ομάδα G πρώτης τάξης q, όπου ισχύει η DDH, δηλ. είναι δύσκολο να διακρίνουμε μεταξύ των (g^a, g^b, g^c) και (g^a, g^b, g^{ab}) όπου $a, b, c \in \mathbb{Z}_q$. Από την G, συμφωνούνται 2 τυχαία στοιχεία (γεννήτορες) g_1, g_2. Επιπλέον έχει συμφωνηθεί και μία συνάρτηση σύνοψης $\mathcal{H}$, η οποία έχει αντίσταση σε συγκρούσεις και δίνει στοιχεία στο $\mathbb{Z}_q$.

Το κρυπτοσύστημα Cramer - Shoup αποτελείται από την κλασική τριάδα αλγορίθμων $(\texttt{KeyGen}, \texttt{Encrypt}, \texttt{Decrypt})$ οι οποίοι λειτουργούν όπως παρακάτω:

`KeyGen`

- Το ιδιωτικό κλειδί είναι τα τυχαία στοιχεία $(x_1, x_2, y_1, y_2, z) \in_R \mathbb{Z}_q$
- Το δημόσιο κλειδί είναι η τριάδα $(A, B, C) = (g_1^{x_1} g_2^{x_2}, g_1^{y_1} g_2^{y_2}, g_1^z)$

`Encrypt`

- Μετασχηματίζεται το μήνυμα ώστε να ανήκει στο G.
- Επιλέγεται $r \in_R \mathbb{Z}_q$
- Το κρυπτοκείμενο είναι η τετράδα $(g_1^r, g_2^r, mC^r, A^r B^{r\alpha})$ όπου $\alpha = \mathcal{H}(g_1^r, g_2^r, mC^r)$

`Decrypt`

- Για μία τετράδα (u_1, u_2, e, v), υπολογίζουμε το $\alpha = \mathcal{H}(u_1, u_2, e)$
- Γίνεται ο έλεγχος αν $v = u_1^{x_1+\alpha y_1} u_2^{x_2+\alpha y_2}$
- Αν είναι επιτυχής τότε προχωρά η αποκρυπτογράφηση ως: $\frac{e}{u_1^z}$
- Σε διαφορετική περίπτωση η αποκρυπτογράφηση αποτυγχάνει.

Σε ένα σωστά κρυπτογραφημένο μήνυμα (u_1, u_2, e, v) θα ισχύει $\alpha = \mathcal{H}(u_1, u_2, e) = \mathcal{H}(g_1^r, g_2^r, mC^r)$. Επιπλέον:

$$\begin{aligned} u_1^{x_1+\alpha y_1} u_2^{x_2+\alpha y_2} &= u_1^{x_1} u_2^{x_2} u_1^{\alpha y_1} u_2^{\alpha y_2} = \\ g_1^{rx_1} g_2^{rx_2} g_1^{r\alpha y_1} g_2^{r\alpha y_2} &= \\ (g_1^{x_1} g_2^{x_2})^r (g_1^{y_1} g_2^{y_2})^r \alpha &= A^r B^{r\alpha} = v \end{aligned}$$

Άρα ο έλεγχος εγκυρότητας της αποκρυπτογράφησης θα περάσει και στην συνέχεια θα ισχύει: $\frac{e}{u_1^z} = \frac{mC^r}{g_1^r z} = \frac{mg_1 z^r}{g_1^{rz}} = m$

Πρακτικά τα στοιχεία των κλειδιών C, z και u_1, e στο κρυπτογράφημα παίζουν τον ίδιο ρόλο που έχουν και στο ElGamal. Το u_2, v μπορούν να θεωρηθούν ως μία απόδειξη μηδενικής γνώσης για να αποφευχθούν πιθανές επιθέσεις CCA. Για να αποδειχθεί αυτό θεωρούμε έναν αντίπαλο ο οποίος στο παιχνίδι CCA μπορεί να μαντέψει το b με μη αμελητέα πιθανότητα και τον χρησιμοποιούμε για να σπάσουμε την υπόθεση DDH [4].

10.7 Ασκήσεις

1. Ο P θέλει να αποδείξει στο V ότι γνωρίζει το αρχικό κείμενο m ενός RSA-κρυπτοκείμενου $c = m^e \bmod n$. Για το λόγο αυτό τρέχουν ένα πρωτόκολλο Zero Knowledge:

   (Î±Í΄) Ο P διαλέγει ένα τυχαίο r_1 με $(r_1, n) = 1$ και υπολογίζει το $r_2 = mr_1^{-1} \bmod n$.

(Î²Í΄) Ο P υπολογίζει τα $x_1 = r_1^e \bmod n$ και $x_2 = r_2^e \bmod n$ και τα στέλνει στο V.

(Î³Í΄) Ο V ελέγχει αν $x_1 x_2 \equiv c \bmod n$.

Συμπληρώστε τα βήματα που λείπουν, ώστε ο V να πείθεται με μεγάλη πιθανότητα ότι ο P δε λέει ψέματα. Προφανώς πρόκειται για ένα πρωτόκολλο Proof of Knowledge, δε χρειάζεται, όμως, να τα ορίσετε ούτε να δώσετε τυπική απόδειξη.

2. Θεωρήστε το ακόλουθο σχήμα πιστοποίησης ταυτότητας (identification scheme): η Αλίκη διαθέτει δύο μυστικούς πρώτους p και q και δημοσιοποιεί το $n = pq$. Επιλέγει επίσης μία τιμή $t \in U(\mathbb{Z}_n)$ και δημοσιοποιεί το $t = s^2 \bmod n$.

   Όποτε η Αλίκη θέλει να πιστοποιήσει τον εαυτό της στον Βασίλη, του ζητάει να υπολογίσει ένα ένα τυχαίο τετραγωνικό υπόλοιπο $x = r^2 \bmod n$ και να της στείλει το $c = xt \bmod n$.

   Στη συνέχεια, η Αλίκη υπολογίζει μια τετραγωνική ρίζα του $c \pmod n$ και στέλνει στον Βασίλη το $c' = cs^{-1} \pmod n$. Ο Βασίλης πείθεται ότι συνομιλεί με την Αλίκη αν $c'^2 \equiv x \pmod n$.

   (i) Έχει το πρωτόκολλο την ιδιότητα της πληρότητας (completeness);

   (ii) Έχει το πρωτόκολλο την ιδιότητα της ορθότητας (soundness);

   (iii) Χρειάζεται να επαναληφθεί το παραπάνω πρωτόκολλο ή αρκεί μία εκτέλεσή του;

   (iv) Έχει κάποιο πρόβλημα ασφάλειας το πρωτόκολλο αυτό;

   (v) Έχει το πρωτόκολλο την ιδιότητα μηδενικής γνώσης (Zero Knowledge); Δικαιολογήστε την απάντησή σας.

3. Θεωρήστε το ακόλουθο σχήμα πιστοποίησης ταυτότητας (identification scheme): η Αλίκη διαθέτει δύο μυστικούς πρώτους p και q, τέτοιους ώστε $p \equiv q \equiv 3 \pmod 4$, και δημοσιοποιεί το $n = pq$ (αυτό μπορεί να γίνεται και μέσω έμπιστης αρχής). Όποτε η Αλίκη θέλει να πιστοποιήσει τον εαυτό της στον Βασίλη, του ζητάει να της στείλει ένα τυχαίο τετραγωνικό υπόλοιπο modulo n, έστω x. Η Αλίκη υπολογίζει μια τετραγωνική ρίζα y του x και την στέλνει στον Βασίλη. Ο Βασίλης ελέγχει αν $y^2 \equiv x \pmod n$: αν ναι, τότε πείθεται ότι συνομιλεί με την Αλίκη.

   (i) Περιγράψτε με ποιον τρόπο μπορεί η Αλίκη να υπολογίζει μια τετραγωνική ρίζα (mod n) του x.

   (ii) Αν όντως συνομιλεί με την Αλίκη, ποια είναι η πιθανότητα να πειστεί ο Βασίλης;

(iii) Αν ο Βασίλης συνομιλεί με την Εύα, που δεν διαθέτει τα p, q, ποια είναι η πιθανότητα να πειστεί;

(iv) Ποιο σοβαρό πρόβλημα ασφάλειας έχει το σύστημα αυτό;

4. Σχεδιάστε ένα σχήμα *από κοινού* ταυτοποίησης (identification), δηλαδή υπάρχουν δύο χρήστες B, C οι οποίοι θέλουν να ταυτοποιηθούν στον χρήστη A, αλλά αυτό πρέπει να γίνεται μόνο με τη συναίνεση και τη συνεργασία των B, C. Υποθέστε ότι υπάρχει μία έμπιστη αρχή η οποία μπορεί να διανέμει αρχικά κλειδιά με ασφάλεια στους A, B, C. Υποθέστε επίσης ότι οι B και C επικοινωνούν μέσω ασφαλούς καναλιού.

   (α) Περιγράψτε το σχήμα και αποδείξτε την πληρότητά του (δηλαδή αν όλοι οι χρήστες είναι έντιμοι η ταυτοποίηση θα επιτύχει).

   (β) Σχολιάστε την ορθότητα του σχήματός σας (δηλαδή, κάποιος που "κρυφακούει" στο κανάλι μεταξύ των A, B ή μεταξύ A, C έχει πολύ μικρή πιθανότητα να ταυτοποιηθεί σωστά ως B ή C.)

   (γ) Τι άλλα θέματα ασφάλειας / ορθότητας μπορεί να υπάρχουν σε ένα τέτοιο σύστημα. Σχολιάστε σε σχέση με το σύστημα που προτείνετε.

10.8 Ηλεκτρονικό Υλικό

- Διαδραστικές Παρουσιάσεις - Video
  - Διαδραστική επίδειξη του προβλήματος του 3-χρωματισμού
  - Απόδειξης μηδενικής γνώσης για SUDOKU
  - Sigma Protocols and Zero-Knowledge, Yehuda Lindel
- Κώδικας
  - Charm Βιβλιοθήκη κρυπτογραφικών πρωτοκόλλων σε Python.

Βιβλιογραφία

[1] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.

[2] David Chaum and Torben P. Pedersen. Wallet databases with observers. In *Proceedings of the 12th Annual International Cryptology Conference on*

Advances in Cryptology, CRYPTO '92, pages 89–105, London, UK, UK, 1993. Springer-Verlag.

[3] Ronald Cramer, Ivan Damgard, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *Proceedings of the 14th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '94, pages 174–187, London, UK, UK, 1994. Springer-Verlag.

[4] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In Hugo Krawczyk, editor, *Advances in Cryptology — CRYPTO '98*, volume 1462 of *Lecture Notes in Computer Science*, pages 13–25. Springer Berlin Heidelberg, 1998.

[5] Amos Fiat and Adi Shamir. How to prove yourself: practical solutions to identification and signature problems. In *Proceedings on Advances in cryptology—CRYPTO '86*, pages 186–194, London, UK, UK, 1987. Springer-Verlag.

[6] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems. *J. ACM*, 38(3):690–728, July 1991.

[7] S Goldwasser, S Micali, and C Rackoff. The knowledge complexity of interactive proof-systems. In *Proceedings of the seventeenth annual ACM symposium on Theory of computing*, STOC '85, pages 291–304, New York, NY, USA, 1985. ACM.

[8] Tatsuaki Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In ErnestF. Brickell, editor, *Advances in Cryptology — CRYPTO' 92*, volume 740 of *Lecture Notes in Computer Science*, pages 31–53. Springer Berlin Heidelberg, 1993.

[9] Edward Yang. Interactive zero knowledge 3-colorability demonstration.

Κεφάλαιο 11

Σύγχρονες Εφαρμογές

11.1 Ηλεκτρονικές Ψηφοφορίες

11.1.1 Εισαγωγή

Το πρόβλημα της ηλεκτρονικής ψηφοφορίας είναι ένα από τα πιο δύσκολα προβλήματα που έχει κληθεί να αντιμετωπίσει η επιστήμη των υπολογιστών. Το γεγονός ότι οι περισσότερες εκλογές διεξάγονται ακόμα με *φυσικό* τρόπο, δείχνει σίγουρα ότι δεν το έχει επιλύσει ακόμα. Η κύρια αιτία για το γεγονός αυτό, είναι ότι οι ψηφοφορίες διακρίνονται από αντιφατικές και αλληλοσυγκρουόμενες απαιτήσεις ασφαλείας, οι οποίες ενισχύονται από χαρακτηριστικά της ίδιας της επιστήμης των υπολογιστών. Οι πιο σημαντικές από τις απαιτήσεις ασφαλείας που έχουν οι ψηφοφορίες είναι η *ακεραιότητα* και η *μυστικότητα*.

Ακεραιότητα Δηλώνει ότι το αποτέλεσμα των εκλογών πρέπει να εκφράζει ακριβώς τη βούληση των ψηφοφόρων. Δηλαδή, κάθε ψήφος πρέπει να αντανακλά τη βούληση του ψηφοφόρου (*cast as intended*) και δεν πρέπει να υπάρχει καμία αλλοίωση είτε στην καταγραφή (*recorded as cast*) είτε στην καταμέτρηση (*counted as cast*). Είναι ευνόητο ότι η ευμετάβλητη φύση του λογισμικού (είτε από δόλο, είτε από προγραμματιστικά λάθη) αλλά και η δυσκολία χειρισμού του, δυσκολεύει την ικανοποίηση αυτής της ιδιότητας. Για το σκοπό αυτό τα διάφορα συστήματα που έχουν προταθεί, χρησιμοποιούν την έννοια της *επαληθευσιμότητας (verifiability)*, δηλαδή επιτρέπουν σε κάθε ψηφοφόρο (individual verifiability) ή σε ενδιαφερόμενες ομάδες ή σε όλους του συμμετέχοντες (universal verifiability) να επαληθεύσουν τη διαδικασία των εκλογών.

Μυστικότητα Επιτρέπει στον ψηφοφόρο να εκφράσει ελεύθερα τη βούλησή του, γνωρίζοντας ότι αυτή δεν θα αποκαλυφθεί σε κανένα. Πιθανοί αντίπαλοι που θέλουν να μάθουν τα περιεχόμενά της, είναι οι καταμετρητές (talliers) και εξωτερικοί εξαναγκαστές (coercers). Υπάρχει και η περίπτωση ο ίδιος ο ψηφοφόρος να θέλει να αποκαλύψει την ψήφο του. Ένα ηλεκτρονικό σύστημα ψηφοφορίας πρέπει τόσο να *προστατεύει* όσο και να επιβάλλει τη *μυστικότητα*. Μπορούμε λοιπόν να αναφερθούμε στα εξής επίπεδα:

- **Ιδιωτικότητα (privacy)**, όταν παρέχεται προστασία από έναν παθητικό αντίπαλο,
- **Αδυναμία απόδειξης (receipt freeness)**, όταν προστατεύεται το σύστημα από τον ψηφοφόρο,
- **Αντίσταση σε εξαναγκασμό (coercion resistance)**, όταν αντιμετωπίζουμε έναν πιο δυνατό αντίπαλο, ο οποίος παρεμβαίνει στον ψηφοφόρο ή στο σύστημα.

Λοιπές Ιδιότητες

- **Έλεγχος Καταλληλότητας (Eligibility):** Στην ψηφοφορία πρέπει να συμμετέχουν μόνο οι ψηφοφόροι που έχουν δικαίωμα και πρέπει να ψηφίζουν συνήθως μόνο μία φορά.
- **Δικαιοσύνη (Fairness):** Δεν είναι πρέπει να αποκαλύπτονται ενδιάμεσα και μερικά αποτελέσματα
- **Διαθεσιμότητα (Availability):** Το σύστημα πρέπει να εξυπηρετεί όλους τους ψηφοφόρους, ώστε να εκφραστεί η βούληση ολόκληρου του εκλογικού σώματος.
- **Αποδοτικότητα (Efficiency):** Η καταμέτρηση των ψήφων πρέπει να γίνεται σε εύλογο χρονικό διάστημα.

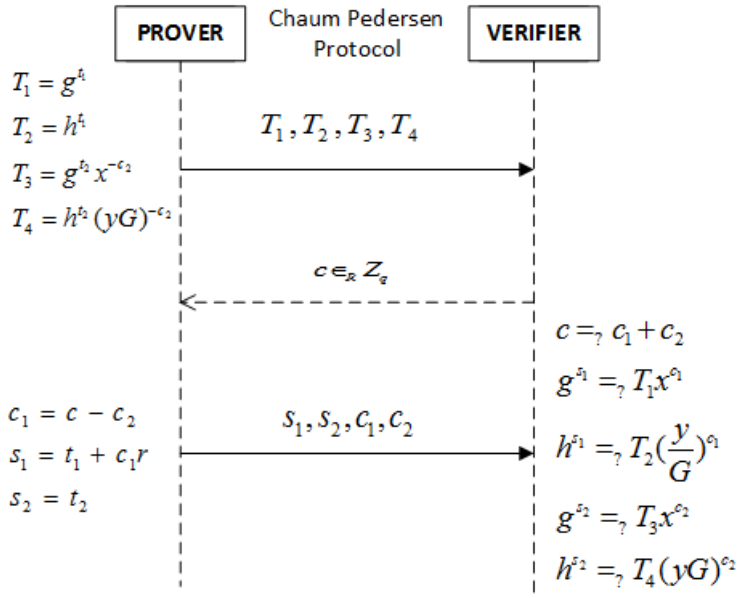
Είναι φανερό ότι κάποιες από τις παραπάνω ιδιότητες είναι αλληλοσυγκρουόμενες. Για παράδειγμα για τον έλεγχο καταλληλότητας χρειάζεται η ταυτότητα του ψηφοφόρου, γεγονός που εγείρει αμφιβολίες για την μυστικότητα της ψήφου. Επιπλέον η ακεραιότητα είναι δύσκολο να επιτευχθεί όταν οι ψήφοι είναι μυστικές. Στον φυσικό κόσμο, οι λύσεις που δίνονται σχετίζονται είτε με φυσικές ιδιότητες των μέσων καταγραφής των ψήφων (φάκελος, ανακάτεμα κάλπης), είτε με μεθόδους ελέγχου πρόσβασης (παραβάν), είτε (κυρίως) με χρήση έμπιστων τρίτων οντοτήτων (δικαστικοί αντιπρόσωποι) και σύγκρουσης συμφερόντων (εκλογικοί αντιπρόσωποι).

Θα περίμενε κανείς πως η κρυπτογραφία στις ηλεκτρονικές ψηφοφορίες θα έπαιζε ρόλο στην υλοποίηση της απαίτησης της μυστικότητας. Πιο σημαντικός όμως είναι ο ρόλος της στην αντικατάσταση των έμπιστων τρίτων οντοτήτων με πρωτόκολλα από τα οποία η εμπιστοσύνη προκύπτει ως αναδυόμενη ιδιότητα. Οι ηλεκτρονικές ψηφοφορίες προτάθηκαν ως μία από τις εφαρμογές των κρυπτοσυστημάτων δημοσίου κλειδιού από τις αρχές της δεκαετίας του 1980 κιόλας. Οι προτάσεις μπορούν να χωριστούν σε τρεις κατηγορίες: τα **ομομορφικά συστήματα**, τα **δίκτυα μίξης** και την **χρήση των τυφλών υπογραφών** τις οποίες και θα περιγράψουμε στις επόμενες ενότητες. Κοινό χαρακτηριστικό και των τριών προσεγγίσεων είναι ότι όλα τα μηνύματα που ανταλλάσσονται 'καταγράφονται' σε έναν *αυθεντικοποιημένο πίνακα ανακοινώσεων*, ο οποίος αναφέρεται ως **bulletin' board** ($\mathcal{BB}$). Τα περιεχόμενα του είναι προσβάσιμα και αποδεκτά από όλους. Για την υλοποίηση του θεωρητικά $\mathcal{BB}$ απαιτούνται πρωτόκολλα συναίνεσης (consensus), αλλά στις διάφορες πρακτικές υλοποιήσεις χρησιμοποιούνται βάσεις δεδομένων.

11.1.2 Ομομορφικά Συστήματα

Τα ομομορφικά συστήματα ηλεκτρονικής ψηφοφορίας προτάθηκαν αρχικά στη δεκαετία του 1980 από τον Josh Benaloh [8]. Βασίζονται στις ομομορφικές ιδιότητες συστημάτων κρυπτογράφησης και διαμοιρασμού κλειδιού. Η βασική ιδέα είναι η κρυπτογράφηση των ψήφων, χρησιμοποιώντας το δημόσιο κλειδί μιας εκλογικής αρχής, οι οποίες για λόγους επαληθευσιμότητας τοποθετούνται στο $\mathcal{BB}$. Στην συνέχεια όλες οι κρυπτογραφημένες ψήφοι συνδυάζονται (πχ. πολλαπλασιάζονται). Λόγω των ομομορφικών ιδιοτήτων το αποτέλεσμα του συνδυασμού δίνει το κρυπτογραφημένο άθροισμα των ψήφων, το οποίο στην συνέχεια η εκλογική αρχή το αποκρυπτογραφεί χρησιμοποιώντας το ιδιωτικό της κλειδί και το ανακοινώνει.

Η παραπάνω λύση έχει το πλεονέκτημα ότι σε όλη τη διάρκεια της καταμέτρησης οι ψήφοι παραμένουν μυστικές. Έχει όμως δύο αρκετά σημαντικά προβλήματα. Καταρχήν, εφόσον η αρχή μπορεί να αποκρυπτογραφήσει το αποτέλεσμα, μπορεί να αποκρυπτογραφήσει και τις μεμονωμένες ψήφους. Η λύση σε αυτό το πρόβλημα είναι ο διαμοιρασμός του κλειδιού αποκρυπτογράφησης σε περισσότερες από μία αρχές, με αντικρουόμενα συμφέροντα, οι οποίες θα συνεργαστούν μόνο για την αποκρυπτογράφηση του αποτελέσματος (χωρίς φυσικά ποτέ να ανακατασκευάσουν πλήρως το ιδιωτικό κλειδί). Το δεύτερο πρόβλημα αφορά τους ψηφοφόρους οι οποίοι θα μπορούσαν αντί να κρυπτογραφήσουν μόνο την ψήφο τους, να κρυπτογραφήσουν το ισοδύναμο 1000 ψήφων αλλοιώνοντας έτσι το αποτέλεσμα. Η λύση στο δεύτερο αυτό πρόβλημα δίνεται χρησιμοποιώντας τις αποδείξεις μηδενικής γνώσης. Κάθε ψηφοφόρος καταθέτει μαζί την ψήφο του και μία απόδειξη για την εγκυρότητα της.



Σχήμα 11.1: Απόδειξη Εγκυρότητας για τη θετική ψήφο [9]

Το πιο πλήρες πρωτόκολλο αυτής της κατηγορίας παρουσιάστηκε από τους Cramer, Genaro και Schoenmakers στο [9]. Εκτός από τα παραπάνω χαρακτηριστικά, είναι βέλτιστο σε ότι αφορά την υπολογιστική πολυπλοκότητα για τον ψηφοφόρο (*vote 'n' go*), η οποία είναι γραμμική ως προς την παράμετρο ασφάλειας του κρυπτοσυστήματος. Κάθε μία από τις εκλογικές αρχές έχει γραμμική πολυπλοκότητα ως προς N το πλήθος των ψηφοφόρων. Βασίζεται στο εκθετικό ElGamal 6.5.2 και έχει χρησιμοποιηθεί στο σύστημα Helios [1].

Συγκεκριμένα το πρωτόκολλο αποτελείται από τις εξής φάσεις:

- **Κατασκευή:** Υποθέτουμε ότι έχουν επιλεγεί κατάλληλα οι παράμετροι του συστήματος. Έστω λοιπόν G ο γεννήτορας. Μία θετική ψήφος αναπαρίσταται ως $m_y = 1$ και μια αρνητική ψήφος ως $m_n = -1$. Ο ψηφοφόρος διαλέγει ένα τυχαίο $b \in \{1, -1\}$ και κρυπτογραφεί ως $(x, y) = (g^r, h^r G^b)$. Για την εγκυρότητα της ψήφου πρέπει να αποδείξει ότι $b = 1$ ή $b = -1$ χωρίς φυσικά να αποκαλύψει ακριβώς την τιμή. Αυτό μπορεί να γίνει με μία διάζευξη αποδείξεων ισοτήτων διακριτών λογαρίθμων συγκεκριμένα ότι $log_g x = log_h(y/G)$ για $b = 1$ ή $log_g x = log_h(yG)$ για $b = -1$. Αυτό μπορεί να γίνει με την παρακάτω παραλλαγή του πρωτοκόλλου των Chaum Pedersen (10.3.2) όπως φαίνεται στο παρακάτω σχήμα:

  Φυσικά, η απόδειξη είναι σε μη διαλογική μορφή χρησιμοποιώντας τον μετασχηματισμό Fiat Shamir. Για να αποφευχθούν διπλοψηφίες η σύνοψη της

δέσμευσης πρέπει να περιέχει ένα μοναδικό αναγνωριστικό για κάθε ψηφοφόρο.

- **Ψηφοφορία:** Ο ψηφοφόρος σχηματίζει την τελική ψήφο $v_i \in \{-1, 1\}$ διαλέγοντας s_i τέτοιο ώστε $v_i = s_i b_i$ και το ανεβάζει στο $\mathcal{BB}$. Ο χωρισμός της διαδικασίας δημιουργίας ψήφου γίνεται σε δύο φάσεις για λόγους αποδοτικότητας. Η αρχική δέσμευση στο b_i η οποία είναι υπολογιστικά δαπανηρή μπορεί να γίνει ανεξάρτητα από την ψηφοφορία. Αντίθετα η επιλογή του s_i γίνεται πριν την οριστική ψήφο, αλλά είναι ταχύτατη.

- **Καταμέτρηση:** Με την λήξη της περιόδου ψηφοφορίας οι καταμετρητές ελέγχουν όλες τις αποδείξεις και αφαιρούν τις ψήφους που δεν αντιστοιχούν σε έγκυρες αποδείξεις. Στην συνέχεια πολλαπλασιάζουν όλα τα κρυπτογραφήματα των έγκυρων ψήφων και σχηματίζουν το γινόμενο $(A, B) = (\prod_{i=1}^{N} g^{r_i}, \prod_{i=1}^{N} h^{r_i} G^{v_i})$. Στη συνέχεια αποκρυπτογραφούν κανονικά κατά El Gamal και υπολογίζουν το $W = \frac{A}{B^x}$.

  Λόγω των ομομορφικών ιδιοτήτων του κρυπτοσυστήματος ισχύει $W = G^{\sum_{i=1}^{N} v_i} = G^T$ όπου T είναι το αποτέλεσμα των εκλογών. Για να το ανακτήσουν πρέπει να υπολογίσουν το διακριτό λογάριθμο $log_G W$. Ένα τέτοιο πρόβλημα είναι φυσικά δύσκολο, αλλά οι παράμετροι σε ένα εκλογικό σύστημα (πλήθος ψηφοφόρων) είναι τέτοιες που ένας τέτοιος υπολογισμός είναι δυνατός. Οπότε μπορεί να γίνει δοκιμή όλων των πιθανών δυνάμεων του G και επιστροφή αυτής που επαληθεύει τον λογάριθμο. Πιο συγκεκριμένα αφού $-N \leq T \leq N$, μπορούν εύκολα να υπολογιστούν όλες οι τιμές $G, G^2, G^3, \cdots$ μέχρι να βρεθεί το W.

Το παραπάνω σχήμα αφορά εκλογές με δύο υποψήφιους (τυπικά ναι-όχι). Μπορεί να επεκταθεί και για $C > 2$ υποψηφίους, όπου κάθε ψηφοφόρος μπορεί να επιλέξει είτε έναν από C είτε περισσότερους (t). Ο πιο απλός τρόπος υλοποίησης, είναι να γίνουν C παράλληλες εκλογές. Κάθε ψηφοφόρος πρέπει δηλαδή να υπερψηφίσει ακριβώς t υποψήφιους και να καταψηφίσει ακριβώς $C - t$. Για την καταμέτρηση θα χρησιμοποιηθούν C μετρητές. Αυτό σημαίνει ότι στο παραπάνω σχήμα θα πρέπει να επιλυθούν C διακριτοί λογάριθμοι. Φυσικά θα πρέπει να αλλάξει και η απόδειξη εγκυρότητας που πρέπει να υποβάλλει ο ψηφοφόρος. Το πρωτόκολλο Chaum Pedersen θα πρέπει να χρησιμοποιηθεί με C γεννήτορες και θα πρέπει να αποδειχθεί ότι ισχύουν ακριβώς t ισότητες.

Μία πιο αποτελεσματική μέθοδος χειρισμού πολλαπλών υποψηφίων προτάθηκε από τον Baudron στο [3]. Σε αυτήν επιλέγεται ένας αριθμός D και χρησιμοποιείται ως μοναδικός μετρητής. Αντί ο ψηφοφόρος να κρυπτογραφήσει τον αριθμό $c \in \{0 \cdots C - 1\}$ για τον αντίστοιχο υποψήφιο, κρυπτογραφεί τον αριθμό D^c. Με τον πολλαπλασιασμό των κρυπτογραφημένων ψήφων προκύπτει το αποτέλεσμα

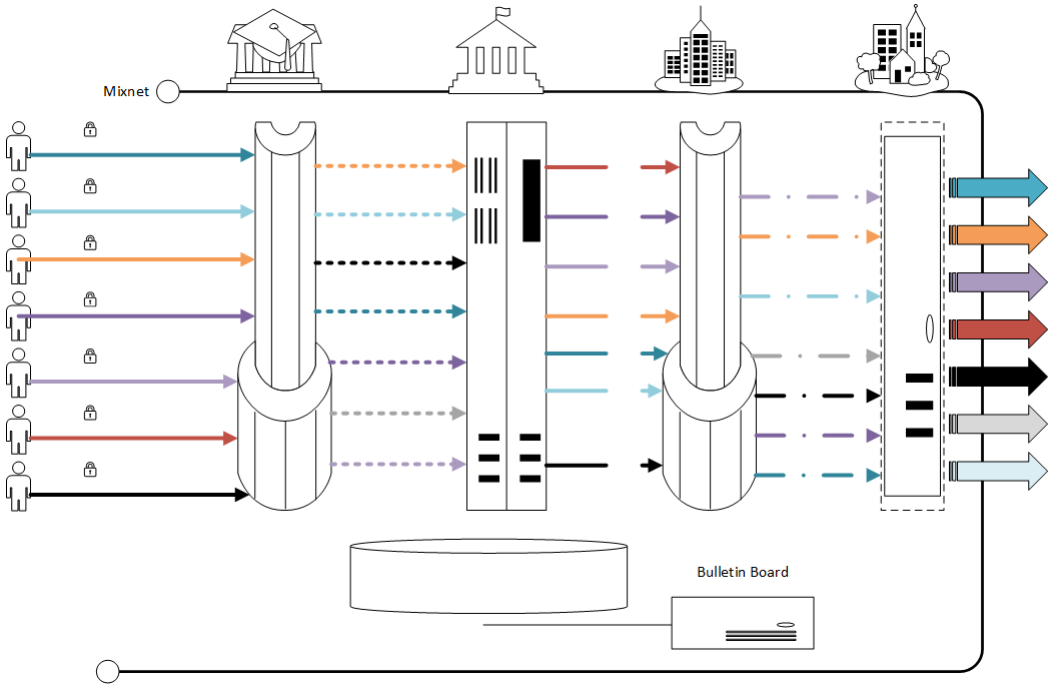
ως ένας αριθμός στο σύστημα αρίθμησης με βάση D, δηλαδή $T = \sum_{c=0}^{C-1} t_c D^c$, ο οποίος όμως θα αποκρυπτογραφηθεί ως G. Μετά την επίλυση του διακριτού λογαρίθμου, κάθε ψηφίο αυτού του αριθμού θα είναι το πλήθος των ψήφων που έλαβε ο συγκεκριμένος υποψήφιος.

Κλείνοντας, αξίζει να αναφερθεί ότι έχουν η προταθεί παρόμοια συστήματα εκλογών τα οποία βασίζονται σε κρυπτογραφικά πρωτόκολλα με τις ίδιες ομομορφικές ιδιότητες, χωρίς να απαιτούν όμως υπολογισμό κάποιου διακριτού λογαρίθμου, όπως το Paillier (βλ. 6.6.2).

