Πανεπιστημιο Πελοποννησου
Τμημα Πληροφορικης και Τηλεπικοινωνιων

Πρόγραμμα Μεταπτυχιακών Σπουδών στην
Επιστήμη και Τεχνολογία Υπολογιστών

Κρυπτοσυστήματα Βασισμένα σε Κώδικες: Κατασκευές και Επιθέσεις

Διπλωματική Εργασία

Συγγραφέας:
Παναγιώτα Σμυρλή

Επιβλέπων:
Νικόλαος Κολοκοτρώνης
Επίκ. Καθηγητής

Μάιος 2015

Ευχαριστίες

Πρώτα απ' όλα θα ήθελα να εκφράσω τη βαθιά μου ευγνωμοσύνη προς στον επιβλέποντά μου, Επίκουρο Καθηγητή κ. Κολοκοτρώνη Νικόλαο. Χωρίς την καθοριστική του συμβολή, η παρούσα διπλωματική εργασία δεν θα είχε επιτευχθεί. Θα ήθελα να τον ευχαριστήσω, για τις συμβουλές και κατευθύνσεις που μου προσέφερε, καθ' όλη την διάρκεια της συνεργασίας μας, αλλά και για το χρόνο του, καθώς ήταν πάντοτε διαθέσιμος όποτε τον χρειάστηκα. Μου εμφύσησε το ενδιαφέρον για το εν λόγω αντικείμενο και με ενέπνευσε για την συνέχιση των σπουδών μου. Είμαι ιδιαίτερα ευτυχής, που η άριστη αυτή συνεργασία θα έχει και συνέχεια.

Θα ήθελα επίσης, να ευχαριστήσω θερμά για τη συνεργασία τα υπόλοιπα μέλη της τριμελούς εξεταστικής επιτροπής, Αναπληρωτή Καθηγητή κ. Βασιλάκη Κωνσταντίνο και Αναπληρωτή Καθηγητή κ. Κούτρα Κωνσταντίνο.

Τέλος, θα ήθελα να ευχαριστήσω τον πιο σημαντικό για μένα άνθρωπο και σύντροφό μου, Γιώργο, που με στήριξε και με στηρίζει σε κάθε αποφασιστικό βήμα της ζωής μου.

Πρόλογος

Η παρούσα διπλωματική εργασία πραγματεύεται την μελέτη *κρυπτοσυστημάτων δημοσίου κλειδιού ή ασύμμετρων κρυπτοσυστημάτων* (public-key cryptosystems). Τα εν λόγω κρυπτοσυστήματα θεωρούν ότι κάθε χρήστης έχει στην κατοχή του ένα ζεύγος κλειδιών, το *ιδιωτικό* και το *δημόσιο*. Το ιδιωτικό κλειδί κρατείται πάντοτε μυστικό, ενώ το δημόσιο διατίθεται ελεύθερα σε όλους τους εν δυνάμει χρήστες ενός ανοικτού και ανασφαλούς δικτύου. Μια από τις βασικές απαιτήσεις ασφαλείας που πρέπει να ικανοποιούν τα εν λόγω κρυπτοσυστήματα, είναι ότι η ανάκτηση του ιδιωτικού κλειδιού από το δημόσιο πρέπει να είναι υπολογιστικά ανέφικτη. Για τον λόγο αυτό, οι κατασκευές των ασύμμετρων κρυπτοσυστημάτων βασίζονται σε δύσκολα μαθηματικά προβλήματα, προερχόμενα, ενδεικτικά, από τον χώρο της θεωρίας αριθμών, της άλγεβρας και της συνδυαστικής, τα οποία ανήκουν στην κλάση υπολογιστικής πολυπλοκότητας NP. Πρόκειται δηλαδή για προβλήματα που δεν επιδέχονται επίλυση σε πολυωνυμικό χρόνο. Χαρακτηριστικά παραδείγματα μεταξύ άλλων, αποτελούν το πρόβλημα της *παραγοντοποίησης μεγάλων ακεραίων* (integer-factoring) [46], *διακριτού λογαρίθμου* (discrete logarithm) [46], *αποκωδικοποίησης τυχαίων γραμμικών κωδίκων* (computational syndrome decoding problem ή CSD) και *κρυφής υποομάδας* (hidden subgroup problem) [38]. Θα εστιάσουμε την μελέτη μας στο CSD πρόβλημα, λόγω της σημαντικότητάς του στην ασφάλεια ασύμμετρων κρυπταλγορίθμων βασιζόμενων σε κώδικες διόρθωσης σφαλμάτων, όπως τα κρυπτοσυστήματα McEliece και Niederreiter. Αξίζει να αναφέρουμε πως υπάρχουν σοβαρές ενδείξεις ότι παρά την αξιοσημείωτη πρόοδο που έχει καταγραφεί την τελευταία δεκαετία στην περιοχή της κβαντομηχανικής, το πρόβλημα CSD δεν επιδέχεται σημαντικής κβαντικής επιτάχυνσης. Ακολουθεί εκτεταμένη μελέτη αλγορίθμων αποκωδικοποίησης για τυχαίους γραμμικούς κώδικες, εστιάζοντας σε μια ιδιαιτέρως σημαντική κλάση αλγορίθμων, που βασίζονται στην *αποκωδικοποίηση συνόλου πληροφορίας* (information set decoding), αλλά και κάποιες επιπλέον εναλλακτικές μεθόδους. Παρατίθεται εν συνεχεία, πλήρης ασυμπτωτική ανάλυση μεταξύ των εξεταζόμενων μεθόδων, καθιστώντας δυνατή τη σύγκριση της εκθετικής τους συμπεριφοράς και τέλος καταγράφονται ανοιχτά προβλήματα και μελλοντικές ερευνητικές κατευθύνσεις.

Παναγιωτα Σμυρλη
Τρίπολη
Μάιος 2015

Abstract

This thesis offers an insight into public-key cryptosystems. It is assumed that every user of the cryptosystems in question possesses a pair of keys, referred to as the private key and the public key. The private key is always kept secret while the public key is known to every user of an open insecure network. An important security requirement involved in such cryptosystems is that the recovery of the private key from the public key is computationally infeasible. For this reason, the construction of asymmetric cryptosystems is based on difficult mathematical problems, indicatively from areas like number theory, algebra and combinatorics, all of which belong to the complexity class NP. In other words, those mathematical problems cannot be solved within polynomial time. Examples of problems in the above areas are integer-factoring, discrete logarithm, computational syndrome decoding or CSD, and the hidden subgroup problem. Our research will focus on the CSD problem due to its significance in the security of code-based cryptosystems, such as McEliece and Niederreiter. It is worth mentioning that despite considerable improvements which have taken place over the last decade in the field of quantum mechanics, the CSD problem cannot be solved via quantum acceleration techniques. What follows is an extensive research into decoding algorithms of random linear codes, focusing on a particularly important class of algorithms called Information Set Decoding algorithms as well as some additional alternative methods. A holistic asymptotic analysis of the examined methods is performed that allows a detailed comparison of the aforementioned methods with respect to their complexity. A number of open problems and future research directions are also presented.

Panagiota Smyrli
Tripoli
May 2015

Περιεχόμενα

Κατάλογος σχημάτων

Κατάλογος πινάκων

Κατάλογος αλγορίθμων

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή

Τα μοντέρνα κρυπτοσυστήματα χωρίζονται σε δύο κατηγορίες, τα κρυπτοσυστήματα ιδιωτικού κλειδιού (ή συμμετρικά κρυπτοσυστήματα) και τα κρυπτοσυστήματα δημοσίου κλειδιού (ή ασύμμετρα κρυπτοσυστήματα), στα οποία και θα εστιάσουμε την μελέτη μας. Τα ασύμμετρα κρυπτοσυστήματα θεωρούν ότι κάθε χρήστης έχει ένα ζεύγος κλειδιών, το ιδιωτικό και το δημόσιο, όπου το τελευταίο είναι διαθέσιμο ή μπορεί να διαμοιραστεί, σε όλους τους εν δυνάμει χρήστες ενός ανοικτού και ανασφαλούς δικτύου. Λόγω της σχέσης που διέπει το ζεύγος κλειδιών, μια από τις βασικές απαιτήσεις ασφαλείας που πρέπει να ικανοποιούν τα εν λόγω κρυπτοσυστήματα, είναι ότι η ανάκτηση του ιδιωτικού κλειδιού από το δημόσιο πρέπει να είναι υπολογιστικά ανέφικτη.

Προκειμένου να καταστεί αυτό εφικτό, οι κατασκευές ασύμμετρων κρυπτοσυστημάτων βασίζονται σε δύσκολα μαθηματικά προβλήματα, που ανήκουν στην κλάση υπολογιστικής πολυπλοκότητας NP (βλ. Κεφ. 2, Ορισμό 2.29), και συνεπώς δεν υπάρχει αλγόριθμος που να επιλύει το πρόβλημα σε πολυωνυμικό χρόνο (ή/και μνήμη). Δύο γνωστά, μεταξύ άλλων, προβλήματα της κλάσης NP είναι το πρόβλημα της παραγοντοποίησης μεγάλων ακεραίων, καθώς επίσης και το πρόβλημα διακριτού λογαρίθμου, τα οποία όπως αποδείχθηκε το 1997 από τον Shor [33], είναι δυνατόν να επιλυθούν σε πολυωνυμικό χρόνο από έναν κβαντικό υπολογιστή και μάλιστα με μικρή πιθανότητα σφάλματος.

Αν και είναι αξιοσημείωτη η πρόοδος που έχει καταγραφεί την τελευταία δεκαετία στην περιοχή της κβαντομηχανικής, επιτρέποντας την ταχύτατη πραγματοποίηση πράξεων μεταξύ λίγων δεκάδων κβαντικών ψηφίων (qubits), προς το παρόν η κατασκευή ενός ολοκληρωμένου κβαντικού υπολογιστή, αποτελεί (πολύ) μακρινό στόχο. Παρόλα αυτά, η ανάπτυξη πλήθους αλγορίθμων που εκμεταλλεύονται ιδιότητες της κβαντομηχανικής για την επιτάχυνση του χρόνου εκτέλεσης συμβατικών αλγορίθμων, ανέδειξε σημαντικές κρυπτογραφικές προκλήσεις. Θέτοντας ως κύριο στόχο την κατασκευή αποδοτικών κρυπτοσυστημάτων βασιζόμενων σε προβλήματα, ικανά να αντιστέκονται σε τέτοιου είδους αλγορίθμους.

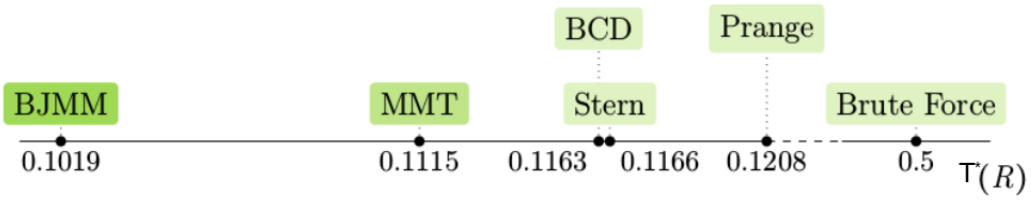
Εκτός από τον πρόσφατα αναδυόμενο τομέα της *κρυπτογραφίας βασισμένης σε πλέγματα* (lattice-based cryptography) [51], οι πιο σοβαρές προτάσεις

προς αυτή την κατεύθυνση, σχετίζονται με την δυσκολία στην αποκωδικοποίηση *τυχαίων γραμμικών κωδίκων* (random linear codes). Ειδικότερα, σύμφωνα με τους Dihn *et al.* [58] για κρυπτοσυστήματα βασισμένα σε κώδικες διόρθωσης σφαλμάτων (βλ. Κεφ. 3), όπως τα κρυπτοσυστήματα McEliece και Niederreiter (βλ. Κεφ. 5), ο σχεδιασμός αποδοτικών κβαντικών αλγορίθμων καθίσταται έως τώρα ανέφικτος. Υπάρχουν ωστόσο, σοβαρές ενδείξεις ότι αποτελεί και μελλοντικά ανέφικτο στόχο, ακόμα και υπό την χρήση ισχυρών τεχνικών, όπως για παράδειγμα η *κβαντική δειγματοληψία Fourier* (quantum Fourier sampling) [39]. Γεγονός που καθιστά την *κρυπτογραφία βασισμένη σε κώδικες* (code-based cryptography) έναν από τους ισχυρότερους ερευνητικούς τομείς στην εποχή της *μετα-κβαντικής κρυπτογραφίας* (post-quantum cryptography) [50].

Μεταξύ των προβλημάτων που ανήκουν στην κλάση υπολογιστικής πολυπλοκότητας NP και δεν επιδέχονται σημαντικής κβαντικής επιτάχυνσης είναι το πρόβλημα αποκωδικοποίησης τυχαίων γραμμικών κωδίκων (CSD) (βλ. Ορισμό 3.10), καθώς και τα συγγενή προβλήματα *εκμάθησης ισοτιμίας με θόρυβο* (Learning parity with noise ή LPN) (βλ. Ενότητα 7.6.1) και *εκμάθησης με σφάλματα* (Learning with errors ή LWE) [57]. Θα επικεντρώσουμε την μελέτη μας στο CSD, το οποίο αποτελεί ένα δύσκολο υπολογιστικά πρόβλημα, που ακόμα και οι πιο αποδοτικοί αλγόριθμοι επίλυσης του, είναι εκθετικού χρόνου. Παρουσιάζει ιδιαίτερο ενδιαφέρον στην θεωρία κωδίκων, λόγω της σχέσης του με την ασφάλεια των ασύμμετρων κρυπταλγορίθμων βασισμένων σε κώδικες διόρθωσης σφαλμάτων και της δυσκολίας αποκωδικοποίησης τυχαίων γραμμικών κωδίκων. Γεγονός που παρακίνησε τον σχεδιασμό ασυμπτωτικά ταχύτερων γενικών αλγορίθμων αποκωδικοποίησης για τυχαίους γραμμικούς κώδικες.

Στα επόμενα κεφάλαια, θα εστιάσουμε την μελέτη μας στην περιγραφή των εν λόγω αλγορίθμων, αναλύοντας την επίδρασή τους στην κρυπτογραφία. Μια ιδιαιτέρως σημαντική κλάση γενικών αλγορίθμων αποκωδικοποίησης, προήλθε το 1962 από τον Prange [1] και ονομάστηκε *απλή αποκωδικοποίηση συνόλου πληροφορίας* (plain information set decoding ή plain ISD). Αλγόριθμοι οι οποίοι λαμβάνουν εκτεταμένου ενδιαφέροντος από πλήθος ερευνητών τα τελευταία χρόνια. Θα αναλύσουμε λεπτομερώς όλες τις βελτιωμένες παραλλαγές της αποκωδικοποίησης συνόλου πληροφορίας, εστιάζοντας κυρίως στις πλέον πρόσφατες. Οι εν λόγω αλγόριθμοι βασίζονται σε μια γενική τεχνική, ικανή να εφαρμοστεί σε πλήθος συνδυαστικών προβλημάτων αναζήτησης, γνωστή ως *τεχνική αναπαράστασης* (representation technique), η οποία εισήχθη το 2010 από τους Howgrave-Graham και Joux [56] (βλ. Ενότητα 6.7).

Τα τελευταία χρόνια, εντός του γενικότερου πλαισίου εισήχθησαν διάφορες παραλλαγές από πολλούς ερευνητές. Σχεδόν τριάντα χρόνια, έπειτα από την αρχική ιδέα του Prange, οι Lee και Brickell [19] και ο Leon [20] εισήγαγαν αλγορίθμους, οι οποίοι βελτιστοποίησαν πολυωνυμικά τον αλγόριθμο του Prange. Λίγο αργότερα ακολούθησε ο Stern, ο οποίος ανέπτυξε τον πρώτο αλγόριθμο που προσέφερε εκθετική βελτίωση στον αλγόριθμο του Prange. Τα επόμενα



Σχήμα 1.1: Σύγκριση των συντελεστών χρονικής πολυπλοκότητας των διαφόρων μεθόδων αποκωδικοποίησης συνόλου πληροφορίας.

είκοσι χρόνια, προτάθηκε πλήθος αλγορίθμων που βελτίωσε σημαντικά την πρακτική απόδοση της αποκωδικοποίησης συνόλου πληροφορίας. Ωστόσο, κανείς εξ' αυτών δεν κατόρθωσε κάποια εκθετική βελτίωση στον αλγόριθμο του Stern. Περισσότερα από είκοσι χρόνια αργότερα στο Crypto'11, οι Bernstein *et al.* [59] παρουσίασαν μια νέα μέθοδο, γνωστή ως *αποκωδικοποίηση σύγκρουσης μπάλας* (ball-collision decoding ή BCD), η οποία κατάφερε πραγματικά να βελτιώσει τον αλγόριθμο του Stern κατά έναν μικρό εκθετικό συντελεστή [60]. Παρότι υπήρξαν δύο προγενέστερες εναλλακτικές μέθοδοι, αντίστοιχης αποδοτικότητας με τον αλγόριθμο του Bernstein, ο αλγόριθμος *αποκωδικοποίησης διαχωρισμού συνδρόμου* (split syndrome decoding) που προτάθηκε το 1991 από τον Dumer [26] και ο αλγόριθμος που προτάθηκε το 2009 από τους Finiasz και Sendrier [55], εντούτοις δεν υπήρξε έως τότε επίσημη απόδειξη που να επιβεβαιώνει την ισοδυναμία τους. Οι May *et al.* [63], απέδειξαν στη συνέχεια την ισοδυναμία αυτή και στο Asiacrypt'11 παρουσίασαν έναν νέο βελτιωμένο αλγόριθμο αποκωδικοποίησης συνόλου πληροφορίας, τον αλγόριθμο MMT-ISD. Πρόκειται για έναν αλγόριθμο, ο οποίος παρέχει μια ιδιαιτέρως σημαντική βελτιστοποίηση έναντι των υπολοίπων μεθόδων. Τέλος, στο Eurocrypt'11 ακολούθησαν οι Becker *et al.* [65] με τον αλγόριθμο BJMM-ISD, παρέχοντας μακράν την καλύτερη εκθετική βελτίωση στον αλγόριθμο του Stern, που έχει καταγραφεί έως σήμερα.

Όπως θα δούμε στην συνέχεια, η αναπαράσταση του ασυμπτωτικού χρόνου εκτέλεσης των γενικών αλγορίθμων αποκωδικοποίησης στην μορφή $2^{T^*(R)n+o(n)}$, όπου $T^*(R)$ εκφράζει τον ασυμπτωτικό *συντελεστή χρονικής πολυπλοκότητας* (time complexity coefficient) (βλ. Κεφ. 3), επιτρέπει την σύγκριση της εκθετικής συμπεριφοράς των αντίστοιχων μεθόδων. Στο Σχ. 1.1 απεικονίζεται η σύγκριση των διαφόρων συντελεστών χρονικής πολυπλοκότητας, για τους πιο σημαντικούς αλγορίθμους αποκωδικοποίησης συνόλου πληροφορίας.

Διάρθρωση της μελέτης

Ακολουθεί μια σύντομη περιγραφή για την οργάνωση και την δομή κάθε κεφαλαίου.

Στο κεφάλαιο 2 εισάγονται βασικοί ορισμοί και έννοιες από το χώρο και της

θεωρητικής πληροφορικής, ιδιαιτέρως της θεωρίας υπολογισμού και της θεωρίας πληροφορίας. Παρέχει στον αναγνώστη τις απαιτούμενες γνώσεις, ώστε να είναι σε θέση να κατανοήσει σε βάθος την περαιτέρω ανάλυση.

Στο κεφάλαιο 3 εισάγονται βασικές έννοιες των γραμμικών τμηματικών κωδίκων (linear block codes), εστιάζοντας στους δυαδικούς γραμμικούς τμηματικούς κώδικες, οι οποίοι χρησιμοποιούν σύμβολα από το αλφάβητο $\mathbb{F}(2)$. Ακολουθεί η περιγραφή της αναπαράστασης κωδίκων και εισάγεται η έννοια του *συνδρόμου* (syndrome), βάσει του οποίου καθορίζεται η ανίχνευση και η διόρθωση σφαλμάτων. Εν συνεχεία, εισάγεται η έννοια της *ελάχιστης απόστασης* (minimum distance) και αποδεικνύονται σχέσεις με την ικανότητα ανίχνευσης και διόρθωσης σφαλμάτων. Κατόπιν, περιγράφεται η *τυπική διάταξη* (standard array) και παρουσιάζεται η εφαρμογή της στην αποκωδικοποίηση γραμμικών τμηματικών κωδίκων. Ακολουθεί ο ορισμός και η περιγραφή του προβλήματος CSD.

Στο κεφάλαιο 4 παρουσιάζονται και αναλύονται οι δυαδικοί ανάγωγοι Goppa κώδικες, οι οποίοι αποτελούν την καλύτερη έως τώρα επιλογή για κρυπτογραφικές εφαρμογές, διασφαλίζοντας τα απαιτούμενα επίπεδα ασφαλείας. Στο Κεφάλαιο 5 περιγράφεται λεπτομερώς το κρυπτοσύστημα δημοσίου κλειδιού του McEliece, το οποίο βασίζεται στην αλγεβρική θεωρία κωδίκων και αποτέλεσε το πρώτο σύστημα που χρησιμοποίησε κώδικες διόρθωσης σφαλμάτων. Στην αρχική του περιγραφή, χρησιμοποιεί Goppa κώδικες και έως σήμερα δεν έχει βρεθεί επίθεση πολυωνυμικού χρόνου, που να σπάει το κρυπτοσύστημα. Κατόπιν, ακολουθεί η περιγραφή του κρυπτοσυστήματος Niederreiter, το οποίο βασίζεται επίσης σε Goppa κώδικες, καθώς και η σχέση ισοδυναμίας μεταξύ των δύο κρυπτοσυστημάτων. Ακολουθεί πλήθος επιθέσεων που προτάθηκαν από πολλούς ερευνητές, θέλοντας να πλήξουν την ασφάλεια των κρυπτοσυστήματων, καθώς επίσης και προτάσεις για βελτιστοποιήσεις σε θέματα ασφαλείας.