11.1.3 Δίκτυα Μίξης

Τα δίκτυα μίξης (mixnets) $\mathcal{MN}$ είναι ένα εργαλείο γενικής χρήσης με στόχο την παροχή ανωνυμίας σε συστήματα ανταλλαγής μηνυμάτων. Προτάθηκε από τον David Chaum στο [4] και έχει για ανώνυμη περιήγηση (πχ. Tor), δημοπρασίες και εκλογές. Αποτελείται από οντότητες οι οποίες ονομάζονται **μίκτες (mix servers)** και τους οποίος θα συμβολίζουμε με $\{M_j\}_{j=1}^{k}$. Κάθε ένας λαμβάνει ένα σύνολο από μηνύματα $\{v_i\}_{i=1}^{n}$ στην είσοδό του, τα οποία αλλοιώνει και ανακατεύει (εφαρμόζοντας) μία τυχαία μετάθεση.Αυτό έχει το αποτέλεσμα στην έξοδο να βγουν ως ${v'}_{\pi(i)=1}^{\pi(i)=n}$. Έτσι ένα μήνυμα καθώς 'ταξιδεύει' στο δίκτυο αλλάζει μορφή και θέση με αποτέλεσμα να μην μπορεί να ανιχνευθεί. Κατά συνέπεια η έξοδος κάθε μίκτη, αλλά και ολόκληρου του δικτύου δεν μπορεί να συσχετιστεί με την είσοδο πετυχαίνοντας ανωνυμία. Συνήθως οι μίκτες τοποθετούνται στην σειρά δηλαδή κάθε ένας επεξεργάζεται όλα τα μηνύματα που έχει εξάγει ο προηγούμενος. Είναι δυνατή και η παράλληλη επεξεργασία, στην οποία κάθε μίκτης ασχολείται με μία διαμέριση των μηνυμάτων (διαφορετική κάθε φορά). Η επικοινωνία γίνεται συνήθως, μέσω του $\mathcal{BB}$, δηλαδή κάθε μίκτης λαμβάνει την είσοδο του από ένα συγκεκριμένο τμήμα του $\mathcal{BB}$ και αποθηκεύει την έξοδο του σε ένα επίσης προκαθορισμένο τμήμα.

Στις ηλεκτρονικές ψηφοφορίες τα μηνύματα τα οποία πρέπει να γίνουν ανώνυμα είναι φυσικά οι ψήφοι. Το $\mathcal{MN}$ θυμίζει έτσι το ανακάτεμα της κάλπης που γίνεται στα εκλογικά κέντρα πριν την καταμέτρηση. Αρχικά οι ψήφοι εισάγονται σε κρυπτογραφημένη μορφή. Η λειτουργία του $\mathcal{MN}$ αφαιρεί την συσχέτιση μεταξύ ψήφου και ψηφοφόρου, οπότε για την καταμέτρηση οι ψήφοι μπορούν να αποκρυπτογραφηθούν. Αυτό είναι και η μεγαλύτερη διαφορά των συγκεκριμένων πρωτοκόλλων από τα ομομορφικά συστήματα που είδαμε προηγουμένως και έχει συνέπειες στις εκλογικές συναρτήσεις οι οποίες μπορούν να υλοποιηθούν από το εκλογικό σύστημα.



Σχήμα 11.2: Λειτουργία Δίκτυου Μίξης

Δίκτυα Μίξης RSA (Decryption Mixnets)

Το πρώτο δίκτυο μίξης του [4], λειτουργεί σε επίπεδα και ονομάζεται και *onion routing*. Κάθε μίκτης έχει ένα ζεύγος κλειδιών RSA (pk_j, sk_j) από τα οποία δημοσιοποιείται το pk_j.

- Κάθε ψηφοφόρος κρυπτογραφεί την ψήφο του χρησιμοποιώντας τα κλειδιά των μικτών σε αντίστροφη σειρά.

$$L_0 = \{Enc_{pk_1}(Enc_{pk_2}(\cdots Enc_{pk_k}(v_i, r_i)\cdots, r_2), r_1)\}_{i=1}^{n}$$

- Κάθε μίκτης αφαιρεί ένα επίπεδο κρυπτογράφησης χρησιμοποιώντας το ιδιωτικό του κλειδί καθώς και την τυχαιότητα που περιέχει.

- Για το ανακάτεμα διαλέγει μία τυχαία μετάθεση και την εφαρμόζει σε κάθε μήνυμα. Το αποτέλεσμα γράφεται στο $\mathcal{BB}$. Για παράδειγμα ο πρώτος μίκτης θα γράψει:

$$L_1 = \{Enc_{pk_2}(\cdots Enc_{pk_k}(v_i, r_i)\cdots, r_2)\}_{i=\pi_1(1)}^{\pi_1(n)}$$

- Η διαδικασία επαναλαμβάνεται. Τελικά η έξοδος του δικτύου μίξης θα είναι:

$$L_k = \{v_i\}_{i=\pi_k\circ\cdots\circ\pi_1(1)}^{\pi_k\circ\cdots\circ\pi_1(n)})$$

- Σε περίπτωση που το δίκτυο μίξης χρησιμοποιείται για ηλεκτρονική ψηφοφορία μπορεί να ξεκινήσει η καταμέτρηση.

Στην όλη διαδικασία μπορούμε να παρατηρήσουμε τα εξής:

- Θεωρητικά αρκεί ένας 'τίμιος' μίκτης για να χαθεί ο συσχετισμός εισόδου εξόδου.

- Ο τελευταίος μίκτης M_k έχει πρόσβαση στο μη κρυπτογραφημένο μήνυμα.

- Η διαδικασία μπορεί να 'μπλοκάρει' αν κάποιος μίκτης δεχθεί ή αποφασίσει άρνηση εξυπηρέτησης (DoS).

- Το πλήθος των κρυπτογραφήσεων και το μέγεθος του κρυπτοκειμένου είναι ανάλογο του αριθμού των μικτών.

Δίκτυα Μίξης με Ανακρυπτογράφηση (Reencryption Mixnets)

Μία παραλλαγή του δικτύου μίξης του Chaum δόθηκε το 1993 [21]. Βασίζεται στην ιδιότητα της αποκρυπτογράφησης που διαθέτει το κρυπτοσύστημα ElGamal, η οποία επιτρέπει σε κάθε μίκτη να τροποποιήσει τα μηνύματα εισόδου χωρίς να χρειάζεται αποκρυπτογράφηση με δικό του κλειδί. Στα δίκτυα μίξης με ανακρυπτογράφηση είναι δυνατές οι εξής δύο παραλλαγές:

- Εφαρμογή μόνο ανακρυπτογράφησης
- Συνδυασμός ανακρυπτογράφησης και αποκρυπτογράφησης

Δεν είναι δυνατή η χρήση δικτύων μίξης μόνο με κρυπτογράφηση αν χρησιμοποιείται το ElGamal (ο λόγος αφήνεται ως άσκηση στον αναγνώστη).

Στην πρώτη περίπτωση για παράδειγμα κάθε μίκτης M_j ακολουθεί τα παρακάτω βήματα:

- Λαμβάνει την είσοδο $L_{j-1} = \{(g^{r_{j-1,i}}, v_i \cdot y^{r_{j-1,i}})\}_{i=1}^n$ από το $\mathcal{BB}$.
- Εφαρμόζει νέα τυχαιότητα με ανακρυπτογράφηση:

$$L_{j-1} = \{(g^{r_{j-1,i}+rj,i}, v_i \cdot y^{r_{j-1,i}+rj,i})\}_{i=1}^n$$

- Εφαρμόζει μία τυχαία μετάθεση π_j

Το κλειδί αποκρυπτογράφησης μοιράζεται σε ένα πλήθος έμπιστων οντοτήτων (μπορεί να είναι και οι ίδιοι οι μίκτες).

Στην δεύτερη περίπτωση:

- Η είσοδος είναι:

$$L_0 = \{(g^{r_{0i}}, v_i(y_1 \cdots y_k)^{r_{0i}})\}_{i=1}^n$$

  Η κρυπτογραφήσεις δεν είναι εμφωλευμένες, αλλά μπορεί να θεωρηθεί ότι έχουν γίνει χρησιμοποιώντας ένα συνδυασμένο δημόσιο κλειδί $\prod_{i=1}^k y_i$ όπου y_i είναι το δημόσιο κλειδί κάθε μίκτη. Κατά συνέπεια το κρυπτοκείμενο αλλά και το πλήθος των κρυπτογραφήσεων δεν εξαρτάται από το πλήθος των μικτών

- Ο M_j αποκρυπτογραφεί μερικώς

$$L_{j-1} = \{(g^{\sum_{t=0}^{j-1} r_{ti}}, v_i \cdot (\prod_{t=j}^k y_t)^{\sum_{t=0}^{j-1} r_{ti}})\}_{i=1}^n$$

διαιρώντας με $g^{x_j \cdot \sum_{t=0}^{j-1} r_t}$ και εφαρμόζει νέα τυχαιότητα r_{ji}:

$$L_j = \{(g^{\sum_{t=0}^{j} r_{ti}}, v_i \cdot (\prod_{t=j+1}^{k} y_t)^{\sum_{t=0}^{j} r_{ti}})\}_{i=1}^{n}$$

- Εφαρμόζει μία τυχαία μετάθεση π_j

Επιθέσεις

Οι επιθέσεις στα δίκτυα μίξης στοχεύουν στην αναίρεση της ιδιότητας της ανωνυμίας. Η βασική ιδέα (οφείλεται στους Pfitzmann [22] αλλά έχει εμφανιστεί με αρκετές παραλλαγές στη βιβλιογραφία) είναι αυτή της *επισήμανσης (tagging)* ενός μηνύματος με τρόπο τέτοιο ώστε να μπορεί να αναγνωριστεί όταν εξέλθει από το δίκτυο. Το μαρκάρισμα επιτυγχάνεται κυρίως με εκμετάλλευση των ομομορφικών ιδιοτήτων του κρυπτοσυστήματος που χρησιμοποιεί το δίκτυο μίξης.

Στην περίπτωση των δικτύων μίξης με ανακρυπτογράφηση για παράδειγμα η επίθεση είναι η εξής:

- Ο επιτιθέμενος $\mathcal{A}$ θέλει να παρακολουθήσει την είσοδο v_i για κάποιον συμμετέχοντα P_i.
- Έτσι ανακτά την αρχική κρυπτογράφηση από το $\mathcal{BB}$ $c_{i0} = (g^R, v_i \cdot (y_1, \cdots, y_k)^R)$ δηλ. ένα ζεύγος (t, u)
- Ο $\mathcal{A}$ για κάποιο γνωστό x παράγει το $c''_{i0} = (t^x, u^x) = (g^{R'x}, v_i^x \cdot (y_j, \cdots, y_k)^{R'x})$ και αντικαθιστά κάποιο άλλο μήνυμα.
- Το $\mathcal{MN}$ θα δώσει στην έξοδο τόσο το v_i^x όσο και v_i λόγω των ομομορφικών ιδιοτήτων
- Ο $\mathcal{A}$ ανακτά όλα τα μηνύματα εξόδου και τα υψώνει στην x.
- Στην συνέχεια ελέγχει τις δύο λίστες για κοινά στοιχεία. Όταν βρει ένα τέτοιο εντοπίζει το μήνυμα που έψαχνε καθώς $v_{\pi(i)}^x = v_i^x$.

Για την επιτυχή έκβαση της παραπάνω επίθεσης, υποθέτουμε ότι ο $\mathcal{A}$ μπορεί να εισάγει μηνύματα της επιλογής του στο $\mathcal{BB}$. Για παράδειγμα, μπορεί να συνεργαστεί με κάποιο νόμιμο χρήστη και να αντικαταστήσει την ψήφο του.

Σχήμα 11.3: Δίκτυο μίξης με ανακρυπτογράφηση 2×2

Επαληθευσιμότητα

Προφανώς ένας αντίπαλος που συνεργάζεται με ένα μίκτη μπορεί να κάνει μεγαλύτερη ζημιά από έναν αντίπαλο που συνεργάζεται μόνο με έναν απλό χρήστη. Μπορεί για παράδειγμα να παρακολουθήσει συγκεκριμένες ψήφους, αλλά και να αντικαταστήσει/παραλείψει ψήφους. Για τον λόγο αυτό οι Sako και Killian [13] εισήγαγαν την έννοια της επαληθευσιμότητας στα δίκτυα μίξης. Το πρωτόκολλο τους χρησιμοποιεί την τεχνική cut-and-choose για την επαλήθευση της λειτουργίας του δικτύου μίξης, δηλαδή κάθε μίκτης αποδεικνύει ότι

- Η μερική αποκρυπτογράφηση ήταν σωστή (αν έγινε)
- Η ανακρυπτογράφηση και η μετάθεση ήταν σωστή (πχ. δεν υπήρχαν παραλείψεις).

Το πρωτόκολλο των Sako και Killian έχει ορθότητα $1/2$ που σημαίνει ότι ένας μίκτης μπορεί να κλέψει χωρίς να γίνει αντιληπτός με πιθανότητα $1/2$. Για να ελαχιστοποιηθεί αυτή η πιθανότητα το πρωτόκολλο μπορεί να επαναληφθεί, αυξάνοντας όμως έτσι τις υπολογιστικές απαιτήσεις.

Στην ενότητα αυτή θα περιγράψουμε για εκπαιδευτικούς λόγους ένα μεταγενέστερο, αλλά απλούστερο πρωτόκολλο επαληθευσιμότητας σε δίκτυα μίξης, το οποίο προτάθηκε από τον Abe [16] και ανεξάρτητα από τους Juel και Jakobsson [15]. Σε αυτά, το δίκτυο μίξης κατασκευάζεται αναδρομικά από μικρότερα στοιχεία τα οποία ονομάζονται *συγκριτές (comparators)*. Ένα τέτοιο στοιχείο, το οποίο μπορούμε να θεωρήσουμε ως ένα 2×2 δίκτυο μίξης, φαίνεται στο παρακάτω σχήμα:

Η είσοδος είναι δύο κρυπτοκείμενα $C_0 = Enc(m_0, r_0)$ και $C_1 = Enc(m_1, r_1)$. Αρχικά εφαρμόζει ανακρυπτογράφηση σε αυτά και προκύπτουν τα $C'_0 = Reenc(C_0) = Enc(m_0, r_0 + r'_0)$ και $C'_1 = Reenc(C_1) = Enc(m_1, r_1 + r'_1)$. Η έξοδος είναι: C'_b και C'_{1-b} όπου $b \in_R \{0, 1\}$.

Η έξοδος δεν επιλέγεται τυχαία αλλά υλοποιείται μια συνάρτηση f τέτοια ώστε:

$$f(x, y) = \begin{cases} (x, y), & x < y \\ (y, x), & x \geq y \end{cases}$$

Αν το κρυπτοσύστημα είναι σημασιολογικά ασφαλές, τότε ένας αντίπαλος δεν μπορεί να διακρίνει ποιο από τα C_0, C_1 κρυπτογραφεί τα m_0 και m_1.

Για να προσθέσουμε επαληθευσιμότητα πρέπει να προστεθεί μία απόδειξη ότι κάθε C'_b είναι ανακρυπτογράφηση είτε του C_0 του C_1. Αυτό θα γίνει σε δύο βήματα.

Πρόταση 11.1. *Το κρυπτογράφημα* $C'_1 = (G'_1, M'1) = (g^u, m'_1 \cdot y^u)$ *είναι ανακρυπτογράφηση* $C_1 = (G_1, M_1) = (g^t, m_1 \cdot y^t)$

Απόδειξη. Το C'_1 είναι ανακρυπτογράφηση του C_1 ανν και τα δύο κρυπτογραφούν το ίδιο μήνυμα, δηλ. $m'_1 = m_1$. Διαιρούμε τα δύο μέρη και έχουμε:

$$\frac{G'_1}{G_1} = \frac{g^u}{g^t} = g^{u-t} \text{ και } \frac{M'_1}{M_1} = \frac{m'y^u}{my^t} = y^{u-t}$$

Ισοδύναμα πρέπει να δείξουμε ότι οι $G'_1/G_1, M'_1/M_1$ έχουν τον ίδιο λογάριθμο με βάσεις g, y αντίστοιχα, δηλ. $log_g \frac{G'_1}{G_1} = log_y \frac{M'_1}{M_1}$. Για τον σκοπό αυτό μπορεί να χρησιμοποιηθεί το πρωτόκολλο Chaum Pedersen που συναντήσαμε στην ενότητα 10.3.2. □

Για την ορθότητα της μετάθεσης πρέπει να δείξουμε ότι το $\{C'_0, C'_1\}$ είναι ανακρυπτογράφηση μια μετάθεσης του $\{C_0, C_1\}$ χωρίς να φανερώσουμε την αντιστοιχία. Ισοδύναμα πρέπει να αποδείξουμε ότι το C'_0 ανακρυπτογραφεί το C_0 και το C'_1 ανακρυπτογραφεί το C_1 *ή* ότι το C'_0 ανακρυπτογραφεί το C_1 και το C'_1 ανακρυπτογραφεί C_0. Συμβολικά δηλαδή:

$$(C'_0 = Reenc(C_0) \wedge C'_1 = Reenc(C_1)) \bigvee (C'_0 = Reenc(C_1) \wedge C'_1 = Reenc(C_0))$$

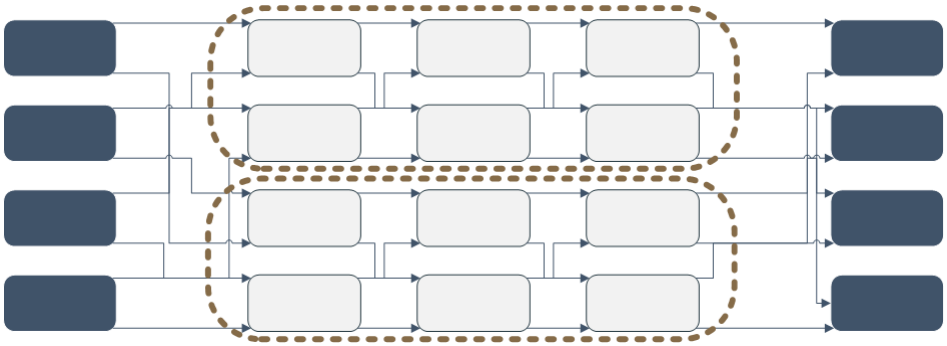
Κάτι τέτοιο μπορεί να επιτευχθεί με την εκτέλεση 4 πρωτοκόλλων Chaum-Pedersen.

Έχοντας κατασκευάσει ένα επαληθεύσιμο 2×2 δίκτυο μίξης μπορούμε να το γενικεύσουμε σε ένα δίκτυο $N \times N$. Ένας τρόπος να γίνει αυτό είναι για παράδειγμα το δίκτυο Benes.

11.1.4 Ψηφοφορίες με Τυφλές Υπογραφές

Η τρίτη προσέγγιση στην υλοποίηση ηλεκτρονικών ψηφοφοριών με χρήση κρυπτογραφίας εκμεταλλεύεται την ανωνυμία που παρέχουν οι ηλεκτρονικές υπογραφές κάτι που άλλωστε ήταν και ένας από τους στόχους του Chaum όταν τις πρότεινε [5]. Το σχήμα του Chaum συνδυάζει μυστικότητα και ατομική επαληθευσιμότητα:

- Ο ψηφοφόρος υποβάλλει μία 'τυφλωμένη' έκδοση του ψηφοδελτίου μαζί με πληροφορίες ταυτότητας.



Σχήμα 11.4: Αναδρομική Κατασκευή Δικτύου Μίξης (Από http://goo.gl/iNiwXB)

- Η εκλογική αρχή επαληθεύει την ταυτότητα του υποψηφίου και ελέγχει αν έχει δικαίωμα ψήφου. Αν η απάντηση είναι θετική υπογράφει ψηφιακά το τυφλωμένο ψηφοδέλτιο και το επιστρέφει στον ψηφοφόρο.
- Ο ψηφοφόρος αφού επαληθεύσει την υπογραφή της αρχής καταθέτει το ψηφοδέλτιο στο $\mathcal{BB}$ *ανώνυμα*.
- Η αρχή λαμβάνει τα υπογεγραμμένα ψηφοδέλτια και επαληθεύει την υπογραφή της.
- Ο ψηφοφόρος μπορεί να επαληθεύσει το ψηφοδέλτιο του εισάγοντας σε αυτό ένα τυχαίο αριθμό που μόνο αυτός γνωρίζει.

Το παραπάνω σχήμα έχει το πρόβλημα του ότι η εκλογική αρχή παίζει κυρίαρχο ρόλο σε όλες τις φάσεις του πρωτοκόλλου. Έτσι για παράδειγμα δεν κατέχει την ιδιότητα της δικαιοσύνης καθώς η αρχή μπορεί να κάνει καταμέτρηση και να μάθει ενδιάμεσα αποτελέσματα πριν την λήξη των εκλογών. Η λύση σε αυτό δόθηκε από τους Fujioka, Okamoto και Ohta στο [11] το οποίο χωρίζει τις λειτουργίες σε 2 οντότητες:

- Την εκλογική αρχή η οποία θα ξέρει την ταυτότητα του ψηφοφόρου και όχι την ψήφο
- Την αρχή καταμέτρησης η οποία γνωρίζει την ψήφο αλλά όχι την ταυτότητα

Συγκεκριμένα υποθέτουμε ότι η εκλογική αρχή έχει ένα δημόσιο και ένα ιδιωτικό κλειδί (e_A, d_A). Ομοίως και ο κάθε ψηφοφόρος διαθέτει τα(e_I, d_I). Το σχήμα προχωράει σε φάσεις:

1. **Προετοιμασία**

- Ο ψηφοφόρος ετοιμάζει την ψήφο του v_i
- Χρησιμοποιώντας τυχαιότητα rc_i, δεσμεύεται σε αυτή δημιουργώντας το $b_i = commit(v_i, rc_i) = g^{rc_i}h^{v_i}$. Η δέσμευση εξασφαλίζει ότι ο ψηφοφόρος δεν θα συμπεριφερθεί διαφορετικά στις φάσεις του πρωτοκόλλου. Επιπλέον καθιστά μη αναγκαία την ενσωμάτωση τυχαιότητας στο ψηφοδέλτιο για λόγους επαληθευσιμότητας.
- Στην συνέχεια χρησιμοποιώντας τυχαιότητα rb_i και το δημόσιο κλειδί της αρχής τυφλώνει το ψηφοδέλτιο $bb_i = blind(b_i, rb_i) = b_i rb_i^{e_A}$.
- Μετά το υπογράφει $sbb_i^I = sign_{d_I}(bb_i)$.
- Στέλνει το μήνυμα (i, bb_i, sbb_i^I) στην εκλογική αρχή όπου με i συμβολίζονται οι πληροφορίες ταυτότητας του ψηφοφόρου.

2. **Εξουσιοδότηση**

- Κατά την παραλαβή η αρχή ελέγχει την υπογραφή του ψηφοφόρου, το δικαίωμα του να ψηφίσει και αν έχει διπλοψηφίσει.
- Αν όλοι οι έλεγχοι είναι επιτυχείς υπογράφει το τυφλωμένο ψηφοδέλτιο $sbb_i^A = sign_{d_A}(bb_i) = b_i^{d_A} rb_i$.
- Τέλος επιστρέφει το sbb_i^A στον ψηφοφόρο i και ανακοινώνει τον συνολικό αριθμό ψηφοφόρων.

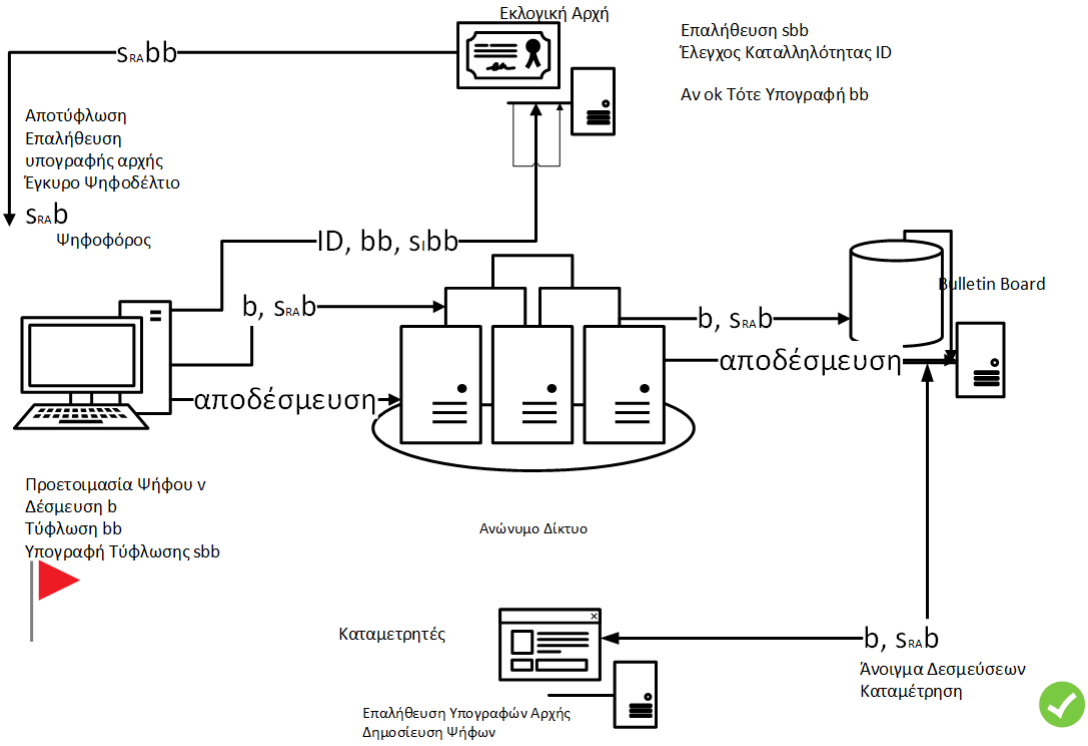
3. **Ψηφοφορία**

- Ο ψηφοφόρος αποτυφλώνει το ψηφοδέλτιο που έλαβε από την αρχή $sb_i^A = unblind(sbb_A^i) = b_i^{d_A}$
- Η υπογραφή επαληθεύεται χρησιμοποιώντας το δημόσιο κλειδί της αρχής. Η υπογραφή είναι επαληθεύσιμη από όλους.
- Η ψήφος γίνεται με αποστολή των b_i, sb_i^A στην αρχή καταμέτρησης. Εδώ πρέπει να χρησιμοποιηθεί ένα ανώνυμο κανάλι ώστε να υπάρξει απόκρυψη στοιχείων που ίσως προδώσουν την ταυτότητα του ψηφοφόρου (πχ. δικτυακές διευθύνσεις).

4. **Συλλογή**

- Η αρχή καταμέτρησης επαληθεύει την υπογραφή της αρχής σε κάθε ψηφοδέλτιο sb_i^A.
- Όσα ψηφοδέλτια πέρασαν τον έλεγχο δημοσιεύονται σε μια λίστα $\{idx, b_i, sb_i^A\}$.

5. **Αποδέσμευση**



Σχήμα 11.5: Σύστημα Ηλ. Ψηφοφορίας με Τυφλές Υπογραφές

- Μετά την λήξη της προθεσμίας ψηφοφορίας κάθε ψηφοφόρος επαληθεύει ότι το πλήθος των ψηφοφόρων που δημοσίευσε η εκλογική αρχή ισούται με το πλήθος των ψηφοδελτίων που δημοσίευσε η αρχή καταμέτρησης.
- Αν όλοι οι έλεγχοι είναι επιτυχείς αποστέλλονται τα idx, rc_i μέσω ενός ανώνυμου καναλιού.

6. **Καταμέτρηση**

- Όλα τα ψηφοδέλτια μπορούν να δημοσιοποιηθούν και όλοι οι ενδιαφερόμενοι μπορούν να κάνουν καταμέτρηση. Τα ψηφοδέλτια είναι ανώνυμα και η ψηφοφορία έχει λήξη οπότε δεν επηρεάζεται το κριτήριο της δικαιοσύνης.

Εφαρμογή σε Ανώνυμη Πιστοποίηση Το παραπάνω σχήμα μπορεί να παρέχει μία πρακτική ιδέα για μια γενικότερη εφαρμογή: Θέλουμε να επιτρέψουμε σε ένα

σύνολο χρηστών να εισέρχονται σε κάποιο σύστημα ανώνυμα, ώστε να υποβάλλουν κάποια αξιολόγηση. Αυτό θα μπορούσε να γίνει ως εξής για παράδειγμα:

- Οι συμμετέχοντες εγγράφονται σε μία αρχή πιστοποίησης, χρησιμοποιώντας τα πραγματικά τους στοιχεία.
- Κατά την υποβολή της αίτησης αξιολόγησης οι συμμετέχοντες δημιουργούν ένα τυχαίο αντικείμενο δεδομένων m και δεσμεύονται σε αυτό χρησιμοποιώντας μια συνάρτηση σύνοψης.
- Το $\mathcal{H}(m)$ τυφλώνεται δίνοντας το b_m.
- Στη συνέχεια υποβάλλουν τα στοιχεία τους στην αρχή πιστοποίησης και τα συσχετίζουν με το b_m.
- Η αρχή υπογράφει το b_m.
- Προκειμένου να υποβάλλει την αξιολόγηση ο κάθε συμμετέχων αποτυφλώνει το υπογεγραμμένο στοιχείο που έλαβε προηγουμένως, αποκτώντας έτσι ένα ανώνυμο αλλά έγκυρο όνομα χρήστη.

11.2 Πρωτόκολλα ανωνυμίας

11.2.1 Εισαγωγή

Στην προηγούμενη ενότητα είδαμε την ανάγκη για παροχή ανωνυμίας στις ηλεκτρονικές ψηφοφορίες. Η ανάγκη αυτή δεν περιορίζεται στην συγκεκριμένη εφαρμογή, αλλά μπορεί να εφαρμοστεί σε μία πληθώρα εφαρμογών στο Internet, όπου υπάρχει η αίσθηση ότι κάθε κίνηση καταγράφεται και συσχετίζεται με αποθηκευμένα προφίλ που διατηρούν κυβερνητικοί και εμπορικοί οργανισμοί. Η συλλογή τόσο μεγάλης ποσότητας πληροφορίας μπορεί να αποτελέσει απειλή για τις ανθρώπινες ελευθερίες, αν χρησιμοποιηθεί από απολυταρχικά καθεστώτα. Για τον σκοπό αυτό ένας από τους στόχους της κρυπτογραφίας είναι η ενίσχυση της δυνατότητας των χρηστών των δικτύων επικοινωνιών να παραμείνουν ανώνυμοι, ακόμα και πριν την διάδοση του Διαδικτύου. Στην ενότητα αυτή θα εξετάσουμε δύο από τα αρχικά πρωτόκολλα που προτάθηκαν για τον σκοπό αυτό, τα οποία προέρχονται από τον ίδιο άνθρωπο, τον David Chaum.

11.2.2 Το πρωτόκολλο Tor

Το πρωτόκολλο The Onion Router (Tor) αποτελεί μια εφαρμογή με λειτουργία ανάλογη με τα δίκτυα μίξης με αποκρυπτογράφηση (decryption mixes - 11.1.3).

Επιτρέπει στους χρήστες ενός δικτύου να επικοινωνήσουν ανώνυμα φτιάχνοντας ένα εσωτερικό δίκτυο μέσω του οποίου δρομολογούνται τα μηνύματα και στο οποίο δεν μπορεί να υπάρξει καταγραφή της συνολικής διαδρομής ενός μηνύματος.

Για να επιτευχθεί αυτό χρησιμοποιείται ένα σύνολο κόμβων του δικτύου R ως αναμεταδότες (relays). Κάθε αναμεταδότης $r \in R$ έχει ένα δημόσιο κλειδί κρυπτογράφησης pk_r. Για να αποσταλεί ένα μήνυμα msg ανώνυμα, ο αποστολέας επιλέγει το σύνολο των κόμβων $\{r_i\}_{i=1}^{n}$ από το R οι οποίοι θα προωθήσουν το μήνυμά του, διαλέγοντας τα δημόσια κλειδιά τους. Στη συνέχεια κατασκευάζει το ζεύγος $m = (addr, msg)$ όπου $addr$ η διεύθυνση του παραλήπτη και το κρυπτογραφεί χρησιμοποιώντας τα δημόσια κλειδιά των αναμεταδοτών με αντίστροφη σειρά. Δηλαδή:

$$o = \texttt{Encrypt}(pk_1 \cdots \texttt{Encrypt}(pk_{n-1}, (pk_n, \texttt{Encrypt}(pk_n, m))) \cdots)$$

Σε κάθε επίπεδο κρυπτογράφησης περιλαμβάνεται το δημόσιο κλειδί του επόμενου αναμεταδότη. Το κρυπτογράφημα αποστέλλεται στον κόμβο r_1, ο οποίος είναι και ο μόνος που γνωρίζει τον αποστολέας. Αυτός αποκρυπτογραφεί το μήνυμα, ανακτά το επόμενο δημόσιο κλειδί και προωθεί το εσωτερικό επίπεδο στον αναμεταδότη r_2. Η διαδικασία συνεχίζεται μέχρι τον κόμβο r_n, ο οποίος προωθεί πλέον το μήνυμα στον παραλήπτη. Παρατηρούμε λοιπόν πως ο πρώτος αναμεταδότης μαθαίνει μόνο τον αποστολέα, ενώ ο τελευταίος μόνο τον παραλήπτη. Με τον τρόπο αυτό διατηρείται η ανωνυμία.