Στο κεφάλαιο 6 εισάγεται η βασική ιδέα των αλγορίθμων αποκωδικοποίησης συνόλου πληροφορίας, που χρησιμοποιούνται για την επίλυση του προβλήματος CSD, στο οποίο βασίζεται η ασφάλεια των κρυπτοσυστημάτων McEliece και Niedereiter και λειτουργούν αξιοποιώντας τον πλεονασμό του κώδικα. Ακολουθεί λεπτομερής περιγραφή πλήθους ISD αλγορίθμων που εισήχθησαν από τους Prange [1], Lee-Brickell [19], Leon [20], Stern [22], Finiasz και Sendrier [55], Bernstein *et al.* [59], Johansson και Löndahl [64], May *et al.* [63] και Becker *et al.* [65]. Στο κεφάλαιο 7 εξετάζονται και αναλύονται εναλλακτικές μέθοδοι αποκωδικοποίησης για την επίλυση του CSD προβλήματος, όπως οι αλγόριθμοι του Al Jabri [43], του Hwang [11], των Levitin και Hartmann [17], των Kalouptsidis και Kolokotronis [62], του Dumer [21], των Barg *et al.* [40] και των Blum et al. [44].

Τέλος, στο κεφάλαιο 8, παρατίθεται λεπτομερώς η πλήρης ασυμπτωτική ανάλυση των ISD αλγορίθμων και της εναλλακτικής μεθόδου του Dumer. καθώς επίσης και η ασυμπτωτική συσχέτιση κάποιων εξ' αυτών. Ακολουθεί ασυμπτωτική

συγκριτική μελέτη του συντελεστή χρονικής πολυπλοκότητας, για τις διάφορες εξεταζόμενες μεθόδους, επιλέγοντας τις βέλτιστες τιμές των εμπλεκόμενων παραμέτρων, ώστε να διασφαλίζονται τα απαιτούμενα επίπεδα ασφαλείας.

ΚΕΦΑΛΑΙΟ 2

Βασικοί ορισμοί

Στο κεφάλαιο αυτό, θα εισάγουμε βασικούς ορισμούς και έννοιες από το χώρο της θεωρητικής πληροφορικής, εστιάζοντας κυρίως στον τομέα της θεωρίας υπολογισμού (Ενότητα 2.4), ο οποίος ασχολείται με την κοστολόγηση των πόρων που απαιτούνται για την αλγοριθμική επίλυση ενός προβλήματος, καθώς επίσης και εκείνον που ασχολείται με την ποσοτικοποίηση της πληροφορίας (Ενότητα 2.5).

2.1 Συναρτήσεις

Ορισμός 2.1 (Μονόδρομη συναρτήση [31]). Μια συνάρτηση $f : X \to Y$ με πεδίο ορισμού το X και σύνολο τιμών το Υ, καλείται *μονόδρομη* (one-way function) αν η f(x) είναι εύκολο να υπολογιστεί για όλα τα $x \in X$, αλλά για ουσιαστικά όλα τα στοιχεία $y \in Y$ είναι υπολογιστικά ανέφικτο να βρούμε κάποιο $x \in X$, έτσι ώστε $f(x) = y$.

Ορισμός 2.2 (Μονόδρομη συνάρτηση καταπακτής [31]). Μια συνάρτηση $f : X \to Y$ με πεδίο ορισμού το X και σύνολο τιμών το Υ, καλείται *μονόδρομη συνάρτηση καταπακτής* (trapdoor one-way function) αν ικανοποιεί το ορισμό 2.1 και επιπλέον έχει την ιδιότητα ότι δοθείσας κάποιας επιπρόσθετης πληροφορίας, (που καλείται: καταπακτή) είναι υπολογιστικά εφικτό, να βρούμε ένα $x \in X$, έτσι ώστε $f(x) = y$, δοθέντος του $y \in Y$.

Ορισμός 2.3 (Μονόδρομη κρυπτογραφική συνάρτηση κατακερματισμού [31]). Μια συνάρτηση h καλείται *μονόδρομη κρυπτογραφική συνάρτηση κατακερματισμού* (cryptographic one-way hash function) όταν δέχεται ως είσοδο μια αυθαίρετου μεγέθους ομάδα δεδομένων $x \in X$, και δίνει ως έξοδο μια καθορισμένου μεγέθους συμβολοσειρά (string) $y \in Y$, έτσι ώστε $y = h(x)$. Η έξοδος αποκαλείται συνήθως *σύνοψη* (digest).

Μια ιδανική κρυπτογραφική συνάρτηση κατακερματισμού έχει τις παρακάτω ιδιότητες:

- Είναι μονόδρομη συνάρτηση, δηλαδή βάσει του ορισμού 2.1:
  - Είναι εύκολο να υπολογιστεί η σύνοψη για οποιαδήποτε είσοδο.
  - Δεν είναι εφικτό να βρεθεί η είσοδος από την σύνοψη. Συγκεκριμένα, δοθέντος του $y \in Y$ είναι υπολογιστικά ανέφικτο να βρεθεί κάποιο $x \in X$, έτσι ώστε $h(x) = y$.
- Δεν είναι υπολογιστικά εφικτό να τροποποιηθεί η είσοδος, χωρίς να αλλάξει η τιμή της σύνοψης. Συγκεκριμένα, δοθέντος του $x \in X$, είναι υπολογιστικά ανέφικτο να βρεθεί μια τιμή $x' \in X$, έτσι ώστε $x' \neq x$ και $h(x') = h(x)$.
- Δεν είναι υπολογιστικά εφικτό να βρεθούν δύο διαφορετικές είσοδοι που δίνουν την ίδια σύνοψη. Συγκεκριμένα, είναι υπολογιστικά ανέφικτο να βρεθούν δυο διαφορετικές τιμές $x', x \in X$, έτσι ώστε $h(x') = h(x)$.

Ορισμός 2.4 (Συναρτήση 1-1 [31]). Μια συνάρτηση $f : X \to Y$ ονομάζεται 1-1 αν κάθε στοιχείο του συνόλου Y, είναι εικόνα μέσω της f, το πολύ ενός στοιχείου του .

2.2 Άλγεβρα

Ορισμός 2.5 (Ομάδα [41]). Μια *ομάδα* (group) $(G, *)$, είναι η αλγεβρική δομή η οποία αποτελείται από ένα σύνολο G εφοδιασμένο, με την δυαδική πράξη $*$ και η οποία ικανοποιεί τις ακόλουθες 3 ιδιότητες:

α. Προσεταιριστική: $a * (b * c) = (a * b) * c$, $\forall a, b, c \in G$.

β. Ουδέτερο στοιχείο: $\exists e \in G : a * e = a$ και $e * a = a$, $\forall a \in G$.

γ. Συμμετρικό στοιχείο: $\forall a \in G$, $\exists a' \in G : a * a' = e$ και $a' * a = e$

Επιπλέον, η ομάδα G είναι αβελιανή ή (αντιμεταθετική) αν: $a * b = b * a$, για $\forall a, b \in G$. Στην περίπτωση που $*$ είναι η πράξη:

- της πρόσθεσης: το e συμβολίζεται με 0 και το a' με $-a$ (αντίθετο)
- του πολλαπλασιασμού: το e συμβολίζεται με 1 και το a' με a^{-1} (αντίστροφο)

Ορισμός 2.6 (Ισομορφισμός ομάδων). Αν $(G, *)$ και $(H, \cdot)$ είναι δύο ομάδες και η συνάρτηση $f : G \to H$ είναι ταυτόχρονα 1-1 και επί, ενώ ισχύει επιπλέον ότι: $f(a * b) = f(a) \cdot f(b), \forall a, b \in G$, τότε έχουμε *ισομορφισμό ομάδων* (group isomorphism).

Ορισμός 2.7 (Δακτύλιος [31]). Ένας *δακτύλιος* (ring) $(F, +, \times)$ είναι μια αλγεβρική δομή, η οποία αποτελείται από ένα σύνολο F εφοδιασμένο με 2 δυαδικές πράξεις, $+$ που συμβολίζει την πρόσθεση και $\times$ που συμβολίζει τον πολλαπλασιασμό αντίστοιχα, ώστε να ικανοποιούνται οι ακόλουθες ιδιότητες:

α. $(F, +)$ είναι μια αβελιανή ομάδα με ουδέτερο στοιχείο 0.

β. Η πράξη $\times$ είναι προσεταιριστική. Οπότε $a \times (b \times c) = (a \times b) \times c$, $\forall a, b, c \in R$.

γ. Υπάρχει ένα πολλαπλασιαστικό ουδέτερο στοιχείο που συμβολίζεται με 1, όπου $1 \neq 0$, έτσι ώστε $1 \times a = a \times 1 = a$, $\forall a \in R$.

δ. Η πράξη $\times$ είναι επιμεριστική ως προς την πρόσθεση. Συνεπώς, $a \times (b \times c) = (a \times b) + (a \times c)$ και $(b + c) \times a = (b \times a) + (c \times a)$, $\forall a, b, c \in R$.

Ένας δακτύλιος λέγεται αντιμεταθετικός αν: $a \times b = b \times a$, $\forall a, b \in R$.

Ορισμός 2.8 (Σώματα [31]). Ένα *σώμα* (field) είναι ένας αντιμεταθετικός δακτύλιος, στον οποίο όλα τα μη-μηδενικά στοιχεία έχουν πολλαπλασιαστικούς αντίστροφους. Εφεξής, θα αναφερόμαστε μόνο στο δυαδικό σώμα $GF(2) = \{0, 1\}$ και επιπλέον η δυαδική πράξη (modulo 2) της πρόσθεσης $\oplus$, θα συμβολίζεται για ευκολία με $+$.

Ορισμός 2.9 (Frobenius αυτομορφισμός). Έστω $\mathbb{F}$ ένα σώμα με χαρακτηριστική p (βλ. [14], ενότητα 2.2). Ο Frobenius *αυτομορφισμός* (automorphism) στο $\mathbb{F}$ είναι μια απεικόνιση από το $\mathbb{F} \to \mathbb{F}$, η οποία απεικονίζει το $a \to a^p$, $\forall a \in \mathbb{F}$.

Ορισμός 2.10 (Διανυσματικός χώρος [31]). Ένας *διανυσματικός χώρος* (vector space) V στο σώμα $\mathbb{F}$ είναι μια αβελιανή ομάδα $(V, +)$ εφοδιασμένη με την πολλαπλασιαστική πράξη $\cdot : \mathbb{F} \times V \to V$, έτσι ώστε $\forall a, b \in \mathbb{F}$ και $v, w \in V$, να ικανοποιούνται οι ακόλουθες ιδιότητες:

α. $a \cdot (u + w) = a \cdot u + a \cdot w$

β. $(a + b) \cdot u = a \cdot u + b \cdot u$

γ. $a \cdot (b \cdot u) = (a \cdot b) \cdot u$

δ. $1 \cdot u = u$

Ορισμός 2.11 (Διανυσματικός υποχώρος [31]). Έστω V ένας διανυσματικός χώρος στο σώμα $\mathbb{F}$. Ένας υποχώρος του V είναι μια προσθετική υποομάδα U στον V η οποία είναι κλειστή ως προς την πράξη του βαθμωτού πολλαπλασιασμού, πχ. $av \in U, \forall a \in \mathbb{F}$ και $\forall v \in U$. Επιπλέον, ένας υποχώρος ενός διανυσματικού χώρου, είναι επίσης διανυσματικός χώρος.

Ορισμός 2.12. Έστω $S = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_n\}$ ένα πεπερασμένο υποσύνολο ενός διανυσματικού χώρου V στο σώμα $\mathbb{F}$ [31].

- Ένας γραμμικός συνδυασμός του S είναι μια έκφραση της μορφής $a_1\boldsymbol{v}_1 + a_2\boldsymbol{v}_2 + \cdots + a_n\boldsymbol{v}_n$ όπου $\forall a_i \in \mathbb{F}$.

- Ο χώρος που παράγει το σύνολο S ονομάζεται θήκη, συμβολίζεται με $\langle S \rangle$ και αποτελείται από το σύνολο όλων των γραμμικών συνδυασμών του S. Επιπλέον, είναι ένας υποχώρος του διανυσματικού χώρου V.

- Τα διανύσματα $\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \cdots, \boldsymbol{v}_n \in V$ ονομάζονται γραμμικά ανεξάρτητα, αν κανένα από αυτά δεν μπορεί να γραφεί ως γραμμικός συνδυασμός των υπολοίπων. Δηλαδή, αν και μόνο αν η εξίσωση:

  $$a_1\boldsymbol{v}_1 + a_2\boldsymbol{v}_2 + \cdots + a_n\boldsymbol{v}_n = 0$$

  έχει μοναδική λύση την $a_i = 0, \forall i = 1, \cdots, n$. Ενώ ονομάζονται γραμμικά εξαρτημένα στην αντίθετη περίπτωση.

- Το υποσύνολο S καλείται βάση του διανυσματικού χώρου V, αν ισχύει:

  a. Η θήκη του S παράγει τον διανυσματικό χώρο V, δηλαδή:

     $$\langle S \rangle = V.$$

  b. Το S είναι γραμμικά ανεξάρτητο σύνολο, δηλαδή τα διανύσματα που περιέχει είναι γραμμικά ανεξάρτητα.

Έστω ότι ο διανυσματικός χώρος V περιέχει μία βάση . Τότε ο αριθμός των στοιχείων της βάσης καλείται διάσταση του χώρου V και συμβολίζεται με $\dim V$.

Ορισμός 2.13 (Πίνακας πλήρους τάξης). Ένας πίνακας $\boldsymbol{A} \in \mathbb{F}^{m \times n}$ είναι:

- *Πλήρους τάξης στηλών* (full column rank) αν και μόνο αν ο A^TA είναι αντιστρέψιμος.

- *Πλήρους τάξης γραμμών* (full row rank) αν και μόνο αν ο AA^T είναι αντιστρέψιμος.

όπου η τάξη του πίνακα A, εκφράζει τον μέγιστο αριθμό των γραμμικά ανεξάρτητων διανυσμάτων του.

Ορισμός 2.14 (Πίνακας Vandermonde). Ένας πίνακας του οποίου οι γραμμές (ή οι στήλες) αποτελούν όρους γεωμετρικής προόδου, ονομάζεται πίνακας Vandermonde και εκφράζεται στην εξής μορφή:

$$\boldsymbol{A} = \begin{bmatrix} 1 & a_1 & a_1^2 & \cdots & a_1^{n-1} \\ 1 & a_2 & a_2^2 & \cdots & a_2^{n-1} \\ 1 & a_3 & a_3^2 & \cdots & a_3^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & a_m & a_m^2 & \cdots & a_m^{n-1} \end{bmatrix}$$

δηλαδή $\boldsymbol{A}_{i,j} = a_i^{j-1}, \forall i, j$.

Ορισμός 2.15 (Πολυωνυμικός δακτύλιος). Αν F είναι ένας αντιμεταθετικός δακτύλιος, τότε ένα πολυώνυμο με άγνωστο x στον δακτύλιο F, εκφράζεται ως εξής:

$$f(x) = a_n x^n + \cdots + a_2 x^2 + a_1 x + a_0$$

όπου $a_i \in F$ και $n \geq 0$. Το στοιχείο a_i καλείται συντελεστής του x^i. Ο μεγαλύτερος ακέραιος m, για τον οποίο $a_m \neq 0$ καλείται βαθμός του $f(x)$ και συμβολίζεται με $\deg f(x)$. Συνεπώς, ο πολυωνυμικός δακτύλιος $F[X]$, διαμορφώνεται από το σύνολο όλων των πολυωνύμων, με άγνωστο x και συντελεστές στο F.

Ορισμός 2.16 (Μονικό πολυώνυμο). Ένα πολυώνυμο βαθμού n λέγεται μονικό (*monic*) όταν ο συντελεστής a_n του μεγιστοβάθμιου όρου είναι μονάδα.

Ορισμός 2.17. Ένα πολυώνυμο $i(x)$ είναι πολλαπλασιαστικός αντίστροφος του $f(x)$ modulo $M(x)$, αν:

$$f(x) \cdot i(x) \equiv 1 \mod M(x).$$

Συγκεκριμένα, αν ο μέγιστος κοινός διαιρέτης $\gcd(f(X), M(X)) = 1$, υπάρχουν δύο πολυώνυμα $r(x)$ και $s(x)$, έτσι ώστε:

$$1 = \gcd(f(x), M(x)) = r(x) \cdot f(x) + s(x) \cdot M(x).$$

Κατ' επέκταση,

$$r(x) \cdot f(x) \equiv 1 \mod M(x)$$

Άρα, το $r(x)$ είναι πολλαπλασιαστικός αντίστροφος του $f(x) \mod M(x)$ και αντίστοιχα το $s(x)$ πολλαπλασιαστικός αντίστροφος του $M(x) \mod f(x)$.

2.3 Θεωρία πιθανοτήτων

Θεώρημα 2.18 (Θεώρημα του Bayes [42]). *Αν A και B δύο γεγονότα και $P(B) > 0$, τότε ισχύει:*

$$\Pr(A \mid B) = \frac{\Pr(A)\Pr(B \mid A)}{\Pr(B)}.$$

Ορισμός 2.19 (Δειγματοχώρος [42]). *Δειγματοχώρος ή δειγματικός χώρος* (sample space) ενός πειράματος τύχης, ονομάζεται το σύνολο όλων των δυνατών αποτελεσμάτων $\omega_i, i = 1, 2, \cdots$ του πειράματος και συμβολίζεται με $\Omega = \{\omega_1, \omega_2, \cdots, \omega_n\}$.

Ορισμός 2.20 (Διακριτή τυχαία μεταβλητή [42]). *Διακριτή τυχαία μεταβλητή* (discrete random variable) X στον δειγματοχώρο Ω, είναι μια μεταβλητή $X : \Omega \to A (A \subseteq \mathbb{R})$ που ορίζεται για κάθε δυνατό αποτέλεσμα ενός πειράματος και για κάθε τέτοιο αποτέλεσμα παίρνει μια ορισμένη τιμή από ένα πεπερασμένο ή άπειρο αριθμήσιμο υποσύνολο των πραγματικών αριθμών.

Ορισμός 2.21 (Συνάρτηση μάζας πιθανότητας [42]). Η *συνάρτηση μάζας πιθανότητας* (probability mass function - PMF) της τυχαίας μεταβλητής X, ορίζεται ως η πραγματική συνάρτηση $p_X : A \to [0,1]$, η οποία εκφράζει την πιθανότητα η τυχαία μεταβλητή X να πάρει μια ορισμένη τιμή από το πεδίο τιμών της:

$$p_X(x) = \mathsf{Pr}(X = x)$$

και ικανοποιεί τις συνθήκες:

$$p_X(x) \geq 0, \qquad \sum_{x \in A} p_X(x) = 1.$$

Ορισμός 2.22 (Στοχαστική ανεξαρτησία [42]). Δύο τυχαίες μεταβλητές X, Y καλούνται στοχαστικά ανεξάρτητες ή απλά ανεξάρτητες όταν:

$$p_{X,Y}(x,y) = p_X(x)p_Y(y), \qquad \forall x, y$$

και κατ' επέκταση

$$p_{X|Y}(x,y) = p_X(x), \qquad \forall x, y \qquad p_Y(y) > 0$$

Ορισμός 2.23 (Μέση τιμή [42]). Η *μέση τιμή* (expected value ή expectation ή mean) μιας τυχαίας μεταβλητής X με ΣΜΠ $p_X(x)$, ορίζεται ως:

$$\mathsf{E}[X] := \sum_x x p_X(x)$$

για $X_1, X_2, \cdots, X_n$ τυχαίες μεταβλητές ισχύει ότι:

$$\mathsf{E}\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} \mathsf{E}[X_i]$$

συγκεκριμένα αν οι τυχαίες μεταβλητές X, Y είναι ανεξάρτητες, σύμφωνα με τον ορισμό 2.22, τότε προκύπτει ότι:

$$\mathsf{E}[XY] = \mathsf{E}[X]\,\mathsf{E}[Y].$$

Θεωρώντας δύο τυχαίες μεταβλητές X και Y και $a \in \mathbb{R}$, μια σημαντική ιδιότητα που ικανοποιεί η $\mathsf{E}[X]$ είναι η εξής:

$$\mathsf{E}[aX + Y] = a\,\mathsf{E}[X] + \mathsf{E}[Y]$$

Ορισμός 2.24 (Διακύμανση [42]). Η *διακύμανση ή διασπορά* (variance) μιας τυχαίας μεταβλητής X, συμβολίζεται συνήθως με $\mathsf{Var}[X]$ και δηλώνει πόσο συγκεντρωμένες γύρω από την μέση τιμή είναι οι τιμές της τυχαίας μεταβλητής:

$$\mathsf{Var}[X] := \mathsf{E}[(X - \mathsf{E}[X]^2] = \mathsf{E}[X^2] - \mathsf{E}[X]^2 \geq 0.$$

Ορισμός 2.25 (Συνδιακύμανση [42]). Η *συνδιακύμανση* (covariance) συμβολίζεται συνήθως με $\mathsf{Cov}(X, Y)$ και εκφράζει το μέτρο του βαθμού συσχετίσεως μεταξύ δύο τυχαίων μεταβλητών X, Y:

$$\mathsf{Cov}[X, Y] := \mathsf{E}[XY] - \mathsf{E}[X]\,\mathsf{E}[Y].$$

Συγκεκριμένα αν οι τυχαίες μεταβλητές X, Y είναι ανεξάρτητες σύμφωνα με τον ορισμό 2.22 η σχέση της συνδιακύμανσης, διαμορφώνεται ως εξής:

$$\mathsf{Cov}[X, Y] = \mathsf{E}[(X - \mathsf{E}[X])(Y - \mathsf{E}[Y])] = \mathsf{E}[[X - \mathsf{E}[X]]\,\mathsf{E}[Y - \mathsf{E}[Y]] = 0.$$

Επιπλέον, για τυχαίες μεταβλητές $X_1, X_2, \cdots, X_n$ υπάρχει η παρακάτω σχέση, που συνδέει την διακύμανση και την συνδιακύμανση. Συγκεκριμένα:

$$\mathsf{Var}\left[\sum_{i=1}^{n} X_i\right] = \sum_{i,j=1}^{n} \mathsf{Cov}[Xi, Xj] = \sum_{i=1}^{n} \mathsf{Var}[X_i] + \sum_{i \neq j} \mathsf{Cov}[X_i, Xj] \tag{2.1}$$

στην περίπτωση όμως, που οι $X_1, X_2, \cdots, X_n$ είναι ανεξάρτητες η 2.1 διαμορφώνεται ως εξής:

$$\mathsf{Var}\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} \mathsf{Var}[X_i].$$

Θεώρημα 2.26 (Ανισότητα Marcov [42]). *Αν μια τυχαία μεταβλητή X μπορεί να πάρει μόνο θετικές τιμές, τότε:*

$$\mathsf{Pr}[|X| \geq a] \leq \frac{\mathsf{E}[|X|]}{a}, \qquad \forall a > 0.$$

Ωστόσο, εφαρμόζοντας την ανισότητα Markov στην τυχαία μεταβλητή $|X - \mathsf{E}[X]|^2$ προκύπτει το ακόλουθο θεώρημα:

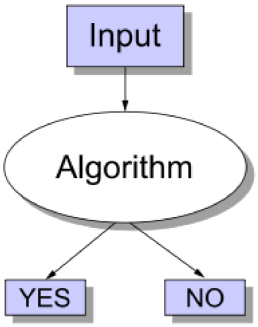
Θεώρημα 2.27 (Ανισότητα Chebychev [42]). *Αν X μια τυχαία μεταβλητή με μέση τιμή $\mathsf{E}[X]$ και διακύμανση $\mathsf{Var}[X]$, τότε:*

$$\mathsf{Pr}[|X - \mathsf{E}[X]| \geq a] \leq \frac{\mathsf{Var}[X]}{a^2}.$$

2.4 Κλάσεις πολυπλοκότητας

Ορισμός 2.28 (Κλάση P [52]). Η κλάση πολυπλοκότητας P, είναι το σύνολο όλων των προβλημάτων απόφασης, τα οποία επιλύονται σε πολυωνυμικό χρόνο.

Ορισμός 2.29 (Κλάση NP [52]). Η κλάση πολυπλοκότητας NP, είναι το σύνολο όλων των προβλημάτων απόφασης, για τα οποία η απάντηση NAI, μπορεί να πιστοποιηθεί σε πολυωνυμικό χρόνο, δοθείσας μιας επιπλέον πληροφορίας, που καλείται *πιστοποιητικό* (certificate).



Σχήμα 2.1: Ένα πρόβλημα απόφασης, έχει δύο πιθανές εξόδους (ναι ή όχι) είτε εναλλακτικά (0 ή 1), για κάθε είσοδο που δέχεται.

Ορισμός 2.30 (Πρόβλημα NP-πλήρες [52]). Ένα πρόβλημα απόφασης L (βλ. Σχ. 2.1) είναι NP-πλήρες αν:

- $L \in \text{NP}$
- $L_1 \leq pL, \forall L_1 \in \text{NP}$.

Η κλάση όλων των NP-πλήρων προβλημάτων ορίζεται ως NPC.
Τα NP-πλήρη προβλήματα, αποτελούν τα δυσκολότερα προβλήματα στην κλάση NP, υπό την έννοια ότι είναι τουλάχιστον τόσο δύσκολα, όσο οποιοδήποτε άλλο πρόβλημα που ανήκει στην κλάση αυτή.

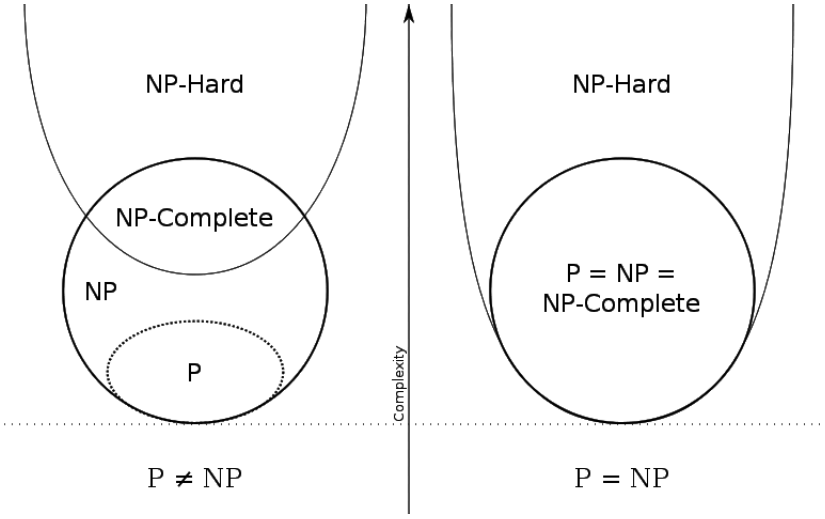
Ορισμός 2.31 (Πρόβλημα NP-δύσκολο [52]). Ένα πρόβλημα L είναι *NP-δύσκολο* (NP-hard) αν και μόνο αν ισχύει κάποια από τις ισοδύναμες συνθήκες:

- $L' \leq L, \forall L' \in \text{NP}$
- $L \in \text{P} \Rightarrow \text{P} = \text{NP}$
- Η ύπαρξη ενός ντετερμινιστικού πολυωνυμικού αλγορίθμου για το πρόβλημα L σημαίνει την ύπαρξη ενός τέτοιου αλγορίθμου, για κάθε πρόβλημα που ανήκει στην κλάση NP.

Ένα πρόβλημα NP είναι NP-πληρες αν ανήκει στην κλάση NP και είναι NP-δύσκολο.

2.5 Θεωρία πληροφορίας

Ορισμός 2.32 (Συνάρτηση δυαδικής εντροπίας [49]). Έστω X μια τυχαία μεταβλητή που μπορεί να λάβει n πιθανές τιμές από ένα σύνολο $Y = \{y_1, y_2, \cdots, y_n\}$ και $p_i = \text{Pr}(X = y_i) \forall i, 1 \leq i \leq n$ η πιθανότητα η τυχαία μεταβλητή X, να λάβει



Σχήμα 2.2: Διάγραμμα Euler για κλάσεις P και NP και για NP-πλήρη και NP-δύσκολα προβλήματα.

την τιμή y_i. Η *συνάρτηση δυαδικής εντροπίας* (binary entropy function) της τυχαίας μεταβλητής X, ορίζεται ως εξής:

$$H(X) = -\sum_{i=1}^{n} p_i \log_2 p_i$$

και εκφράζει τον βαθμό αβεβαιότητας, που συνδέεται με το αποτέλεσμα ενός πειράματος, πριν αυτό διεξαχθεί. Ωστόσο, στην περίπτωση που η τυχαία μεταβλητή X ακολουθεί την κατανομή Bernoulli [42], η συνάρτηση δυαδικής εντροπίας $H_2 : [0,1] \to \mathbb{R}$ διαμορφώνεται ως εξής:

$$H(X) := H_2(p) = -plog_2(p) - (1-p)log_2(1-p) \tag{2.2}$$

όπου με βάση το Σχ. 2.3, παρατηρούμε ότι πρόκειται για μια συνάρτηση συνεχή και συμμετρική γύρω από το $p = \frac{1}{2}$, δηλαδή:

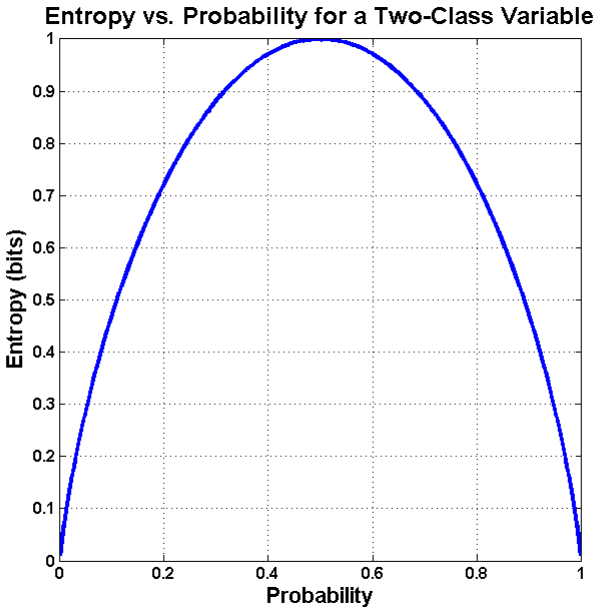
$$H(\frac{1}{2} + p) = H(\frac{1}{2} - p), \qquad p \in [0, \frac{1}{2}]$$

και συγκεκριμένα $H(0) = H(1) = 0$.

Πόρισμα 2.33. *([66], Πόρισμα 2.3.4) Για $a \in [0,1]$, $\beta \in [0,a]$ και επαρκώς μεγάλο n, προκύπτει ότι:*

$$\frac{1}{n} \log \binom{\lfloor an \rfloor}{\lfloor \beta n \rfloor} = aH_2\left(\frac{\beta}{a}\right) \tag{2.3}$$

σύμφωνα με την προσέγγιση Stirling [3].



Σχήμα 2.3: Συνάρτηση δυαδικής εντροπίας

Ισοδύναμα, με βάση τον συμβολισμό Landau $\widetilde{O}$ *[12] η* (2.3) *διαμορφώνεται ως εξής:*

$$\binom{an}{\beta n} = \widetilde{O}(2^{aH_2(\frac{\beta}{a})n}). \tag{2.4}$$

Ορισμός 2.34 (Μπάλα Hamming [66]). Η *μπάλα Hamming* (Hamming ball) είναι μια συμπαγής σφαίρα (μπάλα) ακτίνας $0 \leq r \leq n$ στον n-διάστατο χώρο, με κέντρο το 0.

$$B(n,r) := \left\{ \boldsymbol{e} \in \mathbb{F}_2^n : \mathsf{wt}(e) \leq r \right\}$$

όπου $\boldsymbol{e} = (e_0, e_1, \cdots, e_n)$ ένα διάνυσμα μήκους n και $\mathsf{wt}(\boldsymbol{e})$ ο αριθμός των μη-μηδενικών συνιστωσών του διανύσματος $\boldsymbol{e}$, που ορίζεται ως *βάρος Hamming* (Hamming weight).

Ορισμός 2.35 (Όγκος της μπάλας Hamming [66]). Ο *όγκος* (volume) του $B(n,r)$, συμβολίζεται με $vol_2(n,r)$ και ορίζεται ως εξής:

$$vol_2(n,r) := \sum_{0 \leq i \leq r} \binom{n}{i}.$$

Ακολουθεί μια ασυμπτωτική εκτίμηση του όγκου του $B(n,r)$, όπως αυτή προκύπτει με βάση την συνάρτηση δυαδικής εντροπίας. Συγκεκριμένα:

Λήμμα 2.36. *[66] Έστω* $r \in [0, \frac{1}{2}]$*, τότε:*

$$vol_2(n, rn) = 2^{H_2(r)n} + o(n), \qquad \textit{για} \quad n \to \infty.$$

Απόδειξη. Από τον ορισμό 2.35 έχουμε ότι

$$\frac{vol_2(n,rn)}{2^{H_2(r)n}} = \sum_{0\le i\le rn} \binom{n}{i}(1-r)^n\Big(\frac{r}{1-r}\Big)^{rn}$$

κι εφόσον λοιπόν $r \le \frac{1}{2}$ προκύπτει

$$r \le \frac{1}{2} \le 1-r$$

και κατ' επέκταση

$$\frac{r}{1-r} \le 1.$$

Συνεπώς, η παραπάνω έκφραση φράσσεται από πάνω, ως εξής

$$\frac{vol_2(n,rn)}{2^{H_2(r)n}} = \sum_{0\le i\le rn} \binom{n}{i}(1-r)^n\Big(\frac{r}{1-r}\Big)^{i} = \sum_{0\le i\le rn} \binom{n}{i}(1-r)^{n-i}r^i \le 1$$

κατά συνέπεια,

$$vol_2(n,rn) \le 2^{H_2(r)n}.$$

Τελικά με βάση την (2.4), προκύπτει ότι

$$vol_2(n,rn) \ge \binom{n}{\lfloor rn \rfloor} = 2^{H_2(r)n+o(n)}.$$

■

Ορισμός 2.37 (Σφαίρα Hamming [66]). Η *σφαίρα Hamming* (Hamming sphere) είναι μια σφαίρα στον n-διάστατο χώρο, που περιλαμβάνει όλα τα διανύσματα μήκους n και βάρους Hamming ακριβώς r.

$$W_{n,r} := \left\{ \boldsymbol{e} \in \mathbb{F}_2^n : \mathsf{wt}(e) = r \right\}.$$

Ορισμός 2.38 (Επικαλύπτουσα ακτίνα [66]). Η *επικαλύπτουσα ακτίνα* $\rho(C)$ (covering radius) ενός συνόλου $C \subseteq \mathbb{F}_2^n$, ορίζεται ως η ελάχιστη δυνατή ακτίνα, έτσι ώστε οι μπάλες Hamming, με κέντρο τα στοιχεία του C και ακτίνα ρ, να καλύπτουν ολόκληρο τον χώρο $\mathbb{F}_2^n$.

$$\rho := \rho(C) := \max_{\boldsymbol{x}\in\mathbb{F}_2^n} \mathsf{d}(\boldsymbol{x}, C)$$

όπου $\mathsf{d}(\boldsymbol{x}, C) = \min_{\boldsymbol{y}\in C} \mathsf{d}(\boldsymbol{x}, \boldsymbol{y})$. Καθώς, $\mathsf{d}(\boldsymbol{x}, \boldsymbol{y}) = \mathsf{wt}(\boldsymbol{x} + \boldsymbol{y})$ έχουμε

$$\rho := \rho(C) = \max_{\boldsymbol{x}\in\mathbb{F}_2^n} \min_{\boldsymbol{y}\in C} \mathsf{wt}(\boldsymbol{x} + \boldsymbol{y}) = \max_{\boldsymbol{x}\in\mathbb{F}_2^n} \min_{\boldsymbol{y}\in \boldsymbol{x}+C} \mathsf{wt}(\boldsymbol{y})$$

όπου το $\boldsymbol{x} + C$ ονομάζεται ομοσύνολο (βλ. Κεφ. 3, ενότητα 3.3).

ΚΕΦΑΛΑΙΟ 3

Γραμμικοί τμηματικοί κώδικες

Στην θεωρία κωδίκων [14], τμηματικοί κώδικες ονομάζονται εκείνοι που ανήκουν στην ευρύτερη οικογένεια των *κωδίκων διόρθωσης σφαλμάτων* (error correcting codes) και επιπλέον κωδικοποιούν την πληροφορία σε τμήματα. Δεδομένου ότι, η πληροφορία στα σύγχρονα ψηφιακά συστήματα επικοινωνίας και αποθήκευσης δεδομένων κωδικοποιείται στα δυαδικά ψηφία 0 και 1, θα εστιάσουμε την μελέτη μας στους δυαδικούς *γραμμικούς τμηματικούς κώδικες* (linear block codes), οι οποίοι χρησιμοποιούν σύμβολα από το αλφάβητο $\mathbb{F}(2)$.

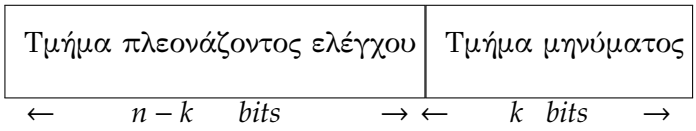
3.1 Αναπαράσταση κωδίκων

Υποθέτουμε λοιπόν, ότι η έξοδος από μια πηγή πληροφορίας είναι μια ακολουθία από τα δυαδικά ψηφία 0 και 1. Στην τμηματική κωδικοποίηση, η δυαδική ακολουθία πληροφορίας κατακερματίζεται σε τμήματα μηνυμάτων καθορισμένου μεγέθους, όπου κάθε τέτοιο τμήμα, συμβολίζεται με $\boldsymbol{m}$ και αποτελείται από k ψηφία πληροφορίας. Υπάρχουν συνολικά 2^k διαφορετικά μηνύματα. Ο κωδικοποιητής, σύμφωνα με κάποιους κανόνες, μετασχηματίζει κάθε μήνυμα εισόδου $\boldsymbol{m}$, σε ένα δυαδικό διάνυσμα $\boldsymbol{c}$, μεγέθους n, όπου $n > k$. Το διάνυσμα αυτό, αναφέρεται ως *κωδική λέξη* (codeword) του μηνύματος $\boldsymbol{m}$. Συνεπώς, κάθε ένα από τα 2^k πιθανά μηνύματα, αντιστοιχίζεται σε μια και μόνο μια κωδική λέξη. Δηλαδή, υπάρχει απεικόνιση 1-1 ανάμεσα στα μηνύματα $\boldsymbol{m}$ και τις κωδικές λέξεις.

Ένας τέτοιος τμηματικός κώδικας μήκους n και διάστασης k, καλείται γραμμικός (n,k) κώδικας αν και μόνο εάν οι 2^k κωδικές λέξεις, σχηματίζουν έναν k-διάστατο υποχώρο του διανυσματικού χώρου $\mathbb{F}_2^n$, όλων των διανυσμάτων μήκους n στο σώμα $\mathbb{F}(2)$, ο οποίος εφεξής θα συμβολίζεται με C. Κατ' επέκταση το (modulo 2) άθροισμα δύο κωδικών λέξεων, αποτελεί επίσης κωδική λέξη. Ο λόγος

$$R = \frac{k}{n}$$

ονομάζεται *ρυθμός πληροφορίας* (information rate ή code rate) του τμηματικού γραμμικού (n,k) κώδικα και ερμηνεύεται, ως ο αριθμός των ψηφίων πληροφορίας



Σχήμα 3.1: Συστηματική δομή κωδικής λέξης

που εισέρχονται στον κωδικοποιητή ανά μεταδιδόμενο σύμβολο. Κάθε διάνυσμα $\boldsymbol{c} \in \mathbb{F}_2^n$, εξαρτάται μόνο από το αντίστοιχο μήνυμα εισόδου $\boldsymbol{m} \in \mathbb{F}_2^k$.

Εφόσον $\dim(C) = k$, είναι δυνατόν να βρούμε k γραμμικά ανεξάρτητες κωδικές λέξεις, $\boldsymbol{g}_0, \boldsymbol{g}_1, \ldots, \boldsymbol{g}_{k-1}$, έτσι ώστε κάθε $\boldsymbol{c} \in C$, να αποτελεί γραμμικό συνδυασμό k κωδικών λέξεων: $\boldsymbol{c} = m_0\boldsymbol{g}_0 + m_1\boldsymbol{g}_1 + \cdots + m_{k-1}\boldsymbol{g}_{k-1}$. Με την βοήθεια των γραμμών ενός $k \times n$ πίνακα, αναπαριστούμε τις k γραμμικά ανεξάρτητες κωδικές λέξεις:

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{g}_0 \\ \boldsymbol{g}_1 \\ \vdots \\ \boldsymbol{g}_{k-1} \end{bmatrix} = \begin{bmatrix} g_{00} & g_{01} & \cdots & g_{0,n-1} \\ g_{10} & g_{11} & \cdots & g_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ g_{k-1,0} & g_{k-1,1} & \cdots & g_{k-1,n-1} \end{bmatrix}. \tag{3.1}$$

Αν $\boldsymbol{m} = [m_0, m_1, \cdots, m_{k-1}]^t$ είναι το μήνυμα που κωδικοποιείται, η αντίστοιχη κωδική λέξη υπολογίζεται ως εξής:

$$\boldsymbol{c}^t = \boldsymbol{m}^t \cdot \boldsymbol{G} = [m_0, m_1, \cdots, m_{k-1}] \begin{bmatrix} \boldsymbol{g}_0 \\ \boldsymbol{g}_1 \\ \vdots \\ \boldsymbol{g}_{k-1} \end{bmatrix} = m_0\boldsymbol{g}_0 + m_1\boldsymbol{g}_1 + \cdots + m_{k-1}\boldsymbol{g}_{k-1}. \tag{3.2}$$

Ο $\boldsymbol{G} \in \mathbb{F}_2^{k\times n}$ ονομάζεται πίνακας γεννήτορας, του κώδικα $C = \left\{\boldsymbol{m}^t \cdot \boldsymbol{G} : \boldsymbol{m} \in \mathbb{F}_2^k\right\}$. Ωστόσο, οποιεσδήποτε k γραμμικά ανεξάρτητες κωδικές λέξεις, ενός (n, k) γραμμικού κώδικα, μπορούν να χρησιμοποιηθούν ώστε να διαμορφώσουν έναν πίνακα γεννήτορα για τον συγκεκριμένο κώδικα.