Αν και το Tor χρησιμοποιείται ευρύτατα από κάθε είδους χρήστες που θέλουν να διατηρήσουν την ανωνυμία τους έχει υπάρχει αντικείμενο κριτικής και καχυποψίας. Για παράδειγμα αν δεν προστατεύεται το μήνυμα που μεταβιβάζεται μπορεί κατά την έξοδό του να ανακτηθεί και λόγω της πληροφορίας που περιέχει να υπάρξει ταυτοποίηση ταυτότητας. Επιπλέον μπορεί εύκολα να παρεμποδιστεί η χρήση του, καθώς για να υπάρξει πρόσβαση στο παράλληλο δικτύο του Tor πρέπει να υπάρχει ένας δημόσιος κατάλογος στον οποίο θα μπορούν να ανακτηθούν οι διευθύνσεις των αναμεταδοτών. Η ύπαρξη αυτού του καταλόγου, όμως επιτρέπει σε οποιονδήποτε θέλει να παρεμποδίσει τους χρήστες του 'κανονικού' δικτύου να χρησιμοποιήσουν το Tor, απλά μπλοκάροντας την αποστολή μηνυμάτων από και προς κάποιους από αυτούς τους κόμβους. Φυσικά αυτή δεν είναι μοναδική επίθεση στο πρωτόκολλο [10].

11.2.3 DC-Net

Μία πολύ πιο ισχυρή λύση στο πρόβλημα της ανωνυμίας δόθηκε από τον ίδιο τον Chaum, μερικά χρόνια αργότερα στο [7]. Στην συγκεκριμένη εργασία, το πρόβλημα της ανωνυμίας μοντελοποιείται ως μια σειρά αλληλεπιδράσεων n παικτών

σε t γύρους. Σε κάθε γύρο ένας παίκτης εκπέμπει το μήνυμά του σε όλους τους υπόλοιπους. Στόχος είναι να μην μπορεί κανείς να καταλάβει ποιος εξέπεμψε στον προηγούμενο γύρο.

Η λύση του Chaum παρέχει ανωνυμία χωρίς προϋποθέσεις (unconditional anonymity). Για να γίνει καλύτερα κατανοητή χρησιμοποιήθηκε ένα απλό παράδειγμα, το οποίο είναι χαρακτηριστικό στη βιβλιογραφία της κρυπτογραφίας και γι' αυτό θα το παραθέσουμε στη συνέχεια:

Κρυπτογράφοι σε Δείπνο Τρεις κρυπτογράφοι δειπνούν σε ένα εστιατόριο όταν ξαφνικά πληροφορούνται ότι ο λογαριασμός πληρώθηκε, είτε από έναν από τους τρεις, είτε από την NSA. Οι κρυπτογράφοι θέλουν να μάθουν ποια από τις δύο περιπτώσεις ισχύει, καθώς δεν νιώθουν άνετα με το να πληρώνει το λογαριασμό η NSA. Από την άλλη δεν θέλουν να μάθει *ποιος* ακριβώς πλήρωσε. Για τον λόγο αυτό σχεδιάζουν το εξής πρωτόκολλο:

- Ανά δύο 'στρίβουν' ένα νόμισμα και από αυτό παράγουν ένα τυχαίο bit. Τελικά παράγονται τα bit $b_{0,1}, b_{0,2}, b_{1,2}$
- Κάθε ένας ανακοινώνει αν τα bit, στα οποία έχει πρόσβαση, είναι ίδια ή όχι. Πιο τυπικά υπολογίζει και δημοσιεύει το:
  - $m_i = b_{i-1,i} \oplus b_{i,i+1}$ αν *δεν* πλήρωσε
  - $m_i = \overline{b_{i-1,i} \oplus b_{i,i+1}}$ αν πλήρωσε
- Το τελικό αποτέλεσμα προκύπτει από το $b = m_0 \oplus m_1 \oplus m_2$

Αν πλήρωσε η NSA τότε:

$$b = m_0 \oplus m_1 \oplus m_2 = b_{0,2} \oplus b_{0,1} \oplus b_{0,1} \oplus b_{1,2} \oplus b_{0,2} \oplus b_{1,2} = 0$$

Αντίθετα αν πλήρωσε κάποιος από τους κρυπτογράφους (για παράδειγμα ο πρώτος):

$$b = m_0 \oplus m_1 \oplus m_2 = \overline{b_{0,2} \oplus b_{0,1}} \oplus b_{0,1} \oplus b_{1,2} \oplus b_{0,2} \oplus b_{1,2} = 1$$

Το παραπάνω πρωτόκολλο δεν διαρρέει καμία πληροφορία για το ποιος πλήρωσε. Συνεχίζοντας το παράδειγμα, ο δεύτερος κρυπτογράφος δεν μπορεί να μάθει ότι πλήρωσε ο πρώτος, καθώς η πληροφορία που γνωρίζει: $b_{0,1}, b_{1,2}$ εξουδετερώνεται κατά το xor και μένει μόνο η πληροφορία που δεν γνωρίζει σε άχρηστη μορφή $b_{0,2} \oplus \overline{b_{0,2}} = 1$. Ομοίως φυσικά και για τον τρίτο κρυπτογράφο.

Η γενίκευση του πρωτοκόλλου σε n παίκτες είναι αρκετά απλή. Κάθε ένας μοιράζεται ένα κλειδί με όλους τους υπόλοιπους. Κατά την ανακοίνωση απλά προσθέτει

$bmod2$ όλα τα bit στα οποία έχει πρόσβαση και ανακοινώνει το αποτέλεσμα ή το αντίθετό του, ανάλογα με το αν δεν πλήρωσε ή όχι.

Μία υλοποίηση των DC-nets η οποία συνδυάζει τα δίκτυα DC-Net με τα δίκτυα μίξης είναι το Dissent.

11.3 Ψηφιακό χρήμα

Με την εξέλιξη της τεχνολογίας στον 20ο αιώνα και την υιοθέτηση των ηλεκτρονικών συναλλαγών προέκυψε η ανάγκη απόδρασης από το χαρτί και χειρισμός των νομισμάτων σε ηλεκτρονική μορφή. Έτσι ήταν εύλογη η ανάγκη δημιουργίας μόνο ηλεκτρονικών νομισμάτων. Κάτι τέτοιο όμως δεν αποδείχθηκε τόσο απλό.

Ένα ηλεκτρονικό νόμισμα είναι μια ακολουθία δυαδικών ψηφίων. Ως τέτοια δεν υπόκειται σε κάποιον φυσικό περιορισμό, οπότε μπορεί να αντιγραφεί δεκάδες φορές καθώς και να δημιουργηθεί εκ του μηδενός. Οι παραπάνω ενέργειες όμως δεν είναι αποδεκτές για οποιοδήποτε νόμισμα καθώς μειώνουν την αξία του.

Στη δεκαετία του 1980 υπήρξαν πολλές προτάσεις για λύση των εγγενών προβλημάτων του ηλεκτρονικού χρήματος βασιζόμενες στο νέο τότε κλάδο της *Κρυπτογραφίας Δημοσίου Κλειδιού* και των δυνατοτήτων που αυτή προσέφερε. Μία από τις πιο σημαντικές δόθηκε από τον David Chaum στα [5] και [6] και βασίζεται στην έννοια της *τυφλής υπογραφής* 7.6.2.

Η παραπάνω προσέγγιση μπορεί να περιγραφεί χρησιμοποιώντας την παρακάτω αναλογία:

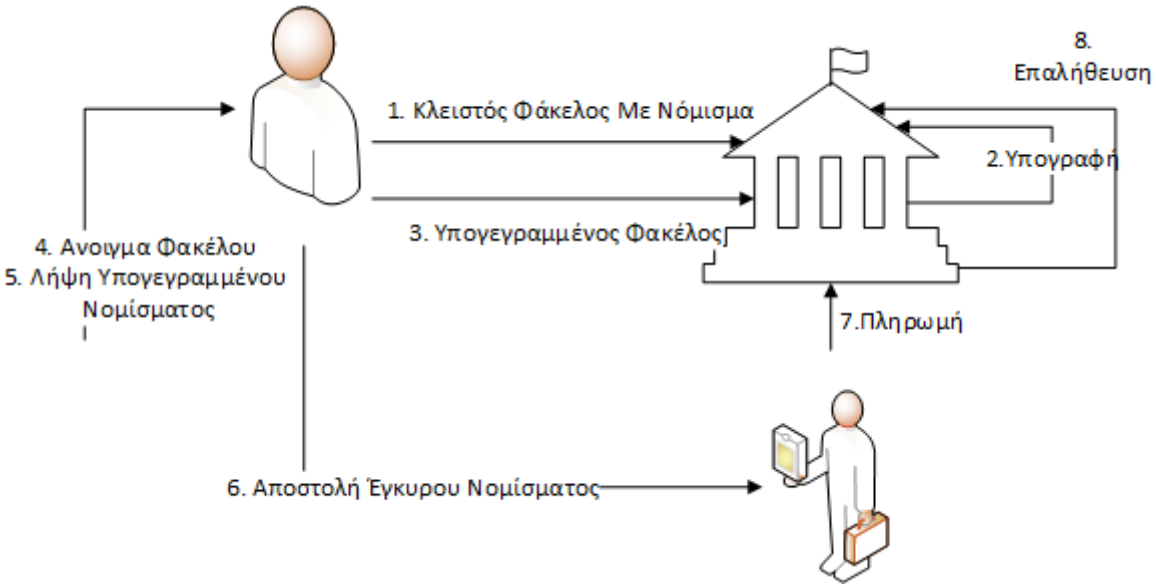
1. Στις τυφλές υπογραφές οι χρήστες θέλουν να πάρουν μία υπογραφή σε κάποια δεδομένα χωρίς ο υπογράφων να μάθει τα δεδομένα αυτά. Στην περίπτωση του νομίσματος τα δεδομένα είναι τα ποσά που ο χρήστης θέλει να ξοδέψει. Ο υπογράφων είναι η τράπεζα η οποία δεν πρέπει να μάθει τον σκοπό για τον οποίο ξοδεύει ο χρήστης τα χρήματα, αλλά πρέπει να εγκρίνει τη συναλλαγή, ελέγχοντας για παράδειγμα ότι ο χρήστης διαθέτει όντως το ποσό στον λογαριασμό του.

2. Ο χρήστης *τοποθετεί* το ποσό σε έναν φάκελο στο οποίο βάζει ένα κομμάτι καρμπόν. Επάνω στο φάκελο βάζει το ποσό που θέλει να ξοδέψει.

3. Η τράπεζα λαμβάνει τον κλειστό φάκελο. Ελέγχει αν το ποσό που πρόκειται να ξοδευτεί υπάρχει στο λογαριασμό του αποστολέα και αν ναι τον υπογράφει. Λόγω του καρμπόν, η υπογραφή μεταφέρεται στο περιεχόμενο του φακέλου.

4. Ο αποστολέας απομακρύνει τον φάκελο και λαμβάνει το υπογεγραμμένο περιεχόμενο με το οποίο πληρώνει τον παραλήπτη.

5. Ο παραλήπτης επαληθεύει την υπογραφή της τράπεζας και έχει πλέον ένα έγκυρο νόμισμα.



Σχήμα 11.6: Παραδοσιακό eCash - Chaum

Στο παραπάνω σενάριο τα δεδομένα που υπογράφονται αντιστοιχούν σε υποδιαιρέσεις του νομίσματος, κάθε μία από τις οποίες έχει την δική της ξεχωριστή υπογραφή, οπότε η τράπεζα και ο παραλήπτης δημιουργούν και επαληθεύουν αντίστοιχα τόσες υπογραφές όσες χρειάζεται για να δημιουργηθεί το ποσό. Αν μάλιστα τα δεδομένα που υπογράφονται είναι μοναδικά (πχ. με κάποιον σειριακό αριθμό) τότε, και με δεδομένο ότι το σύστημα υπογραφών είναι ασφαλές, δεν υπάρχει δυνατότητα πλαστογράφησης. Είναι σημαντικό επίσης να σημειωθεί ότι το παραπάνω σύστημα παρέχει ανωνυμία ως προς την τράπεζα, καθώς επιτρέπει τη συναλλαγή χωρίς να γνωρίζει η τράπεζα τον παραλήπτη.

Το μεγάλο πρόβλημα της παραπάνω προσέγγισης είναι ότι όλες οι συναλλαγές πρέπει να περάσουν μέσα από την κεντρική αρχή της τράπεζας. Αποτελεί έτσι ένα μοναδικό σημείο αποτυχίας του οποίου η κατάρρευση θα οδηγήσει σε καμία δυνατότητα εκτέλεσης συναλλαγές. Επίσης αν και δεν μπορεί να μάθει ποιος πληρώνει ποιον, μπορεί να αρνηθεί τις συναλλαγές κάποιου χρήστη, οδηγώντας σε άρνηση εξυπηρέτησης (Denial Of Service - DoS).

Το ερώτημα το οποίο προκύπτει, είναι αν μπορεί να σχεδιαστεί κάποιο σύστημα στο οποίο δεν χρειάζεται μία κεντρική αρχή και το οποίο έχει τις περισσότερες από τις παραπάνω ιδιότητες. Η απάντηση σε αυτό δόθηκε το 2008 και αξίζει να τις αφιερωθεί μία ξεχωριστή ενότητα.

11.4 Bitcoin

Οι συναρτήσεις σύνοψης (8.1) και οι εφαρμογές τους (8.4.4,8.4.1) έχουν βρει ευρύτατη εφαρμογή σε μια από τις πλέον επαναστατικές επινοήσεις της τελευταίας δεκαετίας, συγκεκριμένα στο κρυπτονόμισμα Bitcoin που προτάθηκε και υλοποιήθηκε από τον Satoshi Nakamoto [1] στην περίφημη εργασία του "Bitcoin: A Peer-to-Peer Electronic Cash System" [17] που δημοσίευσε το 2008 στην κρυπτογραφική λίστα "The Cryptography Mailing list at metzdowd.com".

Το Bitcoin είναι ένα διαρκώς εκτελούμενο πρωτόκολλο, η λειτουργία του οποίου βασίζεται στην πιστοποίηση των συναλλαγών συλλογικά και αποκεντρωμένα, μέσω της επισύναψης κάθε συναλλαγής σε μια μακριά αλυσίδα (*blockchain*) που περιέχει όλες τις συναλλαγές. Κάθε συναλλαγή επικυρώνεται μέσω της χρήσης μιας συνάρτησης σύνοψης (hash function) πάνω στη συνένωση της προηγούμενης συναλλαγής και του δημοσίου κλειδιού του παραλήπτη, με χρήση και ψηφίων τυχαιοποίησης (*nonce*). Κάθε συναλλαγή υπογράφεται με το ιδιωτικό κλειδί του αποστολέα και η υπογραφή μπορεί να επαληθευτεί με το δημόσιο κλειδί του αποστολέα, που περιέχεται στην αμέσως προηγούμενη συναλλαγή. Τελικά, η αλυσίδα λειτουργεί σαν ένας δημόσιος, συλλογικός μηχανισμός χρονοσήμανσης (timestamping) *χωρίς κεντρική αρχή*, αντίθεση με την ενοτητα 8.4.4.

Κομβικό ρόλο στη διαδικασία αυτή παίζει το λεγόμενο "Proof-of-Work" (*απόδειξη έργου*), μάλλον η πλέον εντυπωσιακή ιδέα του Nakamoto: για να είναι έγκυρη η σύνοψη, θα πρέπει να έχει συγκεκριμένη μορφή, με ένα συγκεκριμένο πλήθος αρχικών μηδενικών. Με άλλα λόγια, θα πρέπει το αποτύπωμα να βρίσκεται σε μια συγκεκριμένη (μικρή) περιοχή τιμών. Αυτό κάνει τον υπολογισμό του δύσκολο και χρονοβόρο, εάν βέβαια η συνάρτηση σύνοψης έχει τις επιθυμητές ιδιότητες τυχαιότητας. Έτσι, αν δύο "παίκτες" προσπαθούν ταυτόχρονα να επικυρώσουν μία συναλλαγή, κάποιος θα προηγηθεί του άλλου, και ο χρόνος που θα μεσολαβήσει θα είναι αρκετός ώστε οι υπόλοιποι παίκτες στο δίκτυο να επικυρώσουν μία από τις δύο συναλλαγές. Οι προσωρινές διακλαδώσεις της αλυσίδας είναι βέβαια αναπόφευκτες, αλλά στο Bitcoin υπάρχει ο τρόπος επίλυσης διαφωνιών: υπερισχύει η μακρύτερη αλυσίδα και οι υπόλοιπες απορρίπτονται.

Το Bitcoin συνδυάζει με εκπληκτικά ευφυή τρόπο θεμελιώδεις κρυπτογραφικές

[1] Πιθανότατα ψευδώνυμο, κανείς έως σήμερα δεν ξέρει την πραγματική ταυτότητα του δημιουργού του Bitcoin

τεχνικές όπως οι ψηφιακές υπογραφές, οι συναρτήσεις σύνοψης, τα δέντρα πιστοποίησης γνησιότητας (Merkle authentication trees), και πολλές ακόμη, προκειμένου να επιτύχει νομισματικές συναλλαγές χωρίς κεντρικό έλεγχο. Ακόμη περισσότερο, λειτουργεί σαν δημόσιος, αποκεντρωμένος μηχανισμός χρονοσήμανσης, κάτι οδήγησε ήδη σε διάδοχα κρυπτονομίσματα, όπου πλέον επικυρώνονται και δεδομένα εκτός από συναλλαγές (π.χ. στο Namecoin), ή και υπογράφονται ακόμη και ψηφιακά συμβόλαια (π.χ. στο Ethereum), τα οποία μπορούν να εκτελεστούν αξιόπιστα χωρίς τη συμμετοχή κεντρικών αρχών και δικηγόρων! Ίσως δεν είναι υπερβολή να πούμε ότι τα κρυπτονομίσματα είναι μια νέα επανάσταση ιστορικών διαστάσεων σε εξέλιξη.

Στην ενότητα που ακολουθεί θα περιγράψουμε σταδιακά τη λειτουργία του Bitcoin. Για διδακτικούς λόγους θα ακολουθήσουμε την προσέγγιση των [18] και [20] όπου περιγράφεται η δημιουργία του κρυπτονομίσματος βήμα προς βήμα. Υποθέτουμε ότι κάθε χρήστης του νομίσματος πρέπει να εφοδιαστεί με ένα ζεύγος δημοσίων και ιδιωτικών κλειδιών, ενός κοινού σχήματος ηλεκτρονικών υπογραφών που έχει συμφωνηθεί από όλους. Σε ένα τέτοιο σχήμα κάθε χρήστης μπορείς να ταυτοποιηθεί από τα δημόσια κλειδιά του, που λειτουργούν ως *ψευδώνυμα*.

11.4.1 Δημιουργία του Bitcoin (βήμα - βήμα)

Σχήμα 1 - Δημόσια Κλειδιά Ψευδώνυμα

Ένα απλό ηλεκτρονικό νόμισμα μπορεί να λειτουργήσει ως εξής: Κάθε φορά που κάποιος θέλει να πληρώσει ένα ποσό υπογράφει το ζεύγος (Ποσό, δημόσιο κλειδί παραλήπτη) με το ιδιωτικό κλειδί του και αποστέλλει το μήνυμα. Δηλαδή κάθε χρήστης δημιουργεί από το μηδέν ενά ποσό και το αποστέλλει στον παραλήπτη όπως αυτός προσδιορίζεται από το δημόσιο κλειδί. Φυσικά θα πρέπει να υπάρχουν κάποιοι κανόνες δημιουργίας, οι οποίοι πρέπει να τηρούνται. Αρχικά θα θεωρήσουμε τις συναλλαγές δημιουργίας χρημάτων ειδική περίπτωση και θα ασχοληθούμε με τις υπόλοιπες. Από την στιγμή, τώρα, που κάθε χρήστης έχει λάβει ένα ποσό, μπορεί να το ξοδέψει, δημιουργώντας μια συναλλαγή με ένα ίσο ποσό την οποία και θα απευθύνει σε κάποιο άλλο δημόσιο κλειδί (ή σε περισσότερα) την οποία και θα υπογράψει. Στην ουσία δηλαδή καταστρέφεται το παλιό ποσό και δημιουργείται ένα καινούριο ίδιας αξίας σε άλλους παραλήπτες. Για να είναι έγκυρη η συναλλαγή πρέπει να επαληθεύεται η υπογραφή, αλλά και η συναλλαγή από την οποία προήλθε να είναι με την σειρά της έγκυρη. Οι συναλλαγές δημιουργία χρημάτων είναι εξορισμού έγκυρες μόνο αν έχουν έγκυρες υπογραφές.

Η παραπάνω εκδοχή ενός αποκεντρωμένου ψηφιακού νομίσματος, προσφέρει κάποια επίπεδα προστασίας λόγω των κρυπτογραφικών τεχνικών που χρησιμοποιεί. Έχει όμως ένα βασικό πρόβλημα. Κάθε χρήστης μπορεί να αντιγράψει όσες φο-

ρές τα νομίσματα του (δηλ. τις εισερχόμενες συναλλαγές) και να τις ξοδέψει σε διαφορετικά δημόσια κλειδιά, ακόμα και σε κάποια που ανήκουν στον ίδιο. Το πρόβλημα αυτό ονομάζεται *double spending* και είναι το κυρίαρχο που πρέπει να επιλύσει οποιοδήποτε ψηφιακό νόμισμα.

Σχήμα 2 - Έμπιστη Αρχή και blockchain

Μία λύση στο παραπάνω πρόβλημα θα μπορούσε να δοθεί με την βοήθεια μιας έμπιστης αρχής (TTP). Κάθε συναλλαγή για να είναι έγκυρη πρέπει κατά την πραγματοποίηση της να στέλνεται στην αρχή η οποία ελέγχει ότι δεν υπάρχει απόπειρα double spending. Αν ο έλεγχος είναι επιτυχής η συναλλαγή δημοσιεύεται σε μία λίστα.

Η λίστα συναλλαγών δεν πρέπει να επιτρέπει στην αρχή να αλλάζει την ιστορία, δηλαδή να επεμβαίνει με οποιονδήποτε τρόπο (διαγραφή, αλλοίωση) σε προηγούμενες εγκεκριμένες συναλλαγές. Για τον σκοπό αυτό μπορεί να χρησιμοποιηθεί η αλυσίδα με συνόψεις δείκτες. Δηλαδή η αρχή όταν δημοσιεύει κάποια συναλλαγή, συμπεριλαμβάνει και ένα hash pointer προς την τελευταία δημοσιευμένη συναλλαγή. Με αυτό τον τρόπο οποιαδήποτε εκ των υστέρων τροποποίηση σε κάποιο ενδιάμεσο κόμβο της λίστας θα καταστήσει όλους τους επόμενους άκυρους. Για λόγους απόδοσης, η αρχή μπορεί να ομαδοποιεί τις συναλλαγές αντί να τις δημοσιεύει μία - μία. Έτσι μιλάμε πλέον για μπλοκ έγκυρων συναλλαγών, κάθε ένα από τα οποία δείχνει στο προηγούμενο μπλοκ έγκυρων συναλλαγών. Η αλυσίδα που προκύπτει θα ονομάζεται blockchain και όπως αναφέρθηκε είναι το βασικό συστατικό του Bitcoin.

Εφόσον έχουμε στη διάθεση μας μια έμπιστη αρχή μπορούμε να της εκχωρήσουμε την αρμοδιότητα δημιουργίας νομισμάτων. Επίσης εφ' όσον η αλυσίδα δημοσιεύεται η εγκυρότητα μιας συναλλαγής μπορεί να επιβεβαιωθεί από όλους, αν και με κάποια υπολογιστική επιβάρυνση. Παρά το γεγονός ότι το blockchain αποτρέπει ορισμένα είδη αλλοίωσης, η αρχή μπορεί να κλέψει *έμμεσα*. Για παράδειγμα μπορεί να μπλοκάρει τη δημοσίευση συναλλαγών από κάποιους χρήστες, μη επιτρέποντας τους στην ουσία να ξοδέψουν τα χρήματά τους. Επιπλέον μπορεί να απαιτήσει κάθε συναλλαγή να περιέχει οπωσδήποτε μία μεταβίβαση χρημάτων σε ένα δημόσιο κλειδί της, επιβάλλοντας πρακτικά ένα φόρο συναλλαγών. Επιπλέον όπως είδαμε μπορεί να είναι σημείο αποτυχίας (single point of failure) λόγω της ίδιας της φύσης της. Βέβαια για να είμαστε ακριβείς, παρά το γεγονός ότι υπάρχει κεντρική αρχή το παραπάνω σύστημα δεν είναι συγκεντρωτικό, καθώς δεν χρειάζεται έγκριση για να συμμετάσχει κάποιος στο σύστημα. Η συμμετοχή είναι τοπική διαδικασία, δηλαδή συμβαίνει αποκλειστικά στον υπολογιστή του κάθε χρήστη από την στιγμή που δημιουργεί τα κλειδιά χωρίς να χρειάζεται καμία έγκριση.

Σχήμα 3 - Τυχαία Επιλογή Αρχηγού

Το Bitcoin προκύπτει από το παραπάνω σχήμα αν εξαφανίσουμε την έμπιστη αρχή κατανέμοντας τις λειτουργίες της σε όλους τους χρήστες του συστήματος οι οποίοι πλέον γίνονται ισότιμοι. Συγκεκριμένα πρέπει:

- Όλοι να έχουν τη δυνατότητα δημιουργίας νομισμάτων.
- Όλοι να αποδέχονται τις ίδιες έγκυρες συναλλαγές και τη σειρά με την οποία έγιναν, χωρίς να υπάρχει κίνδυνος διπλοπληρωμών.

Με βάση τα παραπάνω οι χρήστες του Bitcoin, για να κάνουν μία συναλλαγή δεν την στέλνουν σε κάποια αρχή, αλλά την *εκπέμπουν* σε όλο το δίκτυο, το οποίο αποτελείται από *κόμβους* οι οποίοι διατηρούν δύο λίστες από συναλλαγές: αυτές που έχουν επιβεβαιωθεί ως έγκυρες και στις οποίες συμφωνούν όλοι οι κόμβοι, η οποία έχει την μορφή blockchain όπως νωρίτερα, καθώς και τις εκκρεμείς οι οποίες είναι εσωτερικές. Οποιοσδήποτε μπορεί να είναι κόμβος του Bitcoin, αρκεί να μπορεί να εκτελέσει το κατάλληλο λογισμικό ή και υλικό.

Το πρώτο ερώτημα που θα απαντήσουμε είναι το πώς ακριβώς οι κόμβοι του δικτύου φτάνουν στην συναίνεση για τις έγκυρες συναλλαγές: Όλοι οι κόμβοι μαζεύουν συναλλαγές τις οποίες και επικυρώνουν με βάση όσα ξέρουν, δηλ. την λίστα επιβεβαιωμένων συναλλαγών. Όταν μαζευτούν αρκετές επιβεβαιωμένες συναλλαγές δημοσιοποιούνται σε ένα μπλοκ. Από όλους τους κόμβους που δημοσιεύουν μπλοκ, επιλέγεται ένας *τυχαία* από τους υπόλοιπους. Αυτός ο κόμβος θα παίξει το ρόλο του δικτάτορα, με την έννοια ότι το δικό του μπλοκ θα προστεθεί στο blockchain. Αυτός δηλαδή θα υποδείξει για μια φορά ποιες είναι οι έγκυρες συναλλαγές.

Οι υπόλοιποι κόμβοι τώρα παίρνουν το νέο μπλοκ και ξεκινούν να το επεξεργάζονται ως εξής: Για κάθε συναλλαγή που περιέχει:

- Ελέγχουν την υπογραφή με το δημόσιο κλειδί του αποστολέα.
- Ελέγχουν ότι το ποσό που καταστρέφεται είναι ίδιο με το ποσό που δημιουργείται.
- Το ποσό που καταστρέφεται έχει δημιουργηθεί με κάποιο έγκυρο τρόπο.
- Το ποσό που καταστρέφεται δεν έχει ξανακαταστραφεί (διπλοπληρωμές).

Αν όλοι οι παραπάνω έλεγχοι αποδειχθούν έγκυροι, τότε οι κόμβοι *αποδέχονται* το νέο αυτό μπλοκ. Αυτό πρακτικά σημαίνει ότι όταν προτείνουν μελλοντικά το δικό τους επόμενο μπλοκ θα υπάρχει μία σύνοψη-δείκτης που θα δείχνει έστω

και έμμεσα προς το αποδεκτό τρέχον μπλοκ. Επιπλέον σημαίνει ότι οι εκκρεμείς συναλλαγές τους επικυρώνονται με αυτό δεδομένο. Τέλος για να ανταμείψουμε τον τυχαίο κόμβο για την τύχη του, το πρωτόκολλο τον αφήνει να δημιουργήσει ένα καινούριο χρηματικό ποσό από το μηδέν, το οποίο μπορεί να χρησιμοποιήσει όπως θέλει.

Το πρώτο ερώτημα που προκύπτει από την παραπάνω περιγραφή είναι πώς ακριβώς γίνεται η τυχαία επιλογή του κόμβου - δικτάτορα. Μία λύση που θα μπορούσε να δοθεί είναι μία κεντρική και έμπιστη πηγή τυχαιότητας, κάτι που στην κρυπτογραφία ονομάζεται (randomness beacon). Φυσικά κάτι τέτοιο αντιβαίνει στον στόχο μας ο οποίος είναι η κατάργηση των κεντρικών αρχών, αλλά ας το θεωρήσουμε σε αυτή την φάση δεδομένο και θα το καταργήσουμε στο επόμενο βήμα.

Το πραγματικό θέμα με την παραπάνω λύση αφορά το πώς επιτυγχάνεται ο συγχρονισμός της παραπάνω διαδικασίας σε ένα αχανές δίκτυο όπως είναι το Internet, όπου μηνύματα καθυστερούν να φθάσουν, κόμβοι βγαίνουν εκτός λειτουργίας αλλά και όλα δουλεύουν ταυτόχρονα. Τα δύο τελευταία χαρακτηριστικά στην περίπτωση μας είναι κατ' αρχήν θετικά γιατί αν κάποιος κόμβος δεν αποκριθεί σίγουρα θα υπάρχει κάποιος άλλος να προτείνει ένα μπλοκ. Επιπλέον η ταυτόχρονη λειτουργία παρέχει μία άμυνα εναντίον κακόβουλων κόμβων, με την έννοια ότι αν κάποιος κόμβος αρνηθεί μία συναλλαγή, σίγουρα θα υπάρξει κάποιος άλλος να την συμπεριλάβει.

Η παραπάνω προσέγγιση μαζί με τα χαρακτηριστικά του Διαδικτύου παρέχει επιπλέον μια ενδιαφέρουσα λύση στο πρόβλημα των διπλοπληρωμών. Στην πραγματικότητα δεν είναι λύση αλλά επανακαθορισμός του προβλήματος, με την απάντηση να βασίζεται σε τυπικούς κανόνες, με την έννοια του ότι έγκυρη συναλλαγή θεωρείται αυτή που μπαίνει στο blockchain και όχι αυτή την οποία θεωρούμε έγκυρη με βάση τους τυπικούς κανόνες εμπορίου. Για παράδειγμα αν ένας χρήστης αγοράσει ένα αγαθό με μια συναλλαγή και αμέσως μετά μεταφέρει το ίδιο νόμισμα σε μια δική του διεύθυνση, είτε επειδή είναι κακόβουλος και θέλει να διπλοξοδέψει ή επειδή απλά μετάνιωσε, τότε η έγκυρη συναλλαγή είναι απλά αυτή που θα εμφανιστεί στην αλυσίδα. Λόγω των χαρακτηριστικών του διαδικτύου, πχ. καθυστερήσεις, υπάρχει και η πιθανότητα η δεύτερη χρονικά (με τον ανθρώπινο χρόνο) συναλλαγή να καταλήξει να θεωρείται έγκυρη.

Για να προστατευθεί ο έγκυρος παραλήπτης μιας συναλλαγής (πχ. ένας έμπορος) θα πρέπει να παρακολουθεί το δίκτυο του Bitcoin και συγκριμένα να δημοσιευθεί το μπλοκ που περιέχει την μεταφορά του ποσού στη διεύθυνση του. Αν εκείνη την στιγμή παρέχει την υπηρεσία τότε ενδεχομένως να πέσει θύμα της περίπτωσης που περιγράψαμε στην προηγούμενη παράγραφο, λόγω των καθυστερήσεων που εισάγονται στο δίκτυο.

Συμπερασματικά, το παραπάνω σχήμα θα ήταν αποδεκτό μόνο αν υπήρχε τρόπος να επιλεχθεί πραγματικά τυχαία ο κόμβος που προτείνει το επόμενο μπλοκ και αν δεν υπήρχαν καθυστερήσεις στο δίκτυο. Το Bitcoin λύνει και τα δύο αυτά προβλήματα με τον μηχανισμό Proof Of Work που θα δούμε στη συνέχεια.