Επιπλέον, μια επιθυμητή ιδιότητα των γραμμικών τμηματικών κωδίκων είναι, να διατηρούν οι κωδικές λέξεις μια συστηματική μορφή, βάση της οποίας κάθε κωδική λέξη διαιρείται σε 2 τμήματα (όπως απεικονίζεται στο Σχ. 3.1), το *τμήμα μηνύματος* (message part) και το *τμήμα πλεονάζοντος ελέγχου* (redundant checking part). Το τμήμα του μηνύματος αποτελείται από k ψηφία πληροφορίας, ενώ το τμήμα πλεονάζοντος ελέγχου, από $n - k$ *ψηφία ελέγχου ισοτιμίας* (parity check bits). Ωστόσο, ένας γραμμικός τμηματικός κώδικας που ακολουθεί αυτή την δομή, ονομάζεται συστηματικός τμηματικός κώδικας. Ένας γραμμι-

κός συστηματικός (n,k) κώδικας καθορίζεται πλήρως από έναν $k \times n$ πίνακα γεννήτορα $\boldsymbol{G}$, υπό την ακόλουθη μορφή:

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{g}_0 \\ \boldsymbol{g}_1 \\ \vdots \\ \boldsymbol{g}_{k-1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 & p_{00} & p_{01} & \cdots & p_{0,n-k-1} \\ 0 & 1 & \cdots & 0 & p_{10} & p_{11} & \cdots & p_{1,n-k-1} \\ & & \ddots & & & & & \\ 0 & 0 & \cdots & 1 & p_{k-1,0} & p_{k-1,1} & \cdots & p_{k-1,n-k-1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_k & \boldsymbol{P} \end{bmatrix} \tag{3.3}$$

όπου $\boldsymbol{I}_k$ είναι ο μοναδιαίος $k \times k$ πίνακας και $\boldsymbol{P} = [p_{ij}]$ είναι ο πίνακας με στοιχεία $p_{ij} \in \mathbb{F}_2$. Αποδεικνύεται ότι, για κάθε $k \times n$ πίνακα γεννήτορα $\boldsymbol{G}$, με k γραμμικά ανεξάρτητες γραμμές, υπάρχει ένας $(n-k) \times n$ πίνακας $\boldsymbol{H}$ με $n-k$ γραμμικά ανεξάρτητες γραμμές, έτσι ώστε κάθε διάνυσμα που προκύπτει από τις γραμμές του πίνακα $\boldsymbol{G}$ να είναι ορθογώνιο με τις γραμμές του πίνακα $\boldsymbol{H}$ και αντιστρόφως κάθε διάνυσμα που είναι ορθογώνιο με τις γραμμές του πίνακα $\boldsymbol{H}$, να προκύπτει από τις γραμμές του πίνακα $\boldsymbol{G}$. Συνεπώς, μπορούμε να περιγράψουμε έναν (n,k) γραμμικό τμηματικό κώδικα C, που παράγεται από τον πίνακα γεννήτορα $\boldsymbol{G}$, και με έναν εναλλακτικό τρόπο βασισμένο στον πίνακα $\boldsymbol{H}$, που καλείται πίνακας ελέγχου ισοτιμίας.

Σύμφωνα λοιπόν, με τον πίνακα ελέγχου ισοτιμίας $\boldsymbol{H}$, ένα διάνυσμα $\boldsymbol{c}$ μήκους n σε έναν (n,k) κώδικα C , ο οποίος παράγεται από τον πίνακα γεννήτορα $\boldsymbol{G}$, θα αποτελεί κωδική λέξη, αν και μόνο αν:

$$\boldsymbol{H} \cdot \boldsymbol{c} = \boldsymbol{0}$$

Συνεπώς, ο κώδικας C ορίζεται μέσω του πίνακα ελέγχου ισοτιμίας $\boldsymbol{H} \in \mathbb{F}_2^{(n-k)\times n}$, ως εξής: $C = \left\{ \boldsymbol{x} \in \mathbb{F}_2^n : \boldsymbol{H} \cdot \boldsymbol{x} = \boldsymbol{0} \right\}$. Οι 2^{n-k} γραμμικοί συνδυασμοί των γραμμών του πίνακα $\boldsymbol{H}$, σχηματίζουν έναν $(n, n-k)$ γραμμικό κώδικα $C^{\perp}$, του οποίου το σύνολο των δυαδικών διανυσμάτων είναι ορθογώνια με όλα τα στοιχεία του C. Για παράδειγμα, $\forall \boldsymbol{x} \in C$ και $\forall \boldsymbol{y} \in C^{\perp}$, θα ισχύει $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0$. Ο κώδικας $C^{\perp}$ ονομάζεται *δυϊκός κώδικας* (dual code) του C.

Συνεπώς, ο πίνακας ελέγχου ισοτιμίας στην συστηματική του μορφή, εκφράζεται ως εξής:

$$\boldsymbol{H} = \begin{bmatrix} -\boldsymbol{P}^T & \boldsymbol{I}_{n-k} \end{bmatrix} = \begin{bmatrix} p_{00} & p_{10} & \cdots & p_{k-1,0} & 1 & 0 & \cdots & 0 \\ p_{01} & p_{11} & \cdots & p_{k-1,1} & 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & & & & \ddots & \\ p_{0,n-k-1} & p_{1,n-k-1} & \cdots & p_{k-1,n-k-1} & 0 & 0 & \cdots & 1 \end{bmatrix} \tag{3.4}$$

όπου $\boldsymbol{P}^T$ ο ανάστροφος πίνακας του πίνακα $\boldsymbol{P}$. Αν υποθέσουμε ότι, η $\boldsymbol{h}_j$ αποτελεί την j-οστή γραμμή του πίνακα H και η $\boldsymbol{g}_i$ την i-οστή του πίνακα G, το εσωτερικό τους γινόμενο υπολογίζεται ως εξής: $\boldsymbol{h}_j \cdot \boldsymbol{g}_i = p_{ij} + p_{ij} = 0$, για $0 \leq i < k$ και $0 \leq j < n-k$, από όπου τελικά προκύπτει ότι: $\boldsymbol{G} \cdot \boldsymbol{H}^t = \boldsymbol{0}$. Η παραπάνω σχέση,

προκύπτει και από τον ορισμό του δυαδικού γραμμικού κώδικα και του δυϊκού του μέσω των πινάκων, γεννήτορα $\boldsymbol{G}$ και έλεγχου ισοτιμίας $\boldsymbol{H}$. Καθώς, εάν ο (n,k) δυαδικός γραμμικός κώδικας C, προσδιορίζεται μέσω του πίνακα γεννήτορα $\boldsymbol{G}$ και του πίνακα ελέγχου ισοτιμίας $\boldsymbol{H}$, τότε ο $C^{\perp}$ προσδιορίζεται μέσω του πίνακα γεννήτορα $\boldsymbol{H}$ και του πίνακα ελέγχου ισοτιμίας $\boldsymbol{G}$. Συνεπώς, οι πίνακες γεννήτορες των παραπάνω δυϊκών κωδίκων, θα πρέπει να ικανοποιούν την σχέση $\boldsymbol{G} \cdot \boldsymbol{H}^t = \boldsymbol{0}$.

3.2 Σύνδρομο και ανίχνευση σφαλμάτων

Θεωρούμε έναν (n,k) γραμμικό κώδικα με πίνακα γεννήτορα $\boldsymbol{G}$ και πίνακα ελέγχου ισοτιμίας $\boldsymbol{H}$. Έστω $\boldsymbol{c} = (c_0, c_1, \cdots, c_{n-1})^t$ μια κωδική λέξη, που μεταδίδεται μέσω ενός θορυβώδους τηλεπικοινωνιακού καναλιού και $\boldsymbol{x} = (x_0, x_1, \cdots, x_{n-1})^t$ το λαμβανόμενο διάνυσμα που παράγεται στην έξοδο του καναλιού. Λόγω της επίδρασης του θορύβου, ενδέχεται το διάνυσμα $\boldsymbol{x}$ να είναι διαφορετικό από το διάνυσμα $\boldsymbol{c}$. Στην περίπτωση αυτή, από το (modulo 2) άθροισμα των δύο διανυσμάτων προκύπτει το εξής:

$$\boldsymbol{e} = \boldsymbol{x} + \boldsymbol{c} = (e_0, e_1, \cdots, e_{n-1})^t \tag{3.5}$$

όπου $\boldsymbol{e}$ είναι ένα διάνυσμα μήκους n, για το οποίο θα ισχύει ότι $e_i = 1$ για $x_i \neq c_i$ και $e_i = 0$ για $x_i = c_i$, όπου $0 < i \leq n-1$. Ακριβέστερα, πρόκειται για ένα διάνυσμα που υποδεικνύει τις θέσεις στις οποίες το λαμβανόμενο διάνυσμα $\boldsymbol{x}$, διαφέρει από την αντίστοιχη μεταδιδόμενη κωδική λέξη $\boldsymbol{c}$. Δηλαδή $e_i \neq 0$ εάν και μόνο αν υπάρχει σφάλμα στη θέση i. Το διάνυσμα $\boldsymbol{e}$ ονομάζεται *διάνυσμα σφάλματος* (error vector ή error pattern).

Συνεπώς, από τη (3.5), προκύπτει ότι το λαμβανόμενο διάνυσμα στην έξοδο του καναλιού, θα είναι το άθροισμα της μεταδιδόμενης κωδικής λέξης και του διανύσματος σφάλματος. Πιο συγκεκριμένα:

$$\boldsymbol{x} = \boldsymbol{c} + \boldsymbol{e}. \tag{3.6}$$

Προφανώς, ο παραλήπτης του λαμβανόμενου διανύσματος $\boldsymbol{x}$ δεν γνωρίζει ούτε το $\boldsymbol{c}$, αλλά ούτε και το $\boldsymbol{e}$.

Ωστόσο, η διαδικασία που ακολουθείται είναι η εξής: σε πρώτο στάδιο, ο αποκωδικοποιητής λαμβάνοντας το $\boldsymbol{x}$, καθορίζει αν περιέχονται σε αυτό, σφάλματα που προέκυψαν κατά την μετάδοση. Στην συνέχεια, στην περίπτωση ανίχνευσης σφαλμάτων προβαίνει στην διαδικασία διόρθωσής τους αν αυτό είναι εφικτό, διαφορετικά ζητά αναμετάδοση της κωδικής λέξης $\boldsymbol{c}$. Πιο συγκεκριμένα, όταν λαμβάνεται το διάνυσμα $\boldsymbol{x}$ ο αποκωδικοποιητής υπολογίζει την ποσότητα:

$$\boldsymbol{s} = \boldsymbol{H} \cdot \boldsymbol{x} = (s_0, s_1, \cdots, s_{n-k-1})^t \tag{3.7}$$

που καλείται *σύνδρομο* (syndrome) του $\boldsymbol{x}$. Εάν $\boldsymbol{s} \neq \boldsymbol{0}$ ανιχνεύεται η παρουσία σφαλμάτων στον κώδικα και κατ' επέκταση το λαμβανόμενο διάνυσμα $\boldsymbol{x}$ δεν αποτελεί κωδική λέξη. Στην αντίθετη όμως περίπτωση, δηλαδή εάν $\boldsymbol{s} = \boldsymbol{0}$, το διάνυσμα $\boldsymbol{x}$ αποτελεί κωδική λέξη του (n, k) κώδικα και συνεπώς, ο παραλήπτης αποδέχεται το διάνυσμα $\boldsymbol{x}$, σαν την κωδική λέξη που μεταδόθηκε (θεωρώντας ότι δεν υπάρχουν σφάλματα κατά την μετάδοση).

Ως άμεση απόρροια των παραπάνω, υπάρχει το ενδεχόμενο παρότι $\boldsymbol{s} = \boldsymbol{H} \cdot \boldsymbol{x} = \boldsymbol{0}$, να υπάρχουν στον κώδικα σφάλματα τα οποία ονομάζονται *μη-ανιχνεύσιμα σφάλματα* (undetectable error patterns). Αυτό για παράδειγμα συμβαίνει όταν το διάνυσμα σφάλματος $\boldsymbol{e}$ είναι μια μη-μηδενική κωδική λέξη. Ακριβέστερα στην περίπτωση αυτή, το $\boldsymbol{x}$ θα είναι το διανυσματικό άθροισμα δύο κωδικών λέξεων, το οποίο θα αποτελεί, επίσης κωδική λέξη και κατ' επέκταση θα ισχύει ότι: $\boldsymbol{H} \cdot \boldsymbol{x} = \boldsymbol{0}$.

Πιο συγκεκριμένα, ένας (n, k) γραμμικός κώδικας είναι ικανός να ανιχνεύσει $2^n - 2^k$ διανύσματα σφάλματος, μεγέθους n. Από τα $2^n - 1$ πιθανά μη-μηδενικά διανύσματα σφάλματος, υπάρχουν $2^k - 1$, τα οποία ταυτίζονται με τις $2^k - 1$ μη-μηδενικές κωδικές λέξεις, που είναι δυνατόν να περιέρχονται στον κώδικα. Κατά συνέπεια, αν προκληθεί οποιοδήποτε από τα $2^k - 1$ διανύσματα σφάλματος, η μεταδιδόμενη κωδική λέξη $\boldsymbol{c}$, μεταβάλλεται σε μια άλλη κωδική λέξη $\boldsymbol{v}$. Στην περίπτωση αυτή, ο αποκωδικοποιητής αποδέχεται την λαμβανόμενη λέξη $\boldsymbol{v}$ σαν την αρχικά μεταδιδόμενη κωδική λέξη και κατ' επέκταση υπόκειται σε εσφαλμένη αποκωδικοποίηση. Συμπεραίνουμε λοιπόν ότι, υπάρχουν $2^k - 1$ μη-ανιχνεύσιμα διανύσματα σφάλματος.

Ωστόσο για επαρκώς μεγάλο n και αν ο ρυθμός του κώδικα είναι σταθερός, η ποσότητα αυτή θεωρείται γενικά πολύ μικρότερη από 2^n και κατά συνέπεια μόνο ένας πολύ μικρός αριθμός από τα διανύσματα σφάλματος, περνά τελικά από τον αποκωδικοποιητή χωρίς να ανιχνευθεί. Στην αντίθετη όμως περίπτωση κατά την οποία, το διάνυσμα σφάλματος δεν ταυτίζεται με μια μη-μηδενική κωδική λέξη (για την ακρίβεια υπάρχουν $2^n - 2^k$ τέτοια διανύσματα σφάλματος σε έναν (n, k) γραμμικό κώδικα), τότε το λαμβανόμενο διάνυσμα $\boldsymbol{x}$, δεν αποτελεί κωδική λέξη και κατ' επέκταση έχουμε $\boldsymbol{s} \neq \boldsymbol{0}$. Κατά συνέπεια, ο αποκωδικοποιητής ανιχνεύει ότι προκλήθηκε σφάλμα κατά την μετάδοση.

Εύκολα διαπιστώνουμε ότι, το σύνδρομο $\boldsymbol{s}$ που υπολογίζεται από το λαμβανόμενο διάνυσμα $\boldsymbol{x}$, εξαρτάται τελικά, μόνο από το διάνυσμα σφάλματος $\boldsymbol{e}$ και όχι από την μεταδιδόμενη κωδική λέξη $\boldsymbol{c}$. Καθώς σύμφωνα με την (3.6), προκύπτει το εξής:

$$\boldsymbol{s} = \boldsymbol{H} \cdot \boldsymbol{x} = \boldsymbol{H} \cdot (\boldsymbol{c} + \boldsymbol{e}) = \boldsymbol{H} \cdot \boldsymbol{c} + \boldsymbol{H} \cdot \boldsymbol{e}. \tag{3.7b}$$

Ωστόσο γνωρίζουμε ότι ισχύει $\boldsymbol{H} \cdot \boldsymbol{c} = \boldsymbol{0}$, εφόσον το $\boldsymbol{c}$ αποτελεί κωδική λέξη. Κατά

συνέπεια, καταλήγουμε στην ακόλουθη σχέση μεταξύ του συνδρόμου και του διανύσματος σφάλματος:

$$\boldsymbol{s} = \boldsymbol{H} \cdot \boldsymbol{e}. \tag{3.8}$$

Αν ο πίνακας ελέγχου ισοτιμίας $\boldsymbol{H}$, εκφράζεται στην συστηματική του μορφή, όπως υποδηλώνει η (3.4), τότε βάσει της (3.8) προκύπτει η ακόλουθη γραμμική σχέση μεταξύ των ψηφίων του συνδρόμου, με τα αντίστοιχα ψηφία του διανύσματος σφάλματος:

$$s_i = e_i + \sum_{j=0}^{k-1} p_{ji} e_{n-k+j}, \qquad i = 0, \cdots, n-k-1. \tag{3.9}$$

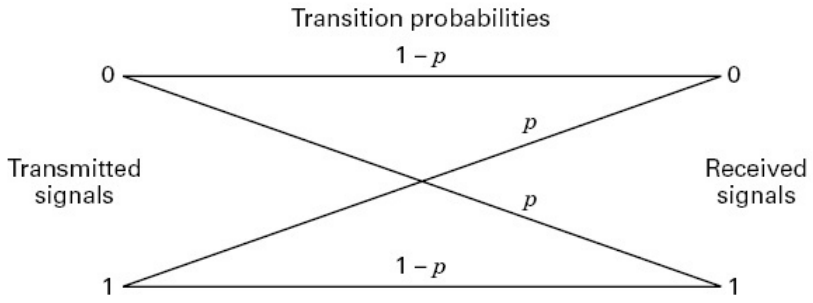
Συμπεραίνουμε λοιπόν, ότι τα ψηφία του συνδρόμου αποτελούν γραμμικό συνδυασμό των ψηφίων του διανύσματος σφάλματος και ως εκ τούτου, μπορούν να χρησιμοποιηθούν για τη διόρθωση των σφαλμάτων.

Με μια πρώτη ματιά θεωρούμε ότι, οποιαδήποτε μέθοδος διόρθωσης συνίσταται στην λύση του συστήματος των $n-k$ γραμμικών εξισώσεων, όπως προκύπτει από τις (3.8) και (3.9). Καθώς, εφόσον βρεθεί το διάνυσμα σφάλματος $\boldsymbol{e}$, το διάνυσμα $\boldsymbol{x}+\boldsymbol{e}$ μπορεί να θεωρηθεί ως η αρχική μεταδιδόμενη κωδική λέξη. Δυστυχώς όμως, ο καθορισμός του πραγματικού διανύσματος σφάλματος $\boldsymbol{e}$, δεν αποτελεί ιδιαίτερα απλή διαδικασία. Για την ακρίβεια, το σύστημα των $n-k$ γραμμικών εξισώσεων, όπως προκύπτει από την (3.9), δεν έχει μια μοναδική λύση αλλά 2^k λύσεις, καθώς υπάρχουν 2^k πιθανά διανύσματα σφάλματος, που αθροίζουν στο ίδιο σύνδρομο σύμφωνα με την (3.7b) (με την διαφορά όμως, ότι μόνο ένα από αυτά αποτελεί το πραγματικό διάνυσμα σφάλματος).

Συνεπώς, ο αποκωδικοποιητής θα πρέπει να καθορίσει το διάνυσμα σφάλματος από ένα σύνολο από 2^k πιθανά τέτοια διανύσματα. Προκειμένου λοιπόν, να ελαχιστοποιηθεί η πιθανότητα σφάλματος κατά την αποκωδικοποίηση, επιλέγεται ως πραγματικό διάνυσμα σφάλματος το πιο πιθανό διάνυσμα που ικανοποιεί τις γραμμικές εξισώσεις που προκύπτουν από την (3.9). Πιο συγκεκριμένα, στην περίπτωση ενός *δυαδικού συμμετρικού καναλιού* - BSC (binary symmetric channel ή BSC) (βλ. Σχ. 3.2), ως πιο πιθανό διάνυσμα σφάλματος, θεωρείται αυτό με τον μικρότερο αριθμό μη-μηδενικών ψηφίων [14].

3.3 Ελάχιστη απόσταση

Θα αναλύσουμε μια ιδιαίτερα σημαντική παράμετρο ενός (n,k) τμηματικού κώδικα που ονομάζεται *ελάχιστη απόσταση* (minimum distance). Πρόκειται για μια παράμετρο βάσει της οποίας, καθορίζεται η ικανότητα της *τυχαίας ανίχνευσης σφαλμάτων* (random-error-detecting) ενός κώδικα, καθώς επίσης και η ικανότητα της *τυχαίας διόρθωσης σφαλμάτων* (random-error-correcting) του



Σχήμα 3.2: Το BSC αποτελεί ένα δυαδικό κανάλι, στο οποίο ο αποστολέας αποστέλλει ένα ψηφίο (0 ή 1). Δεδομένου ότι η μετάδοση δεν είναι τέλεια και το κανάλι είναι θορυβώδες, ενδέχεται το ψηφίο που θα λάβει τελικά ο παραλήπτης, να είναι διαφορετικό από την αρχική μετάδοση του αποστολέα.

κώδικα.

Έστω λοιπόν $\boldsymbol{c} = (c_0, c_1, \cdots, c_{n-1})^t$, ένα διάνυσμα μήκους n, το *βάρος Hamming* (Hamming weight) ή απλά *βάρος* (weight) του $\boldsymbol{c}$ που συμβολίζεται με $\mathsf{wt}(\boldsymbol{c})$, ορίζεται ως ο αριθμός των μη-μηδενικών συνιστωσών του διανύσματος $\boldsymbol{c}$. Αντίστοιχα, στην περίπτωση δύο διανυσμάτων μήκους n $\boldsymbol{c}$, $\boldsymbol{v}$, η απόσταση Hamming ή απλά απόσταση μεταξύ των διανυσμάτων αυτών συμβολίζεται με $\mathsf{d}(\boldsymbol{c}, \boldsymbol{v})$ και ορίζεται ως ο αριθμός των θέσεων στις οποίες διαφέρουν. Η απόσταση Hamming αποτελεί μια μετρική που ικανοποιεί την ακόλουθη τριγωνική ανισότητα.

$$\mathsf{d}(\boldsymbol{c}, \boldsymbol{v}) + \mathsf{d}(\boldsymbol{v}, \boldsymbol{x}) \geq \mathsf{d}(\boldsymbol{c}, \boldsymbol{x}), \qquad \forall \boldsymbol{c}, \boldsymbol{v}, \boldsymbol{x} \in \mathbb{F}_2^n. \tag{3.10}$$

Από τον ορισμό της απόστασης Hamming και την modulo-2 πρόσθεση συνεπάγεται ότι η απόσταση Hamming μεταξύ δυο διανυσμάτων μήκους n $\boldsymbol{c}, \boldsymbol{v}$, ισούται με το βάρος Hamming του αθροίσματος των εν λόγω διανυσμάτων.

$$\mathsf{d}(\boldsymbol{c}, \boldsymbol{v}) = \mathsf{wt}\,(\boldsymbol{c} + \boldsymbol{v})\,. \tag{3.11}$$

Κατά συνέπεια, δοθέντος ενός τμηματικού κώδικα C η ελάχιστη απόσταση μεταξύ δύο κωδικών λέξεων $\boldsymbol{c}, \boldsymbol{v}$, συμβολίζεται με d_{min} και ορίζεται ως εξής:

$$\mathsf{d}_{min} = \min\{\mathsf{d}(\boldsymbol{c}, \boldsymbol{v}) : \boldsymbol{c}, \boldsymbol{v} \in C, \boldsymbol{c} \neq \boldsymbol{v}\}\,. \tag{3.12}$$

Στην περίπτωση που ο C είναι ένας γραμμικός τμηματικός κώδικας, το άθροισμα δύο κωδικών λέξεων, αποτελεί επίσης κωδική λέξη. Κατ' επέκταση, βάσει της (3.11) προκύπτει ότι η απόσταση Hamming μεταξύ δύο κωδικών λέξεων στον κώδικα C, είναι ίση με το βάρος Hamming μιας άλλης κωδικής λέξης του C και άρα:

$$\mathsf{d}_{min} = \min\{\mathsf{wt}(\boldsymbol{x}) : \boldsymbol{x} \in C, \boldsymbol{x} \neq 0\}\,. \tag{3.13}$$

Η παράμετρος wt_{min} ονομάζεται *ελάχιστο βάρος* (minimum weight) του γραμμικού κώδικα C και οδηγούμαστε στο ακόλουθο θεώρημα.

Θεώρημα 3.1. *Η ελάχιστη απόσταση ενός γραμμικού τμηματικού κώδικα είναι ίση με το ελάχιστο βάρος των μη-μηδενικών κωδικών λέξεων, που περιέχονται στον κώδικα και αντίστροφα.*

Επιπλέον, στο θεώρημα που ακολουθεί, αναδεικνύεται η σχέση του ελαχίστου βάρους ενός γραμμικού τμηματικού κώδικα, με τον πίνακα ελέγχου ισοτιμίας $\boldsymbol{H}$.

Θεώρημα 3.2. *Έστω C ένας (n,k) γραμμικός τμηματικός κώδικας με πίνακα ελέγχου ισοτιμίας $\boldsymbol{H}$. Για κάθε κωδική λέξη με βάρος Hamming l, υπάρχουν l στήλες του πίνακα $\boldsymbol{H}$ των οποίων το άθροισμα είναι ίσο με το μηδενικό διάνυσμα. Αντιστρόφως, αν υπάρχουν l στήλες του $\boldsymbol{H}$, των οποίων το άθροισμα είναι το μηδέν, τότε υπάρχει μια κωδική λέξη βάρους Hamming l, που ανήκει στον κώδικα C.*

Απόδειξη. Αρχικά εκφράζουμε τον πίνακα ελέγχου ισοτιμίας στην ακόλουθη μορφή:

$$\boldsymbol{H} = \left[\boldsymbol{h}_0^t, \boldsymbol{h}_1^t, \cdots, \boldsymbol{h}_{n-1}^t\right]$$

όπου το $\boldsymbol{h}_i$ αναπαριστά την i-οστή στήλη του $\boldsymbol{H}$. Στην συνέχεια, θεωρούμε το διάνυσμα $\boldsymbol{c} = (c_0, c_1, \cdots, c_{n-1})^t$, το οποίο αποτελεί κωδική λέξη βάρους l. Συνεπώς, το $\boldsymbol{c}$ θα έχει l μη-μηδενικές συνιστώσες $c_{i_1}, c_{i2}, \cdots, c_{i_l} = 1$, όπου $0 \leq i_1 < i_2 \cdots < i_l \leq n-1$. Δεδομένου λοιπόν, ότι το $\boldsymbol{c}$ αποτελεί κωδική λέξη για τον κώδικα C, ισχύει ότι

$$\begin{aligned}\mathbf{0} &= \boldsymbol{H} \cdot \boldsymbol{c} \\ &= \boldsymbol{h}_0^t \boldsymbol{c}_0 + \boldsymbol{h}_1^t \boldsymbol{c}_1 + \cdots + \boldsymbol{h}_{n-1}^t \boldsymbol{c}_{n-1} \\ &= \boldsymbol{h}_{i_1}^t + \boldsymbol{h}_{i_2}^t + \cdots + \boldsymbol{h}_{i_l}^t\end{aligned}$$

Και αντίστροφα, υποθέτουμε ότι οι $\boldsymbol{h}i_1, \boldsymbol{h}_{i_2}, \cdots, \boldsymbol{h}i_l$, αποτελούν l στήλες του πίνακα ελέγχου ισοτιμίας $\boldsymbol{H}$, έτσι ώστε να ισχύει

$$\boldsymbol{h}_{i_1} + \boldsymbol{h}_{i_2} + \cdots + \boldsymbol{h}_{i_l} = \mathbf{0}. \tag{3.14}$$

Στην συνέχεια, θεωρούμε ένα διάνυσμα $\boldsymbol{x} = (x_0, x_1, \cdots, x_{n-1})^t$ μήκους n, με βάρος Hamming l και μη-μηδενικές συνιστώσες τις $x_{i_1}, x_{i_2}, \cdots, x_{i_l}$. Το εσωτερικό γινόμενο του $\boldsymbol{x}$ με τον πίνακα ελέγχου ισοτιμίας $\boldsymbol{H}$, εκφράζεται ως εξής

$$\begin{aligned}\boldsymbol{H} \cdot \boldsymbol{x} &= \boldsymbol{h}_0^t \boldsymbol{x}_0 + \boldsymbol{h}_1^t \boldsymbol{x}_1 + \cdots + \boldsymbol{h}_{n-1}^t \boldsymbol{x}_{n-1} \\ &= \boldsymbol{h}_{i_1}^t + \boldsymbol{h}_{i_2}^t + \cdots + \boldsymbol{h}_{i_l}^t.\end{aligned}$$

Σύμφωνα λοιπόν, με την (3.14) προκύπτει το εξής: $\boldsymbol{H} \cdot \boldsymbol{x} = \mathbf{0}$. Συνεπώς, το διάνυσμα $\boldsymbol{x}$ αποτελεί κωδική λέξη του κώδικα C, με $\mathsf{wt}(\boldsymbol{x}) = l$. ■

3.3.1 Ικανότητα ανίχνευσης και διόρθωσης σφαλμάτων

Εάν η ελάχιστη απόσταση ενός τμηματικού κώδικα C είναι d_{min}, οποιεσδήποτε δύο κωδικές λέξεις του C, θα διαφέρουν μεταξύ τους τουλάχιστον κατά d_{min} θέσεις. Αυτό συνεπάγεται ότι, ένα διάνυσμα σφάλματος βάρους το πολύ $\mathsf{d}_{min}-1$, δεν είναι ικανό να μετατρέψει μια κωδική λέξη σε μια άλλη. Συνεπώς, ένας τμηματικός κώδικας είναι δυνατόν να ανιχνεύσει όλα τα διανύσματα σφάλματος βάρους το πολύ $\mathsf{d}_{min}-1$. Ωστόσο, δεν μπορεί να ανιχνεύσει όλα τα διανύσματα σφάλματος που περιέχουν d_{min} σφάλματα, καθώς υπάρχει τουλάχιστον ένα ζεύγος κωδικών λέξεων, οι οποίες διαφέρουν μεταξύ τους κατά d_{min} θέσεις. Αυτό ισχύει και για διανύσματα σφάλματος με περισσότερα από d_{min} σφάλματα.

Κατά συνέπεια, η *ικανότητα ανίχνευσης τυχαίων σφαλμάτων* (random-error-detecting-capability) ενός τμηματικού κώδικα, με ελάχιστη απόσταση d_{min} είναι: $\mathsf{d}_{min}-1$. Η ελάχιστη απόσταση d_{min} ενός κώδικα ενδέχεται να είναι είτε περιττός, είτε άρτιος αριθμός. Έστω t ένας θετικός ακέραιος, έτσι ώστε

$$2t+1 \leq \mathsf{d}_{min} \leq 2t+2. \tag{3.15}$$

Ένας τμηματικός κώδικας με ελάχιστη απόσταση d_{min} εγγυάται διόρθωση όλων των διανυσμάτων σφάλματος με $t = \lfloor(\mathsf{d}_{min}-1)/2\rfloor$ ή λιγότερα σφάλματα, όπου $\lfloor x\rfloor$ συμβολίζει τον μεγαλύτερο ακέραιο, που είναι μικρότερος ή ίσος του $\boldsymbol{x}$. Η παράμετρος t, ονομάζεται *διορθωτική ικανότητα τυχαίων σφαλμάτων* (random-error-correcting-capability) του κώδικα ή απλά διορθωτική ικανότητα.

Έστω $\boldsymbol{c}$ η πραγματικά μεταδιδόμενη κωδική λέξη και v μια άλλη κωδική λέξη που ανήκει στον κώδικα C, έτσι ώστε

$$\mathsf{d}(\boldsymbol{c}, v) \geq \mathsf{d}_{min} \geq 2t+1, \tag{3.16}$$

Υποθέτουμε ότι, ένα διάνυσμα σφάλματος που αποτελείται από $t' = \frac{\mathsf{d}_{min}-1}{2}$ σφάλματα, προκαλείται κατά την μετάδοση της κωδικής λέξης $\boldsymbol{c}$. Στην περίπτωση αυτή το λαμβανόμενο διάνυσμα $\boldsymbol{x}$, θα διαφέρει από την μεταδιδόμενη κωδική λέξη $\boldsymbol{c}$ κατά t' θέσεις και συνεπώς, $\mathsf{d}(\boldsymbol{c}, \boldsymbol{x}) = \mathsf{wt}(\boldsymbol{c}+\boldsymbol{x}) = t'$. Δεδομένου λοιπόν ότι, $\mathsf{d}(\boldsymbol{c}, \boldsymbol{x}) = t'$ και συνδυάζοντας τις (3.10) και (3.16), προκύπτει ότι:

$$\mathsf{d}(v, \boldsymbol{x}) \geq 2t+1-t',$$

Αν $t' \leq t$,

$$\mathsf{d}(v, \boldsymbol{x}) > t,$$

Συμπεραίνουμε λοιπόν ότι, αν προκληθεί οποιοδήποτε διάνυσμα σφάλματος με t ή λιγότερα σφάλματα, η απόσταση Hamming του λαμβανόμενου διανύσματος $\boldsymbol{x}$ από την μεταδιδόμενη κωδική λέξη $\boldsymbol{c}$, θα είναι μικρότερη συγκριτικά, με οποιαδήποτε άλλη κωδική λέξη περιέχεται στον κώδικα C. Κατ' επέκταση το

λαμβανόμενο διάνυσμα $\boldsymbol{x}$ αποκωδικοποιείται στην πραγματικά μεταδιδόμενη κωδική λέξη $\boldsymbol{c}$. Επιπλέον στην περίπτωση αυτή, τα σφάλματα διορθώνονται και η αποκωδικοποίηση που τελικά προκύπτει, είναι σωστή. Αντιθέτως, ο κώδικας δεν είναι ικανός να διορθώσει διανύσματα σφάλματος με t ή περισσότερα σφάλματα.

3.4 Αποκωδικοποίηση συνδρόμου

Έστω C ένας (n,k) γραμμικός τμηματικός κώδικας και $\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots, \boldsymbol{c}_{2_k}$ οι κωδικές λέξεις που περιέχονται στον κώδικα. Ανεξάρτητα από το ποια κωδική λέξη μεταδίδεται μέσω ενός καναλιού, το λαμβανόμενο διάνυσμα $\boldsymbol{x}$ μπορεί να είναι οποιοδήποτε από τα διανύσματα του $\mathbb{F}_2^n$. Κάθε σχήμα αποκωδικοποίησης στον παραλήπτη, χρησιμοποιεί μια μέθοδο διαμέρισης των 2^n πιθανών λαμβανόμενων διανυσμάτων (η διαμέριση αυτή βασίζεται στην γραμμική δομή του κώδικα), σε 2^k ξένα υποσύνολα, έτσι ώστε μια κωδική λέξη $\boldsymbol{c}_i$ να περιέχεται σε ένα υποσύνολο, όπου $1 \leq i \leq 2^k$. Συνεπώς, θα πρέπει κάθε υποσύνολο, να αποτελεί 1-1 αντιστοίχιση για κάθε μια από τις κωδικές λέξεις $\boldsymbol{c}_i$. Αν το λαμβανόμενο διάνυσμα $\boldsymbol{x}$ βρίσκεται σε ένα τέτοιο υποσύνολο, αποκωδικοποιείται στην αντίστοιχη κωδική λέξη $\boldsymbol{c}_i$. Ωστόσο, η αποκωδικοποίηση είναι σωστή μόνο αν το λαμβανόμενο διάνυσμα $\boldsymbol{x}$, αντιστοιχεί στην κωδική λέξη που μεταδόθηκε.

Πιο συγκεκριμένα, αρχικά τοποθετούμε τις 2^k κωδικές λέξεις του κώδικα C, σε μια γραμμή με την μηδενική λέξη $\boldsymbol{c}_1 = \mathbf{0}$, να αποτελεί το πρώτο (αριστερότερο) στοιχείο της γραμμής αυτής. Από τα υπόλοιπα $2^n - 2^k$ διανύσματα μήκους n, επιλέγουμε ένα διάνυσμα $\boldsymbol{e}_2$ και το τοποθετούμε κάτω από το μηδενικό διάνυσμα $\boldsymbol{c}_1$. Στην συνέχεια, σχηματίζουμε την δεύτερη γραμμή προσθέτοντας το διάνυσμα $\boldsymbol{e}_2$, σε κάθε κωδική λέξη που βρίσκεται στην πρώτη γραμμή και τοποθετώντας το άθροισμα $\boldsymbol{e}_2 + \boldsymbol{c}_i$ κάτω από την αντίστοιχη κωδική λέξη $\boldsymbol{c}_i$. Στο επόμενο βήμα επιλέγουμε ένα αχρησιμοποίητο διάνυσμα $\boldsymbol{e}_3$ το οποίο και προσθέτουμε σε κάθε κωδική λέξη $\boldsymbol{c}_i$. Το άθροισμα $\boldsymbol{e}_3 + \boldsymbol{c}_i$ τοποθετείται κάτω από την αντίστοιχη κωδική λέξη $\boldsymbol{c}_i$. Η διαδικασία επαναλαμβάνεται, έως ότου χρησιμοποιηθούν όλα τα διανύσματα σφάλματος $\boldsymbol{e}_j$, $2 \leq j \leq 2^{n-k}$. Ο $2^{n-k} \times 2^k$ πίνακας που τελικά προκύπτει, ονομάζεται *τυπική διάταξη* (standard array) και απεικονίζεται στο Σχ.3.3. Συγκεκριμένα, οι 2^{n-k} γραμμές της τυπικής διάταξης καλούνται *ομοσύνολα* (cosets) για τον κώδικα C. Το διάνυσμα $\boldsymbol{e}_l$ της πρώτης στήλης καλείται *αντιπρόσωπος ομοσυνόλου* (coset leader ή coset representative).

Οι 2^k στήλες της τυπικής διάταξης, χρησιμοποιούνται στην αποκωδικοποίηση του κώδικα C. Η αποκωδικοποίηση είναι σωστή αν και μόνο αν, το διάνυσμα σφάλματος $\boldsymbol{e}_l$ που προκαλείται από το κανάλι επικοινωνίας, είναι ένας αντιπρόσωπος ομοσυνόλου. Για τον λόγο αυτό, οι 2^{n-k} αντιπρόσωποι ομοσυνόλου (συμπεριλαμβανομένου και του μηδενικού διανύσματος) ονομάζονται *διορθώσιμα*

$$\begin{array}{ccccccc} \boldsymbol{e}_1 = \boldsymbol{c}_1 = 0 & \boldsymbol{c}_2 & \cdots & \boldsymbol{c}_i & \cdots & \boldsymbol{c}_{2^k} \\ \boldsymbol{e}_2 & \boldsymbol{e}_2 + \boldsymbol{c}_2 & \cdots & \boldsymbol{e}_2 + \boldsymbol{c}_i & \cdots & \boldsymbol{e}_2 + \boldsymbol{c}_{2^k} \\ \vdots & & & & & \vdots \\ \boldsymbol{e}_l & \boldsymbol{e}_l + \boldsymbol{c}_2 & \cdots & \boldsymbol{e}_l + \boldsymbol{c}_i & \cdots & \boldsymbol{e}_l + \boldsymbol{c}_{2^k} \\ \vdots & & & & & \vdots \\ \boldsymbol{e}_{2^{n-k}} & \boldsymbol{e}_{2^{n-k}} + \boldsymbol{c}_2 & \cdots & \boldsymbol{e}_{2^{n-k}} + \boldsymbol{c}_i & \cdots & \boldsymbol{e}_{2^{n-k}} + \boldsymbol{c}_{2^k} \end{array}$$

Σχήμα 3.3: Τυπική διάταξη ενός (n, k) γραμμικού κώδικα

διανύσματα σφάλματος (correctable error patterns). Ακριβέστερα ως αντιπρόσωπος ομοσυνόλου, επιλέγεται το διάνυσμα ελαχίστου βάρους σε ένα ομοσύνολο (συγκριτικά με τα υπόλοιπα διαθέσιμα διανύσματα σφάλματος).