Σχήμα 4 - Επιλογή με Proof Of Work

Όπως προαναφέραμε σε ένα κατανεμημένο σύστημα όπως είναι το Bitcoin, δεν μπορούμε να βασιστούμε σε οποιαδήποτε κεντρική αρχή για οτιδήποτε, ακόμα και για την τυχαιότητα που χρειάζεται στην επιλογή κόμβου. Η λύση που δίνεται στο παρακάτω πρόβλημα είναι ο μηχανισμός που ονομάζεται *Proof Of Work* στο Bitcoin. Συγκεκριμένα, ο κόμβος που θα προτείνει το επόμενο μπλοκ θα πρέπει μαζί με αυτό να δώσει και μια απόδειξη ότι έλυσε ένα πρόβλημα, του οποίου στην πραγματικότητα η λύση είναι τυχαίο να βρεθεί. Για παράδειγμα, θα μπορούσε να προβλέψει το αποτέλεσμα της τιμής μιας συνάρτησης σύνοψης, που όπως προαναφέραμε συμπεριφέρεται ως τυχαία μεταβλητή. Όποιος καταφέρει και υπολογίζει το *Proof Of Work* δημοσιεύει το επόμενο block και γίνεται υποψήφιος για να εισπράξει την ανταμοιβή που ορίζεται σε 25 bitcoins (BTC). Το συγκεκριμένο ποσό, θα μπορούσε να πει κανείς, πως ανήκει στη δεύτερη γενιά ανταμοιβής. Αρχικά κάθε miner λάμβανε 50BTC. Κάθε 210.000 μπλοκ η αμοιβή μειώνεται στο μισό. Κάποια στιγμή το 2016 η αμοιβή θα μειωθεί ξανά στα 12.5BTC. Υπολογίζεται ότι περίπου στο 2140 θα έχουν παραχθεί όλα τα bitcoins που προβλέπονται. Τότε η αμοιβή των miners θα προκύπτει ως αμοιβή εκκαθάρισης συναλλαγής.

Πιο συγκεκριμένα, το πρόβλημα που καλείται να λύσει ο κάθε κόμβος προκειμένου να προτείνει το επόμενο block είναι το εξής:

Ποια είναι η τιμή τυχαιότητας (nonce), η οποία αν δωθεί ως τιμή εισόδου στην συνάρτηση σύνοψης SHA256 μαζί με μία σύνοψη των συναλλαγών του μπλοκ και την σύνοψη δείκτη του τελευταίου αποδεκτού μπλοκ, θα έχει αποτέλεσμα μικρότερο από κάποια προκαθορισμένη τιμή;

Δηλαδή ζητείται ο υπολογισμός της τιμής:

$$f(h, block, \epsilon) = nonce : \mathcal{H}(nonce||h||block) \leq \epsilon$$

Καταρχήν, πρέπει να παρατηρήσουμε ότι το παραπάνω πρόβλημα είναι δύσκολα υπολογίσιμο καθώς για οποιαδήποτε είσοδο η συνάρτηση σύνοψης συμπεριφέρεται τυχαία πρακτικά. Προσεγγίζεται έτσι η απαίτηση που είχαμε στο σχήμα 3, για την τυχαία επιλογή ενός κόμβου.

Επιπλέον δεν υπάρχουν υπολογιστικές συντομεύσεις. Η μόνη στρατηγική που υπάρχει για την εύρεση του nonce, είναι η εξαντλητική δοκιμή (brute force) τιμών μέχρι να βρεθεί κάποια αποδεκτή. Η διαδικασία αυτή λέγεται *Bitcoin mining*

και οι κόμβοι που συμμετέχουν σε αυτό *miners*.

Το mining μπορεί να είναι υπολογιστικά δύσκολο, ταυτόχρονα όμως είναι εύκολα *επαληθεύσιμο* (ανήκει δηλαδή στην κλάση προβλημάτωνNP. Οι τιμές $h, block, \epsilon$ είναι δημόσια διαθέσιμες και αν δοθεί το nonce μπορεί με υπολογιστική ευκολία να ελεγχθεί από *οποιονδήποτε* ότι είναι κάτω από την επιλεγμένη τιμή. Άρα όταν κάποιος κόμβος προτείνει κάποιο μπλοκ, οι υπόλοιποι μπορούν εύκολα να επαληθεύσουν το *Proof Of Work*.

Επιπλέον πρέπει να παρατηρήσουμε ότι με την τιμή ϵ ελέγχεται στην ουσία η δυσκολία του προβλήματος. Αν η τιμή της είναι μεγάλη τότε πολλές τιμές θα ικανοποιούν αυτή τη σχέση και άρα θα υπάρχει πολύ μεγάλη πιθανότητα να βρεθούν. Αντίθετα μια μικρή τιμή θα δυσκόλευε περισσότερο την εύρεση της απόδειξης. Η τιμή του ϵ δεν είναι σταθερή, αλλά αλλάζει με στόχο ο χρόνος εξόρυξης να είναι σταθερός (περίπου 10 λεπτά). Ο επανακαθορισμός της τιμής γίνεται περίπου κάθε 2 εβδομάδες.

Από την παραπάνω περιγραφή είναι φανερό ότι η πιθανότητα εύρεσης του proof of work εξαρτάται από την υπολογιστική ισχύ του κάθε miner σε σχέση φυσικά με την υπολογιστική ισχύ όλων των κόμβων. Έτσι είναι φανερό ότι αυτό που παίζει ρόλο είναι η υπολογιστική ισχύς των κόμβων που συμπεριφέρονται *σωστά* σε σχέση με τους υπόλοιπους κόμβους. Αν δηλαδή κάποιος κόμβος καταφέρει να ελέγξει πάνω από το 50% της υπολογιστικής ισχύος του δικτύου τότε έχει αντίστοιχη πιθανότητα να γίνει αποδεκτή η πρόταση του για το επόμενο μπλοκ.

Για να συλλεχθεί η ανταμοιβή από κάποιον miner πρέπει το μπλοκ που πρότεινε να μπει στο block chain. Για να συμβεί αυτό πρέπει να προστεθούν νέα μπλοκ μετά από αυτό από άλλους κόμβους. Στην παρούσα μορφή αυτό συμβαίνει μετά από 6 μπλοκ, δηλαδή περίπου μετά από μία ώρα. Με αυτή την λογική κάθε miner έχει συμφέρον να προσπαθεί να επεκτείνει την μεγαλύτερη αλυσίδα που έχει δει μέχρι εκείνη τη στιγμή, γιατί αλλιώς θα χρειασθούν περισσότερα μπλοκ για να συλλεχθεί η ανταμοιβή. Έτσι είναι συχνό το φαινόμενο ένας miner να εγκαταλείπει την επεξεργασία ενός μπλοκ γιατί δημοσιεύτηκε κάποιο άλλο. Επίσης είναι συχνό το φαινόμενο δύο miner να δημοσιεύουν σχεδόν ταυτόχρονα δύο νέα μπλοκ, οπότε το ένα από τα δύο τελικά θα εγκαταλειφθεί (orphan block). Φυσικά οι συναλλαγές που περιέχει είτε θα δημοσιευτούν από τον ίδιο αργότερα, είτε θα δημοσιευτούν από άλλο κόμβο.

11.4.2 Πρακτικά Θέματα

Στην ενότητα αυτή θα περιγράψουμε πώς χρησιμοποιείται το Bitcoin από τους χρήστες και άλλες πρακτικές πτυχές του πρωτοκόλλου.

Χρήστες

Κάθε χρήστης μπορεί να συμμετάσχει στο δίκτυο με δύο ρόλους: Μπορεί να είναι είτε απλός πελάτης του πρωτοκόλλου, δηλαδή απλά να στέλνει και να λαμβάνει χρήματα. Για τον σκοπό αυτό πρέπει να εγκαταστήσει στον υπολογιστή του το σχετικό λογισμικό, το οποίο ονομάζεται *Bitcoin wallet*. Εναλλακτικά, μπορεί να χρησιμοποιήσει εφαρμογές παγκοσμίου ιστού, για τον σκοπό αυτό σε αναλογία δηλαδή με τους πελάτες ηλεκτρονικού ταχυδρομείου.

Όλοι οι πελάτες bitcoin έχουν παρόμοιες λειτουργίες: Αρχικά επιτρέπουν στους χρήστες να δημιουργήσουν ένα 'πορτοφόλι', το οποίο είναι απλά η συλλογή δημοσίων και ιδιωτικών κλειδιών - αρχικά δημιουργείται επίσης και ζεύγος από ένα ιδιωτικό και ένα δημόσιο κλειδί. Το δημόσιο κλειδί, όπως προαναφέραμε λειτουργεί ως ψευδώνυμο του χρήστη και του επιτρέπει να λάβει bitcoins. Για την ακρίβεια το ψευδώνυμο δεν είναι το ίδιο το δημόσιο κλειδί, αλλά αφού περάσει από τις συναρτήσεις σύνοψης SHA256 και RIPEMD160. Το αποτέλεσμα ονομάζεται *διεύθυνση* και ξεκινάει από 1 ή 3. Το ιδιωτικό κλειδί πρέπει να προστατεύεται από τους χρήστες καθώς η απώλεια του, ουσιαστικά ισοδυναμεί με την απώλεια όλων των χρημάτων που έχουν αποσταλεί στο αντίστοιχο δημόσιο κλειδί. Μία πρακτική, η οποία θα εξηγηθεί αναλυτικά στη συνέχεια, θέλει κάθε συναλλαγή να γίνεται με καινούριο ζεύγος κλειδιών και υλοποιείται από όλους σχεδόν τους πελάτες Bitcoin.

Σε κάθε περίπτωση εφόσον δημιουργηθούν τα κλειδιά, οι χρήστες μπορούν να συμμετέχουν σε συναλλαγές. Πρώτα όμως πρέπει να αποκτήσουν bitcoins. Για τον σκοπό αυτόν, στην περίπτωση πάντα των απλών πελατών, πρέπει να τα ανταλλάξουν με κάποιο *παραδοσιακό* νόμισμα σε ανταλλακτήρια. Τέτοια είναι το Bitstamp για την Ευρώπη και το Coinbase για τις ΗΠΑ. Φυσικά μια τέτοια συναλλαγή, μπορεί να γίνει και με ανεπίσημο τρόπο, πχ. μέσω του κοινωνικού περίγυρου. Για να πραγματοποιηθεί μια συναλλαγή, ο αποστολέας απλά χρειάζεται να συμπληρώσει την διεύθυνση του παραλήπτη και το ποσό. Για να προφυλαχθούν οι χρήστες από λάθη, οι διάφοροι πελάτες δίνουν τη δυνατότητα εξαγωγής των διευθύνσεων σε QR codes και εισαγωγή τους μέσω φωτογραφικής μηχανής. Προαιρετικά, μπορούν να εισάγουν ένα *τέλος συναλλαγής (transaction fee)* για να επεξεργαστεί η συναλλαγή από τους miners ταχύτερα.

Σε αυτό το σημείο, πρέπει να αναφέρουμε την σχέση του Bitcoin με τα παραδοσιακά νομίσματα. Τη στιγμή που γράφεται αυτό το κείμενο 1BTC ισούται περίπου με 245EUR. Τον τελευταίο χρόνο μάλιστα το Bitcoin έχει χάσει μεγάλο μέρος της αξίας τους λόγω διαφόρων συμβάντων ασφαλείας που θα αναλύσουμε στη συνέχεια. Από την παραπάνω ισοτιμία προκύπτει ότι 1BTC έχει αρκετά μεγάλη αξία. Έτσι πρακτικά χρησιμοποιούνται οι υποδιαιρέσεις του. Η μικρότερη υποδιαίρεση ονομάζεται satoshi και είναι ίσο με 10^{-8} BTC.

Εκτός από τους χρήστες πελάτες, υπάρχει μια πιο απλή κατηγορία χρηστών του Bitcoin. Ονομάζονται κόμβοι SPV (Simple Payment Verification). Τέτοιο χρήστες είναι οι χρήστες που περιμένουν να λάβουν πληρωμές όπως για παράδειγμα οι έμποροι. Στόχος τους είναι να δουν πότε μια πληρωμή θεωρείται έγκυρη, ώστε να προωθήσουν στην παραλαβή των προϊόντων στους πελάτες τους. Για τον σκοπό αυτό κατεβάζουν μόνο τις επικεφαλίδες των μπλοκ στην αλυσίδα και όχι ολόκληρο το block. Στην περίπτωση που θέλουν να ελέγξουν αν έχει γίνει μία πληρωμή σε μια διεύθυνση επικοινωνούν με πλήρεις κόμβους κάνοντας μια αναζήτηση για τη διεύθυνση τους. Αν βρεθεί συναλλαγή σε κάποιο μπλοκ που την αφορά τότε οι πλήρεις αυτοί κόμβοι επιστρέφουν ένα μονοπάτι merkle για αυτό. Ο πελάτης SPV τότε επαληθεύει ότι η ρίζα περιέχεται στο μπλοκ που έχει κατεβάσει ο ίδιος ανεξάρτητα, και ότι το μονοπάτι στο merkle tree επαληθεύει τη συναλλαγή. Αν όλοι αυτοί οι έλεγχοι επαληθευτούν τότε ο έμπορος είναι αρκετά σίγουρος ότι θα πληρωθεί και μπορεί να αποστείλει τα προϊόντα. Ένας τέτοιος κόμβος είναι πολύ ελαφρύς στη λειτουργία. Η όλη λειτουργία μπορεί να υλοποιηθεί με ανταλλαγή μηνυμάτων περίπου 1KB/block, αντί για 1MB/block.

Miners

Εναλλακτικά, κάποιος χρήστης μπορεί να συμμετάσχει στο πρωτόκολλο ως εξυπηρετητής ή πλήρης κόμβος, και να συμμετάσχει ενεργά στο mining. Για τον σκοπό αυτό υπάρχει σχετικό λογισμικό, αλλά για λόγους ταχύτητας πολλές φορές χρησιμοποιείται υλικό για τη λειτουργία αυτή.

Οι πλήρεις κόμβοι διατηρούν ολόκληρη την αποδεκτή λίστα συναλλαγών η οποία έχει μέγεθος περίπου 30GB. Επιπλέον κάθε miner διατηρεί μία λίστα από εκκρεμείς συναλλαγές. Όταν ξεκινάει να δημιουργήσει ένα μπλοκ επιλέγει κάποιες από αυτές με βάση κριτήρια όπως αξία, ηλικία και αμοιβή για τον ίδιο. Η αμοιβή προκύπτει ως διαφορά εισόδου και εξόδου. Η πρώτη συναλλαγή που μπαίνει σε κάθε μπλοκ ονομάζεται coinbase transaction ή generation transaction και είναι αυτή που θα οδηγήσει στην πληρωμή του miner για την υπολογιστική του πρόταση.

Από την στιγμή που θα εξορυχθεί ένα νέο μπλοκ, η πρώτη ενέργεια κάθε κόμβου είναι να το μεταδώσει στους γείτονες του στο δίκτυο. Κάθε ένας από τους τελευταίους ξεκινάει την επαλήθευση του μπλοκ με βάση συγκεκριμένους συντακτικούς και λογικούς κανόνες (πχ. το μέγεθος να είναι συγκεκριμένο ή να υπάρχει μόνο μία coinbase transaction). Στη συνέχεια προσπαθεί να το εισαγάγει στην κύρια αλυσίδα, κοιτώντας την σύνοψη δείκτη που περιέχει. Τις περισσότερες φορές θα δείχνει στο πιο πρόσφατο μπλοκ της αλυσίδας. Σε διαφορετική περίπτωση θα κατασκευαστεί μία δεύτερη αλυσίδα (μπορεί να υπάρχουν ήδη και περισσότερες). Σε κάθε περίπτωση τις συγκρίνει ώστε να διαπιστώσει ποια περιέχει την περισσότερη δουλειά (συνήθως είναι αυτή που έχει τα περισσότερα μπλοκ). Αυτή γίνεται

πλέον κύρια. Όλοι οι κόμβοι έχουν την ίδια πολιτική με αποτέλεσμα σταδιακά όλοι να συμφωνούν στην ίδια αλυσίδα, δηλαδή στην ίδια ιστορία συναλλαγών.

Συναλλαγές

Οι συναλλαγές είναι το πιο σημαντικό συστατικό του πρωτοκόλλου του Bitcoin καθώς αποτελούν τα περιεχόμενα της αλυσίδας συναλλαγών. Κάθε συναλλαγή έχει μία ή περισσότερες εισόδους και μία ή περισσότερες εξόδους, και εκπέμπεται (σταδιακά) σε όλο το δίκτυο. Το χρηματικό ποσό (balance) που όλοι οι χρήστες συμφωνούν ότι κατέχει ο κάθε συναλλασσόμενος, είναι το άθροισμα των εξόδων όλων των συναλλαγών που απευθύνονται στα δημόσια κλειδιά που κατέχει και δεν έχουν ξοδευτεί. Αυτό αναφέρεται στο Bitcoin ως *UTXO (Unspent Transaction Output)* και υπολογίζεται συνέχεια από τους πελάτες για να παρουσιαστεί στους χρήστες το υπόλοιπό τους.

Τεχνικά κάθε συναλλαγή έχει περίπου 400 bytes. Το μικρό αυτό μέγεθος επιτρέπει τη μετάδοση τους σε οποιοδήποτε είδος δικτύου, είτε στο Internet, είτε μέσω Bluetooth και NFC σε πύλες εισόδου στο Bitcoin. Κάθε μπλοκ περιέχει εώς και 1000 συναλλαγές μέχρι το όριο του 1MB . Χρειάζεται πολύ συχνά να επαληθεύεται το αν μία συναλλαγή περιέχεται σε ένα μπλοκ χωρίς μεταβολές. Η αποδοτική απάντηση σε αυτό το ερώτημα, αποκλείει την σειριακή αναζήτηση λόγω της συχνότητας με την οποία γίνεται και οδηγεί στην χρήση των δένδρων Merkle που περιγράψαμε στο 8.5.

Εσωτερικά, η δομή δεδομένων μιας συναλλαγής είναι πολύ απλή. Περιέχει μία επικεφαλίδα, μία ή περισσότερες εισόδους και μία ή περισσότερες εξόδους, που αναπαριστούν τα χρηματικά ποσά που ανταλλάσσονται. Τα ποσά αυτά αναφέρονται ως ακέραια πολλαπλάσια των satoshi. Τέλος μια συναλλαγή περιέχει ένα ακόμα πεδίο το οποίο ονομάζεται *Locktime* που αναφέρει από ποια στιγμή και μετά μπορεί να προστεθεί η συναλλαγή στο δίκτυο.

Από τα παραπάνω προκύπτει ότι τα χρηματικά ποσά στο Bitcoin είναι *αμετάβλητα* (immutable). Δηλαδή, δεν τροποποιούνται, μόνο καταστρέφονται και δημιουργούνται. Έτσι αν κάποιος χρήστης έχει UTXO 20BTC και θέλει να ξοδέψει το 1BTC, θα πρέπει να βρει από τις εισερχόμενες συναλλαγές αυτή που είναι πιο κοντά στο ποσό και να την προσθέσει ως είσοδο σε μία συναλλαγή. Από αυτήν δεν θα μπορεί να ξοδέψει τμήμα μιας συναλλαγής. Θα πρέπει να καταστρέψει το ποσό, να πληρώσει το 1BTC και το αποτέλεσμα να το μεταφέρει ως έξοδο, να πληρώσει δηλαδή τον εαυτό του, ώστε πιστωθεί στο υπόλοιπο του ως ένα ακόμα UTXO.

Οι είσοδοι μιας συναλλαγής είναι αναφορές σε UTXO. Οι έξοδοι κάθε συναλλαγής έχουν και οι ίδιες εσωτερική δομή. Σε αυτήν περιλαμβάνεται μία περιορισμένη scripting γλώσσα, η οποία μπορεί να χρησιμοποιηθεί για τη δημιουργία απλών

'προγραμμάτων'. Το πιο σημαντικό από αυτά αναφέρεται ως *locking script ή επιβάρυνση* και καθορίζει την συνθήκη που πρέπει να ικανοποιείται για να ξοδευτούν τα bitcoins. Βασίζεται σε στοίβα, όπου τοποθετούνται τιμές. Οι εντολές τις αποτιμούν. Για να είναι έγκυρη η συναλλαγή πρέπει η μόνη τιμή που θα έχει απομείνει στη στοίβα να είναι το TRUE, όπως φαίνεται στο παρακάτω παράδειγμα:

2 7 OP_ADD 3 OP_SUB 1 OP_ADD 7 OP_EQUAL

Αυτό αποτιμάται σε TRUE, άρα η συναλλαγή είναι έγκυρη. Ένα πιο ρεαλιστικό παράδειγμα είναι το:

OP_DUP OP_HASH160 OP_EQUAL OP_CHECKSIG

το οποίο επαληθεύει ότι η υπογραφή μιας συναλλαγής έγινε από τον κάτοχο του ιδιωτικού κλειδιού της διεύθυνσης αποδοχής. Η αποτίμηση του θα ακολουθήσει τα εξής βήματα.

- Αρχικά η στοίβα είναι κενή [].
- Στην συνέχεια προστίθεται η υπογραφή [signature].
- Μετά το δημόσιο κλειδί [signature, public_key]
- το οποίο και αντιγράφεται [signature, public_key, public_key]
- Στο αντίγραφο εφαρμόζεται η συνάρτηση σύνοψης [signature, public_key, hash160(public_key)]
- Στην συνέχεια ελέγχεται προστίθεται η διεύθυνση του παραλήπτη [signature, public_key, hash160(public_key), hash160(public_key)]
- Ελέγχεται αν η διεύθυνση ταυτίζεται με τη σύνοψη του δημοσίου κλειδιού οπότε και αφαιρούνεται από τη στοίβα [signature, public_key]
- Τέλος ελέγχεται αν το δημόσιο κλειδί επαληθεύει την υπογραφή. Αν όλα πάνε καλά στην στοίβα παραμένη η τιμή [True] η οποία επαληθεύει και τη συναλλαγή.

Μία πολύ ενδιαφέρουσα εντολή που παρέχει η παραπάνω γλώσσα είναι η OP_RETURN. Η εντολή αυτή επιτρέπει την δημιουργία 40bytes από *μη-χρηματικά δεδομένα* (δεν υπολογίζονται στο UTXO) και την εισαγωγή τους στο blockchain. Η ύπαρξη του έχει δώσει την δυνατότητα χρήστης της υποδομής του bitcoin και για άλλους σκοπούς. Τέλος, όπως προαναφέρουμε η διαφορά μεταξύ των εισόδων και των εξόδων παρακρατείται από τους miners ως αποζημίωση για την επεξεργασία τους.

Συναλλαγές Με Εγγυητές (Multisig / Escrow Transactions)

Από την περιγραφή που προηγήθηκε γίνεται φανερό ότι οι συναλλαγές στο Bitcoin είναι αμετάκλητες, δηλαδή δεν υπάρχει τρόπος να ακυρωθούν από την στιγμή που αποτελούν μέρος της κοινά αποδεκτής αλυσίδας. Πολλές φορές όμως είναι απαραίτητος ένας μηχανισμός ανάκλησης ώστε να υπάρχει προστασία από τόσο του αγοραστή όσο και του πωλητή από λάθη, αλλά και από προσπάθειες εξαπάτησης.

Ο μηχανισμός αυτός στο Bitcoin είναι η χρήση συναλλαγών με πολλαπλές υπογραφές (multisig) ή με εγγυητές. Υπάρχουν αρκετές παραλλαγές. Οι κυριότερες είναι:

Συναλλαγές 2-από-2: Εδώ ο αγοραστής πρέπει να συνεννοηθεί με τον πωλητή ώστε να δημιουργήσουν μία διεύθυνση η οποία να χρειάζεται υπογραφές και από τους δύο. Στη συνέχεια η πληρωμή γίνεται από τον αγοραστή σε αυτή τη διεύθυνση. Εν συνεχεία ο πωλητής αποστέλλει το προϊόν. Όταν αυτό παραληφθεί και οι δύο χρησιμοποιώντας τα κλειδιά τους υπογράφουν μία συναλλαγή η οποία στέλνει τα νομίσματα στον πωλητή. Αν η συναλλαγή ακυρωθεί το ποσό στέλνεται πίσω στον αποστολέα.

Συναλλαγές 2-από-3: Εδώ οι συναλλασσόμενοι πρέπει να βρουν έναν έμπιστο εγγυητή (όπως το clearcoin). Στη συνέχεια δημιουργούν μία διεύθυνση η οποία χρειάζεται 2 από 3 ιδιωτικά κλειδιά τα οποία θα διανεμηθούν στον αγοραστή στον πωλητή και τον εγγυητή. Η διαδικασία που ακολουθεί είναι: Ο αγοραστής στέλνει το ποσό σε αυτή τη διεύθυνση. Ο πωλητής στέλνει το προϊόν. Αν όλα πάνε καλά και συμφωνήσουν αγοραστής και πωλητής τότε υπογράφουν μία συναλλαγή η οποία αποστέλλει το ποσό από τη διεύθυνση στον πωλητή.

Ουσιαστικά μέχρι εδώ δεν υπάρχει η επέμβαση του εγγυητή και όλα βαίνουν καλώς χωρίς να υπάρχει καμία διαφοροποίηση από τις συναλλαγές 2-από-2. Η χρησιμότητα των συναλλαγών αυτών και συγκεκριμένα του εγγυητή γίνεται εμφανής σε περίπτωση που κάτι πάει στραβά. για διάφορους λόγους. Αν για παράδειγμα το προϊόν δεν παραδοθεί, τότε ο πωλητής παραπονιέται στον εγγυητή. Αυτός επαληθεύει το γεγονός αυτό και υπογράφει μία συναλλαγή στέλνοντας το ποσό από την κοινή διεύθυνση πίσω στον αγοραστή. Από την άλλη, αν το προϊόν όντως φθάσει αλλά ο αγοραστής δεν θέλει να πληρώσει τότε ο εγγυητής συνασπίζεται με τον πωλητή και υπογράφουν μία συναλλαγή η οποία στέλνει το ποσό στον τελευταίο.

11.4.3 Προβλήματα - Επιθέσεις

Για να κατανοήσουμε καλύτερα την λειτουργία του Bitcoin θα περιγράψουμε κάποιες πιθανές επιθέσεις αλλά και προβλήματα που έχουν προκύψει από την χρήση του Bitcoin. Οι επιθέσεις που θα αναφέρουμε παρακάτω δεν σχετίζονται με τους κρυπτογραφικούς αλγορίθμους που χρησιμοποιούνται, οι οποίοι είναι μαθηματικά ασφαλείς, υπό τις γνωστές βέβαια υποθέσεις.

Επιθέσεις στη συναίνεση

Αν ένα μεγάλο πλήθος από miners με σημαντική δύναμη στη δυνατότητα εξόρυξης συνεργαστεί, μπορεί να επιτεθεί στο πρωτόκολλο του Bitcoin και να καταφέρει να διπλοξοδέψει κάποιο νόμισμα ή να αρνηθεί να εξυπηρετήσει κάποιον χρήστη. Αρχικά αυτή η επίθεση ονομάστηκε επίθεση 51%, γιατί είχε μεγάλες πιθανότητες επιτυχίες αν ένας συνασπισμός είχε υπό τον έλεγχο του το 51% της δύναμης εξόρυξης. Έχουν υπάρξει όμως και ερευνητικές εργασίες που επιτυγχάνουν το ίδιο ποσοστό με ποσοστό 30%.

Ας δούμε όμως πως λειτουργεί αυτή η επίθεση με ένα παράδειγμα:

Έστω ένας χρήστης ο οποίος θέλει να εξαπατήσει έναν πωλητή αγαθών. Αφού λοιπόν συμφωνήσουν στην τιμή ο αγοραστής αποφασίζει να δημιουργήσει μία συναλλαγή η οποία θα μεταφέρει κάποια UTXO σε μια διεύθυνση του πωλητή.

Ο πωλητής τώρα έχει τις εξής επιλογές. Μπορεί να αποστείλει αμέσως τα αγαθά, να περιμένει μέχρι η συναλλαγή να εμφανιστεί σε κάποιο καινούριο μπλοκ που μόλις έχει εξορυχθεί ή να περιμένει να εξορυχτούν κάποια μπλοκ μετά από αυτό που περιέχει την συναλλαγή του. Η επίθεση 51% είναι δυνατή στις 2 πρώτες περιπτώσεις, ενώ απαιτεί σημαντικά περισσότερη υπολογιστική ισχύ στις υπόλοιπες.

Αμέσως μετά την απελευθέρωση των αγαθών λοιπόν στις 2 πρώτες περιπτώσεις, ο αγοραστής δημιουργεί μία νέα συναλλαγή μεταφοράς του ίδιου ποσού από τα ίδια UTXO σε μια διεύθυνση που ο ίδιος ελέγχει. Αν ο αγοραστής ελέγχει σημαντική δύναμη εξορυξης μπορεί να εξάγει ένα μπλοκ που θα περιέχει τη δεύτερη συναλλαγή. Φυσικά αυτό θα πρέπει να επεκτείνει το αμέσως προηγούμενο μπλοκ από αυτό που περιέχει την έγκυρη συναλλαγή. Αν η δύναμη εξόρυξης είναι πάνω από το 51%, τότε παραπάνω από τα μισά νέα μπλοκ που εξάγονται θα προέρχονται από τον ίδιο και άρα θα επεκτείνουν το δεύτερο μπλοκ, μεγαλώνοντας αυτή την αλυσίδα η οποία σε λίγο θα γίνει κυρίαρχη. Ως αποτέλεσμα ο έμπορος θα έχει στείλει τα αγαθά, χωρίς όμως να πιστωθεί η συναλλαγή στο λογαριασμό του.

Η επίθεση αυτή μπορεί να πραγματοποιηθεί αν ο έμπορος περιμένει να εμφανιστούν αρκετά μπλοκ μετά το δικό του, προτού αποστείλει τα αγαθά. Μία γενικά αποδεκτή τιμή είναι να περάσουν 6 μπλοκ. Μία παραλλαγή της παραπάνω επίθεσης αφορά την άρνηση εξυπηρέτησης σε συγκεκριμένες διευθύνσεις, με αποτέ-

λεσμα να μην μπορούν ποτέ οι συναλλαγές τους να εμφανιστούν στη μεγαλύτερη αλυσίδα.

Συγκεντρωτισμός

Στην αρχή της ιστορίας του Bitcoin οποιοσδήποτε χρήστης μπορούσε να συμμετάσχει ως miner στο πρωτόκολλο, χρησιμοποιώντας λογισμικό το οποίο θα μπορούσε να εκτελέσει οποιοσδήποτε υπολογιστής στον χρόνο αδράνειάς του. Αυτό καθιστούσε τους συμμετέχοντες πραγματικά ισότιμους. Με την εξέλιξη του ενδιαφέροντος για το πρωτόκολλο, υπήρξε μεγαλύτερος ανταγωνισμός για το ποιος θα καταφέρει να εξορύξει πρώτος ένα μπλοκ ώστε να καρπωθεί την αμοιβή. Για τον σκοπό αυτό αρχικά χρησιμοποιήθηκαν κάρτες γραφικών και στη συνέχεια ειδικό υλικό το οποίο απαιτεί σημαντικές χρηματικές επενδύσεις τόσο για την αγορά όσο και για τη λειτουργία του (ηλ. ενέργεια). Έτσι στις μέρες μας μεγάλες εταιρείες κατασκευής υλικού κυριαρχούν στο Bitcoin, το οποίο από πρωτόκολλο όπου όλοι οι χρήστες ήταν ισότιμοι, κατέληξε σε ιεραρχικό σύστημα με τους απλούς χρήστες να μην μπορούν ουσιαστικά να συμμετέχουν στον πυρήνα του πρωτοκόλλου.

Για την ακρίβεια το 2014 είναι σχεδόν αδύνατο οποιοσδήποτε να καταφέρει να εξορύξει κάποιο μπλοκ χωρίς την χρήση ειδικού υλικού. Πράγματι:

- Το 2009 κάθε miner υπολόγιζε 8MHashes/sec (CPU Mining)
- Το 2010 η δυνατότητα αυξήθηκε 14000 φορές δηλαδή 116 GHashes/sec (GPU Mining)
- Το 2011 9 THashes/sec (FPGA mining)
- Το 2012 23 THashes/sec (ASIC mining)
- Το 2013 10 PHashes/sec
- Το 2014 150 PHashes/sec

Αυτό σημαίνει ότι για να καταφέρει ένας απλός χρήστης να εξορύξει ένα μπλοκ από τον υπολογιστή του χρησιμοποιώντας CPU Mining είναι 2 χιλιετίες περίπου ενώ μέσω κάρτας γραφικών χρειάζεται περίπου 98 χρόνια. Έτσι η εξόρυξη μπλοκ τα τελευταία χρόνια γίνεται μέσω εξειδικευμένου υλικου (Application Specific Integrated Circuits - *ASICs*) οι οποίες γίνονται όλο και πιο πυκνές. Επειδή μάλιστα το κόστος προμήθειας και λειτουργίας τέτοιου υλικού (πχ. λογαριασμοί ηλεκτρικού ρεύματος) γίνεται ασύμφορο ακόμα και για οργανισμούς, υπάρχουν συνεργασίες οι οποίες συνδυάζουν υλικό και μοιράζουν τα κέρδη. Αυτές αναφέρονται ως

mining pools. Πρακτικά ένας απλός χρήστης μπορεί να συμμετάσχει στο πρωτόκολλο μόνο αν συμμετάσχει σε κάποιο mining pool.