Θεώρημα 3.3. *Το λαμβανόμενο διάνυσμα αποκωδικοποιείται στην κοντινότερη κωδική λέξη $\boldsymbol{c}_i$, άρα για κάθε αντιπρόσωπο ομοσυνόλου, η αποκωδικοποίηση που βασίζεται στην τυπική διάταξη είναι αποκωδικοποίηση ελάχιστης απόστασης (minimum distance decoding ή complete decoding ή full decoding - MDD) ή ισοδύναμα αποκωδικοποίηση μεγίστης πιθανοφάνειας (maximum likelihood decoding - MLD).*

Απόδειξη. Έστω $\boldsymbol{x}$ το λαμβανόμενο διάνυσμα στην έξοδο του καναλιού. Υποθέτουμε ότι, το $\boldsymbol{x}$ βρίσκεται στην i-οστη γραμμή και l- οστή στήλη της τυπικής διάταξης. Συνεπώς, αποκωδικοποιείται στην κωδική λέξη $\boldsymbol{c}_i$, άρα:

$$\boldsymbol{x} = \boldsymbol{e}_l + \boldsymbol{c}_i.$$

Κατ' επέκταση, η απόσταση Hamming μεταξύ του $\boldsymbol{x}$ και του $\boldsymbol{c}_i$, θα είναι:

$$d(\boldsymbol{x}, \boldsymbol{c}_i) = \mathsf{wt}(\boldsymbol{x} + \boldsymbol{c}_i) = \mathsf{wt}(\boldsymbol{e}_l). \tag{3.17}$$

Στην συνέχεια, εξετάζουμε την απόσταση Hamming, μεταξύ του λαμβανόμενου διανύσματος $\boldsymbol{x}$ και οποιασδήποτε άλλης κωδικής λέξης (έστω $\boldsymbol{c}_s$) που περιέχεται στον κώδικα C. Συνεπώς:

$$\mathsf{d}(\boldsymbol{x}, \boldsymbol{c}_s) = \mathsf{wt}(\boldsymbol{x} + \boldsymbol{c}_s) = \mathsf{wt}(\boldsymbol{e}_l + \boldsymbol{c}_i + \boldsymbol{c}_s).$$

Εφόσον $\boldsymbol{c}_i \neq \boldsymbol{c}_s$ (όπως προκύπτει από την κατασκευή της τυπικής διάταξης), το άθροισμα $\boldsymbol{c}_i + \boldsymbol{c}_s$, είναι μια μη-μηδενική κωδική λέξη, έστω $\boldsymbol{c}_j$ και συνεπώς:

$$\mathsf{d}(\boldsymbol{x}, \boldsymbol{c}_s) = \mathsf{wt}(\boldsymbol{e}_l + \boldsymbol{c}_j). \tag{3.18}$$

Καθώς το $\boldsymbol{e}_l$ και το $\boldsymbol{e}_l + \boldsymbol{c}_j$ ανήκουν στο ίδιο ομοσύνολο και $\mathsf{wt}(\boldsymbol{e}_l) < \mathsf{wt}(\boldsymbol{e}_l + \boldsymbol{c}_j)$. Συνδυάζοντας τις (3.17) και (3.18) προκύπτει τελικά:

$$\mathsf{d}(\boldsymbol{x}, \boldsymbol{c}_i) \leq \mathsf{d}(\boldsymbol{x}, \boldsymbol{c}_j), \forall \boldsymbol{c}_j \neq \boldsymbol{c}_i.$$

■

3.4.1 Αποκωδικοποίηση μεγίστης πιθανοφάνειας

Δοθείσας της λαμβανόμενης λέξης $x \in \mathbb{F}_2^n$ στην έξοδο του καναλιού, ο αποκωδικοποιητής μεγίστης πιθανοφάνειας, επιλέγει ως εκτιμώμενη είσοδο $\widehat{c}$ την κωδική λέξη $c \in C$, η οποία μεγιστοποιεί την πιθανότητα, να ληφθεί το διάνυσμα x, δεδομένου ότι εστάλει το διάνυσμα c. Δηλαδή, ο αποκωδικοποιητής για κάθε λαμβανόμενη λέξη x, υπολογίζει:

$$\Pr(x \mid c)$$

το οποίο με βάση το Θεώρημα του Bayes (βλ. θεώρημα 2.18) ισούται με

$$\Pr(x \mid c) = \frac{\Pr(c \mid x)\Pr(x)}{\Pr(c)}. \tag{3.19}$$

Ωστόσο η $\Pr(x)$, είναι ανεξάρτητη από τον κανόνα αποκωδικοποίησης, καθώς το λαμβανόμενο διάνυσμα x, αποτελεί την είσοδο στη διαδικασία αποκωδικοποίησης. Επιπλέον και η $\Pr(c)$, είναι ανεξάρτητη από τον κανόνα αποκωδικοποίησης, θεωρώντας ότι όλες οι κωδικές λέξεις, είναι εξίσου πιθανές να μεταδοθούν. Συνεπώς:

$$\widehat{c} = \arg\max_{c \in C} \Pr(x \mid c) = \arg\max_{c \in C} \Pr(c \mid x)$$

Ωστόσο, σε ένα Δυαδικό Συμμετρικό Κανάλι, ο αποκωδικοποιητής μεγίστης πιθανοφάνειας επιλέγει ως εκτιμώμενη είσοδο $\widehat{c}$ την κωδική λέξη $c \in C$, η οποία ελαχιστοποιεί την απόσταση Hamming $\mathrm{d}(x, c)$, μεταξύ του λαμβανόμενου διανύσματος x και της κωδικής λέξης c. Κατ' επέκταση, η αποκωδικοποίηση αυτή για ένα δυαδικό συμμετρικό κανάλι συνήθως, ονομάζεται και αποκωδικοποίηση ελάχιστης απόστασης. Συνεπώς

$$\widehat{c} = \arg\min_{c \in C} \mathrm{d}(c, x).$$

Θεώρημα 3.4. *Όλα τα 2^k διανύσματα μήκους n, σε ένα ομοσύνολο έχουν το ίδιο σύνδρομο. Τα σύνδρομα για διαφορετικά ομοσύνολα είναι διαφορετικά.*

Απόδειξη. Θεωρούμε το ομοσύνολο, $e_j + C$ του οποίου ο αντιπρόσωπος ομοσυνόλου είναι το e_j. Το σύνδρομο ενός διανύσματος στο ομοσύνολο, υπολογίζεται ως εξής:

$$\boldsymbol{H} \cdot (\boldsymbol{e}_j + \boldsymbol{c}_i) = \boldsymbol{H} \cdot \boldsymbol{e}_j + \boldsymbol{H} \cdot \boldsymbol{c}_i = \boldsymbol{H} \cdot \boldsymbol{e}_j$$

διότι $\boldsymbol{H} \cdot \boldsymbol{c}_i = \boldsymbol{0}$ (εφόσον το $\boldsymbol{c}_i$ αποτελεί κωδική λέξη του C). Ουσιαστικά, η παραπάνω ισότητα αποδεικνύει ότι, το σύνδρομο οποιουδήποτε διανύσματος σε ένα ομοσύνολο, είναι ίσο με το σύνδρομο του αντιπροσώπου του ομοσυνόλου. Κατά συνέπεια, όλα τα διανύσματα σε ένα ομοσύνολο, έχουν το ίδιο σύνδρομο. Αντιθέτως, τα διανύσματα που ανήκουν σε διαφορετικά ομοσύνολα, έχουν διαφορετικά σύνδρομα. Συνεπώς, υπάρχει αντιστοίχιση 1-1 ανάμεσα στον αντιπρόσωπο ομοσυνόλου και το αντίστοιχο σύνδρομο. ■

3.5 Τυχαίοι κώδικες

Προκειμένου να συγκρίνουμε εν συνέχεια, την ασυμπτωτική πολυπλοκότητα των αλγορίθμων αποκωδικοποίησης συνόλου πληροφορίας (βλ. Κεφ. 8), θα εστιάσουμε την μελέτη μας σε κώδικες αυξανόμενου μήκους n και διάστασης $k := k(n)$ και συγκεκριμένα σε τυχαίους γραμμικούς κώδικες, σταθερού ρυθμού πληροφορίας $0 < R < 1$, σταθερής ελάχιστης απόστασης και διορθωτικής ικανότητας $0 < W < 1/2$, για τους οποίους ισχύει

$$\lim_{n\to\infty} \frac{k(n)}{n} = R \tag{3.20}$$

$$\lim_{n\to\infty} \frac{d(n)}{n} = D \tag{3.21}$$

$$\lim_{n\to\infty} \frac{\omega(n)}{n} = W \tag{3.22}$$

όπου $R, D, W \in \mathbb{R}$ αντίστοιχα.

Έστω $C(n,k)$ ένας τυχαίος γραμμικός κώδικας, με πίνακα ελέγχου ισοτιμίας[1] $\boldsymbol{H} \in \mathbb{F}_2^{(n-k)\times k}$. Για καθορισμένο διάνυσμα $\boldsymbol{x} \in \mathbb{F}_2^n$ και σύνδρομο $\boldsymbol{s} \in \mathbb{F}_2^{(n-k)\times n}$, προκύπτει

$$\mathsf{Pr}[\boldsymbol{Hx} = \boldsymbol{s}] = \frac{1}{2^{n-k}} \tag{3.23}$$

συγκεκριμένα, όλες οι $n-k$ εξισώσεις είναι ανεξάρτητες και για κάθε μία από αυτές ($\langle \boldsymbol{h}_i, \boldsymbol{x}\rangle = s_i$), η πιθανότητα είναι $\frac{1}{2}$. Ισοδύναμα, η (3.23) διατυπώνεται ως

$$\mathsf{Pr}[\boldsymbol{x} \in C] = \frac{1}{2^{n-k}}. \tag{3.24}$$

Συγκεκριμένα, με βάση την (3.24), αποδεικνύεται εύκολα το ακόλουθο λήμμα.

Λήμμα 3.5. *[66] Έστω $S \subset \mathbb{F}_2^n$ και $N(S) := |S \cap C|$[2]. Εάν υποθέσουμε ότι, $0 \notin S$, τότε*

i) $\mathsf{E}[N(S)] = \frac{|S|}{2^{n-k}}$

ii) $\mathsf{Var}[N(S)] \leq \mathsf{E}[N(S)]$

Απόδειξη. Έστω $\sigma := |S|$, όπου $S = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_\sigma\}$ και $N(S) = \sum \chi_i$ (όπου η τυχαία μεταβλητή $\chi_i = 1$ αν και μόνο αν $\boldsymbol{x_i} \in C$). Σύμφωνα με την (3.24):

$$\mathsf{E}[\chi_i] = \frac{1}{2^{n-k}}$$

[1] Για επαρκώς μεγάλο n και $R > 0$, ο $\boldsymbol{H} \in \mathbb{F}_2^{(n-k(n))\times n}$ (του οποίου τα στοιχεία επιλέγονται ομοιόμορφα, με τυχαίο τρόπο), θα είναι πλήρους τάξης γραμμών $n-k$.

[2] Η $N(S)$ αποτελεί μια τυχαία μεταβλητή, η οποία εκφράζει τον αριθμό των κωδικών λέξεων, που περιέχονται στο σύνολο S.

συνεπώς από την *γραμμικότητα αναμενόμενων τιμών* [53], προκύπτει η (*i*). Εν συνεχεία, θα αποδείξουμε την (*ii*) με βάση τον ορισμό 2.24 και την (2.1). Συγκεκριμένα

$$\mathsf{Var}[N(S)] = \sum_{i=1}^{\sigma} \mathsf{Var}[\chi_i] + \sum_{i \neq j} \mathsf{Cov}[\chi_i, \chi_j]. \tag{3.25}$$

Αν $\boldsymbol{x}_i$ και $\boldsymbol{x}_j$ είναι διανύσματα γραμμικά ανεξάρτητα, τότε οι χ_i και χ_j αποτελούν ανεξάρτητες τυχαίες μεταβλητές και συνεπώς $\mathsf{Cov}[\chi_i, \chi_j] = 0$ [42]. Επομένως, η (3.25) διαμορφώνεται ως εξής

$$\sum_{i=1}^{\sigma} \mathsf{E}[\chi_i] - \mathsf{E}[\chi_i]^2 = \frac{\sigma}{2^{n-k}}\Big(1 - \frac{1}{2^{n-k}}\Big)$$

όπου $\chi_i = \chi_i^2$. Κατ' επέκταση προκύπτει η (*ii*). ■

Ορισμός 3.6 (Σχετική απόσταση Gilbert - Varshamov [40]). Έστω $0 < \mathrm{R} < 1$. Η *σχετική απόσταση Gilbert - Varshamov* (relative Gilbert - Varshamov distance ή GV) $D_{GV}(R) \in \mathbb{R}$, αποτελεί μοναδική λύση της εξίσωσης

$$H_2(x) = 1 - R, \qquad 0 \leq x \leq \frac{1}{2}. \tag{3.26}$$

Θεώρημα 3.7. *Σχεδόν όλοι οι γραμμικοί κώδικες συναντούν την σχετική απόσταση Gilbert - Varshamov. Για την ακρίβεια, για σχεδόν όλους τους κώδικες, με ρυθμό πληροφορίας R, ισχύει*

$$\mathsf{d}(C) \geq \lfloor D_{GV}(R)n \rfloor.$$

Απόδειξη. Έστω $\epsilon > 0$ και $\omega := \lfloor (D_{GV}(R) - \epsilon)n \rfloor$ (όπου $\omega = t$ η διορθωτική ικανότητα του κώδικα C). Συνδυάζοντας το Λήμμα 2.36, βάσει του οποίου

$$vol_2(n, \omega) = 2^{H_2(\frac{\omega}{n})n + o(n)}, \qquad \omega = rn$$

και τον ορισμό 3.6, προκύπτει

$$H_2\left(\frac{\omega}{n}\right) \leq H_2(D_{GV}(R) - \epsilon) = 1 - R - \epsilon'.$$

για κάποιο $\epsilon' > 0$. Κατ' επέκταση, εφαρμόζοντας το Λήμμα 3.5

$$\begin{aligned} \mathsf{E}[N_\omega] &= \frac{vol_2(n, \omega)}{2^{n-k}} \\ &= \frac{2^{H_2(\frac{\omega}{n})n + o(n)}}{2^{n-k}} \\ &\overset{(3.20)}{=} 2^{n(H_2(\frac{\omega}{n}) - (1-R)) + o(n)} \\ &= 2^{-\epsilon' n + o(n)} \end{aligned} \tag{3.27}$$

όπου η τυχαία μεταβλητή N_ω, εκφράζει το αναμενόμενο πλήθος των κωδικών λέξεων που ανήκουν στην μπάλα Hamming $B(n, \omega)$ (βλ. ορισμό 2.34). Ωστόσο, στην

περίπτωση κατά την οποία $\omega > \lfloor (D_{GV}(R))n \rfloor$, η (3.27) διαμορφώνεται αντίστοιχα, ως εξής:

$$\mathsf{E}[N_\omega] = 2^{\epsilon n + o(n)}$$

εφόσον

$$H_2\left(\frac{\omega}{n}\right) = 1 - R + \epsilon, \qquad \epsilon > 0 \tag{3.28}$$

επιπλέον,

$$\begin{aligned}
\mathsf{E}[N_\omega]^2 &= \frac{vol_2(n,\omega)}{(2^{n-k})^2} \\
&= \frac{2^{H_2(\frac{\omega}{n})n + o(n)}}{(2^{n-k})^2} \\
&\overset{(3.20)}{=} 2^{n(H_2(\frac{\omega}{n}) - 2(1-R)) + o(n)} \\
&\overset{(3.28)}{=} 2^{n(1-R+\epsilon-2+2R)} = 2^{-n} \cdot 2^{Rn} \cdot 2^{\epsilon n}
\end{aligned}$$

συνεπώς, με βάση το λήμμα 3.5, προκύπτει ότι:

$$\mathsf{Var}[N_\omega] = \mathsf{E}[N_\omega] - \mathsf{E}[N_\omega]^2 = 2^{\epsilon n} \cdot (1 - 2^{(R-1)n})$$

και από την ανισότητα του Chebychev (βλ. ορισμό 2.27)

$$\mathsf{Pr}[|N_\omega - \mathsf{E}[N_\omega]| \geq 2\,\mathsf{E}[N_\omega]] \leq \frac{\mathsf{Var}[N_\omega]}{(2\,\mathsf{E}[N_\omega])^2} = \frac{2^{\epsilon n} \cdot (1 - 2^{-(1-R)n})}{(2 \cdot 2^{\epsilon n})^2} \simeq \frac{2^{-\epsilon n}}{4}. \tag{3.29}$$

Διαπιστώνουμε λοιπόν ότι, η N_ω συγκεντρώνεται γύρω από την $\mathsf{E}[N_\omega]$, για σχεδόν όλους τους κώδικες. ■

Παρατηρούμε, ωστόσο, ότι για $\omega \geq \frac{n}{2}$ το λήμμα 2.36, δεν μπορεί πλέον να εφαρμοστεί και στην περίπτωση αυτή υπολογίζεται η εξής απλή εκτίμηση: $\mathsf{E}[N_\omega] = 2^{nR - o(n)}$ (δηλαδή σχεδόν όλες οι κωδικές λέξεις περιέχονται στο $B(n,\omega)$). Αυτό προφανώς προκύπτει, διότι όσο αυξάνεται το ω, αυξάνεται και το $B(n,\omega)$, με αποτέλεσμα να περιέχονται σε αυτό, σχεδόν όλα τα διανύσματα. Συμπεραίνουμε λοιπόν τα ακόλουθα.

Πόρισμα 3.8. *Για σχεδόν όλους τους γραμμικούς κώδικες, με ρυθμό R, προκύπτει:*

$$N_\omega = \begin{cases} 2^{n(H_2(\frac{\omega}{n}) - (1-R)) + o(n)}, & D_{GV}(R)n < \omega < \frac{n}{2} \\ 2^{nR - o(n)}, & \frac{n}{2} \leq \omega \leq n \\ 0, & \omega < D_{GV}(R)n \end{cases}$$

Θεώρημα 3.9 (Θεώρημα 3.4, [36]). *Για σχεδόν όλους τους κώδικες ρυθμού πληροφορίας R, ισχύει*

$$\rho(C) = D_{GV}(R)n + o(n).$$

3.6 Υπολογιστικό πρόβλημα αποκωδικοποίησης συνδρόμου

Ορισμός 3.10. Το *υπολογιστικό πρόβλημα αποκωδικοποίησης συνδρόμου* (computational syndrome decoding ή CSD), ορίζεται ως εξής: δοθέντος του πίνακα ελέγχου ισοτιμίας $\boldsymbol{H} \in \mathbb{F}_2^{(n-k)\times k}$, του συνδρόμου $\boldsymbol{s} \in \mathbb{F}_2^{(n-k)}$ και του αντίστοιχου βάρους $\omega \in \mathbb{N}$, αναζητείται διάνυσμα $\boldsymbol{e} \in \mathbb{F}_2^n$ (εάν υπάρχει), έτσι ώστε $\boldsymbol{He} = \boldsymbol{s}$ και επιπλέον $\mathsf{wt}(\boldsymbol{e}) \leq \omega$. Το πρόβλημα συμβολίζεται με $CSD(\boldsymbol{H}, \boldsymbol{s}, \omega)$.

Παρατήρηση 3.11. Με βάση τον ορισμό 3.10 δεν απαιτείται να βρεθεί λύση ελαχίστου βάρους, αντιθέτως κάθε λύση με $\mathsf{wt}(\boldsymbol{e}) \leq \omega$ είναι αποδεκτή.

Ο *χρόνος εκτέλεσης*[3] (running time) ενός αλγορίθμου για την επίλυση του CSD προβλήματος, εξαρτάται από τις παραμέτρους n, k, ω και συμβολίζεται με $T(n, k, \omega)$. Συγκεκριμένα, όλοι οι αλγόριθμοι αποκωδικοποίησης που χρησιμοποιούνται για την επίλυση του CSD, είναι εκθετικού χρόνου εκτέλεσης (ακόμα και για κατάλληλη επιλογή παραμέτρων). Η σύγκριση όσον αφορά την απόδοση τους, συνήθως πραγματοποιείται για μεγάλους κώδικες, ρυθμού πληροφορίας R και διορθωτικής ικανότητας $\omega = \lfloor Wn \rfloor$ για $W > 0$. Το 1990 οι Coffey και Goodman [24], εισήγαγαν μια μετρική σύγκρισης των εν λόγω αλγορίθμων, που αποκαλείται *συντελεστής πολυπλοκότητας* (complexity coefficient) και δίνεται από τον ακόλουθο τύπο:

$$T^*(R, W) := \lim_{n\to\infty} \frac{1}{n} \log T(n, \lfloor Rn \rfloor, \lfloor Wn \rfloor) \tag{3.30}$$

η μετρική αυτή ωστόσο, αγνοεί τους εκάστοτε πολυωνυμικούς συντελεστές $p(n)$, δηλαδή $\lim \frac{1}{n} \log p(n) = 0$.

3.6.1 Παράμετρος ω

Συγκεκριμένα, η δυσκολία επίλυσης όσον αφορά το CSD πρόβλημα, για μεγάλους κώδικες με ρυθμό πληροφορίας R, έγκειται στην επιλογή της παραμέτρου ω. Το πρόβλημα είναι χαμηλής δυσκολίας σε δύο περιπτώσεις:

- η παράμετρος ω είναι σταθερή
- η παράμετρος ω είναι αρκετά μεγάλη ($\omega > n/2$)

Για την ακρίβεια, στην ασυμπτωτική μελέτη επίλυσης του CSD προβλήματος διακρίνουμε 3 περιπτώσεις. Αρχικά, την περίπτωση κατά την οποία $W < D_{GV}(R)$, γεγονός που συνεπάγεται μοναδικότητα της επιθυμητής λύσης, καθώς η μπάλα Hamming $B(n, \omega)$ αυξάνεται αναλογικά με το ω και εκθετικά με το n (με βάση την απόδειξη του θεωρήματος 3.7). Υπάρχει δηλαδή μοναδική κοντινότερη κωδική λέξη στην οποία αποκωδικοποιείται το λαμβανόμενο διάνυσμα (εφόσον βρισκόμαστε εντός της διορθωτικής ικανότητας - MDD).

[3] Εκφράζει τον αριθμό των πράξεων που εκτελούνται ανά μονάδα χρόνου.

Επιπλέον, η περίπτωση κατά την οποία $W > D_{GV}(R)$, όπου με βάση το πόρισμα 3.8 προκύπτει ένας εκθετικός αριθμός λύσεων (αυξάνεται τόσο το ω όσο και το $B(n, \omega)$), με αποτέλεσμα να περιέχονται σε αυτό περισσότερες κωδικές λέξεις. Συνεπώς, η αποκωδικοποίηση του λαμβανόμενου διανύσματος καθίσταται ευκολότερη, χωρίς ωστόσο να είναι και απαραιτήτως σωστή.

Τέλος, η δυσκολότερη εκ των 3 περιπτώσεων κατά την οποία $W = D_{GV}(R)$, αποτελεί οριακή τιμή καθώς η αποκωδικοποίηση του λαμβανόμενου διανύσματος εμπίπτει είτε στην 1η είτε στην 2η περίπτωση. Εν συνεχεία και με βάση τις παραπάνω περιπτώσεις, θα εξετάσουμε την συμπεριφορά των πιο γνωστών αλγορίθμων αποκωδικοποίησης, όσον αφορά την επίλυση του CSD προβλήματος (για σταθερά n, R και μεταβλητό W) - *αλγόριθμοι αποκωδικοποίησης συνόλου πληροφορίας* (information set decoding ή ISD) (βλ. Κεφ. 6, ενότητα 1).

3.6.1.1 Συσχέτιση με το MDD πρόβλημα

Η μελέτη της ασυμπτωτικής πολυπλοκότητας του MDD προβλήματος, περιορίζεται στους αλγορίθμους αποκωδικοποίησης που μπορούν να διορθώσουν μέχρι $D_{GV}(R)n$ σφάλματα, σύμφωνα με το θεώρημα 3.9. Συνεπώς, σφάλματα (που προκύπτουν κατά την αποκωδικοποίηση) βάρους μέχρι και $\Omega := [D_{GV}(R + \epsilon)n], \forall \epsilon > 0$ είναι επαρκή για την επίλυση του MDD προβλήματος. Έστω λοιπόν, $\mathcal{A}$ ένας αλγόριθμος επίλυσης για το CSD πρόβλημα, με συντελεστή πολυπλοκότητας $T^*_{\mathcal{A}}(R, W)$. Ως εκ τούτου, η κατ' επανάληψη επίκληση του αλγορίθμου $\mathcal{A}$ το πολύ Ω φορές (για αυξανόμενες τιμές $\omega = 1, 2, \cdots$), επιτρέπει την εύρεση μιας λύσης ελαχίστου βάρους για το MDD πρόβλημα. Κατά συνέπεια, ο συντελεστής πολυπλοκότητας για την επίλυση του MDD αλγορίθμου, φράσσεται από πάνω και τελικά διαμορφώνεται ως εξής: $T^*_{\mathcal{A}}(R, D_{GV} + \epsilon)$.