Ο συγκεντρωτισμός λοιπόν της δύναμης του πρωτοκόλλου σε μερικούς δυνατούς παίκτες έχει ενοχλήσει αρκετούς στην κοινότητα του Bitcoin οι οποίοι πέρα από ζήτημα αρχών θεωρούν ότι με αυτόν τον τρόπο μειώνεται η ασφάλεια του πρωτοκόλλου, όπως γίνεται φανερό και από την προηγούμενη ενότητα. Έχουν προτείνει λοιπόν προβλήματα πάνω στα οποία θα βασίζεται το proof of work στα οποία η ύπαρξη εξειδικευμένου υλικού δεν θα αποτελεί συντριπτικό πλεονέκτημα. Τέτοια είναι:

- Προβλήματα στα οποία η δυσκολία βασίζεται σε πρόσβαση στην μνήμη της οποίας η σχέση κόστους απόδοσης είναι πιο σταθερή (memory hard puzzles - scrypt).

- Προβλήματα όπως η εύρεση πρώτων αριθμών (primecoin), στα οποία η απαιτούμενη επεξεργαστική ισχύς δεν μπορεί να υλοποιηθεί τόσο εύκολα σε υλικό και να γίνει αντικείμενο παράλληλης επεξεργασίας.

- Εναλλακτικοί τρόποι επίτευξης συναίνεσης όπως για παράδειγμα *proof of stake*. Σε αυτούς τους τρόπους η πιθανότητα δημιουργίας νέου μπλοκ δεν εξαρτάται από την αναλογία επεξεργαστικής ισχύς του miner σε σχέση με αυτή του συνολικού δικτύου, αλλά από την αναλογία πλούτου (σε Bitcoin) του κάθε miner, σε σχέση με το συνολικό δίκτυο. Η ιδέα είναι ότι κάθε miner που είναι αρκετά 'πλούσιος' ώστε να έχει σημαντική πιθανότητα να επηρεάσει το δίκτυο έχει κίνητρο να συμπεριφέρεται 'έντιμα', καθώς θα ζημιωθεί περισσότερο από την κατάρρευση του νομίσματος που θα προκύψει από επιθέσεις και τη μαζική φυγή χρηστών που θα επακολουθήσει.

Ανωνυμία

Η ανωνυμία είναι ένα θέμα το οποίο έχει διχάσει την κοινότητα του Bitcoin. Αν και σε πολλές πηγές το Bitcoin έχει χαρακτηριστεί ως ανώνυμο, στην πραγματικότητα δεν παρέχει μία τέτοια ιδιότητα. Η ψευδαίσθηση αυτή προκύπτει εύλογα, καθώς όπως είδαμε νωρίτερα, οι συναλλαγές στο Bitcoin γίνονται μεταξύ διευθύνσεων, χωρίς να αποκαλύπτονται τα πραγματικά στοιχεία των χρηστών. Οι χρήστες είναι δηλαδή γνωστοί με τα ψευδώνυμα τους. Επιπλέον όπως είδαμε τα περισσότερα προγράμματα επιτρέπουν την πραγματοποίηση συναλλαγών με διαφορετική διεύθυνση - ψευδώνυμο κάθε φορά.

Η ανωνυμία λοιπόν θα ήταν εφικτή αν ήταν όντως αδύνατο να συνδυαστούν μεταξύ τους οι διάφορες διεύθυνσεις καθώς και να συνδεθούν με κάποιο φυσικό

πρόσωπο. Δυστυχώς όμως *η ίδια η φύση+ του Bitcoin αποτρέπει κάτι τέτοιο, καθώς όλες οι συναλλαγές είναι δημόσια διαθέσιμες στο blockchain. Το πρόβλημα είναι ότι πρόσφατα επιτεύγματα της επιστήμης της πληροφορικής δίνουν δυνατότητας απο-ανωνυμοποίησης (deanonymization) τέτοιων συναλλαγών, ειδικά όταν μπορούν να χρησιμοποιηθούν μεγάλα ποσά δεδομένων (large datasets). Το πιο εντυπωσιακό τέτοιο επίτευγμα έγινε σε μια εργασία που δημοσιεύτηκε το 2008 [19], όπου οι συγγραφείς κατάφεραν να συσχετίσουν ανώνυμους χρήστες που είχαν δημοσιεύσει κριτικές ταινιών στον ιστότοπο Netflix με τα προφίλ τους στην διαδικτυακή βάση δεδομένων IMDB.

Άρα η επεξεργασία είναι πολύ εύκολο να γίνει και όντως έχει πραγματοποιηθεί σε εργασίες όπως η [23]. Σε αυτές έγινε προσπάθεια να συσχετιστούν μεταξύ τους διαφορετικές διευθύνσεις μέσα στο πρωτόκολλο, αλλά και εκτός αυτού ώστε να ταυτοποιηθούν χρήστες. Για να επιτευχθεί αυτό έγιναν συναλλαγές σε Bitcoin και προσπάθησαν να ταξινομηθούν οι συναλλασσόμενοι.

- Ένας πολύ απλός τρόπος σύνδεσης είναι αυτό στο οποίο κάποιος χρήστης μπορεί να συνδυάσει UTXO που ανήκουν στον ίδιο ως εισόδους σε μία συναλλαγή ώστε να καλυφθεί ένα ποσό. Αυτό αμέσως προδίδει ότι οι διευθύνσεις αυτές ανήκουν στην ίδια οντότητα, καθώς πρέπει να υπάρχουν τα αντίστοιχα ιδιωτικά κλειδια ώστε να παραχθούν οι υπογραφές.

- Ένας άλλος τρόπος σύνδεσης χρηστών αφορά τις διευθύνσεις που μπορούν να χρησιμοποιηθούν για ρέστα. Αυτές συνήθως δημιουργούνται αυτόματα από τα προγράμματα πελάτες. Αν κάποιος όμως δει περισσότερες συναλλαγές, τότε οι διευθύνσεις που έχουν μόνο μία είσοδο είναι διευθύνσεις για ρέστα και ανήκουν στον ιδιοκτήτη του ιδιωτικού κλειδιού της συναλλαγής.

- Οι διάφορες υπηρεσίες όπως για παράδειγμα τα ανταλλακτήρια νομισμάτων ή υπηρεσίες τζόγου αποτελούν σημεία συγκέντρωσης χρηστών και μπορούν να τους προδώσουν.

Πολλές περισσότερες πληροφορίες μπορούν να προκύψουν αν συνδυαστούν οι συναλλαγές με εξωτερικές του πρωτοκόλλου πληροφορίες.

- Ορισμένες υπηρεσίες συσχετίζουν δημόσια κλειδιά με δικτυακές πληροφορίες όπως για παράδειγμα διευθύνσεις IP.

- Πολλοί χρήστες δημοσιοποιούν δημόσια κλειδιά τους σε διάφορους ιστότοπους, fora, blogs ώστε να λαμβάνουν δωρεές. Είναι πλέον αποδεκτό ότι από την στιγμή που κάποιος χρήστης συνδέει μια διεύθυνση Bitcoin με κάποιο φυσικό χαρακτηριστικό η ανωνυμία του χάνεται.

Κλείνοντας αυτή την ενότητα, θα πρέπει να τονίσουμε ότι δεν είναι ξεκάθαρο αν οποιοδήποτε νόμισμα θα πρέπει να έχει ως ιδιότητα του την ανωνυμία, η οποία παρέχει ελευθερία μεν, αλλά διευκολύνει και αθέμιτες δραστηριότητες. Για παράδειγμα, η ανωνυμία στα μετρητά (φυσικό νόμισμα) χρησιμοποιείται τόσο για προστασία από επεξεργασία των αγορών αλλά και ξέπλυμα χρήματος. Κάτι τέτοιο είναι πιο δύσκολο στις ηλεκτρονικές αγορές στις οποίες εμπλέκονται κεντρικές αρχές όπως με πιστωτικές κάρτες, αλλά εκεί είναι πολύ πιο εύκολο να φτιάξει κάποιος το προφίλ ενός καταναλωτή.

11.5 Συσκότιση Κώδικα

Η συσκότιση κώδικα είναι ένα πρόβλημα που έχει προέλευση από την τεχνολογία λογισμικού. Με απλά λόγια, στοχεύει στην διανομή προγραμμάτων με τρόπο ώστε να μπορούν να εκτελεστούν από τους χρήστες, χωρίς όμως να μπορούν να καταλάβουν (από τον πηγαίο κώδικα ή από reverse engineering του εκτελέσιμου κώδικα) πώς αυτά λειτουργούν. Μία τέτοια δυνατότητα έχει πολλές πρακτικές εφαρμογές:

- Προστασία αλγορίθμων που αποτελούν (εμπορικά) μυστικά.
- Προστασία κλειδιών τα οποία χρησιμοποιούνται σε κάποιο σημείο ενός προγράμματος. Ένα τέτοιο παράδειγμα αποτελούν τα κρυπτογραφικά κλειδιά που υπάρχουν σε λογισμικό ή υλικό προστασίας πνευματικών δικαιωμάτων (DRM) και χρησιμοποιούνται για την αποκρυπτογράφηση και αναπαραγωγή του. Γενικότερα μπορεί να χρησιμοποιηθεί για υδατογράφηση λογισμικού, δηλαδή ενσωμάτωση δεδομένων σε κάποιο πρόγραμμα τα οποία διαφέρουν ανά κάτοχο.
- Προστασία προγραμμάτων τα οποία διανέμονται σε μορφές bytecode (Java, .Net) οι οποίες είναι πολύ εύκολο να αναλυθούν.
- Γενικότερη προστασία λογισμικού από την ανάλυση του από μη εξουσιοδοτημένους χρήστες. Η δυνατότητα αυτή μπορεί να χρησιμοποιηθεί και από συγγραφείς κακόβουλου λογισμικού ώστε να αποτρέψουν τη δημιουργία αντιμέτρων.

11.5.1 Συσκότιση Μαύρου Κουτιού

Οι τεχνικές συσκότισης κώδικα έχουν παρουσιάσει εξέλιξη ανάλογη με αυτή της κρυπτογραφίας, με την έννοια ότι ξεκίνησαν από ευρετικές μεθόδους οι οποίες αν

και έχουν αρκετά σημαντικά αποτελέσματα δεν μπορούν να *αποδειχθούν* άτρωτες. Έτσι και σε αυτό το πεδίο είναι απαραίτητη η έννοια της αποδείξιμης ασφάλειας (συσκότισης), η οποία ξεκίνησε στο [2] και ορίζεται με την έννοια της προσομοίωσης.

Ο πρώτος ορισμός της συσκότισης αφορά προσομοίωση με χρήση μαύρου κουτιού:

Ορισμός 11.2. Ένας συσκοτιστής κώδικα $\mathcal{O}$ είναι μία πιθανοτική μηχανή Turing η οποία έχει τις παρακάτω τρεις ιδιότητες:

1. **Διατήρηση Λειτουργικότητα** $\forall M, x\, \mathcal{O}(M)(x) = M(x)$. Για κάθε μηχανή Turing M, η $\mathcal{O}(M)$ υπολογίζει την ίδια συνάρτηση με την M.

2. **Πολυωνυμική Επιβάρυνση** $\forall M, p\, |\mathcal{O}(M)| \leq p(|M|)$ και $time(\mathcal{O}(M)(x)) \leq p(time(M(x)))$ Η περιγραφή της $\mathcal{O}(M)$ καθώς και ο χρόνος εκτέλεσης της είναι το πολύ πολυωνυμικά μεγαλύτερος από αυτό της M.

3. **Προσομοίωση Με Μαύρο Κουτί** $\forall PPT\, A,\ \exists PPT\, S$:

$$Pr[A(\mathcal{O}(M)) = 1] - Pr[A(S^M(1^{|M|}) = 1] \leq negl(|M|)$$

   Οτιδήποτε μπορεί να υπολογιστεί αποδοτικά από το $\mathcal{O}(\mathcal{M})$ μπορεί να υπολογιστεί χρησιμοποιώντας το M ως μαντείο.

Ο παραπάνω ορισμός σημαίνει με απλά λόγια ότι ένα συσκοτισμένο πρόγραμμα $\mathcal{O}(P)$ δεν δίνει καμία επιπλέον πληροφορία στο χρήστη του, από την πληροφορία που θα έδινε η αλληλεπίδραση με το P σε ένα μαύρο κουτί.

Το βασικό αποτέλεσμα των [2] είναι ότι με τον παραπάνω ορισμό:

Θεώρημα 11.3. *Υπάρχει μια οικογένεια συναρτήσεων οι οποίες έχουν ιδιότητες με τιμή 0,1 για τις οποίες τα προγράμματα που τις υπολογίζουν δεν μπορούν να συσκοτιστούν κατά το μοντέλο μαύρου κουτιού.*

Απόδειξη. Θα δείξουμε ότι ένα πρόγραμμα που έχει συσκοτιστεί δίνει μη αμελητέο πλεονέκτημα στον υπολογισμό της ιδιότητας σε σχέση με την πρόσβαση σε μαύρο κουτί.

Έστω ένας συσκοτιστής $\mathcal{O}$ και η παρακάτω μηχανή Turing:

$$C_{a,b}(x) = \begin{cases} b, & x = a \\ 0^k, & x \neq= a \end{cases}$$

Ορίζουμε τώρα το κατηγόρημα:

$$D_{a,b}(C) = \begin{cases} 1b, & C(a) = b \\ 0^k, & C(a) \neq b \end{cases}$$

Επειδή η D είναι μη υπολογίσιμη στον παραπάνω ορισμό, αν η D δεν έχει τερματίσει μετά από πολυωνυμικό αριθμό βημάτων τότε επιστρέφει 0.

Έστω τώρα ο αντίπαλος $\mathcal{A}$ ο οποίος με είσοδο 2 μηχανές Turing τρέχει τη δεύτερη με είσοδο την πρώτη. Δηλαδή: $\mathcal{A}(C, D) = D(C)$.

Επειδή η συσκότιση *διατηρεί τη λειτουργικότητα* έχουμε :

$$\mathcal{A}(\mathcal{O}(C), \mathcal{O}(D)) = \mathcal{O}(D)(\mathcal{O}(C)) = D(C) \text{ και κατά συνέπεια:}$$
$$Pr[\mathcal{A}(\mathcal{O}(C_{a,b}), \mathcal{O}(D_{a,b})) = 1] = 1$$

Από την άλλη, για έναν PPT αλγόριθμο $\mathcal{S}$ ο οποίος έχει μόνο πρόσβαση μαντείου στις $C_{a,b}, D_{a,b}$ ισχύει:
$Pr[\mathcal{S}^{C_{a,b},D_{a,b}}(1^k) = 1] - Pr[\mathcal{S}^{Z_k,D_{a,b}}(1^k) = 1] \leq neql(k)$ όπου $Z_k(x) = 0^k$.
Δηλαδή με πρόσβαση μαντείου μόνο, ο $\mathcal{S}$ δεν θα πάρει πρακτικά ποτέ την έξοδο 1 από την D γιατί πρακτικά ποτέ δεν θα τύχει η είσοδος της C να είναι a.

Επειδή η συσκότιση *διατηρεί τη λειτουργικότητα* έχουμε πάλι:

$$Pr[\mathcal{A}(\mathcal{O}(Z_k), \mathcal{O}(D_{a,b})) = 1] = 0 \text{ και κατά συνέπεια:}$$
$$Pr[\mathcal{A}(\mathcal{O}(C_{a,b}), \mathcal{O}(D_{a,b})) = 1] - Pr[\mathcal{A}(\mathcal{O}(Z_k), \mathcal{O}(D_{a,b})) = 1] = 1$$

Άρα σύμφωνα με τον ορισμό ο $\mathcal{O}$ δεν είναι συσκοτιστής. □

Ο προσεκτικός αναγνώστης θα παρατηρήσει ότι ο αντίπαλος της παραπάνω απόδειξης δέχεται δύο συσκοτισμένες μηχανές. Για να ικανοποιηθεί ακριβώς ο ορισμός πρέπει ο αντίπαλος να δέχεται μόνο μία. Κάτι τέτοιο είναι εύκολο να γίνει καθώς οι περιγραφές τους μπορούν να γραφτούν στην ταινία μιας μηχανής Turing χωρισμένες από ένα ειδικό σύμβολο.

Μετά το συγκεκριμένο αρνητικό αποτέλεσμα, οι ερευνητικές κατευθύνσεις στο συγκεκριμένο θέμα κινήθηκαν προς δύο κατευθύνσεις. Η πρώτη διατήρησε το συγκεκριμένο μοντέλο ορισμού της συσκότισης συμπληρωμένο με κάποιες υποθέσεις, ενώ η δεύτερη προσπάθησε να ελέγξει αν η συσκότιση είναι δυνατή σε ένα διαφορετικό μοντέλο.

11.5.2 Συσκότιση Με Υποθέσεις

Ένα θετικό αποτέλεσμα σε αυτή την κατεύθυνση παρουσιάστηκε στο [14] στο οποίο αποδείχθηκε η δυνατότητα συσκότισης συναρτήσεων *σημείου*, δηλ. λογικών συναρτήσεων οι οποίες έχουν την τιμή 1 σε ένα μόνο σημείο του πεδίου ορισμού τους, ενώ στα υπόλοιπα έχουν την τιμή 0, *υπό την προϋπόθεσή* ύπαρξης τυχαίων

μαντείων ή μη αντιστροφής μιας κατηγορίας μεταθέσεων. Η συγκεκριμένη κατηγορία συναρτήσεων έχουν και πρακτική χρησιμότητα καθώς μοντελοποιούν τον έλεγχο συνθηματικών, όπου για κάθε χρήστη λαμβάνουν την τιμή αληθής μόνο για το έγκυρο συνθηματικό.

Η βασική ιδέα της απόδειξης αυτής είναι η εξής: Σε περίπτωση που το συνθηματικό είναι αρκετά τυχαίο και ο αντίπαλος δεν το γνωρίζει, τότε πρακτικά η πιθανότητα να το μαντέψει και κατά συνέπεια η συνάρτηση να είναι αληθής είναι το ίδιο αμελητέα είτε αλληλεπιδρά με τον συσκοτισμένο κώδικα είτε με το τυχαίο μαντείο.

Πιο συγκεκριμένα, μία συνάρτηση σημείου ορίζεται ως εξής:

$$P_a(x) = \begin{cases} 1, & x = a \\ 0, & x \neq a \end{cases}$$

Για την συσκότιση με την υπόθεση τυχαίων μαντείων πρέπει να τροποποιήσουμε λίγο τον ορισμό, αντικαθιστώντας την ακριβή υλοποίηση της λειτουργίας με μία προσεγγιστική:

$$Pr[\exists x \in \{0,1\}^* : \mathcal{O}^R(M)(x) = M(x)] \leq negl(k)$$

Τα υπόλοιπα χαρακτηριστικά του ορισμού παραμένουν αμετάβλητα. Έτσι προκύπτει το :

Θεώρημα 11.4. *Για τυχαίο μαντείο* $R : \{0,1\}^k \to \{0,1\}^{2k}$*, αν* $\mathcal{O}(P_a)$ *είναι ένα πρόγραμμα που περιέχει την τιμή* $r = R(a)$ *και λειτουργεί ως:* $\mathcal{O}(P_a)(x) = \begin{cases} 1, & R(x) = r \\ 0^k, & R(x) \neq r \end{cases}$ *Τότε ο* $\mathcal{O}(P_a)$ *είναι συσκοτιστής για το* P*.*

Απόδειξη. Σχετικά με την χρονική πολυπλοκότητα είναι πολύ εύκολο να δει κανείς ότι ισχύει η δεύτερη ιδιότητα καθώς το τυχαίο μαντείο απαντάει σε ένα βήμα.

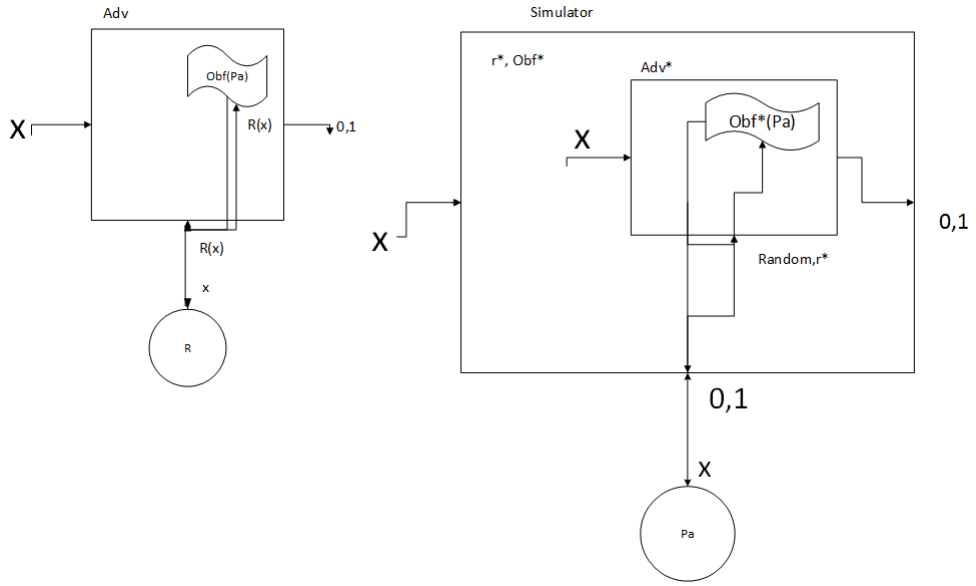
Για την προσεγγιστική λειτουργικότητα έχουμε:

$Pr[\exists x \neq a : R(x) = R(a)] \leq \sum_x [R(x) = R(a)] = \frac{2^k - 1}{2^{2k}} = negl(k)$

Τέλος για την ιδιότητα του μαύρου κουτιού, διεξάγουμε την εξής προσομοίωση:

Από την μία ο $\mathcal{A}$

- λαμβάνει μία τυχαία είσοδο x
- την οποία προωθεί στην $\mathcal{O}(P_a)$
- η οποία ρωτάει το R.



Σχήμα 11.7: Προσομοίωση Συσκότισης σημείου

- Η απάντηση είναι $R(x)$
- Η $\mathcal{O}(P_a)$ ελέγχει αν $R(x) = r$
- και απαντάει στην έξοδο 0 ή 1 αντίστοιχα
- Αυτή θα είναι και η απάντηση του $\mathcal{A}$

Κατά συνέπεια $Pr[A^R(\mathcal{O}^R)(M)) = 1] = Pr[x = a] = negl(k)$.
Από την άλλη ο $\mathcal{S}$ δεν γνωρίζει το a. Κατά συνέπεια δεν γνωρίζει ούτε το r ούτε την $\mathcal{O}(P_a)$ αφού έχει πρόσβαση στην P_a μόνο ως μαντείο.

- Διαλέγει ένα τυχαίο $r^* \in \{0,1\}^{2k}$
- Προετοιμάζει μία συσκότιση του $\mathcal{O}^*$ όπου το r έχει αντικατασταθεί με το r^*
- Χρησιμοποιεί εσωτερικά τον $\mathcal{A}$ ως εξής:
  - Όταν ο $\mathcal{A}$ ρωτήσει το R τότε ο $\mathcal{S}$ προωθεί την ερώτηση στο P_a. Αν το τελευταίο απαντήσει 1, τότε ο $\mathcal{S}$ επιστρέφει το r^*, αλλιώς μια τυχαία συμβολοσειρά. Ούτε ο $\mathcal{A}$ δεν ξέρει το a οπότε δεν μπορεί να καταλάβει τη διαφορά και εκτελεί κανονικά την $\mathcal{O}^*$ με το r.
  - Η απάντηση προωθείται στον $\mathcal{S}$ και μετά στην έξοδο.

Κατά συνέπεια έχουμε: $Pr[(S)^{P_a}(1^{|M|}) = 1] = Pr[x = a] = negl(k)$. Δηλαδή οι προσομοιώσεις είναι ταυτόσημες.

□

Είναι πολύ ενδιαφέρον να παρατηρήσει κανείς ότι η κατηγορία αυτή των συναρτήσεων σημείου αντιστοιχεί στην καλή πρακτική ελέγχου συνθηματικών όπου ο έλεγχος γίνεται μέσω μιας συνάρτησης σύνοψης η οποία δέχεται ως είσοδο το συνθηματικό και μια τυχαία συμβολοσειρά (salt). Στην απόδειξη η συνάρτηση σύνοψης μοντελοποιείται ως τυχαίο μαντείο.

11.5.3 Συσκότιση Μη Διακρισιμότητας

Η συσκότιση μη διακρισιμότητας (indistinguishability obfuscation) προτάθηκε στο [2] ως ένας εναλλακτικός τρόπος ορισμού της συσκότισης, αντί για τη χρήση μαύρου κουτιού.

Ορισμός 11.5. Ένας συσκοτιστής μη διακρισιμότητας $i\mathcal{O}$ για μια οικογένεια κυκλωμάτων C εγγυάται ότι για δύο ισοδύναμα κυκλώματα $C_1, C_2 \in C$ οι κατανομές των συσκοτίσεων $i\mathcal{O}(C_1), i\mathcal{O}(C_2)$ θα είναι υπολογιστικά αδιαχώριστες.

Η πολύ σημαντική ιδιότητα των συγκεκριμένων συσκοτιστών είναι ότι ένας αποδοτικός τέτοιο συσκοτιστής κρύβει την μέγιστη πληροφορία για την είσοδο του από ότι οποιοσδήποτε συσκοτιστής με συγκεκριμένο μέγεθος. Η κατασκευή των συσκοτιστών προτάθηκε το 2013 στο [12], αλλά η περιγραφή της είναι εκτός των σκοπών του παρόντος έργου.

11.5.4 Εφαρμογές στην Κρυπτογραφία

Κλείνοντας την συγκεκριμένη ενότητα, αξίζει να αναφερθούμε στην επίδραση της ύπαρξης 'ασφαλούς' συσκότισης στην ίδια την κρυπτογραφία, πέρα δηλαδή από το λογισμικό που αναφέραμε νωρίτερα.

Μία τέτοια εφαρμογή είναι η δημιουργία αλγορίθμων κρυπτογράφησης δημοσίου κλειδιού από αλγόριθμους κρυπτογράφησης ιδιωτικού κλειδιού. Αν Enc_k είναι μια συμμετρική συνάρτηση κρυπτογράφησης με κλειδί k, τότε η $\mathcal{O}(Enc_k)$ μπορεί να δίνεται σε όποιον χρήστη θέλει να κρυπτογραφήσει ένα μήνυμα, χωρίς να υπάρχει κίνδυνος διαρροής του κλειδιού. Φυσικά η συνάρτηση αποκρυπτογράφησης θα είναι διαθέσιμη μόνο στον χρήστη που διαθέτει το k.

Επιπλέον η ασφαλής συσκότιση δίνει έναν τρόπο κατασκευής πλήρως ομομορφικών κρυπτοσυστημάτων. Για να υπολογίσουμε οποιαδήποτε συνάρτηση μεταξύ δύο κρυπτοκειμένων τα οποία έχουν κρυπτογραφηθεί με το ίδιο δημόσιο κλειδί

εφαρμόζουμε τα εξής βήματα: Αρχικά συσκοτίζουμε την συνάρτηση αποκρυπτογράφησης (με ενσωματωμένο το ιδιωτικό κλειδί) και την εφαρμόζουμε στα κρυπτοκείμενα. Στη συνέχεια υπολογίζουμε τη συνάρτηση και κρυπτογραφούμε το αποτέλεσμα. Τα παραπάνω βήματα συσκοτίζονται και έτσι μπορούν να διανεμηθούν με ασφάλεια.

11.6 Ηλεκτρονικό Υλικό

- Διαδραστικές Παρουσιάσεις - Video
  - Verifying elections with cryptography: Ομιλία Ben Adida, δημιουργού του Helios
- Υλοποιήσεις Συστημάτων Ηλεκτρονικών Ψηφοφοριών
  - Ηλ. Ψηφοφορίες Ζευς
  - Ηλ. Ψηφοφορίες Helios
- Dissent: accountable anonymous group communication
- Εναλλακτικά κρυπτονομίσματα βασισμένα στο bitcoin
  - Namecoin
  - Litecoin
  - Ethereum
- Δοκιμαστικός κώδικας για ηλεκτρονικές ψηφοφορίες
  - Υλοποίηση Ομομορφικού Συστήματος Ηλ. Ψηφοφορίας
  - Υλοποίηση Συστήματος Ηλ. Ψηφοφορίας Με Δίκτυα Μίξης

Βιβλιογραφία

[1] Ben Adida. Helios: web-based open-audit voting. In *Proceedings of the 17th conference on Security symposium*, SS'08, pages 335–348, Berkeley, CA, USA, 2008. USENIX Association.

[2] Boaz Barak, Oded Goldreich, Rusell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *Advances in Cryptology — CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 1–18. Springer Berlin Heidelberg, 2001.

[3] Olivier Baudron, Pierre-Alain Fouque, David Pointcheval, Jacques Stern, and Guillaume Poupard. Practical multi-candidate election system. In *Proceedings of the twentieth annual ACM symposium on Principles of distributed computing*, PODC '01, pages 274–283, New York, NY, USA, 2001. ACM.

[4] David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–88, 1981.

[5] David Chaum. Blind signatures for untraceable payments. In D. Chaum, R.L. Rivest, and A.T. Sherman, editors, *Advances in Cryptology Proceedings of Crypto 82*, pages 199–203, 1983.

[6] David Chaum. Security without identification: transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10):1030–1044, October 1985.

[7] David Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of cryptology*, 1(1):65–75, 1988.

[8] Josh D. Cohen and Michael J. Fischer. A robust and verifiable cryptographically secure election scheme (extended abstract). In *FOCS*, pages 372–382, 1985.

[9] Ronald Cramer, Rosario Gennaro, and Berry Schoenmakers. European transactions on telecommunications. In *Proceedings of the 17th conference on Security symposium*, pages 103–118. Springer-Verlag, 1997.

[10] Joan Feigenbaum and Bryan Ford. Seeking anonymity in an internet panopticon. *arXiv preprint arXiv:1312.5307*, 2013.

[11] Atsushi Fujioka, Tatsuaki Okamoto, and Kazuo Ohta. A practical secret voting scheme for large scale elections. In *Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques: Advances in Cryptology*, ASIACRYPT '92, pages 244–251, London, UK, UK, 1993. Springer-Verlag.

[12] Sanjam Garg, Mariana Raykova, Craig Gentry, Amit Sahai, Shai Halevi, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *In FOCS*, 2013.

[13] Joe Kilian and Kazue Sako. Receipt-free MIX-type voting scheme - a practical solution to the implementation of a voting booth. In *Proceedings of EUROCRYPT 1995*. Springer-Verlag, 1995.

[14] Benjamin Lynn, Manoj Prabhakaran, and Amit Sahai. Positive results and techniques for obfuscation. In *In EUROCRYPT '04*, 2004.

[15] Jakobsson Markus and Juels Ari. Millimix: Mixing in small batches. Technical report, Center for Discrete Mathematics & Theoretical Computer Science, 1999.

[16] ABE Masayuki. Mix-networks on permutation networks. In *ASIACRYPT'99*, page 258. Springer, 1999.

[17] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System.

[18] Arvind Narayanan, Andrew Miller, Ed Felten, Steven Goldfeder, Shivam Agarwal, and Joseph Bonneau. Princeton online bitcoin course. Διαθέσιμο στο https://piazza.com/princeton/spring2015/btctech/home.

[19] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, pages 111–125. IEEE, 2008.

[20] Michael Nielsen. How the Bitcoin protocol actually works | DDI.

[21] Choonsik Park, Kazutomo Itoh, and Kaoru Kurosawa. Efficient anonymous channel and all/nothing election scheme. In *EUROCRYPT*, pages 248–259, 1993.

[22] B. Pfitzmann. Breaking an efficient anonymous channel. In *Advances in Cryptology—EUROCRYPT'94*, pages 332–340. Springer, 1995.

[23] Fergal Reid and Martin Harrigan. An Analysis of Anonymity in the Bitcoin System. In Yaniv Altshuler, Yuval Elovici, Armin B. Cremers, Nadav Aharony, and Alex Pentland, editors, *Security and Privacy in Social Networks*, pages 197–223. Springer New York, 2013.

Κεφάλαιο 12

Προηγμένα Θέματα

Στην ενότητα αυτή θα αναφερθούμε σε σχήματα και πρωτόκολλα τα οποία είτε έχουν πολύ μεγάλη σημασία στις σύγχρονες κρυπτογραφικές εφαρμογές, είτε αναμένεται να διαδραματίσουν σημαντικό ρόλο στο μέλλον. Η αναλυτική περιγραφή κάθε ενός από αυτά θα μπορούσε άνετα να καταλάβει τόμους. Οπότε σε αυτό το κεφάλαιο δεν σκοπεύουμε να κάνουμε αναλυτική περιγραφή των προχωρημένων αυτών εννοιών. Στόχος μας είναι να δώσουμε ένα σημείο εκκίνησης ώστε ο ενδιαφερόμενος αναγνώστης να προχωρήσει από εκεί.