ΚΕΦΑΛΑΙΟ 4

Δυαδικοί ανάγωγοι Goppa κώδικες

Υπάρχουν πολλοί λόγοι που καθιστούν σημαντικούς τους Goppa κώδικες στην κρυπτογραφία [47]. Συγκεκριμένα, η γνώση του *πολυωνύμου γεννήτορα* (generating polynomial) καθιστά αποδοτική την διόρθωση σφαλμάτων (καθώς χωρίς την γνώση του δεν υπάρχουν γνωστοί αποδοτικοί αλγόριθμοι). Επιπλέον, είναι εύκολο να υπολογιστεί το κάτω φράγμα για την ελάχιστη απόσταση.

4.1 Ορισμός

Έστω m, t θετικοί ακέραιοι και

$$g(X) = \sum_{i=0}^{t} g_i X^i \in \mathbb{F}_{2^m}[X] \tag{4.1}$$

είναι ένα *ανάγωγο* (irreducible) μονικό πολυώνυμο (βλ. ορισμό 2.16) βαθμού t που καλείται Goppa πολυώνυμο. Επιπλέον, έστω $\boldsymbol{L} = (\gamma_0, \cdots, \gamma_{n-1}) \in \mathbb{F}_{2^m}^n$ είναι ένα σύνολο n διαφορετικών στοιχείων, έτσι ώστε: $g(\gamma_i) \neq 0, \forall 0 \leq i < n$. Δηλαδή, κανένα από τα στοιχεία του συνόλου L, δεν αποτελεί ρίζα του πολυωνύμου $g(X)$. Τα παραπάνω ορίζουν τον $C(\boldsymbol{L}, g(X))$ δυαδικό ανάγωγο Goppa κώδικα. Για κάθε διάνυσμα $\boldsymbol{c} = (c_0, \cdots, c_{n-1})^t \in \mathbb{F}_2^n$, το σύνδρομο ορίζεται ως εξής

$$S_c(X) = -\sum_{i=0}^{n-1} \frac{\boldsymbol{c}_i}{g(\gamma_i)} \frac{g(X) - g(\gamma_i)}{X - \gamma_i} \bmod g(X). \tag{4.2}$$

Εφόσον το σύνολο όλων των διανυσμάτων $\boldsymbol{c} = (c_0, \cdots, c_{n-1})^t \in \mathbb{F}_2^n$, αποτελούν κωδικές λέξεις του $C(L, g(X))$ δυαδικού ανάγωγου Goppa κώδικα, θα ισχύει ότι:

$$S_c(X) = 0 \tag{4.3}$$

ή ισοδύναμα

$$S_c(X) \equiv \sum_{i=0}^{n-1} \frac{\boldsymbol{c}_i}{X - \gamma i} \equiv 0 \bmod g(X) \tag{4.4}$$

βάσει της ιδιότητας: $[(a(x) \bmod f(x)][b(x) \bmod f(x)] \bmod f(x) = a(x)b(x) \bmod f(x)$. Συνεπώς, ο κώδικας C εκφράζεται ως

$$\begin{aligned} C(\boldsymbol{L}, g(X)) &= \left\{ \boldsymbol{c} \in \mathbb{F}_2^n | S_{\boldsymbol{c}}(X) = 0 \right\} \\ &= \left\{ \boldsymbol{c} \in \mathbb{F}_2^n | S_{\boldsymbol{c}}(X) \equiv 0 \bmod g(X) \right\}. \end{aligned}$$

4.2 Κατασκευή του πίνακα ελέγχου ισοτιμίας

Για να κατασκευάσουμε τον πίνακα ελέγχου ισοτιμίας για τον κώδικα C, θεωρούμε την ακόλουθη σχέση:

$$\frac{g(X) - g(\gamma_i)}{X - \gamma_i} \overset{(4.1)}{=} \sum_{j=0}^{t} g_j \frac{X^j - \gamma_i^j}{X - \gamma_i} = \sum_{j=0}^{t} g_i \sum_{s=0}^{j-1} X^s (\gamma^{i-1-s}) = \sum_{s=0}^{t-1} X^s \sum_{j=s+1}^{t} g_j \gamma_i^{j-1-s}, \forall 0 \leq i < n. \tag{4.5}$$

Αντικαθιστώντας λοιπόν, την (4.5) στην (4.2) και συνδυάζοντας την (4.3) προκύπτει ότι

$$\sum_{i=0}^{n-1} \left(\frac{1}{g(\gamma_i)} \sum_{j=s+1}^{t} g_j \gamma_i^{j-1-s} \right) \boldsymbol{c}_i = 0, \qquad \forall s = 0, \cdots, t-1.$$

Κατ' επέκταση, εφόσον $\boldsymbol{H} \cdot \boldsymbol{c} = \boldsymbol{0}$, ο πίνακας ελέγχου ισοτιμίας $\boldsymbol{H}$ μπορεί να γραφεί στην ακόλουθη μορφή

$$\boldsymbol{H} = \begin{bmatrix} g_t g(\gamma_0)^{-1} & \cdots & g_t g(\gamma_{n-1})^{-1} \\ (g_{t-1} + g_t \gamma_0) g(\gamma_0)^{-1} & \cdots & (g_{t-1} + g_t \gamma_{n-1}) g(\gamma_{n-1})^{-1} \\ \vdots & \ddots & \vdots \\ (\sum_{j=1}^{t} g_j \gamma_0^{j-1}) g(\gamma_0)^{-1} & \cdots & (\sum_{j=1}^{t} g_j \gamma_{n-1}^{j-1}) g(\gamma_{n-1})^{-1} \end{bmatrix} = \boldsymbol{XYZ}$$

όπου $\boldsymbol{X}$ ένας τετραγωνικός $t \times t$ πίνακας, που δίνεται από

$$\boldsymbol{X} = \begin{bmatrix} g_t & 0 & 0 & \cdots & 0 \\ g_{t-1} & g_t & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_1 & g_2 & g_3 & \cdots & g_t \end{bmatrix}$$

$\boldsymbol{Y}$ ένας $t \times n$ Vandermonde πίνακας (βλ. ορισμό 2.14) της μορφής

$$\boldsymbol{Y} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \gamma_0 & \gamma_1 & \cdots & \gamma_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_0^{t-1} & \gamma_1^{t-1} & \cdots & \gamma_{n-1}^{t-1} \end{bmatrix}$$

και $\boldsymbol{Z}$ ένας διαγώνιος $n \times n$ πίνακας της μορφής

$$\boldsymbol{Z} = \begin{bmatrix} \frac{1}{g(\gamma_0)} & & & \\ & \frac{1}{g(\gamma_1)} & & \\ & & \ddots & \\ & & & \frac{1}{g_{\gamma(n-1)}} \end{bmatrix}.$$

Όπως παρατηρούμε τα στοιχεία του πίνακα $\boldsymbol{H}$ ανήκουν στο σώμα $\mathbb{F}_2^m$. Αν ερμηνεύσουμε το σώμα $\mathbb{F}_2^m$ σαν έναν m-διάστατο διανυσματικό χώρο στο δυαδικό σώμα $\mathbb{F}_2$ (βλ. [14] ενότητα 2.4), μπορούμε να πούμε ότι ο πίνακας $\boldsymbol{H} \in \mathbb{F}_2^{mt \times n}$.

Οι γραμμές του πίνακα $\boldsymbol{H}$ παράγουν ένα διανυσματικό χώρο V, υποχώρο του $\mathbb{F}_2^n$. Από την σχέση $\boldsymbol{H} \cdot \boldsymbol{c} = \boldsymbol{0}$, προκύπτει ότι ο κώδικας C είναι ένας διανυσματικός χώρος, δυϊκός στον V. Κατ' επέκταση ο πίνακας γεννήτορας $\boldsymbol{G}$, προκύπτει υπολογίζοντας την βάση του διανυσματικού αυτού χώρου, καθώς οι γραμμές του αποτελούν τα διανύσματα βάσης του κώδικα C (βλ. ορισμό 2.12). Συνεπώς, προκύπτει ότι ο (n, k) Goppa κώδικας, θα έχει διάσταση $k \geq n - mt$, καθώς ο $\boldsymbol{G} \in \mathbb{F}_2^{k \times n}$ και ο $\boldsymbol{H} \in \mathbb{F}_2^{mt \times n}$ και επιπλέον ο πίνακας $\boldsymbol{H}$ είναι πλήρους τάξης γραμμών (βλ. ορισμό 2.13) δηλαδή, $mt = n - k$.

4.3 Υπολογισμός της ελάχιστης απόστασης

Έστω $\boldsymbol{c} = (c_0, \cdots, c_{n-1})^t \in C$ και $T_c = \mathsf{supp}(\boldsymbol{c}) = \{i : \boldsymbol{c}_i = 1\}$, όπου με $\mathsf{supp}(\boldsymbol{c})$ συμβολίζουμε το σύνολο των δεικτών που αντιστοιχούν στις μη-μηδενικές συνιστώσες του διανύσματος $\boldsymbol{c} \in \mathbb{F}_2^n$. Ορίζουμε

$$\sigma_c(X) = \prod_{j \in Tc} (X - \gamma_j) \in \mathbb{F}_2^m[X]. \tag{4.6}$$

Στην συνέχεια υπολογίζουμε την παράγωγο του $\sigma_c(X)$, από όπου προκύπτει ότι:

$$\sigma'_{\boldsymbol{c}}(X) = \sum_{i \in T_c} \prod_{j \in T_c \setminus \{i\}} (X - \gamma_j). \tag{4.7}$$

Συνδυάζοντας λοιπόν, τις (4.4), (4.6) και (4.7) προκύπτει:

$$\sigma_{\boldsymbol{c}}(X) S_{\boldsymbol{c}}(X) \equiv \sigma'_{\boldsymbol{c}}(X) \mod g(X). \tag{4.8}$$

Εφόσον $g(\gamma_i) \neq 0, \forall 0 \leq i < n$, θα έχουμε $\gcd(\sigma_{\boldsymbol{c}}(X), g(X)) = 1$. Κατ' επέκταση το $\sigma_{\boldsymbol{c}}(X)$, είναι αντιστρέψιμο modulo g(X) (βλ. ορισμό 2.17), από όπου προκύπτει

$$\frac{\sigma'_{\boldsymbol{c}}(X)}{\sigma_{\boldsymbol{c}}(X)} \equiv S_{\boldsymbol{c}}(X) \mod g(X)$$

και σύμφωνα με την (4.3) καταλήγουμε στη σχέση

$$\sigma'_{\boldsymbol{c}}(X) \equiv 0 \mod g(X), \qquad \forall \boldsymbol{c} \in \mathbb{F}_2^n. \tag{4.9}$$

Χρησιμοποιώντας ωστόσο, τον Frobenius αυτομορφισμό (βλ. ορισμό 2.9) στο σώμα $\mathbb{F}_2^m$ και με βάση την παρακάτω ιδιότητα μεταξύ πολυωνύμων στο $\mathbb{F}_2$ (βλ. [14] ενότητα 2.3):

$$f(X) = \sum_{i=0}^{n} f_i X^i \mapsto (f(X))^2 = \sum_{i=0}^{n} f_i^2 X^{2i}$$

προκύπτει ότι το σύνολο των πολυωνύμων που περιέχονται στον πολυωνυμικό δακτύλιο $F_{2^m}[X^2]$, είναι *τέλεια τετράγωνα* (perfect squares) $F_{2^m}[X]$. Συνεπώς, αναδιατυπώνοντας τις (4.6) και (4.7) ως

$$\sigma_{\boldsymbol{c}}(X) = \sum_{i=1}^{n} \sigma_i X^i \quad \text{και} \quad \sigma'_{\boldsymbol{c}}(X) = \sum_{i=1}^{n} i\sigma_i X^{i-1}$$

αντίστοιχα, παρατηρούμε ότι το $\sigma'_{\boldsymbol{c}}(X)$ είναι πολυώνυμο τέλειου τετραγώνου, καθώς $i\sigma_i X^{i-1} = 0, \forall i$ άρτιο. Επιπλέον, η (4.8) μπορεί να γραφεί ως εξής:

$$\sigma'_{\boldsymbol{c}}(X) \equiv 0 \mod g^2(X), \qquad \forall \boldsymbol{c} \in \mathbb{F}_2^n.$$

Άρα, $\forall \boldsymbol{c} \in C \setminus \{0\}$, έχουμε ότι:

$$\mathsf{wt}(\boldsymbol{c}) = \deg(\sigma_{\boldsymbol{c}}) \geq 1 + \deg(\sigma'_{\boldsymbol{c}}) \geq 2\deg(g) + 1.$$

Συμπεραίνουμε λοιπόν, ότι η ελάχιστη απόσταση ενός Goppa κώδικα C, που παράγεται από ένα ανάγωγο πολυώνυμο βαθμού t, είναι τουλάχιστον $2t + 1$.

4.4 Διόρθωση σφαλμάτων

Εφόσον λοιπόν, η ελάχιστη απόσταση ενός Goppa κώδικα C είναι τουλάχιστον $2t + 1$, θα περιγράψουμε έναν αλγόριθμο διόρθωσης το πολύ t σφαλμάτων. Υποθέτουμε λοιπόν ότι, το $\boldsymbol{m} \in C$ και ότι το $\boldsymbol{e} \in \mathbb{F}_2^n$ αποτελεί ένα διάνυσμα σφάλματος, βάρους $\mathsf{wt}(\boldsymbol{e}) = t$. Συνεπώς

$$\boldsymbol{c} = \boldsymbol{m} + \boldsymbol{e}.$$

Δοθέντος του λαμβανόμενου διανύσματος $\boldsymbol{c}$, θα υπολογίσουμε τα $\boldsymbol{m}$, $\boldsymbol{e}$. Υπολογίζουμε αρχικά, την σχέση μεταξύ των συνδρόμων. Εφόσον, το $\boldsymbol{m}$ αποτελεί κωδική λέξη του κώδικα C, προκύπτει ότι το σύνδρομο $S_{\boldsymbol{m}}(X) \equiv 0 \mod g(X)$ και κατ' επέκταση:

$$S_{\boldsymbol{c}}(X) \equiv S_{\boldsymbol{e}}(X) \mod g(X).$$

Στην συνέχεια, υπολογίζουμε το *πολυώνυμο εντοπισμού σφαλμάτων* (error locator polynomial) $\sigma_{\boldsymbol{e}}(X)$. Συγκεκριμένα, για $T_e = \mathsf{supp}(\boldsymbol{e})$, θα έχουμε:

$$\sigma_{\boldsymbol{e}}(X) = \prod_{j \in T_e} (X - \gamma_j) \in \mathbb{F}_2^m(X)$$

όπου με βάση την (4.8), προκύπτει ότι:

$$\sigma_{\boldsymbol{e}}(X)S_{\boldsymbol{e}}(X) \equiv \sigma'_{\boldsymbol{e}}(X) \mod (X). \tag{4.10}$$

Χωρίζουμε το $\sigma_{\boldsymbol{e}}(X)$ σε πολυώνυμα τετραγώνου και περιττού βαθμού ως εξής

$$\sigma_{\boldsymbol{e}}(X) = a^2(X) + X\beta^2(X).$$

Κατ' επέκταση η παράγωγος του $\sigma e(X)$, αναλύεται ως εξής:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}X}\sigma_{\boldsymbol{e}}(X) &= \frac{\mathrm{d}}{\mathrm{d}X}a^2(X) + \frac{\mathrm{d}X}{\mathrm{d}X}\beta^2(X) + X\frac{\mathrm{d}}{\mathrm{d}X}\beta^2(X)\\ &= 2\frac{\mathrm{d}}{\mathrm{d}X}a(X) + \beta^2(X) + 2X\frac{\mathrm{d}}{\mathrm{d}X}\beta(X)\\ &= \beta^2(X)\end{aligned}$$

Εφόσον λοιπόν, $\sigma'_{\boldsymbol{e}}(X) = \beta^2(X)$ η (4.10) αναδιατυπώνεται ως εξής

$$\beta^2(X)(XS_{\boldsymbol{e}}(X) + 1) \equiv a^2(X)S_{\boldsymbol{e}}(X) \mod g(X). \tag{4.11}$$

Υποθέτουμε ότι, το διάνυσμα σφάλματος $\boldsymbol{e}$ δεν αποτελεί κωδική λέξη του κώδικα C και κατ' επέκταση το σύνδρομο $S_{\boldsymbol{e}}(X) \neq 0 \mod g(X)$. Συνεπώς, υπάρχει πολυώνυμο $T(X)$ έτσι ώστε: $T(X) = S_{\boldsymbol{e}}^{-1}(X)$. Επομένως, η (4.11) γράφεται ως

$$\beta^2(X)(X + T(X)) \equiv a^2(X) \mod g(X). \tag{4.12}$$

Εφόσον λοιπόν η χαρακτηριστική είναι 2, τότε κάθε στοιχείο που ανήκει στο σώμα $\mathbb{F}_2^{mt}$ θα έχει μοναδική τετραγωνική ρίζα (βλ. [14] ενότητα 2.2). Έστω λοιπόν, $\tau(X) \in \mathbb{F}_2^m[X]$ η τετραγωνική ρίζα του πολυωνύμου $T(X)+X$, που ικανοποιεί

$$\tau(X)\tau(X) \equiv T(X) + X \mod g(X).$$

Συνεπώς, η (4.12) αναδιατυπώνεται ως εξής

$$\beta(X)\tau(X) \equiv a(X) \mod g(X) \Leftrightarrow a(X) + \beta(X)\tau(X) \equiv 0 \mod g(X). \tag{4.13}$$

Για την επίλυση της (4.13) μπορεί να χρησιμοποιηθεί ο *επεκτεταμένος αλγόριθμος του Ευκλείδη* (Extended Euclidean Algorithm) (βλ. [5], κεφ. 12, ενότητα 8). Από υπόθεση $\deg(\sigma_e) \leq t$, θα δείξουμε ότι $\deg(a) \leq \lfloor t/2 \rfloor$ και $\deg(\beta) \leq \lfloor (t-1)/2 \rfloor$. Εκτελώντας την 1η επανάληψη του αλγορίθμου για $i = 1$, προκύπτει ότι:

$$r_1(X) = r_{-1}(X) - q_1(X)r_0(X) \Leftrightarrow g(X) - q_1(X)a_0(X) \tag{4.14}$$

$$\beta_1(X) = \beta_{-1}(X) + q_1(X)\beta_0(X) \Leftrightarrow \beta_1(X) = q_1(X). \tag{4.15}$$

Συνεπώς, η (4.14) γράφεται ως εξής

$$r_1(X) = g(X) - \beta_1(X)a_0(X) \tag{4.16}$$

Αλγ. 4.1 Ο επεκτεταμένος αλγόριθμος του Ευκλείδη

είσοδος: $g(X), \tau(X)$

αρχικοποίηση: $i = 0$

1: $r_{-1}(X) = a_{-1}(X) = g(X)$
2: $r_0(X) = a_0(X) = \tau(X)$
3: $\beta_{-1}(X) = 0,\ \beta_0(X) = 1$
4: **repeat**
5: $\quad i = i + 1$
6: $\quad r_i(X) = r_{i-2}(X) - q_i(X) r_{i-1}(X)$ » $\deg(r_i) < \deg(r_{i-1})$
7: $\quad \beta_i(X) = \beta_{i-2}(X) - q_i(X)\beta_{i-1}(X)$
8: $\quad a_i(X) = r_i(X)$
9: **until** $\deg(r_i) < \lfloor (t+1)/2 \rfloor$

έξοδος: $\sigma_e(X) = c^2((a_i(X))^2 + X(\beta_i(X))^2), c \in \mathbb{F}_2^m$ » $\sigma_e(X)$ μονικό πολυώνυμο

και λόγω ότι $a_1(X) = r_1(X)$, η (4.16) αναδιατυπώνεται ως

$$a_1(X) = g(X) - \beta_1(X) a_0(X). \tag{4.17}$$

Συνεπώς, από την (4.17) προκύπτει

$$\deg(g) = \deg(\beta_1) + \deg(a_0)$$

και

$$\deg(\beta_1) = \deg(g) - \deg(a_0). \tag{4.18}$$

Στην γενική περίπτωση, σε κάποιο k βήμα του αλγορίθμου η (4.18) γράφεται ως εξής

$$\deg(\beta_k) = \deg(g) - \deg(a_{k-1}).$$

Παρατηρούμε λοιπόν, ότι σε κάθε βήμα του αλγορίθμου όσο αυξάνεται ο βαθμός του $\beta(X)$, μειώνεται ο βαθμός του $a(X)$. Συνεπώς, υπάρχει ένα μοναδικό σημείο κατά τον υπολογισμό του αλγορίθμου, όπου τα δύο πολυώνυμα $a(X)$ και $\beta(X)$, βρίσκονται κάτω από το αντίστοιχο όριο. Ακριβέστερα, εκτελούμε τον αλγόριθμο, μέχρις ότου το $\deg(a)$ να πέσει κάτω από $\lfloor (t+1)/2 \rfloor$ για πρώτη φορά, δηλαδή

$$\deg(a_k) \leq \lfloor (t+1)/2 \rfloor - 1 \leq \lfloor t/2 \rfloor$$

και επιπλέον,

$$\deg(\beta_k) = \deg(g) - \deg(a_{k-1}) \leq t - \lfloor (t+1)/2 \rfloor = \lfloor (t-1)/2 \rfloor.$$