12.1 Κβαντική Κρυπτογραφία

12.1.1 Κβαντικοί Υπολογισμοί

Κατά τη δεκαετία του 1930 μεγάλες μορφές της επιστημονικής κοινότητας, όπως ο Alan Turing, έθεσαν τις βάσεις της θεωρητικής πληροφορικής. Αυτές οι θεωρίες θέτουν κάποια όρια για τους αλγορίθμους που εκτελούνται σε πρότυπες υπολογιστικές μηχανές. Το συγκλονιστικό είναι ότι στις θεωρίες αυτές βασίστηκε η κατασκευή μιας "σύγχρονης" υπολογιστικής μηχανής (παρόμοια στη λειτουργία με τους σημερινούς υπολογιστές) που πρωτοεμφανίστηκε τη δεκαετία του '50. Από τότε οι εξελίξεις στον τομέα της κατασκευής υπολογιστών υπήρξαν ραγδαίες. Έτσι περνώντας από τεχνολογία λυχνιών και DLSI κυκλώματα έχουμε σήμερα φτάσει σε ένα σημείο που τα δομικά στοιχεία των υπολογιστών είναι τόσο μικρά ώστε να επηρεάζονται ήδη από τους νόμους της κβαντομηχανικής. Η εξέλιξη αυτή γέννησε μία νέα γενιά επιστημόνων που οραματίζονταν ότι ίσως αυτές οι επιδράσεις θα μπορούσαν να χρησιμοποιηθούν για να επιταχύνουν τους υπολογισμούς.

Ο Richard Feynman ήταν εκείνος που πρώτος παρουσίασε μία ιδέα για το πώς ένα κβαντικό σύστημα θα μπορούσε θεωρητικά να χρησιμοποιηθεί για την πραγματο-

ποίηση υπολογισμών. Στη συνέχεια ο David Deutsch το 1985 έκανε μία ριζοσπαστική δημοσίευση, όπου περιέγραφε το πώς κάθε φυσική διαδικασία θα μπορούσε να μοντελοποιηθεί θεωρητικά με τέλειο τρόπο με χρήση ενός *κβαντικού υπολογιστικού συστήματος*. Ένα τέτοιο *κβαντικό υπολογιστικό σύστημα* μπορεί, όπως αναφέρει, να πραγματοποιήσει διαδικασίες αδύνατες για έναν "κλασσικό" υπολογιστή, π.χ. παραγωγή πραγματικά τυχαίων ακεραίων. Η βασικότερη ιδιότητά του είναι η ικανότητα να χρησιμοποιεί το φαινόμενο του *κβαντικού παραλληλισμού*, για να πραγματοποιεί κάποιους υπολογισμούς σε χρόνο πολύ μικρότερο από τον "κλασσικό" υπολογιστή. Ας σταθούμε όμως λίγο στις βασικές αρχές αυτού του *κβαντικού υπολογιστή*.

Στο «κλασσικό» μοντέλο υπολογιστή το βασικό δομικό στοιχείο, το bit, μπορεί να βρίσκεται σε μία από τις καταστάσεις 0 και 1. Από την άλλη μεριά ένα *κβαντικό bit (qubit)* μπορεί να βρίσκεται όχι μόνο σε μία από αυτές τις δύο καταστάσεις, αλλά και σε μία *υπέρθεσή* τους! Σ' αυτή τη *σύμφωνη κατάσταση*, το qubit, υπάρχει σαν 0 και 1 ταυτόχρονα! Ας θεωρήσουμε για παράδειγμα έναν καταχωρητή από 3 bit. Αυτός μπορεί να χρησιμοποιηθεί για αναπαράσταση ενός εκ των αριθμών από το 0 ως το 7 σε κάθε χρονική στιγμή. Αν τώρα θεωρήσουμε έναν καταχωρητή από τρία qubit, παρατηρούμε ότι αν κάθε qubit είναι σε υπέρθεση, ο καταχωρητής μπορεί μπορεί να αναπαριστά όλους τους αριθμούς από το 0 μέχρι το 7 ταυτόχρονα. Γενικότερα είναι θεωρητικά δυνατό ένα κβαντικός υπολογιστής n qubit, να βρίσκεται ταυτόχρονα σε 2^n καταστάσεις. Αυτό θα σήμαινε για παράδειγμα ότι θα μπορούσαν να αναπαραστήσει ταυτόχρονα όλα τα κλειδιά ενός κρυπτοσυστήματος.

Δεν πρέπει εντούτοις να νομίσουμε ότι ο *κβαντικός υπολογιστής* θα "εκτελεί" - δωρεάν - τους σημερινούς αλγορίθμους σε λιγότερο χρόνο. Δηλαδή ακόμα και αν καταφέρουμε και κατασκευάσουμε την υπέρθεσή 2^n καταστάσεων - από το οποίο βρισκόμαστε αρκετά μακριά, πρέπει να βρεθεί ένας τρόπος να διεξαχθεί ένας υπολογισμός με αυτές. Πιστεύεται ότι με τον υπολογισμό αυτό, δεν θα μπορούν να λυθούν NP πλήρη προβλήματα.

12.1.2 Ο αλγόριθμος του Shor

Από την άλλη, μπορεί να δημιουργηθούν νέοι αλγόριθμοι για την επίλυση προβλημάτων που να εκμεταλλεύονται τις νέες ιδιότητες μιας τέτοιας *κβαντικής υπολογιστικής μηχανής* και να λύνουν προβλήματα με καινούριους - ίσως θεαματικούς τρόπους. Ένα παράδειγμα τέτοιου αλγορίθμου δημιουργήθηκε στα εργαστήρια της AT&T Bell από τον Peter Shor μόλις το 1994 [1]. Ο αλγόριθμος του Shor (σχήμα 12.1) επιλύει το πρόβλημα της παραγοντοποίησης (FACTORING) σε *πολυωνυμικό χρόνο* και παρουσιάζεται παρακάτω:

Ο παραπάνω αλγόριθμος με κάποιες τροποποιήσεις μπορεί να επιλύσει με εκθε-

Υποθέτουμε ότι θέλουμε παραγοντοποιήσουμε τον ακέραιο N. Ο αλγόριθμος ακολουθεί τα παρακάτω βήματα:

1. Επίλεξε έναν ψευδοτυχαίο αριθμό $a < N$
2. Υπολόγισε το $gcd(a, N)$ (π.χ. χρησιμοποιώντας τον ευκλείδειο αλγόριθμο)
3. Αn ο $gcd(a, N) \neq 1$, τότε αυτός είναι ένας μη τετριμένος διαιρέτης του , αλλιώς πήγαινε στο βήμα 4.
4. Υπολόγισε την περίοδο r της συνάρτησης:

$$f(x) = a^x \bmod N$$

(Εδώ είναι η καινοτομία του Shor αφού για τον υπολογισμό του r χρησιμοποιεί ένα *κβαντικό υπολογιστικό μοντέλο* που τον πραγματοποιεί σε πολυωνυμικό χρόνο. Η περιγραφή εντούτοις του μοντέλου αυτού ξεφεύγει από του στόχους αυτού του βιβλίου.)

5. Αν ο r είναι περιττός, πήγαινε στο βήμα 1
6. Αν $a^{r/2} \equiv -1 \pmod{N}$, πήγαινε στο βήμα 1
7. Οι παράγοντες του N είναι $gcd(a^{r/2} \pm 1, N)$

Σχήμα 12.1: Ο αλγόριθμος του Shor για το FACTORING

τική βελτίωση το πρόβλημα Διακριτού Λογαρίθμου (DLOG) σε ομάδες πρώτων και σε ελλειπτικές καμπύλες. Στην συμμετρική κρυπτογραφία υπάρχει ο αλγόριθμος του Grover ([2]) ο οποίος όμως επιφέρει πολυωνυμική βελτίωση στο AES.

Από τα παραπάνω γίνεται φανερό ότι η πιθανή κατασκευή κβαντικών υπολογιστών θα έχει σημαντικές συνέπειες στους κρυπτογραφικούς αλγόριθμους που χρησιμοποιούνται σήμερα. Για να αντιμετωπιστεί αυτή η απειλή έχουν προταθεί δύο κατευθύνσεις:

- Η *Κβαντική Κρυπτογραφία (Quantum Cryptography)* η οποία προσπαθεί χρησιμοποιώντας τις κρυπτογραφικές δυνατότητες των κβαντικών υπολογιστών να αντιμετωπίσει τις κρυπταναλυτικές δυνατότητες τους.

- Η *Μέτα-Κβαντική Κρυπτογραφία (Post-Quantum Cryptography)* η οποία προσπαθεί να σχεδιάσει κρυπτοσυστήματα με βάση τεχνικές οι οποίες δεν επηρεάζονται από τους κβαντικούς υπολογιστές. Μία τέτοια προσέγγιση είναι η κρυπτογραφία που βασίζεται στα δικτυωτά (lattices) με την οποία θα ασχοληθούμε στην ενότητα 12.4.

12.2 Ελλειπτικές Καμπύλες

Στην ενότητα αυτή θα ασχοληθούμε με τις *ελλειπτικές καμπύλες (Elliptic Curves - EC)* και τις κρυπτογραφικές τους εφαρμογές. Οι ελλειπτικές καμπύλες, ως αντικείμενο της θεωρίας αριθμών και της αλγεβρικής γεωμετρίας, έχουν μελετηθεί για περισσότερο από δύο αιώνες και η θεωρία που έχει αναπτυχθεί γύρω τους είναι ιδιαίτερα πλούσια σε αποτελέσματα. Ωστόσο την τελευταία εικοσαετία το ενδιαφέρον της ακαδημαϊκής κοινότητας για τις ελλειπτικές καμπύλες έχει αυξηθεί σημαντικά, τόσο λόγω των εφαρμογών τους στην κρυπτογραφία, όσο και της άμεσης σχέσης της θεωρίας ελλειπτικών καμπυλών με την απόδειξη του τελευταίου θεωρήματος του Fermat.

Η σημασία των ελλειπτικών καμπυλών στην κρυπτογραφία συνοψίζεται στο γεγονός ότι τα σημεία τους μπορούν να αντικαταστήσουν τα σημεία του $\mathbb{Z}_p^*$ με πολύ μεγάλη επιτυχία καθώς:

- Το πρόβλημα του διακριτού λογάριθμου στην ομάδα που σχηματίζουν αυτά δεν επιδέχεται υποεκθετικούς αλγορίθμους όπως οι αλγόριθμοι index calculus της ενότητας 4.8

- Κατά συνέπεια μπορούμε να πετύχουμε ίδια και καλύτερα επίπεδα ασφάλειας χρησιμοποιώντας μικρότερες παραμέτρους ασφαλείας, δηλαδή λιγότερα bits κλειδιών και κατά συνέπεια πιο αποδοτικές λειτουργίες.

Τα μαθηματικά που απαιτούνται για τις ελλειπτικές καμπύλες είναι αρκετά πολύπλοκα και έχουν αναπτυχθεί σε βάθος αιώνων. Για τον λόγο αυτό η παρουσίαση τους ξεφεύγει από τους σκοπούς ενός εγχειριδίου κρυπτογραφίας. Επιπλέον όπως αναφέρεται και στο [3] η περιγραφή των εφαρμογών τους μπορεί να γίνει αφαιρώντας τις 'εξειδικευμένες λεπτομέρειες' και απλά υποθέτοντας μια κυκλική ομάδα όπου κάποιο πρόβλημα είναι δύσκολο.

Για τον σκοπό αυτό η μαθηματική περιγραφή θα είναι περιορισμένη. Για περισσότερες λεπτομέρειες ο αναγνώστης μπορεί να διατρέξει σε εγχειρίδια όπως το [4] και το [5].

12.2.1 Ελλειπτικές καμπύλες πάνω από σώματα

Έστω $\mathbb{F}$ ένα σώμα. Για αυτή την ενότητα θα θεωρούμε ότι το $\mathbb{F}$ είναι είτε το $\mathbb{R}$, είτε το πεπερασμένο σώμα q στοιχείων $GF(q)$, όπου $q = p^r$ και p πρώτος.

Ορισμός 12.1. Έστω το σώμα $\mathbb{F}$ με χαρακτηριστική διάφορη του 2 και του 3, και έστω x^3+ax+b, με $a, b \in \mathbb{F}$, ένα πολυώνυμο 3ου βαθμού χωρίς πολλαπλές ρίζες[1]. Τότε μια ελλειπτική καμπύλη πάνω από το $\mathbb{F}$ είναι το σύνολο των σημείων (x, y) με $x, y \in \mathbb{F}$, τα οποία ικανοποιούν την εξίσωση

$$, y^2 = x^3 + ax + b \tag{12.1}$$

μαζί με ένα στοιχείο $\mathcal{O}$, το οποίο θα ονομάζουμε «σημείο στο άπειρο».

Αν το $\mathbb{F}$ είναι σώμα χαρακτηριστικής 2, τότε μια ελλειπτική καμπύλη πάνω από το $\mathbb{F}$ είναι το σύνολο των σημείων, τα οποία ικανοποιούν είτε μια εξίσωση της μορφής

$$y^2 + cy = x^3 + ax + b\ ,\quad a, b, c \in \mathbb{F}\ ,$$

είτε μια εξίσωση της μορφής

$$y^2 + xy = x^3 + ax + b\ ,\quad a, b \in \mathbb{F}\ ,$$

(εδώ δε μας ενδιαφέρει αν το πολυώνυμο 3ου βαθμού στο δεξί μέλος έχει πολλαπλές ρίζες) μαζί με ένα «σημείο στο άπειρο» $\mathcal{O}$.

Αν το $\mathbb{F}$ είναι σώμα χαρακτηριστικής 3, τότε μια ελλειπτική καμπύλη πάνω από το $\mathbb{F}$ είναι το σύνολο των σημείων, τα οποία ικανοποιούν μια εξίσωση της μορφής ,

$$y^2 = x^3 + ax^2 + bx + c\ ,\quad a, b, c \in \mathbb{F} \tag{12.2}$$

(εδώ το πολυώνυμο 3ου βαθμού στο δεξί μέλος *δεν* έχει πολλαπλές ρίζες) μαζί με ένα "σημείο στο άπειρο" $\mathcal{O}$.

Παρατήρηση 21. Υπάρχει μια γενική εξίσωση μέσω της οποίας αναπαριστούμε τις ελλειπτικές καμπύλες πάνω από οποιοδήποτε σώμα $\mathbb{F}$:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6. \tag{12.3}$$

Η 12.3, που ονομάζεται εξίσωση Weierstrass μετασχηματίζεται στις υπόλοιπες ανάλογα με την χαρακτηριστική του $\mathbb{F}$.

[1] Στην περίπτωση (12.1) η συνθήκη των πολλαπλών ριζών είναι ισοδύναμη με την πιο εύχρηστη συνθήκη $4a^3 + 27b^2 \neq 0$.

Παρατήρηση 22. Αν γράψουμε την (12.1) (ή τις αντίστοιχες πχ.(12.2)) στη μορφή $F(x,y) = 0$, δηλαδή $F(x,y) = y^2 - x^3 - ax - b$, τότε ένα σημείο (x,y) της καμπύλης θα λέγεται *ομαλό* (ή *μη ιδιάζον*) αν τουλάχιστον μία από τις μερικές παραγώγους [2] $\partial F/\partial x,\ \partial F/\partial y$ είναι μη μηδενική στο (x,y).
Εύκολα προκύπτει ότι η συνθήκη της μη ύπαρξης πολλαπλών ριζών στο πολυώνυμο 3ου βαθμού στο δεξί μέλος των παραπάνω εξισώσεων, ισοδυναμεί με την απαίτηση όλα τα σημεία της καμπύλης να είναι ομαλά.

12.2.2 Ελλειπτικές καμπύλες πάνω από το $\mathbb{R}$

Όπως προαναφέραμε στην εισαγωγή, η βασική ιδιότητα των ελλειπτικών καμπυλών που τις κάνει πολύτιμα κρυπτογραφικά εργαλεία είναι πως μας δίνουν με φυσικό τρόπο αβελιανές ομάδες με δομή τέτοια ώστε το DLOG να είναι υπολογιστικά δύσκολο. Πιο συγκεκριμένα, το σύνολο των σημείων μίας ελλειπτικής καμπύλης $\mathcal{E}$ εφοδιασμένο με μια πράξη πρόσθεσης (της οποίας τον ορισμό και την ερμηνεία θα δούμε παρακάτω) αποτελεί αβελιανή ομάδα, την οποία θα συμβολίζουμε $G(\mathcal{E})$.

Για τις εφαρμογές στην κρυπτογραφία που θα εξετάσουμε δε χρειαζόμαστε παρά μόνο ελλειπτικές καμπύλες πάνω από πεπερασμένα σώματα. Ωστόσο στο κομμάτι που ακολουθεί θα θεωρήσουμε $\mathbb{F} = \mathbb{R}$ προκειμένου να δοθεί ένας γεωμετρικός ορισμός για την πρόσθεση των σημείων της καμπύλης και να γίνει διαισθητικά σαφές ότι αυτά αποτελούν αβελιανή ομάδα. Έτσι σε όσα ακολουθούν σε αυτή την υποενότητα μία ελλειπτική καμπύλη θα είναι μία συνήθης καμπύλη στο επίπεδο μαζί με το "σημείο στο άπειρο" $\mathcal{O}$. Στο Σχήμα 12.2 φαίνονται τέσσερις ελλειπτικές καμπύλες για διάφορες τιμές των παραμέτρων a και b.

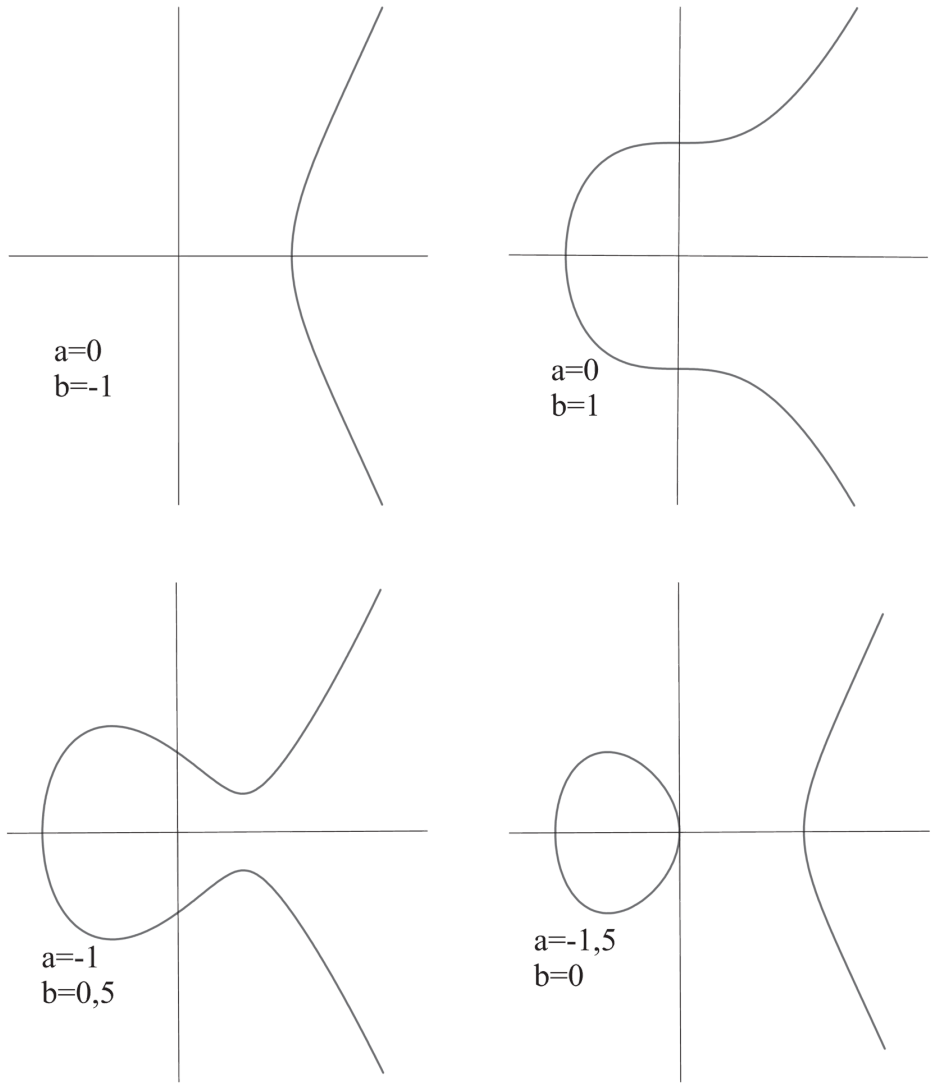
Ορισμός 12.2. Έστω $\mathcal{E}$ μία ελλειπτική καμπύλη πάνω από το $\mathbb{R}$, και έστω P, Q δύο σημεία πάνω στην $\mathcal{E}$. Τότε ορίζουμε το $-P$ (αντίθετο του P) και το άθροισμα $P + Q$ ως ακολούθως:

1. Αν $P = \mathcal{O}$, τότε ορίζουμε $-P = \mathcal{O}$ και $P + Q = Q$, δηλαδή το $\mathcal{O}$ είναι το ουδέτερο στοιχείο της ομάδας των σημείων της καμπύλης. Στα παρακάτω θεωρούμε $P \neq \mathcal{O} \neq Q$.

2. Το $-P$ είναι το σημείο με την ίδια τετμημένη και αντίθετη τεταγμένη από το P. Από τον ορισμό της ελλειπτικής καμπύλης είναι προφανές πως το $(x,-y)$ ανήκει στην καμπύλη αν και μόνο αν το (x,y) ανήκει στην καμπύλη.

3. Αν τα P, Q έχουν διαφορετικές τετμημένες, τότε είναι σχετικά εύκολο να παρατηρήσουμε ότι η ευθεία $\ell = \overline{PQ}$ τέμνει την καμπύλη ακριβώς σε ένα

[2] Η παράγωγος πολυωνύμου στο σώμα $\mathbb{F}$ ορίζεται με τον κανόνα nx^{n-1} και όχι μέσω ορίων, αφού για να έχει νόημα η έννοια του ορίου πρέπει να υπάρχει μια μετρική ορισμένη στο $\mathbb{F}$.



Σχήμα 12.2: Τέσσερις ελλειπτικές καμπύλες πάνω από το $\mathbb{R}$.

ακόμα σημείο R (εκτός εάν η ℓ εφάπτεται στο P, οπότε παίρνουμε $R = P$, ή στο Q, οπότε παίρνουμε $R = Q$). Ορίζουμε τότε $P + Q = -R$. Η γεωμετρική κατασκευή του σημείου $P + Q$ από τα P και Q γίνεται σαφέστερη στο Παράδειγμα 1.

4. Αν $Q = -P$, τότε ορίζουμε $P + Q = \mathcal{O}$.

5. Τέλος, αν $P = Q$, η εφαπτομένη στο P θα τέμνει την καμπύλη ακριβώς σε ένα ακόμα σημείο R (εκτός εάν το P είναι σημείο καμπής, οπότε παίρνουμε $R = P$). Ορίζουμε τότε $P + Q = -R$.

Παράδειγμα 17. Έστω η ελλειπτική καμπύλη $y^2 = x^3 - x$. Στο Σχήμα 12.3 αριστερά φαίνεται μια συνηθισμένη περίπτωση πρόσθεσης σημείων P, Q. Για να βρούμε το $P + Q$ επεκτείνουμε το ευθύγραμμο τμήμα $\overline{PQ}$ έως ότου τμήσει την καμπύλη στο R και έπειτα παίρνουμε το συμμετρικό σημείο ως προς τον άξονα των x. Στο Σχήμα 12.3 δεξιά τα P, Q συμπίπτουν, επομένως το $P + Q = 2P$ θα είναι το συμμετρικό του σημείου τομής της καμπύλης και της εφαπτομένης στο P.

Στη συνέχεια θα δείξουμε ότι πράγματι η ευθεία ℓ που διέρχεται από τα P και Q τέμνει την ελλειπτική καμπύλη $\mathcal{E}$ ακριβώς σε ένα ακόμα σημείο. Ταυτόχρονα, θα υπολογίσουμε τις συντεταγμένες του $P + Q$ βάσει των συντεταγμένων των P και Q.

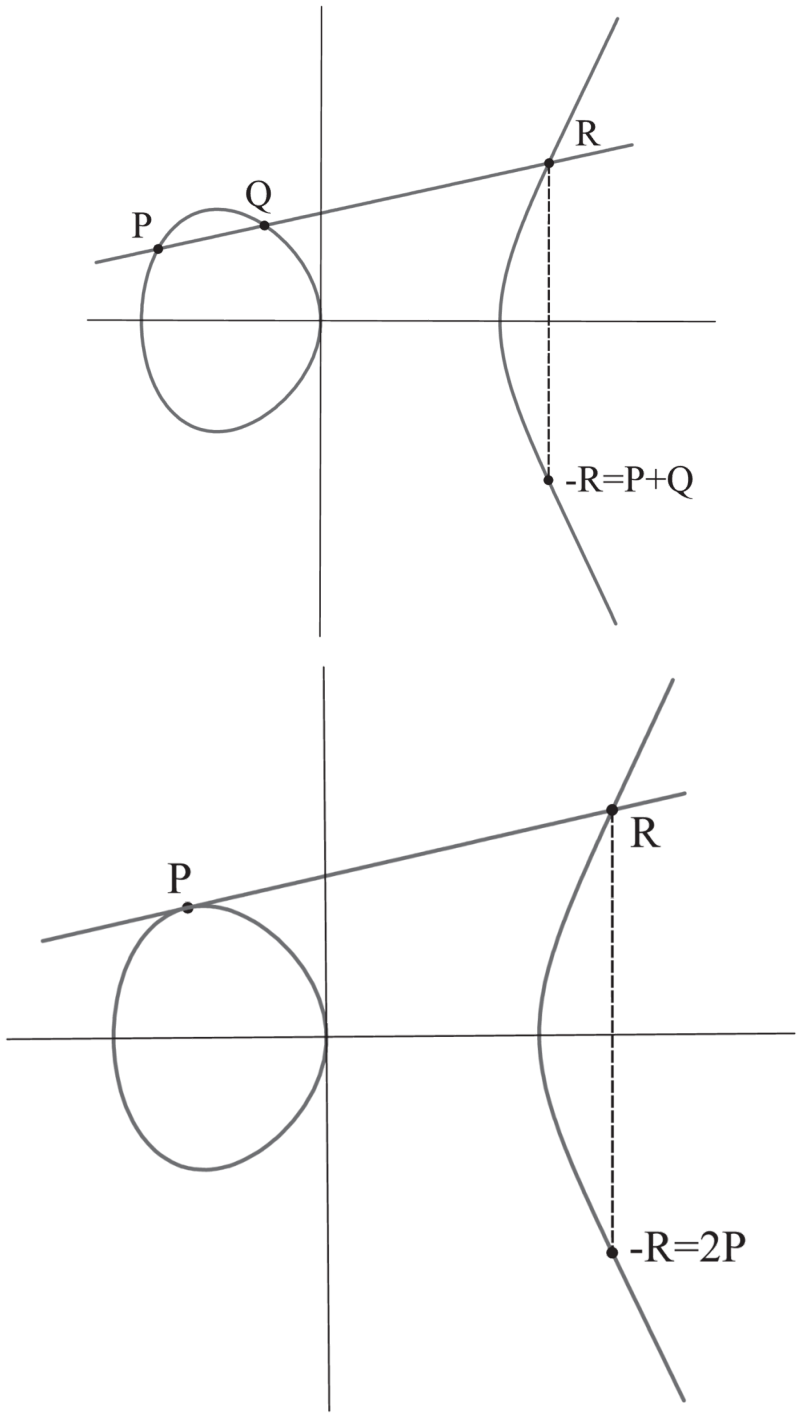
Έστω (x_1, y_1), (x_2, y_2) οι συντεταγμένες των σημείων P και Q αντίστοιχα, τα οποία ανήκουν στην ελλειπτική καμπύλη $\mathcal{E}$: $y^2 = x^3 + ax + b$, και (x_3, y_3) οι συντεταγμένες του $P + Q$. Οι περιπτώσεις του ορισμού της πρόσθεσης που χρειάζεται να εξετάσουμε είναι οι 3 και 5. Θα ξεκινήσουμε με την 3. Έστω $y = \beta x + \gamma$ η εξίσωση της ευθείας ℓ που διέρχεται από τα P και Q (η ℓ δεν είναι κατακόρυφη). Τότε $\beta = (y_2 - y_1)/(x_2 - x_1)$ και $\gamma = y_1 - \beta x_1$. Ένα σημείο της ℓ βρίσκεται πάνω στην $\mathcal{E}$ αν και μόνο αν $(\beta x + \gamma)^2 = x^3 + ax + b$, υπάρχει δηλαδή ένα σημείο τομής για κάθε ρίζα της κυβικής εξίσωσης

$$x^3 - \beta^2 x^2 + (a - 2\beta\gamma)x + b - \gamma^2 = 0.$$

Γνωρίζουμε ήδη ότι τα x_1, x_2 είναι ρίζες της εξίσωσης αφού $(x_1, \beta x_1 + \gamma)$, $(x_2, \beta x_2 + \gamma)$ είναι οι συντεταγμένες των P και Q. Επομένως[3] η τρίτη ρίζα είναι $x_3 = \beta^2 - x_1 - x_2$.

Από τον ορισμό της πρόσθεσης προκύπτει ότι $P + Q = (x_3, -(\beta x_3 + \gamma))$. Αν εκφράσουμε τελικά τα x_3, β και γ συναρτήσει των x_1, x_2, y_1, y_2 έχουμε:

[3] Είναι γνωστό πως το άθροισμα των ριζών του πολυωνύμου $a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$ ισούται με $\frac{-a_{n-1}}{a_n}$ (Vieta's Formulas).



Σχήμα 12.3: Πρόσθεση των σημείων P και Q, όταν $P \neq Q$ και όταν $P = Q$.

$$x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2\ , \quad y_3 = -y_1 + \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3). \qquad (12.4)$$

Η περίπτωση 5 είναι παρόμοια, με τη διαφορά ότι το β θα είναι τώρα η παράγωγος dy/dx στο P. Δηλαδή $\beta = (3x_1^2 + a)/2y_1$ και άρα για τις συντεταγμένες του $2P$ ισχύει:

$$x_3 = \left(\frac{3x_1^2 + a}{2y_1}\right)^2 - 2x_1\ , \quad y_3 = -y_1 + \left(\frac{3x_1^2 + a}{2y_1}\right)(x_1 - x_3). \qquad (12.5)$$

Παράδειγμα 18. Έστω τα σημεία $P = (-3, 9)$ και $Q = (-2, 8)$ της ελλειπτικής καμπύλης $y^2 = x^3 - 36x$. Αντικαθιστώντας $x_1 = -3$, $y_1 = 9$, $x_2 = -2$, $y_2 = 8$ στην 12.2.2 παίρνουμε $x_3 = 6$ και στη συνέχεια $y_3 = 0$, δηλαδή $P + Q = (6, 0)$. Παρόμοια, αν αντικαταστήσουμε $x_1 = -3$, $y_1 = 9$ και $a = -36$ στην 12.2.2 παίρνουμε $2P = (\frac{25}{4}, \frac{-35}{8})$.

Το γεγονός ότι το σύνολο των σημείων μιας ελλειπτικής καμπύλης εφοδιασμένο με την πρόσθεση όπως την ορίσαμε εδώ, αποτελεί ομάδα, αποδεικνύεται αρκετά δύσκολα και ο ενδιαφερόμενος αναγνώστης παραπέμπεται στη σχετική βιβλιογραφία [5].

Προτού εξετάσουμε ελλειπτικές καμπύλες ορισμένες σε πεπερασμένα σώματα, οφείλουμε να δώσουμε μία μαθηματική ερμηνεία για το "σημείο στο άπειρο" $\mathcal{O}$. Όπως είδαμε, είναι εξορισμού το ουδέτερο στοιχείο της ομάδας $G(\mathcal{E})$, ενώ γραφικά θα πρέπει να το φανταστούμε ως το τρίτο σημείο τομής μεταξύ της ελλειπτικής καμπύλης και κάθε κατακόρυφης ευθείας. Ένας πιο φυσικός τρόπος να εισάγουμε το $\mathcal{O}$ είναι ο ακόλουθος.

Στο σύνολο των τριάδων (X, Y, Z) (όχι όλα ταυτόχρονα μηδέν) θεωρούμε τη σχέση ισοδυναμίας $(X, Y, Z) \sim (\lambda X, \lambda Y, \lambda Z)$, δηλαδή δύο τριάδες είναι ισοδύναμες αν η μία είναι (βαθμωτό) πολλαπλάσιο της άλλης. Μία κλάση ισοδυναμίας $[(X, Y, Z)]_\sim$ λέγεται προβολικό σημείο. Ονομάζουμε *προβολικό επίπεδο* το σύνολο των προβολικών σημείων. Αν ένα σημείο (X, Y, Z) έχει $Z \neq 0$ τότε υπάρχει μοναδικό σημείο $(x, y, 1)$, τέτοιο ώστε $(X, Y, Z) \in [(x, y, 1)]_\sim$ (απλά θέτουμε $x = X/Z, y = Y/Z$). Επομένως ,θα μπορούσαμε να πούμε πως το προβολικό επίπεδο περιέχει όλα τα σημεία (x, y) του συνήθους (αφφινικού) επιπέδου, καθώς και τα προβολικά σημεία για τα οποία $Z = 0$. Τα τελευταία βρίσκονται πάνω σε μία ευθεία, την αποκαλούμενη *επ'άπειρον ευθεία*.

Κάθε εξίσωση καμπύλης $F(x, y) = 0$ στο αφφινικό επίπεδο, αντιστοιχεί σε μία εξίσωση $\tilde{F}(X, Y, Z) = 0$, που ικανοποιείται από τα αντίστοιχα προβολικά σημεία: απλά αντικαθιστούμε το x με X/Z, το y με Y/Z και πολλαπλασιάζουμε

με κατάλληλη δύναμη του Z. Για παράδειγμα, αν εφαρμόσουμε αυτή την διαδικασία στην εξίσωση 12.1 θα πάρουμε την αντίστοιχη «προβολική εξίσωση» $Y^2Z = X^3 + aXZ^2 + bZ^3$. Η τελευταία ικανοποιείται προφανώς για κάθε σημείο (X, Y, Z) για το οποίο το $(X/Z, Y/Z)$ ικανοποιεί την 12.1. Ας εξετάσουμε ωστόσο ποια σημεία της επ'άπειρον ευθείας ικανοποιούν την εξίσωση. Θέτοντας $Z = 0$, η εξίσωση γίνεται $0 = X^3$, δηλαδή $X = 0$. Όμως η μόνη κλάση ισοδυναμίας με $X = Z = 0$ είναι το προβολικό σημείο $[(0, 1, 0)]_\sim$. Ακριβώς αυτό το σημείο αποκαλούμε "σημείο στο άπειρο". Το $\mathcal{O}$ δηλαδή είναι το σημείο τομής του άξονα $y'y$ με την επ'άπειρον ευθεία.

12.2.3 Ελλειπτικές καμπύλες πάνω από το $GF(p)$

Για το υπόλοιπο του κεφαλαίου θα θεωρούμε κάθε ελλειπτική καμπύλη $\mathcal{E}$ ορισμένη πάνω από το πεπερασμένο σώμα $GF(p)$, όπου p πρώτος. Έτσι η σχέση 12.1 γίνεται:

$$\mathcal{E} = \{(x, y) \in GF(p) : y^2 = x^3 + ax + b \bmod p,$$
$$a, b \in GF(P) : 4 * a^3 + 27 * b^2 \neq 0 \pmod p\} \cup \mathcal{O}$$

Εύκολα παρατηρούμε πως μια τέτοια ελλειπτική καμπύλη μπορεί να έχει το πολύ $2p + 1$ σημεία, το σημείο $\mathcal{O}$ μαζί με $2p$ ζεύγη $(x, y),\ x, y \in GF(p)$, που ικανοποιούν την 12.1 Αυτό συμβαίνει επειδή για κάθε ένα από τα p πιθανά x υπάρχουν το πολύ δύο y ώστε να ικανοποιείται η εξίσωση της ελλειπτικής καμπύλης. Καθώς μόνο τα μισά από τα στοιχεία του $GF(p)^*$ έχουν τετραγωνική ρίζα, είναι φυσικό να περιμένουμε ότι υπάρχουν περίπου τα μισά από αυτά τα $2p + 1$ σημεία στην καμπύλη. Πιο συγκεκριμένα, ισχύει το παρακάτω σημαντικό θεώρημα του Hasse.

Θεώρημα 12.3. *Θέωρημα Hasse Έστω $\#\mathcal{E}$ το πλήθος των σημείων μίας ελλειπτικής καμπύλης $\mathcal{E}$ ορισμένης πάνω από το $GF(p)$. Τότε ισχύει*

$$p + 1 - 2\sqrt{p} \leq \#\mathcal{E} \leq p + 1 + 2\sqrt{p}.$$

Ο ακριβής υπολογισμός του $\#\mathcal{E}$ είναι πιο δύσκολος, υπάρχει ωστόσο αποδοτικός αλγόριθμος, ο αλγόριθμος του Schoof [6] (χρονικής πολυπλοκότητας $O((\log p)^8)$), οποίος έχει πρακτική εφαρμογή για πρώτους p αρκετών εκατοντάδων ψηφίων. Δεδομένου ότι μπορούμε να υπολογίσουμε το $\#\mathcal{E}$, θα μας ενδιέφερε να γνωρίζουμε περισσότερα για τη δομή της $G(\mathcal{E})$. Πιο συγκεκριμένα μας ενδιαφέρει να βρούμε μια "μεγάλη" κυκλική υποομάδα της $G(\mathcal{E})$ (η ίδια η $G(\mathcal{E})$ κατά κανόνα δεν είναι κυκλική). Παρακάτω θα συμβολίζουμε με q την τάξη της υποομάδας αυτής.

Για να βρούμε την τάξη q της υποομάδας που παράγεται από ένα σημείο G μπορούμε να εκμεταλλευτούμε το θεώρημα Lagrange (2.34) και να εκτελέσουμε τα παρακάτω βήματα:

- Υπολογίζουμε το $\#\mathcal{E}$ (την τάξη δηλ. της ελλειπτικής καμπύλης με τον αλγόριθμο του Schoof
- Βρίσκουμε τον μικρότερο διαιρέτη q του $\#\mathcal{E}$ ώστε $q \cdot G = \mathcal{O}$

12.2.4 Εφαρμογές στην κρυπτογραφία δημοσίου κλειδιού

Στην ενότητα αυτή, θα ορίσουμε αρχικά το πρόβλημα του διακριτού λογαρίθμου στις ελλειπτικές καμπύλες (Elliptic Curve Discrete Logarithm Problem) και, θα δώσουμε τα ανάλογα κρυπτοσυστημάτων στις ελλειπτικές καμπύλες.

Ορισμός 12.4. Το *Πρόβλημα Διακριτού Λογαρίθμου στις Ελλειπτικές Καμπύλες* (ECDLP) είναι το παρακάτω:

Δίνονται: Μία ελλειπτική καμπύλη $\mathcal{E}$ ορισμένη πάνω από το $GF(p)$, ένα σημείο τής G (η βάση του διακριτού λογαρίθμου) και ένα σημείο της Y.

Ζητείται: Να βρεθεί, αν υπάρχει, ακέραιος x τέτοιος ώστε $xG = Y$.

Το ECDLP φαίνεται να είναι δυσκολότερο από το DLOG. Μάλιστα για το δεύτερο υπάρχει, όπως είδαμε, αλγόριθμος υποεκθετικού χρόνου, ενώ για το πρώτο όχι. Έτσι, αν και δεν υπάρχει απόδειξη που να μας εξασφαλίζει ότι το ECDLP είναι πράγματι δυσκολότερο από το DLOG, με τα σημερινά δεδομένα έχουμε εξίσου καλή ασφάλεια με αρκετά μικρότερο μήκος κλειδιού όταν χρησιμοποιούμε κρυπτοσυστήματα ελλειπτικών καμπυλών. Ας δούμε κάποια από αυτά σε πιο πρακτικό επίπεδο από την προηγούμενη ενότητα.

Για τους παρακάτω αλγόριθμους πρέπει να συμφωνηθούν κάποιες παράμετροι. Συγκεκριμένα:

- Μία ελλειπτική καμπύλη $\mathcal{E}$ η οποία θα καθορίζεται από κάποιες παραμέτρους a, b.
- Ένας πρώτος p που καθορίζει το μέγεθος του πεπερασμένου σώματος.
- Η τάξη q της κυκλικής υποομάδας
- Ένα σημείο βάσης G για την παραγωγή της υποομάδας

Το ανάλογο της ανταλλαγής κλειδιού Diffie-Hellman

Υποθέτουμε πως η Alice και ο Bob θέλουν να συμφωνήσουν σε ένα κλειδί, το οποίο αργότερα θα χρησιμοποιούν για να επικοινωνούν με ένα συμμετρικό κρυπτοσύστημα. Το κλειδί τους θα κατασκευαστεί από κάποιο σημείο P της $\mathcal{E}$. Ο σκοπός είναι να επιλεγεί το P με τέτοιο τρόπο, ώστε η επικοινωνία της Alice και του Bob να γίνεται δημόσια, χωρίς ωστόσο να γνωρίζει οποιοσδήποτε άλλος, εκτός απ' τους δύο τους, ποιο είναι το P.

Αφού η Alice και ο Bob έχουν επιλέξει δημόσια το σημείο βάσης G το οποίο μπορούμε να θεωρήσουμε πως παίζει το ρόλο του γεννήτορα g στο "κλασσικό" Diffie-Hellman. Εδώ όμως δε μας ενδιαφέρει αν το G είναι γεννήτορας της $G(\mathcal{E})$, η οποία, όπως είπαμε ήδη, πιθανόν να μην είναι κυκλική. Μάλιστα εδώ δεν χρειαζόμαστε ούτε καν την ακριβή τιμή του $\#\mathcal{E}$. Το μόνο που θέλουμε είναι η κυκλική υποομάδα που παράγεται από το G να είναι μεγάλη, κατά προτίμηση της ίδιας τάξης μεγέθους με τη $G(\mathcal{E})$.

Στη συνέχεια η Alice επιλέγει έναν ακέραιο a της ίδιας τάξης μεγέθους με το q, τον οποίο γνωρίζει μόνο αυτή. Υπολογίζει το $aG \in \mathcal{E}$ και το δημοσιοποιεί. Ο Bob κάνει το ίδιο. Επιλέγει έναν ακέραιο b και δημοσιοποιεί το $bG \in \mathcal{E}$. Το μυστικό κλειδί που θα χρησιμοποιούν στη συνέχεια είναι το $P = abG \in \mathcal{E}$. Προφανώς και οι δύο μπορούν να υπολογίσουν το P, για παράδειγμα ο Bob γνωρίζει το aB και το μυστικό του b, άρα $P = abG = baG$. Οποιοσδήποτε όμως εκτός από την Alice και τον Bob γνωρίζει μόνο τα aG και bG (και το G φυσικά). Για να υπολογίσει λοιπόν ένας τρίτος το P δε φαίνεται να έχει άλλη λύση από το να βρει το a γνωρίζοντας τα aG και G (ή το b γνωρίζοντας τα bG και G), δηλαδή να λύσει το ECDLP στην $\mathcal{E}$.

Το ανάλογο του κρυπτοσυστήματος ElGamal

Όπως και στο Diffie-Hellman παραπάνω, ξεκινάμε από μία συμφωνημένη πλειάδα παραμέτρων. Επιπλέον κάθε χρήστης επιλέγει έναν ακέραιο x, τον οποίο κρατάει μυστικό (ιδιωτικό κλειδί), ενώ υπολογίζει και δημοσιοποιεί το σημείο $Y = xG$ (δημόσιο κλειδί).

Για να στείλει η Alice το μήνυμα P_m στον Bob, επιλέγει τυχαία έναν ακέραιο k και στέλνει το ζεύγος σημείων $(kG, P_m + kY_B)$, όπου Y_B το δημόσιο κλειδί του Bob. Για να διαβάσει το μήνυμα ο Bob, πολλαπλασιάζει το πρώτο σημείο με x_B και αφαιρεί το αποτέλεσμα από το δεύτερο. Πράγματι

$$P_m + k(Y_B) - x_B(kG) = P_m.$$

Ένα τρίτο πρόσωπο που μπορεί να λύσει το ECDLP στην $\mathcal{E}$, μπορεί φυσικά να υπολογίσει το x_B από τα γνωστά Y_B καιG και να διαβάσει το P_m.

Παράδειγμα 19. Ας δούμε ένα παράδειγμα της κρυπτογράφησης ElGamal, χρησιμοποιώντας την ελλειπτική καμπύλη $y^2 = x^3 + x + 6$ ορισμένη πάνω από το $\mathbb{Z}_{11}$. Εδώ $\#\mathcal{E} = 13$. Υποθέτουμε πως $G = (2, 7)$ και πως το ιδιωτικό κλειδί που επιλέγει ο Bob είναι το $x_B = 7$, επομένως το δημόσιο κλειδί του θα είναι

$$Y_B = 7(2, 7) = (7, 2)$$

Έτσι η συνάρτηση κρυπτογράφησης για τον Bob είναι

$$e_K(P_m, k) = (k(2, 7), P_m + k(7, 2)),$$

όπου το P_m είναι το σημείο της καμπύλης που θέλουμε να στείλουμε και k ακέραιος, που χωρίς βλάβη της γενικότητας μπορούμε να θεωρήσουμε μεταξύ του 0 και του 12. Ο περιορισμός αυτός του k προκύπτει από τη σχέση $13 \cdot B = \mathcal{O}$. Ας υποθέσουμε πως η Alice θέλει να στείλει το μήνυμα $P_m = (10, 9)$ (το οποίο είναι σημείο της $\mathcal{E}$). Αν επιλέξει αυθαίρετα $k = 3$, τότε θα υπολογίσει

$$\begin{aligned} y_1 &= 3(2, 7) \\ &= (8, 3) \end{aligned}$$

και

$$\begin{aligned} y_2 &= (10, 9) + 3(7, 2) \\ &= (10, 9) + (3, 5) \\ &= (10, 2) \end{aligned}$$

και στέλνει επομένως το $y = ((8, 3), (10, 2))$. Στη συνέχεια ο Bob αποκρυπτογραφεί το y ως εξής:

$$\begin{aligned} P_m &= (10, 2) - 7(8, 3) \\ &= (10, 2) - (3, 5) \\ &= (10, 2) + (3, 6) \\ &= (10, 9), \end{aligned}$$

παίρνοντας έτσι το αρχικό μήνυμα.

Το ανάλογο του DSA

Αντίστοιχα με το ElGamal μπορούμε να προσαρμόσουμε και τον αλγόριθμο ψηφιακών υπογραφών του 7.5 στις ελλειπτικές καμπύλες.

Όπως και πριν κάθε χρήστης έχει ένα μυστικό κλειδί x και ένα δημόσιο $Y = xG$ όπου G το σημείο βάσης στο οποίο έχουν συμφωνήσει. Επίσης υπάρχει διαθέσιμη και μία συνάρτηση σύνοψης $\mathcal{H}$.

Ο αλγόριθμος δημιουργίας υπογραφών είναι ο εξής:

- Υπολογισμός σύνοψης του μηνύματος $h = \mathcal{H}(M)$ και προσαρμογή της στο $[0, \cdots, q - 1]$

- Επιλογή τυχαίου αριθμού k στο σύνολο $[0, \cdots, q-1]$
- Υπολογισμός του σημείου $P = (r, y_P) = kG$. Αν $r = 0 \pmod q$ τότε επιλέγεται καινούριο k και η διαδικασία επαναλαμβάνεται.
- Υπολογισμός του $s = k^{-1}(h + rx) \bmod q$
- Αν $s = 0$ τότε επανάληψη της διαδικασίας
- Η υπογραφή είναι το ζεύγος (r, s)

Ο αλγόριθμος επαλήθευσης των υπογραφών είναι ο εξής:

- Υπολογισμός του $u_1 = s^{-1}h \bmod q$
- Υπολογισμός του $u_2 = s^{-1}r \bmod q$
- Υπολογισμός του σημείου $P' = u_1G + u_2Y$
- Η υπογραφή είναι έγκυρη αν $r = x_p \pmod q$

Για την ορθότητα της επαλήθευσης παρατηρούμε ότι κατά την επαλήθευση ουσιαστικά υπολογίζουμε το ίδιο σημείο με διαφορετικούς τρόπους.

$$P' = u_1G + u_2Y = s^{-1}(h+rx)G = k(h+rx)^{-1}(h+rx)G = kG = P \quad (12.6)$$

Η τιμή του k πρέπει να διατηρείται κρυφή κατά τη δημιουργία των υπογραφών αλλά και να αλλάζει κάθε φορά που δημιουργείται νέα υπογραφή (βλ. και άσκηση 1).

12.3 Ζεύξεις και Διγραμμικές Απεικονίσεις

Στην ενότητα αυτή θα ασχοληθούμε με τις ζεύξεις, οι οποίες αποτελούν ένα πολύ σημαντικό σύγχρονο κρυπτογραφικό εργαλείο με πολλές εφαρμογές.

12.3.1 Εισαγωγή

Μία ζεύξη είναι μία συνάρτηση η οποία αντιστοιχεί στοιχεία από μία ομάδα πηγή $\mathcal{G}$ σε μία ομάδα προορισμό $\mathcal{G}_T$. Το χαρακτηριστικό τους που μας ενδιαφέρει στην Κρυπτογραφία είναι ότι ενώ στο $\mathcal{G}$ κάποια προβλήματα είναι δύσκολα, μπορούν να γίνουν εύκολα μέσω της ζεύξης στο $\mathcal{G}_T$. Συγκεκριμένα:

Ορισμός 12.5. Μία ζεύξη (pairing) είναι μία αποδοτικά υπολογίσιμη συνάρτηση $e : \mathcal{G} \times \mathcal{G} \to \mathcal{G}_T$ η οποία είναι:

- Διγραμμική (bilinear): $e(g^a, g^b) = e(g, g)^{ab}$ όπου $g \in \mathcal{G}$ $a, b \in \mathbb{Z}$
- Μη εκφυλισμένη (non-degenerate): Αν $\mathcal{G} =< g >$ τότε $\mathcal{G}_T =< e(g, g) >$

Ένα τέτοιο παράδειγμα αποτελεί η αντιστοίχιση σημείων ελλειπτικής καμπύλης σε ένα πεπερασμένο σώμα (δηλ. $\mathcal{E}(GF(p)) \times \mathcal{E}(GF(p)) \to GF(p^a)$) Από τον παραπάνω ορισμό προκύπτει ότι οι ζεύξεις είναι 'συμμετρικές':

$$e(g^a, h^b) = e(g^b, h^a) = e(g, h)^{ab} \tag{12.7}$$

Το πρόβλημα DLOG (4.8) μπορεί να αναχθεί από το $\mathcal{G}$ στο $\mathcal{G}_T$. Πράγματι αν θέλουμε να υπολογίσουμε τον διακριτό λογάριθμο x ενός στοιχείου $y = g^x$ μπορούμε να το κάνουμε ελέγχοντας τις ζεύξεις $e(g, y)$ και $e(g, g)^x$. Στην μία ομάδα όμως αυτό μπορεί να είναι πιο εύκολο από την άλλη, κάτι που είχε ανησυχήσει αρχικά πολλούς κρυπτογράφους.

Σε κάθε περίπτωση οι ζεύξεις καθιστούν το DDH (6.5.1) εύκολα υπολογίσιμο. Για να δούμε αν $g^c = g^{ab}$ μπορούμε να υπολογίζουμε (αποδοτικά) το $e(g^a, g^b) = e(g, g)^{ab}$ και να το συγκρίνουμε με το $e(g, g^c) = e(g, g)^c$

Αυτό σημαίνει ότι το DDH πρέπει να αντικατασταθεί. Για τον σκοπό αυτό έχει οριστεί το *διγραμμικό ανάλογο του* ως εξής:

Ορισμός 12.6. Διγραμμικό Πρόβλημα Απόφασης των Diffie-Hellman (Bilinear Decisional Diffie Hellman Problem (BDDH))

Δίνονται: δύο στοιχεία $h, g \in \mathcal{G}$ και τα στοιχεία $g^\alpha, g^\beta, e(h, g)^c$.
Ζητείται: Ισχύει $c = \alpha\beta$;

Μπορεί να αποδειχθεί ότι $BDDHP \Rightarrow DDHP$ στο $\mathcal{G}_T$ (άσκηση 2).

12.3.2 Τριμερής Ανταλλαγή Κλειδιού

Μία από τις πρώτες εφαρμογές των ζεύξεων στην κρυπτογραφία δόθηκε από τον A. Joux στο [7] και αφορούσε την ανταλλαγή κλειδιών Diffie Hellman μεταξύ τριών οντοτήτων.

Αν υποθέσουμε ότι δουλεύουμε σε μία κυκλική ομάδα με γεννήτορα g και οι τρεις οντότητες A, B, C έχουν ζευγάρια ιδιωτικών - δημοσίων κλειδιών $(x_A, y_A = g^{x_A}), (x_B, y_B = g^{x_B}), (x_C, y_C = g^{x_C})$. Μπορεί να συμφωνηθεί ένα κοινό κλειδί μεταξύ τους ως εξής:

Αυτό μπορεί να γίνει σε τρεις γύρους ως εξής:

1. Ο A στέλνει το y_A στον B, ο B στέλνει το y_B στον C, ο C στέλνει το y_C στον A (κυκλικά).
2. Ο A υπολογίζει το $t_A = y_C^{x_A} = g^{x_C x_A}$, ο B υπολογίζει το $t_B = y_A^{x_B} = g^{x_B x_A}$ και ο C υπολογίζει το $t_C = y_B^{x_C} = g^{x_B x_C}$
3. Ο A στέλνει το t_A στον B, ο B στέλνει το t_B στον C, ο C στέλνει το t_C στον A (πάλι κυκλικά).
4. Όλοι υπολογίζουν το κοινό κλειδί ως εξής:
   - Ο A με $t_C^{x_A} = g^{x_B x_C x_A}$
   - Ο B με $t_A^{x_B} = g^{x_C x_A x_B}$
   - Ο C με $t_B^{x_C} = g^{x_A x_B x_C}$

Στο [7] προτάθηκε το ίδιο πρωτόκολλο με ένα γύρο ανταλλαγής μηνυμάτων χρησιμοποιώντας ζεύξεις. Υποθέτουμε δύο ομάδες $\mathcal{G}_1$, $\mathcal{G}_2$ με τάξη ένα πρώτο p και μία συμμετρική διγραμμική ζεύξη $e : \mathcal{G}_1 \times \mathcal{G}_1 \to \mathcal{G}_2$.

- Όλοι οι συμμετέχοντες εκπέμπουν τα δημόσια κλειδιά τους $y_A = g^{x_A}, y_B = g^{x_B}, y_C = g^{x_C}$.
- Με την βοήθεια της ζεύξης το κοινό κλειδί μπορεί να υπολογιστεί ως εξής:
  - $e(g^{x_B}, g^{x_C})^{x_A} = e(g, g)^{x_B x_C x_A}$
  - $e(g^{x_A}, g^{x_C})^{x_B} = e(g, g)^{x_A x_C x_B}$
  - $e(g^{x_A}, g^{x_B})^{x_C} = e(g, g)^{x_A x_B x_C}$

12.3.3 Εφαρμογές

Η εργασία του Joux προκάλεσε το ενδιαφέρον των κρυπτογράφων για τις ζεύξεις και οδήγησε σε αρκετές εφαρμογές τους.

Σύντομες υπογραφές διακριτού λογαρίθμου

Η πρώτη τέτοια εφαρμογή προτάθηκε από τους Boheh, Lynn και Shacham στο [8]. Προσπαθεί να παρουσιάσει υπογραφές που βασίζονται στο DLOG, αλλά έχουν μικρότερο μέγεθος από αυτές που αναφέραμε στις ενότητες 7.2 και 7.5 οι οποίες αποτελούνται από δύο ή περισσότερους ακέραιους μεγέθους όσο η τάξη της επιλεγμένης ομάδας. Συγκεκριμένα οι υπογραφές του [8] έχουν μέγεθος όσο ένας ακέραιος και είναι συγκρίσιμες με τις υπογραφές RSA (7.1).

Με απλά λόγια το σχήμα υπογραφών BLS λειτουργεί ως εξής:

- Οι συμμετέχοντες συμφωνούν σε μία διγραμμική απεικόνιση e μεταξύ των ομάδων $(\mathcal{G}, \mathcal{G}_T)$ όπου το CDH είναι δύσκολο στο $\mathcal{G}$. Υποθέτουμε πως το $\mathcal{G}$ παράγεται από το G και έχει τάξη n

- Το ιδιωτικό κλειδί του αποστολέα είναι ένας ακέραιος $a \in [1, n-1]$ ενώ το δημόσιο είναι το σημείο $A = aG$ (με προσθετικό συμβολισμό).

- Για να παραχθεί η υπογραφή σε ένα μήνυμα m παράγεται η σύνοψη $M = \mathcal{H}(m)$ και υπολογίζεται το στοιχείο $S = aM$ το οποίο είναι και η υπογραφή.

- Για την επαλήθευση αρκεί να ελεγχθεί ότι το $\mathcal{G}, M, A, S$ ικανοποιεί το DDHP το οποίο όπως είπαμε είναι εύκολο μέσω της ζεύξης. Αρκεί να ελέγξουμε δηλαδή αν $e(\mathcal{G}, A) = e(M, S)$

Από την άλλη η πλαστογράφηση παραμένει δύσκολη καθώς πρέπει να λυθεί το CDH στο $\mathcal{G}$.

12.3.4 Κρυπτογράφηση με βάση την ταυτότητα

Όπως είδαμε στην ενότητα 9.1 για να λειτουργήσει πρακτικά η κρυπτογραφία δημοσίου κλειδιού απαιτεί την ύπαρξη μίας υποδομής για την διανομή των δημοσίων κλειδιών και κυρίως εμπιστοσύνη σε αυτή την υποδομή. Προτού ακόμα εφαρμοστούν οι πρώτες υποδομές δημοσίων κλειδιών ο Shamir [9] πρότεινε μια εναλλακτική λύση, την *κρυπτογράφηση με βάση την ταυτότητα (Identity Based Encryption (IBE))*. Η βασική ιδέα του σχήματος αφορά την χρήση των ταυτοτήτων των χρηστών για δημόσια κλειδιά.

Συγκεκριμένα, θεωρούμε πως, κάθε χρήστης ενός IBE κρυπτοσυστήματος διαθέτει μία ταυτότητα, όπως για παράδειγμα μιας διεύθυνσης ηλεκτρονικού ταχυδρομείου. Μία έμπιστη τρίτη οντότητα αναλαμβάνει να δημιουργήσει το ιδιωτικό κλειδί από την ταυτότητα και να το παραδώσει ασφαλώς στον χρήστη. Οποιοσδήποτε χρήστης θέλει να κρυπτογραφήσει ένα μήνυμα, το κάνει με βάση την ταυτότητα του παραλήπτη και το δημόσιο κλειδί της έμπιστης αρχής (ακόμα και πριν την δημιουργία του ιδιωτικού κλειδιού).

Η πρώτη πρακτική υλοποίηση IBE δόθηκε από τους Boneh και Franklin στο [10] μέσω ζεύξεων. Λειτουργεί ως εξής:

- Οι συμμετέχοντες συμφωνούν σε μία διγραμμική απεικόνιση e μεταξύ των ομάδων $(\mathcal{G}, \mathcal{G}_T)$ όπου το BDDH είναι δύσκολο. Υποθέτουμε πως το $\mathcal{G}$ παράγεται από το G και έχει τάξη n. Επίσης υποθέτουμε δύο συναρτήσεις σύνοψης $\mathcal{H}_{\mathcal{G}}, \mathcal{H}$ στο $\mathcal{G}$ και στο $\mathtt{M}$ αντίστοιχα.

- Η έμπιστη αρχή έχει ως ιδιωτικό κλειδί το $t \in [1, \cdots, n-1]$ και δημόσιο κλειδί το $T = tG$.
- Για την παραγωγή του ιδιωτικού κλειδιού χρησιμοποιείται η συνάρτηση σύνοψης $\mathcal{H}_G$ και παράγεται το $x = t\mathcal{H}_G(ID)$
- Για την κρυπτογράφηση ενός μηνύματος προς τον χρήστη ID, ο αποστολέας:
  - υπολογίζει αρχικά το $Y = \mathcal{H}_G(ID)$
  - Όπως και στο Elgamal επιλέγεται ένα τυχαίο $r \in [1, n-1]$ και υπολογίζει το $R = rG$. Στην συνέχεια εφαρμόζεται η ζεύξη και υπολογίζεται το $C = m \oplus \mathcal{H}(e(Y,T)^r)$
  - Το κρυπτοκείμενο είναι το ζεύγος (R, C).
- Η αποκρυπτογράφηση γίνεται χρησιμοποιώντας το ιδιωτικό κλειδί και τη ζεύξη $m = c \oplus \mathcal{H}(e(x,R))$

Η ορθότητα του κρυπτοσυστήματος βασίζεται στη διγραμμικότητα της ζεύξης:

$$e(x,R) = e(tY, rG) = e(Y,G)^{rt} = e(Y,tG)^r = e(Y,T)^r$$

Η ασφάλεια του κρυπτοσυστήματος βασίζεται στο BDDH.

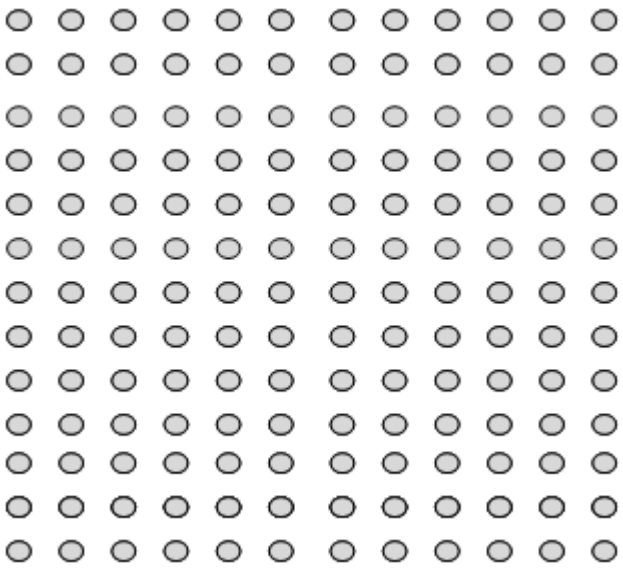
Ο προσεκτικός αναγνώστης θα προσέξει ότι η IBE έχει και αυτή ανάλογα προβλήματα με το PKI, όπως για παράδειγμα τη διανομή του δημοσίου κλειδιού της έμπιστης αρχής ή την ασφαλή διανομή των ιδιωτικών κλειδιών στους χρήστες. Μια αναλυτική σύγκριση μπορεί να βρεθεί στο [11].

12.4 Δικτυωτά (Lattices)

Τα δικτυωτά είναι γεωμετρικά αντικείμενα με πλούσια δομή τα οποία έχουν μαθηματική ιστορία πάνω από 200 έτη ξεκινώντας από τον Gauss, τον Hermite και τον Minkowski τον 19ο αιώνα. Μάλιστα κατά μία έννοια, μπορεί να ειπωθεί ότι η μελέτη των δικτυωτών (με μία διάσταση) ξεκίνησε από τον Ευκλείδη, με τον ΜΚΔ.

Ένα *δικτυωτό* είναι ένα σύνολο σημείων στον n-διάστατο χώρο με περιοδική δομή. Πιο τυπικά, δικτυωτό είναι οποιοσδήποτε γραμμικός συνδυασμός με ακέραιους συντελεστές μιας σειράς γραμμικώς ανεξάρτητων διανυσμάτων $\{\mathbf{b_i}\}_{i=1}^n$ τα οποία ονομάζονται *βάση* του δικτυωτού:

$$\mathcal{L}(\mathbf{b_1}, \cdots, \mathbf{b_n}) = \sum_{i=1}^n x_i \cdot b_i, \quad x_i \in \mathbb{Z}$$



Σχήμα 12.4: Δισδιάστατο δικτυωτό

Πολλές φορές βολεύει ο συμβολισμός δικτυωτών χρησιμοποιώντας πίνακες, οπότε ο παραπάνω ορισμός γίνεται:

$$\mathcal{L}(B) = B \cdot \mathbf{x}, \quad \mathbf{x} \in \mathbb{Z}^n$$

Κάθε δικτυωτό μπορεί να παραχθεί από περισσότερες από μία βάσεις. Μάλιστα αποδεικνύεται πως αν **U** ένας unimodular πίνακας ισχύει:

$$\mathcal{L}(B) = \mathcal{L}(B \cdot U)$$

Η παραπάνω ιδιότητα χρησιμοποιείται ώστε να κατασκευαστούν κρυπτοσυστήματα δημοσίου κλειδιού, όπου το δημόσιο κλειδί είναι μία βάση, ενώ το ιδιωτικό κάποια άλλη.

Το σημαντικότερο πρόβλημα των δικτυωτών, πάνω στο οποίο χτίζονται οι εφαρμογές τους στην Κρυπτογραφία είναι το *Πρόβλημα του Μικρότερου Διανύσματος (Shortest Vector Problem - SVP) και η προσεγγιστική παραλλαγή του.*

Ορισμός 12.7. Shortest Vector Problem SVP

Δίνεται μία βάση B. Να βρεθεί το μικρότερο μη μηδενικό διάνυσμα στο $\mathcal{L}(B)$.

Ορισμός 12.8. α - Shortest Vector Problem SVP

Δίνεται μία βάση B. Να βρεθεί το ένα διάνυσμα το οποίο είναι το πολύ $\alpha(n)$ φορές το ελάχιστο μη μηδενικό διάνυσμα στο $\mathcal{L}(B)$, όπου n η διάσταση του δικτυωτού.

Φυσικά, πρέπει να δοθεί ένα μέτρο της απόστασης ώστε να ορίσουμε τι σημαίνει μικρότερο διάνυσμα ή ελάχιστη απόσταση.

Ένα σχετικό πρόβλημα είναι και το Πρόβλημα του Κοντινότερου Διανύσματος (Closest Vector Problem - CVP) και η προσεγγιστική παραλλαγή του.

Ορισμός 12.9. Closest Vector Problem - SVP

Δίνεται μία βάση B και ένα διάνυσμα $\mathbf{v}$. Να βρεθεί το σημείο $\mathbf{p} \in \mathcal{L}(B)$, πιο κοντά στο $\mathbf{v}$.

Το πρώτο σημαντικό επίτευγμα χρήσης των δικτυωτών στην Πληροφορική είναι ο αλγόριθμος LLL [12] για την προσέγγιση του μικρότερου διανύσματος (shortest vector) σε κάποιο δικτυωτό. Ο συγκεκριμένος αλγόριθμος έχει πάρα πολλές εφαρμογές στην πληροφορική όπως για παράδειγμα την παραγοντοποίηση πολυωνύμων στο $\mathbb{Q}$ αλλά και την επίλυση ακέραιων γραμμικών προγραμμάτων.

Στην κρυπτογραφία τα πλέγματα χρησιμοποιήθηκαν αρχικά για κρυπτανάλυση. Η χρήση των δικτυωτών για κρυπτογραφία έγινε φανερή με βάση την εργασία του Ajtai [13]. Το μεγάλο πλεονέκτημα που ανέδειξε η συγκεκριμένη εργασία σχετίζεται με την συζήτηση που είχαμε στην ενότητα 6.3 για την χρήση NP-Hard προβλημάτων στην κρυπτογραφία. Πιο συγκεκριμένα, όπως είδαμε, η ασφάλεια των κρυπτοσυστημάτων προκύπτει από την αναγωγή σε αυτήν της ασφάλειας δύσκολων υπολογιστικών προβλημάτων. Δηλαδή, αν καταφέρει κάποιος να σπάσει το κρυπτοσύστημα, μπορεί να επιλύσει το δύσκολο πρόβλημα. Εδώ υπάρχει όμως ένα κενό: Μέχρι τώρα δεν προσδιορίσαμε, αν η αναγωγή εξαρτάται από το στιγμιότυπο του προβλήματος του οποίου τη δυσκολία ανάγουμε στην παραβίαση του κρυπτοσυστήματος μας. Για παράδειγμα, όπως είδαμε τόσο στο RSA, όσο και στο DLOG πρέπει να υπάρξει προσεκτική επιλογή των παραμέτρων, αλλιώς το αντίστοιχο πρόβλημα μπορεί να είναι μια εξειδικευμένη εύκολα επιλύσιμη περίπτωση. Αν, δηλαδή κάποιος, 'σπάσει το RSA', αυτό δεν σημαίνει κατ' ανάγκη ότι μπορεί να παραγοντοποιήσει εύκολα οποιονδήποτε αριθμό. Αντίθετα, τα πλέγματα έχουν την ιδιότητα ότι αν 'σπάσει το κρυπτοσύστημα' τότε μπορεί να επιλυθεί *οποιοδήποτε στιγμιότυπο* του δύσκολου προβλήματος.

Πιο συγκεκριμένα τώρα τα κρυπτοσυστήματα με πλέγματα βασίζονται σε δύο προβλήματα:

Το πρόβλημα των *Σύντομων Ακέραιων Λύσεων (Short Integer Solutions - SIS)*

Ορισμός 12.10. Δίνονται τα διανύσματα $\mathbf{a_1}, \cdots, \mathbf{a_m} \in Z_q^n$

Να βρεθούν μη τετριμμένες λύσεις $z_1, \cdots, z_m \in \{-1, 0, 1\}$ λύσεις ώστε:

$$\sum_{i=1}^{m} \mathbf{a_i} z_i = 0$$

Είναι φανερό ότι αν δεν υπήρχε ο περιορισμός $z_i \in \{-1, 0, 1\}$, τότε θα μπορούσαμε εύκολα να λύσουμε το παραπάνω πρόβλημα χρησιμοποιώντας τη μέθοδο του Gauss. Επιπλέον, η δυσκολία εύρεσης λύσης στο πρόβλημα αυτό συνεπάγεται την ύπαρξη συναρτήσεων σύνοψης χωρίς συγκρούσεις.

Πράγματι θεωρούμε συγκεκριμένο $A = (\mathbf{a_1}, \cdots, \mathbf{a_m})$ και ορίζουμε μία συνάρτηση σύνοψης ως εξής:

$$\mathcal{H}_A(z_1||z_2||\cdots||z_m) = \sum_{i=1}^{m} \mathbf{a_i} z_i$$

Υποθέτουμε ότι $m > n \log q$.Έστω ότι μπορούμε να βρούμε μία σύγκρουση στην $\mathcal{H}_A$ δηλ. $x_1, \cdots, x_m$ και $y_1, \cdots, y_m$ ώστε;

$$\mathcal{H}_A(x_1||x_2||\cdots||x_m) = \mathcal{H}_A(y_1||y_2||\cdots||y_m)$$

Εύκολα βλέπει κάνεις ότι:

$$\sum_{i=1}^{m} \mathbf{a_i}(x_i - y_i) = 0$$

που σημαίνει ότι βρέθηκε μία λύση στο SIS.

Το πρόβλημα της *Μάθησης Με Λάθη (Learning with Errors - LWE)*

Ορισμός 12.11. Δίνονται τα διανύσματα $\mathbf{a_1}, \cdots, \mathbf{a_m} \in Z_q^n$ και διανύσματα $\mathbf{b_1}, \cdots, \mathbf{b_m} \in Z_q^n$ ώστε για κάθε ένα από αυτά να ισχύει:

$$\mathbf{a_i} =< \mathbf{s}, \mathbf{a_i} > + \mathbf{e_i}$$

Να βρεθεί το διάνυσμα $\mathbf{s} \in Z_q^n$.

Φυσικά χωρίς το $\mathbf{e_i}$ που ονομάζεται θόρυβος, πάλι θα μπορούσε να υπάρξει λύση χρησιμοποιώντας τη μέθοδο του Gauss.

Η έκδοση απόφασης του παραπάνω προβλήματος, θέλει την αναγνώριση ζευγών διανυσμάτων $(\mathbf{a_i}, \mathbf{b_i})$ που δημιουργήθηκαν με την παραπάνω μέθοδο από ζεύγη που δημιουργήθηκαν τυχαία.

Είναι αξιοσημείωτο πως η παράμετρος ασφαλείας στην κρυπτογραφία με δικτυωτά, είναι μόνο το n, δηλ. η διάσταση του δικτυωτού.

12.5 Πλήρως Ομομορφική Κρυπτογραφία

12.5.1 Εισαγωγή

Στην ενότητα 9.5 αναφερθήκαμε στην δυνατότητα αρκετών κρυπτοσυστημάτων να επιτρέπουν ένα (περιορισμένο) σύνολο λειτουργιών πάνω σε κρυπτοκείμενα

χωρίς βέβαια πρόσβαση στο κλειδί αποκρυπτογράφησης. Κάτι τέτοιο έγινε αντιληπτό από την αρχή των κρυπτοσυστημάτων δημοσίου κλειδιού [14] ίσως με την παρατήρηση ότι το RSA είναι πολλαπλασιαστικά ομομορφικό. Βέβαια το συγκεκριμένο είδος ομομορφικής κρυπτογραφίας είχε αρκετούς περιορισμούς, καθώς όπως είδαμε τα διάφορα 'παραδοσιακά' κρυπτοσυστήματα επιτρέπουν ένα μόνο είδος επεξεργασίας πάνω στα κρυπτογραφημένα δεδομένα (πρόσθεση ή πολλαπλασιασμός).

Η δυνατότητα πραγματοποίησης *οποιουδήποτε υπολογισμού* πάνω σε κρυπτογραφημένα δεδομένα είναι φυσικό ότι έχει πληθώρα εφαρμογών. Από το [14] κιόλας οι Rivest, Adleman και Dertouzos πρότειναν την υλοποίηση αναζήτησης πάνω σε κρυπτογραφημένα δεδομένα τα οποίο είναι αποθηκευμένα σε έναν (ενδεχομένως) κακόβουλο εξυπηρετητή. Ο χρήστης κωδικοποιεί το ερώτημα αναζήτησης με τέτοιο τρόπο ώστε όταν ο εξυπηρετητής το αποτιμήσει, θα λάβει ως απάντηση ένα κρυπτογραφημένο αποτέλεσμα.

Τέτοιες εφαρμογές έχουν τεράστια σημασία στη σύγχρονη εποχή με την δυνατότητα υπολογισμού στο νέφος (*cloud computing*), όπου διάφοροι πάροχοι υπηρεσιών έχουν συγκεντρώσει τεράστια υπολογιστική ισχύ (επεξεργαστική ή αποθηκευτική), την οποία νοικιάζουν στους διάφορους χρήστες. Τυπικά τα διάφορα δεδομένα διατηρούνται κρυπτογραφημένα, αποκρυπτογραφούνται όμως όταν οι χρήστες θέλουν να τα επεξεργαστούν. Αυτό θέτει τεράστια προβλήματα ιδιωτικότητας, καθώς ο πάροχος μπορεί να είναι κακόβουλος ή να χρειαστεί να παρέχει τα διάφορα δεδομένα σε κυβερνητικούς φορείς. Αυτό που θα θέλαμε ιδανικά, θα ήταν να εκμεταλλευτούμε την υπολογιστική ισχύ του νέφους, χωρίς όμως να θυσιάσουμε την ιδιωτικότητα των δεδομένων μας. Με άλλα λόγια, θέλουμε να επιτρέψουμε την πλήρη επεξεργασία των δεδομένων, δηλαδή να πραγματοποιήσουμε οποιεσδήποτε λειτουργίες σε αυτά, χωρίς όμως να δώσουμε πρόσβαση σε αυτά, ώστε για παράδειγμα, να μπορούμε να ελέγξουμε αν ένα email είναι ανεπιθύμητο (spam), χωρίς όμως να εξετάσουμε τα περιεχόμενα του. Κάτι τέτοιο αν και φαίνεται οξύμωρο, είναι θεωρητικά δυνατό με την *Πλήρως Ομομορφική Κρυπτογραφία - Fully Homomorphic Encryption (FHE)*. Το πρώτο τέτοιο κρυπτοσύστημα προτάθηκε από τον Craig Gentry στα [15, 16].

Άλλες εφαρμογές της πλήρως ομομορφικής κρυπτογραφίας, σε θεωρητικό επίπεδο, αφορούν όλα τα είδη ασφαλούς υπολογισμού μεταξύ δύο ή περισσότερων συμμετεχόντων. Για παράδειγμα, στην ενότητα 9.4.5 είδαμε πως για να τον Ασφαλή Υπολογισμό Συνάρτησης απαιτείται πολυπλοκότητα επικοινωνίας ανάλογη της πολυπλοκότητας του κυκλώματος που περιγράφει τη συνάρτηση. Χρησιμοποιώντας FHE, μπορούμε να έχουμε το ίδιο αποτέλεσμα με ανταλλαγή μόνο των εισόδων και των εξόδων. Γενικότερα φαίνεται ότι η FHE μπορεί να καταστήσει την πολυπλοκότητα επικοινωνίας ανεξάρτητη του είδους του ασφαλούς υπολογισμού.

12.5.2 Ορισμός και ιδιότητες

Ένα πλήρως ομομορφικό κρυπτοσύστημα, τυπικά, είναι μια τετράδα αποδοτικών αλγορίθμων (`KeyGen`, `Encrypt`, `Decrypt`, `Eval`), όπου οι τρεις πρώτοι διαδραματίζουν το ρόλο που περιγράψαμε στο 1.3. Ο αλγόριθμος `Eval` συσχετίζεται με μια οικογένεια συναρτήσεων $\mathcal{F}$, όπου για κάθε συνάρτηση $f \in \mathcal{F}$ με είσοδο ένα σύνολο κρυπτοκειμένων $\{c_i\}_{i=1}^n$ υπολογίζει ένα κρυπτοκείμενο c που αποκρυπτογραφείται στο $f(m_1, \cdots, m_n)$. Δηλαδή:

$$\texttt{Eval}(f, \texttt{Encrypt}_K(m_1), \cdots, \texttt{Encrypt}_K(m_n)) = \texttt{Encrypt}_K(f(m_1, \cdots, m_n))$$

Το κρυπτοκείμενο c δεν πρέπει να διαφέρει από τα κανονικά κρυπτοκείμενα, ούτε στο μέγεθος ούτε στο χρόνο αποκρυπτογράφησης και είναι αντελώς ανεξάρτητο από την f.

Η συνάρτηση f αναπαρίσταται συνήθως ως ένα λογικό κύκλωμα. Ένας λόγος για αυτό είναι για να μην διαρρεύσουν 'λεπτομέρειες' για την εκτέλεση του υπολογισμού. Είναι απαραίτητο λοιπόν η αναπαράσταση της f να μην αντανακλά την πορεία του υπολογισμού, πχ. ότι παραλείπονται τιμές σε κάποιες περιπτώσεις. Επίσης με την αναπαράσταση με ένα λογικό κύκλωμα σταθερού μεγέθους για την υλοποίηση της f, αρκεί να υλοποιηθεί η `Eval` για τα συστατικά στοιχεία του κυκλώματος δηλαδή τις λογικές πύλες.

Είναι εύκολο να δει κανείς από τον παραπάνω ορισμό ότι ένα ομομορφικό κρυπτοσύστημα είναι εξ' ορισμού εύπλαστο (malleable). Άρα δεν διαθέτει ασφάλεια IND-CCA2. Σε ότι αφορά όμως της απαιτήσεις σημασιολογικής ασφάλειας (IND-CPA), ένα ομομορφικό κρυπτοσύστημα δεν διαφέρει καθόλου από ένα παραδοσιακό.

Για να κατανοήσουμε καλύτερα τον τρόπο λειτουργίας των διαφόρων σχημάτων ομομορφικής κρυπτογραφίας, πρέπει να αλλάξουμε λίγο την οπτική που έχουμε για τα διάφορα κρυπτοσυστήματα, χρησιμοποιώντας έννοιες από τη Θεωρία Πληροφορίας.

Συγκεκριμένα, μπορούμε να φανταστούμε την διαδικασία κρυπτογράφησης ως την προσθήκη θορύβου στο αρχικό μήνυμα. Ο θόρυβος αυτός είναι 'ελεγχόμενος' και μπορεί να διορθωθεί με την αποκρυπτογράφηση (από τον κάτοχο του αντίστοιχου κλειδιού), με τρόπο ανάλογο με έναν κώδικα διόρθωσης λαθών. Αν βέβαια ο θόρυβος μεγαλώσει, τότε κανένας δεν μπορεί να προχωρήσει στην αποκρυπτογράφηση. Το πρόβλημα στην ομομορφική κρυπτογραφία, είναι ότι η αποτίμηση διαφόρων συναρτήσεων πάνω στα κρυπτοκείμενα αυξάνει τον θόρυβο. Έτσι υπάρχει ένα όριο στο πλήθος των συναρτήσεων που μπορούμε να αποτιμήσουμε ή στο βάθος του κυκλώματος που τις αναπαριστά, καθιστώντας μερικώς (somewhat) και όχι πλήρως ομομορφικό ένα τέτοιο κρυπτοσύστημα.

Η λύση που έδωσε ο Gentry στο πρόβλημα αυτό στο πρόβλημα αυτό βασίζεται στην παρατήρηση πως και η ίδια η αποκρυπτογράφηση είναι μια υπολογίσιμη συνάρτηση. Κατά συνέπεια μπορούμε να την τροφοδοτήσουμε στην `Eval`, θέτοντας $\texttt{Decrypt} = f$ στην σχέση 12.5.2. Η αποτίμηση της `Decrypt` θα μειώσει το θόρυβο, οπότε θα μπορέσουμε στη συνέχεια να αποτιμήσουμε και άλλες συναρτήσεις. Όταν ο θόρυβος κοντεύει να γίνει ξανά μη ανεκτός, τότε αποτιμούμε ξανά την συνάρτηση αποκρυπτογράφησης. Η διαδικασία επαναλαμβάνεται μέχρι να εφαρμόσουμε όλες τις συναρτήσεις που θέλουμε στα κρυπτοκείμενα. Φυσικά η παραπάνω περιγραφή προϋποθέτει κατάλληλο χειρισμό των κλειδιών αποκρυπτογράφησης και ενσωμάτωσης τους στα κρυπτοκείμενα, κάτι που μπορεί να γίνει λόγω της σημασιολογικής ασφάλειας που παρέχει το κρυπτοσύστημα.

Τα παραπάνω ισχύουν βέβαια με την προϋπόθεση ότι η `Eval` μπορεί να χειριστεί σε επίπεδο θορύβου την αποκρυπτογράφηση μαζί με μία επιπλέον συνάρτηση (ώστε να προχωρήσει λίγο η επεξεργασία). Ένα κρυπτοσύστημα με αυτή την ιδιότητα ονομάζεται *εκκινήσιμο - bootstrappable*. Η μεγάλη επιτυχία του [16] είναι ακριβώς ότι έδειξε πως από οποιοδήποτε εκκινήσιμο κρυπτοσύστημα μπορεί να κατασκευαστεί ένα πλήρως ομομορφικό.

12.5.3 Γενικευμένη Κατασκευή

Ας περιγράψουμε με περισσότερη σαφήνεια το πώς μπορούμε να κατασκευάσουμε ένα πλήρως ομομορφικό κρυπτοσύστημα $\mathcal{CS}^*$ από ένα μερικώς ομομορφικό $\mathcal{CS}$, το οποίο ορίζεται από τους αλγορίθμους $(\texttt{KeyGen}, \texttt{Encrypt}, \texttt{Decrypt}, \texttt{Eval})$ και υποστηρίζει την αποτίμηση συναρτήσεων $f \in \mathcal{F}_{\mathcal{CS}}$.

Υποθέτουμε ότι το $\mathcal{CS}$ είναι εκκινήσιμο, δηλαδή μπορεί να αποτιμήσει την `Decrypt` μαζί με μία f. Για να αποτιμήσουμε μία f^* στο $\mathcal{CS}^*$ την αναπαριστούμε ως κύκλωμα με την βοήθεια των f που μπορούν να αποτιμηθούν από το cs και το οργανώνουμε σε l επίπεδα. Το δημόσιο κλειδί του $\mathcal{CS}^*$ είναι μια ακολουθία δημοσίων κλειδιών $\{pk_i\}_{i=1}^{l+1}$ μαζί με τις κρυπτογραφήσεις των αντίστοιχων ιδιωτικών κλειδιών $\{\bar{sk}_i = \texttt{Encrypt}(pk_{i+1}, sk_i)\}_{i=1}^{l}$ με το κλειδί του επόμενου επιπέδου. Λόγω της σημασιολογικής ασφάλειας, οι κρυπτογραφήσεις των ιδιωτικών κλειδιών δεν διαρρέουν καμία πληροφορία γι' αυτά.

Αρχικά κρυπτογραφούμε το μήνυμα m ως $c_1 \leftarrow \texttt{Encrypt}(pk_1, m)$. Σε κάθε επίπεδο υλοποιούμε το κύκλωμα d_f που υλοποιεί την συνάρτηση αποκρυπτογράφησης μαζί με τις επιπλέον πύλες.

Στη συνέχεια εκτελούμε την διαδικασία $Recrypt(pk_2, \texttt{Decrypt}, \bar{sk}_2, c_1)$:

- Παράγουμε το κρυπτοκείμενο $\bar{c_1} = \texttt{Encrypt}(pk_2, c_1)$. Βλέπουμε ότι το μήνυμα στην πραγματικότητα κρυπτογραφείται 2 φορές με διαφορετικά δημόσια κλειδιά.

- Επιστρέφουμε το $c \leftarrow \texttt{Eval}(pk_2, d_f, \bar{sk}_1, \bar{c}_1)$. Εδώ αφαιρούμε το *εσωτερικό επίπεδο* κρυπτογράφησης. Το c είναι μια ανακρυπτογράφηση του αρχικού μηνύματος με το κλειδί του επιπέδου 2 όμως.

Η αλλαγή κρυπτογράφησης μειώνει το θόρυβο του κρυπτοκειμένου αν η `Decrypt` αφαιρεί περισσότερο θόρυβο από όσο προσθέτει η `Eval`. Η διαδικασία επαναλαμβάνεται αναδρομικά, μέχρι να υλοποιηθεί η f^*. Το τελικό αποτέλεσμα του κυκλώματος θα είναι το αποτέλεσμα της f^* κρυπτογραφημένο με το δημόσιο κλειδί pk_{l+1} το οποίο μπορεί να αποκρυπτογραφηθεί με το αντίστοιχο ιδιωτικό sk_{l+1}. Φυσικά αν το d_f θέλει περισσότερα από ένα ορίσματα, αυτά πρέπει να δίνονται στην $Recrypt$ κρυπτογραφημένα κάθε φορά με τα αντίστοιχα δημόσια κλειδιά.

12.5.4 Μία υλοποίηση

Η παραπάνω διαδικασία κατασκευής πλήρως ομομορφικών κρυπτοσυστημάτων έχει χρησιμοποιηθεί σε κατασκευές όπως αυτή του Gentry [17] που αξιοποιεί πλέγματα και των Smart και Vercauteren [18] που χρησιμοποιούν πολυώνυμα.

Στη συνέχεια θα περιγράψουμε μία διαφορετική και πιο προσιτή προσέγγιση, η οποία προτάθηκε από τους Van Dijk, Gentry κλπ. στο [19]. Χρησιμοποιεί αποκλειστικά ακέραιους αριθμούς και πράξεις modulo.

Το αρχικό κρυπτοσύστημα $\mathcal{CS}$ έχει παράμετρο ασφαλείας λ και αποτελείται από τους παρακάτω αλγορίθμους:

- $\texttt{KeyGen}(1^\lambda) = p$, ένας τυχαίος μονός ακέραιος λ^2 bit.
- $\texttt{Encrypt}(p, m) = m' + pq$ όπου m' ένας τυχαίος αριθμός λ bit με $m' = m \bmod 2$ και q ένας τυχαίος αριθμός λ^5 bit.
- $\texttt{Decrypt}(p, c) = (c \bmod p) \bmod 2$ ή ισοδύναμα
- $\texttt{Decrypt}(p, c) = (c \bmod 2) \oplus (c \div p \bmod 2)$ (με $\div$ συμβολίζουμε την ακέραια διαίρεση) ή ακόμα πιο εύκολα:
- $\texttt{Decrypt}(p, c) = LSB(c) \oplus LSB(c \div p)$ όπου φυσικά η LSB δίνει το λιγότερο σημαντικό bit της παραμέτρου της.

Παρατηρούμε ότι όλα τα κρυπτοκείμενα απέχουν m' από τα πολλαπλάσια του p. Η απόσταση αυτή είναι ο θόρυβος του κρυπτοκειμένου. Παρά την ύπαρξη του, η αποκρυπτογράφηση δουλεύει σωστά καθώς διατηρείται η ισοτιμία με το αρχικό μήνυμα. Αποδεικνύεται πώς το $\mathcal{CS}$ είναι σημασιολογικά ασφαλές αν το πρόβλημα του *Προσεγγιστικού Μέγιστου Κοινού Διαιρέτη* είναι δύσκολο:

Ορισμός 12.12. Προσεγγιστικός ΜΚΔ
Δίνεται: Μία λίστα από ακέραιους της μορφής $x_i = q_i \cdot p_i + r_i$
Ζητείται: Το p

Το κρυπτοσύστημα είναι ομομορφικό, καθώς μπορεί να χειριστεί τις συναρτήσεις `Add`, `Mult`, οι οποίες μπορούν να εκφραστούν ως κύκλωμα με πύλες `XOR` και `AND`, τροφοδοτώντας έτσι την `Eval` του ορισμού 12.5.2:

- $\texttt{Add}(c_1, c_2) = c_1 + c_2 = (m'_1 + m'_2) + p(q_1 + q_2)$
- $\texttt{Mult}(c_1, c_2) = c_1 \cdot c_2 = m1 \cdot m_2 + pq'$, όπου q' ένας ακέραιος.

Όπως φαίνεται από τις παραπάνω σχέσεις, οι ομομορφικές λειτουργίες αυξάνουν το θόρυβο των κρυπτοκειμένων. Η αποκρυπτογράφηση δουλεύει σωστά όσο ο θόρυβος του αποτελέσματος είναι μικρότερος από το κλειδί.

Για να κατασκευάσουμε το $\mathcal{CS}^*$ από το $\mathcal{CS}$ πρέπει να δούμε αν είναι εκκινήσιμο, δηλαδή αν μπορεί να αποτιμήσει την συνάρτηση αποκρυπτογράφησης. Για ευκολία θα εξετάσουμε την δεύτερη ισοδύναμη μορφή της. Είναι σαφές, λοιπόν, ότι μπορεί να αποτιμήσει τις λειτουργίες LSB, αφού αρκεί να εξάγει το τελευταίο bit της εισόδου. Άρα λοιπόν μένει το αν μπορεί να αποτιμηθεί η πράξη $c \div p$ ή ισοδύναμα να μπορεί να υπολογιστεί το $1/p$. Αυτό *δεν* μπορεί να γίνει άμεσα, αλλά έμμεσα ακολουθώντας τα δύο παρακάτω βήματα:

Αρχικά μετατρέπουμε το $\mathcal{CS}$ σε κρυπτοσύστημα δημοσίου κλειδιού. Το ιδιωτικό κλειδί θα είναι το p. Το δημόσιο κλειδί θα είναι μία λίστα με μήκος πολυωνυμικό στην παράμετρο λ από ακέραιους που κρυπτογραφούν το 0. Είναι φανερό λοιπόν ότι αν δεν αυξήσουν υπερβολικά τον θόρυβο, δεν θα παίξουν κανένα ρόλο στη διαδικασία της αποκρυπτογράφησης.

Στη συνέχεια θα χρησιμοποιηθεί μία κρυπτογραφική τεχνική η οποία σχεδιάστηκε για να επιτρέψει σε συσκευές με μικρές υπολογιστικές δυνατότητες να εκτελούν πολύπλοκους υπολογισμούς [20]. Θα συμπεριληφθεί έτσι μία υπόδειξη (hint) για το ιδιωτικό κλειδί μέσα στο δημόσιο τέτοια ώστε να 'ελαφρυνθεί' η αποτίμηση της αποκρυπτογράφησης, χωρίς όμως να μπορεί να υπάρξει παραβίαση της ασφάλειας. Η υπόδειξη θα είναι ένα σύνολο ακέραιων που κάποιο υποσύνολό του να αθροίζει στο $1/p$.

Το κρυπτοσύστημα λοιπόν θα έχει ως εξής:

- $\texttt{KeyGen}(1^\lambda)$
  - Παράγουμε το ιδιωτικό κλειδί p και το δημόσιο κλειδί pk για το $\mathcal{CS}$ όπως περιγράψαμε νωρίτερα.
  - Δημιουργούμε ένα διάνυσμα $\mathbf{y}$ β αριθμών $y_i \in \{0, 1\}$ τέτοιο ώστε να περιέχει ένα αραιό υποσύνολο S πλήθους α ώστε $\sum_{i \in S} y_i = \frac{1}{p} \bmod 2$.

  - Το ιδιωτικό κλειδί είναι το $sk^* = (p, S)$ ενώ το δημόσιο είναι το $pk^* = (pk, \mathbf{y})$

- $\texttt{Encrypt}(pk*, m)$
  - Αρχικά εφαρμόζουμε την $\texttt{Encrypt}(pk, m)$ λαμβάνοντας ένα αρχικό κρυπτοκείμενο c.
  - Δημιουργούμε το β διάνυσμα $\mathbf{z}$ με $z_i = cy_i \bmod 2$
  - Το κρυπτοκείμενο είναι το $c^* = (c, \mathbf{z})$
- $\texttt{Decrypt}(sk^*, c^*) = (LSB(c)) \oplus LSB(\sum_i z_i)$

Αποδεικνύεται ότι και με αυτήν αλλαγή το κρυπτοσύστημα γίνεται εκκινήσιμο. Για να είναι σημασιολογικά ασφαλές πρέπει επιπλέον να είναι δύσκολο και το πρόβλημα Low-weight subset sum (SSSP): Για ένα σύνολο β αριθμών και ένα δεδομένο s να βρεθεί ένα αραιό (με α στοιχεία υποσύνολο που να έχει άθροισμα s. Για περισσότερες λεπτομέρειες για την παραπάνω διαδικασία, ο ενδιαφερόμενος αναγνώστης μπορεί να ανατρέξει στα [16, 15, 19, 20].

12.6 Ασκήσεις

1. Να εξηγήσετε γιατί κατά τη διάρκεια της δημιουργίας υπογραφών που περιγράψαμε στην ενότητα 12.2.4 πρέπει να διατηρείται κρυφή η τυχαία τιμή k και να αλλάζει κάθε φορά που δημιουργείται νέα υπογραφή.

2. Να αποδείξει ότι το BDDH συνεπάγεται το DDH στην ομάδα προορισμού της ζεύξης. Ισχύει το ίδιο αν η ζεύξη ειναι 'τριγραμμική';

3. Διαθέτει το IBE σχήμα των Boneh, Franklin την ιδιότητα IND-CCA; Επιχειρηματολογήστε.

12.7 Ηλεκτρονικό Υλικό

- Προσομοιωτής κβαντικής ανταλλαγής κλειδιού - QKDSimulator
- Ένας ηλεκτρονικός οδηγός για ελλειπτικές καμπύλες από τον Andrea Corbellini. Περιλαμβάνει οπτικοποίηση για:
  - Πρόσθεση σε ελλειπτικές καμπύλες στους πραγματικούς και σε πεπερασμένα σώματα

  - Πολλαπλασιασμός σε ελλειπτικές καμπύλες στους πραγματικούς και σε πεπερασμένα σώματα
  - Κώδικας Python για τις λειτουργίες που αναφέραμε στην ενότητα 12.2

- 3ο Χειμερινό Σχολείο για Ελλειπτικές καμπύλες και Pairings από το Πανεπιστήμιο Bar-Ilan. Διαφάνειες και Βιντεοσκοπημένες Διαλέξεις.

- 2ο Χειμερινό Σχολείο για Κρυπτογραφία με Πλέγματα. Διαφάνειες και Βιντεοσκοπημένες Διαλέξεις.

Βιβλιογραφία

[1] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM journal on computing*, 26(5):1484–1509, 1997.

[2] Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219. ACM, 1996.

[3] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography (Chapman & Hall/Crc Cryptography and Network Security Series)*. Chapman & Hall/CRC, 2007. ISBN 1584885513.

[4] Ian F. Blake, Gadiel Seroussi, and Nigel P. Smart. *Elliptic curves in cryptography*. Cambridge University Press, Cambridge, 2000. ISBN 0–521–65374–6.

[5] Joseph H. Silverman. *The Arithmetic of Elliptic Curves (Graduate Texts in Mathematics)*. Springer, 1985. ISBN 0387962034.

[6] Rene Schoof. Counting points on elliptic curves over finite fields, 1995.

[7] Antoine Joux. A one round protocol for tripartite diffie-hellman. In *Algorithmic Number Theory, 4th International Symposium, ANTS-IV, Leiden, The Netherlands, July 2-7, 2000, Proceedings*, pages 385–394, 2000. doi: 10.1007/10722028_23.

[8] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the weil pairing. *Journal of Cryptology*, 17(4):297–319, 2004. ISSN 0933-2790. doi: 10.1007/s00145-004-0314-9.

[9] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Proceedings of CRYPTO 84 on Advances in Cryptology*, pages 47–53, New York, NY, USA, 1985. Springer-Verlag New York, Inc. ISBN 0-387-15658-5.

[10] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. In *Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '01, pages 213–229, London, UK, UK, 2001. Springer-Verlag. ISBN 3-540-42456-3.

[11] Kenneth G Paterson and Geraint Price. A comparison between traditional public key infrastructures and identity-based cryptography. *Information Security Technical Report*, 8(3):57–72, 2003.

[12] Arjen Klaas Lenstra, Hendrik Willem Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4): 515–534, 1982.

[13] Miklós Ajtai. Generating hard instances of lattice problems. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 99–108. ACM, 1996.

[14] Ronald L Rivest, Len Adleman, and Michael L Dertouzos. On data banks and privacy homomorphisms. 1978.

[15] Craig Gentry. Computing arbitrary functions of encrypted data. *Communications of the ACM*, 53(3):97–105, 2010.

[16] Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009. crypto.stanford.edu/craig.

[17] Craig Gentry et al. Fully homomorphic encryption using ideal lattices. 2009.

[18] Nigel P Smart and Frederik Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In *Public Key Cryptography–PKC 2010*, pages 420–443. Springer, 2010.

[19] Marten Van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In *Advances in cryptology–EUROCRYPT 2010*, pages 24–43. Springer, 2010.

[20] Tsutomu Matsumoto, Koki Kato, and Hideki Imai. Speeding up secret computations with insecure auxiliary devices. In Shafi Goldwasser, editor, *Advances in Cryptology — CRYPTO' 88*, volume 403 of *Lecture Notes in*

Computer Science, pages 497–506. Springer New York, 1990. ISBN 978-0-387-97196-4. doi: 10.1007/0-387-34799-2_35.

Ευρετήριο