Τελικά, μέσα από τον προσδιορισμό των ριζών του $\sigma_e(X)$, καθορίζουμε τα $\boldsymbol{m}, \boldsymbol{e}$. Αναλυτικότερα:

Αλγ. 4.2 Ο αλγόριθμος διόρθωσης σφαλμάτων ενός Goppa κώδικα

είσοδος: C, $\boldsymbol{c}$ » C δυαδικός ανάγωγος Goppa κώδικας και c = m + e

1: $S_c(X) = \sum_{i=0}^{n-1} \frac{c_i}{X-\gamma_i} \mod g(X)$ » Υπολογισμός του συνδρόμου του $\boldsymbol{c}$
2: **if** $S_c(X) \equiv 0 \mod g(X)$ **then**
3: **return** $(\boldsymbol{c}, 0)$ » Δεν υπάρχει σφάλμα, το $c \in C$
4: **else**
5: $T(X) \equiv S_c^{-1}(X) \mod g(X)$
6: $\tau(X) \equiv \sqrt{T(X) + X} \mod g(X)$ » Υπάρχουν σφάλματα, το $\boldsymbol{c} \notin C$
7: **end**
8: $\sigma_e(X) = ExtendedEuclidean(i, r_i(X), \beta_i(X))$
9: **for** $i = 0$ to $n - 1$ **do** » Προσδιορισμός των ριζών του $\sigma_e(X)$
10: **if** $\sigma(\gamma_i) = 0$ **then**
11: $e_i = 1$
12: **else**
13: $e_i = 0$
14: **end**
15: **end**
16: $\boldsymbol{m} = \boldsymbol{c} + \boldsymbol{e}$

έξοδος: διανύσματα $\boldsymbol{e}$, $\boldsymbol{m}$

Στην συνέχεια, θα αναλύσουμε τον χρόνο εκτέλεσης του αλγορίθμου 4.2. Συγκεκριμένα, για τον υπολογισμό του $S_c(X)$ με βάση τον πίνακα ελέγχου ισοτιμίας $\boldsymbol{H}$, απαιτούνται $(n-k)n$ δυαδικές πράξεις. Επιπλέον, για τον υπολογισμό του $T(X)$ χρησιμοποιώντας τον αλγόριθμο 4.1, απαιτούνται $O(t^2m^2)$ δυαδικές πράξεις, καθώς οι υπολογισμοί είναι modulo $g(X)$ ενός πολυωνύμου βαθμού t και συντελεστών μήκους m. Τέλος, οι πράξεις που απαιτούνται κατά το τελευταίο βήμα το αλγορίθμου, για την εύρεση όλων των ριζών του πολυωνύμου εντοπισμού σφαλμάτων, είναι $n(tm^2 + tm)$. Κατ' επέκταση οι συνολικές δυαδικές πράξεις που απαιτούνται κατά την εκτέλεση του αλγορίθμου είναι $O(n \cdot t \cdot m^2)$.

ΚΕΦΑΛΑΙΟ 5

Κρυπτοσυστήματα δημοσίου κλειδιού βασισμένα σε κώδικες

Το 1978, ο McEliece [9] πρότεινε ένα *κρυπτοσύστημα δημοσίου κλειδιού* (McEliece public-key cryptosystem), το οποίο βασίζεται στην αλγεβρική θεωρία κωδίκων και αποτέλεσε το πρώτο σύστημα που χρησιμοποίησε κώδικες διόρθωσης σφαλμάτων. Η ιδέα αυτού του κρυπτοσυστήματος, βασίζεται στο γεγονός ότι το πρόβλημα αποκωδικοποίησης ενός τυχαίου γραμμικού κώδικα, ανήκει στην κλάση των NP-δύσκολων προβλημάτων [8]. Πρόκειται για ένα σύστημα απλό και ιδιαίτερα γρήγορο, όσον αφορά τη διαδικασία της κρυπτογράφησης και της αποκρυπτογράφησης. Ακόμα και σήμερα δεν έχει βρεθεί επίθεση πολυωνυμικού χρόνου, να σπάει το κρυπτοσύστημα του McEliece, στην αρχική του μορφή. Παρόλα αυτά, δεν χρησιμοποιείται ευρέως, καθώς ο ρυθμός πληροφορίας που διαθέτει είναι χαμηλός (κοντά στο 0.5) και απαιτεί μεγάλους δυαδικούς πίνακες, ως μυστικό και ιδιωτικό κλειδί αντίστοιχα. Στην αρχική του περιγραφή χρησιμοποιεί Goppa κώδικες [47], αν και είναι δυνατή η χρήση και άλλων οικογενειών κωδίκων. Ενδεχομένως όμως, στην περίπτωση αυτή να μην διασφαλίζεται η επιθυμητή ασφάλεια του συστήματος [45]. Ο πιο αποδοτικός αλγόριθμος αποκωδικοποίησης σφαλμάτων, όταν στο κρυπτοσύστημα χρησιμοποιούνται Goppa κώδικες, είναι ο αλγόριθμος του Patterson [6].

5.1 Περιγραφή του κρυπτοσυστήματος McEliece

Έστω C ένας (n,k) δυαδικός ανάγωγος Goppa κώδικας. Το κρυπτοσύστημα McEliece κάνει χρήση των παραμέτρων $n,t \in \mathbb{N}$, όπου $t \ll n$, οι οποίες καθορίζουν το μήκος του χρησιμοποιούμενου κώδικα και τη διορθωτική του ικανότητα, αντίστοιχα. Η ελάχιστη απόσταση του κώδικα είναι $\mathsf{d}_{min} \geq 2t+1$.

Δεδομένων των παραμέτρων $n,t \in \mathbb{N}$, παράγονται οι παρακάτω πίνακες οι οποίοι χρησιμοποιούνται για την κατασκευή, τόσο του δημοσίου όσο και του ιδιωτικού κλειδιού του κρυπτοσυστήματος. Πιο συγκεκριμένα:

- Ο πίνακας $\boldsymbol{G}$, ο οποίος είναι ένας $k \times n$ τυχαίος πίνακας γεννήτορας (στην συστηματική του μορφή) για τον κώδικα C.

- Ο πίνακας $\boldsymbol{S}$, ο οποίος είναι ένας $k \times k$ τυχαίος αντιστρέψιμος πίνακας στο $\mathbb{F}_2^k$.

- Και ο πίνακας $\boldsymbol{P}$, ο οποίος είναι ένας $n \times n$ τυχαίος πίνακας αντιμετάθεσης με $p_{ij} \in \mathbb{F}_2$, όπου σε κάθε γραμμή ή σε κάθε στήλη του πίνακα, υπάρχει μια μόνο μονάδα.

Βάσει των παραπάνω, το δημόσιο κλειδί του κρυπτοσυστήματος, υπολογίζεται από τον $k \times n$ πίνακα $\boldsymbol{G}_{pub} = \boldsymbol{SGP}$.

Επιπλέον, το ιδιωτικό κλειδί του κρυπτοσυστήματος είναι το $(\boldsymbol{S}, D_C, \boldsymbol{P})$, όπου D_C είναι ένας αποτελεσματικός αλγόριθμος αποκωδικοποίησης για τον κώδικα C.

Παρατηρούμε ότι, προκειμένου να αποκρυφτεί η πραγματική δομή του κώδικα C, εισάγεται ένα trapdoor information (βλ. ορισμό 2.2) που αποτελείται από τους πίνακες: $(\boldsymbol{S}, \boldsymbol{G}, \boldsymbol{P})$, παράγοντας με αυτόν τον τρόπο έναν φαινομενικά τυχαίο κώδικα, ισοδύναμο του αρχικού. Ακριβέστερα, μέσω ενός κρυφού τέτοιου κώδικα, παρέχεται στον παραλήπτη η δυνατότητα ανάκτησης του *απλού ή αρχικού κειμένου* (plaintext), από τα *κρυπτοκείμενα* (ciphertexts), τα οποία αλλοιώνονται σκοπίμως από τον αποστολέα, με την προσθήκη τυχαίων σφαλμάτων, κατά την διαδικασία της κρυπτογράφησης του απλού κειμένου, με σκοπό την παρεμπόδιση ενός ενδεχόμενου κρυπταναλυτή. Συγκεκριμένα, ο αποστολέας εκτελεί την διαδικασία της κρυπτογράφησης ως εξής:

Κρυπτογράφηση: Για να κρυπτογραφηθεί κάποιο αρχικό κείμενο $\boldsymbol{m} \in \mathbb{F}_2^k$, επιλέγεται τυχαία ένα διάνυσμα σφάλματος $\boldsymbol{e} \in \mathbb{F}_2^n$, βάρους $\mathrm{wt}(\boldsymbol{e}) = t$. Το κρυπτοκείμενο $\boldsymbol{c} \in C$, υπολογίζεται ως εξής:

$$\boldsymbol{c} = \boldsymbol{m}\boldsymbol{G}_{pub} + \boldsymbol{e}. \tag{5.1}$$

Όσο μεγαλύτερο είναι το βάρος t, τόσο δυσκολότερη καθίσταται η διαδικασία της κρυπτανάλυσης, από έναν ενδεχόμενο κρυπταναλυτή. Επιπλέον στην περίπτωση κατά την οποία, δεν υπήρχε το διάνυσμα σφάλματος $\boldsymbol{e}$, ο κρυπταναλυτής θα μπορούσε εύκολα, με την βοήθεια της απαλοιφής του Gauss και γνωρίζοντας το δημόσιο κλειδί $\boldsymbol{G}_{pub}$, να ανακτήσει το αρχικό μήνυμα $\boldsymbol{m}$. Ο παραλήπτης μπορεί να ανακτήσει αποτελεσματικά, το μήνυμα $\boldsymbol{m}$ από το κρυπτοκείμενο $\boldsymbol{c}$, χρησιμοποιώντας τον ακόλουθο αλγόριθμο αποκρυπτογράφησης:

Αποκρυπτογράφηση: Ο παραλήπτης υπολογίζει πρώτα το $\boldsymbol{cP}^{-1}$, όπου $\boldsymbol{P}^{-1}$ είναι ο αντίστροφος πίνακας του πίνακα μεταθέσεων $\boldsymbol{P}$ ως εξής

$$\begin{aligned}\widetilde{\boldsymbol{c}} = \boldsymbol{cP}^{-1} &= \left(\boldsymbol{mG}_{\boldsymbol{pub}} + \boldsymbol{e}\right)\boldsymbol{P}^{-1} \\ &= \boldsymbol{mG}_{\boldsymbol{pub}}\boldsymbol{P}^{-1} + \boldsymbol{eP}^{-1} = \boldsymbol{mSGPP}^{-1} + \boldsymbol{eP}^{-1} = \boldsymbol{mSG} + \boldsymbol{eP}^{-1}\end{aligned}$$

όπου $\widetilde{\boldsymbol{m}} = \boldsymbol{mS}$ και $\widetilde{\boldsymbol{e}} = \boldsymbol{eP}^{-1}$, από όπου τελικά προκύπτει

$$\widetilde{\boldsymbol{c}} = \widetilde{\boldsymbol{m}}\boldsymbol{G} + \widetilde{\boldsymbol{e}}.$$

Επειδή το βάρος του $\widetilde{\boldsymbol{e}}$ είναι ίδιο με το βάρος του $\boldsymbol{e}$, ο παραλήπτης χρησιμοποιεί τον αλγόριθμο αποκωδικοποίησης του αρχικού κώδικα C:

$$D_C\left(\widetilde{\boldsymbol{m}}\boldsymbol{G} + \widetilde{\boldsymbol{e}}\right) = \widetilde{\boldsymbol{m}}\boldsymbol{G}, \quad \forall \boldsymbol{e} \in \mathbb{F}_2^n \quad \text{και} \quad \forall \boldsymbol{m} \in \mathbb{F}_2^k$$

προκειμένου να ανακτήσει το μήνυμα $\widetilde{\boldsymbol{m}} = \boldsymbol{mS}$. Τελικά, ανακτά το αρχικό μήνυμα, υπολογίζοντας το $\boldsymbol{m} = \widetilde{\boldsymbol{m}}\boldsymbol{S}^{-1}$.

5.2 Περιγραφή του κρυπτοσυστήματος Niederreiter

Το κρυπτοσύστημα δημοσίου κλειδιού Niederreiter, αποτελεί ένα *κρυπτοσύστημα σακιδίου* (knapsack - type cryptosystem) (βλ. [16] ενότητα 2) το οποίο βασίζεται επίσης σε Goppa κώδικες. Έστω C ένας (n,k) δυαδικός ανάγωγος Goppa κώδικας. Το κρυπτοσύστημα Niederreiter κάνει χρήση των παραμέτρων $n,t \in \mathbb{N}$, όπου $t \ll n$, οι οποίες καθορίζουν το μήκος του χρησιμοποιούμενου κώδικα και τη διορθωτική του ικανότητα, αντίστοιχα. Η ελάχιστη απόσταση του κώδικα είναι η $\mathsf{d}_{min} \geq 2t+1$.

Δεδομένων των παραμέτρων $n,t \in \mathbb{N}$, παράγονται οι παρακάτω πίνακες οι οποίοι χρησιμοποιούνται για την κατασκευή, τόσο του δημοσίου όσο και του ιδιωτικού κλειδιού του κρυπτοσυστήματος.

- Ο πίνακας $\boldsymbol{H}$, ο οποίος είναι ένας $(n-k)\times n$ τυχαίος πίνακας ελέγχου ισοτιμίας (στην συστηματική του μορφή) για τον κώδικα C.
- Ο πίνακας $\boldsymbol{M}$, ο οποίος είναι ένας $(n-k)\times(n-k)$ τυχαίος αντιστρέψιμος πίνακας στο $\mathbb{F}_2^{n-k}$.
- Και ο πίνακας $\boldsymbol{P}$, ο οποίος είναι ένας $n\times n$ τυχαίος πίνακας αντιμετάθεσης με $p_{ij} \in \mathbb{F}_2$.

Κατά συνέπεια, το δημόσιο κλειδί του κρυπτοσυστήματος, υπολογίζεται από τον $n\times(n-k)$ πίνακα $\boldsymbol{H}_{pub} = \boldsymbol{MHP}$. Επιπλέον, το ιδιωτικό κλειδί του κρυπτοσυστήματος είναι το $(\boldsymbol{P}, D_C, \boldsymbol{M})$, όπου D_C είναι ένας αποτελεσματικός αλγόριθμος αποκωδικοποίησης συνδρόμου, για τον κώδικα C.

Παρομοίως με το κρυπτοσύστημα McEliece, προκειμένου να αποκρυφτεί η πραγματική δομή του κώδικα C, εισάγεται μια καταπακτή που αποτελείται από τους πίνακες: $(\boldsymbol{M}, \boldsymbol{H}, \boldsymbol{P})$, παράγοντας με αυτόν τον τρόπο έναν φαινομενικά τυχαίο κώδικα (ισοδύναμο του αρχικού). Συγκεκριμένα, ο αποστολέας εκτελεί την διαδικασία της κρυπτογράφησης ως εξής:

Κρυπτογράφηση: Ένα αρχικό κείμενο $\boldsymbol{m} \in \mathbb{F}_2^k$, αναπαρίσταται ως ένα διάνυσμα $\boldsymbol{e} \in \mathbb{F}_2^n$, βάρους $\mathrm{wt}(\boldsymbol{e}) = t$ και προκειμένου να κρυπτογραφηθεί, υπολογίζεται το σύνδρομο $\boldsymbol{s}$ από τη σχέση

$$\boldsymbol{s}^t = \boldsymbol{H}_{pub} \cdot \boldsymbol{e}^t.$$

Κατ' επέκταση, ο παραλήπτης μπορεί να ανακτήσει αποτελεσματικά, το μήνυμα $\boldsymbol{m}$ από το σύνδρομο $\boldsymbol{s}$, χρησιμοποιώντας τον ακόλουθο αλγόριθμο αποκρυπτογράφησης:

Αποκρυπτογράφηση: Ο παραλήπτης υπολογίζει:

$$\boldsymbol{M}^{-1}\boldsymbol{s}^t = \boldsymbol{H}\boldsymbol{P}\boldsymbol{e}^t$$

και στην συνέχεια εφαρμόζει τον αλγόριθμο αποκωδικοποίησης συνδρόμου D_C του αρχικού κώδικα C, προκειμένου να ανακτήσει το $\boldsymbol{P}\boldsymbol{e}^t$. Τελικά, ανακτά το αρχικό μήνυμα, υπολογίζοντας το $\boldsymbol{e} = \boldsymbol{P}^{-1}\boldsymbol{P}\boldsymbol{e}$.

Σύμφωνα με τον Yuan Xing Li *et al.* [30], το κρυπτοσύστημα Niederreiter και το κρυπτοσύστημα McEliece είναι ισοδύναμα ως προς την ασφάλειά τους. Συνεπώς, ένας κρυπταναλυτής που μπορεί να σπάσει το ένα, μπορεί ισοδύναμα να σπάσει και το άλλο και αντίστροφα. Τα δύο κρυπτοσυστήματα, βασίζονται στον ίδιο C (n, k) γραμμικό κώδικα διόρθωσης σφαλμάτων. Επιπλέον, δοθέντος του δημοσίου κλειδιού $\boldsymbol{G}_{pub}$, καθώς και της εξίσωσης κρυπτογράφησης (5.1) του κρυπτοσυστήματος McEliece και πολλαπλασιάζοντας με το δημόσιο κλειδί $\boldsymbol{H}_{pub}$ του κρυπτοσυστήματος Niederreiter, προκύπτει

$$\boldsymbol{c}^t\boldsymbol{H}_{pub} = \boldsymbol{m}^t\boldsymbol{G}_{pub}^t\boldsymbol{H}_{pub} + \boldsymbol{e}^t\boldsymbol{H}_{pub} = \boldsymbol{e}^t\boldsymbol{H}_{pub} \equiv \boldsymbol{s}^t$$

καθώς, $\boldsymbol{G}_{pub}^t\boldsymbol{H}_{pub} = \boldsymbol{0}$. Συμπεραίνουμε λοιπόν, ότι η παραπάνω σχέση ταυτίζεται με την εξίσωση κρυπτογράφησης του κρυπτοσυστήματος Niederreiter. Με την χρήση γραμμικών μετασχηματισμών, αποδεικνύεται και το αντίστροφο.

5.3 Ασφάλεια του κρυπτοσυστήματος McEliece

Ο McEliece [9] πρότεινε κάποιες πιθανές επιθέσεις, οι οποίες θα μπορούσαν να πλήξουν την ασφάλεια του κρυπτοσυστήματος. Ακολούθως περιγράφεται, μια πιθανή μορφή επίθεσης.

Ο κρυπταναλυτής θα μπορούσε να επιχειρήσει να ανακτήσει τον πίνακα γεννήτορα $\boldsymbol{G}$ του κώδικα C, γνωρίζοντας το δημόσιο κλειδί $\boldsymbol{G}_{pub}$, ώστε να μπορεί στην συνέχεια να χρησιμοποιήσει, τον αλγόριθμο αποκωδικοποίησης του Patterson. Ωστόσο, οι προοπτικές της επίθεσης αυτής δεν είναι ιδιαίτερα ελπιδοφόρες για τον κρυπταναλυτή, ιδιαίτερα στην περίπτωση κατά την οποία, τόσο το μήκος το κώδικα n όσο η διορθωτική του ικανότητα t, παίρνουν αρκετά μεγάλες τιμές. Καθώς στην περίπτωση αυτή, υπάρχουν πολλοί πιθανοί υποψήφιοι πίνακες: $\boldsymbol{G}, \boldsymbol{S}, \boldsymbol{P}$.

Μια δεύτερη μορφή επίθεσης, θα μπορούσε να αποτελεί η εξής. Να επιχειρήσει ο κρυπταναλυτής να ανακτήσει το αρχικό κείμενο $\boldsymbol{m}$ από το κρυπτοκείμενο $\boldsymbol{c}$, χωρίς να είναι απαραίτητη η εκμάθηση, του πίνακα γεννήτορα $\boldsymbol{G}$. Η επίθεση αυτή φαίνεται πιο ελπιδοφόρα συγκριτικά με την προηγούμενη, ωστόσο το βασικό πρόβλημα που προκύπτει, είναι η αποκωδικοποίηση ενός τυχαίου γραμμικού (n, k) κώδικα, υπό την παρουσία t σφαλμάτων. Καθώς το γενικό πρόβλημα αποκωδικοποίησης γραμμικών κωδίκων, ανήκει στην κλάση προβλημάτων NP -πλήρες (βλ. ορισμό 2.30). Κατ' επέκταση, αν οι παράμετροι του χρησιμοποιούμενου κώδικα παίρνουν αρκετά μεγάλες τιμές, η επίθεση καθίσταται υπολογιστικά ανέφικτη.

Επιπλέον, μια αρκετά ελπιδοφόρα επίθεση θα μπορούσε να αποτελεί η εξής: Να επιλέξει ο κρυπταναλυτής k, από τις n συντεταγμένες του κρυπτοκειμένου

$$\begin{aligned}\boldsymbol{c} &= (c_0, c_1, \cdots, c_{n-1}) \\ &= (\underbrace{c_0, \cdots, c_{k-1}}_{\widehat{c}}, \underbrace{c_k, \cdots, c_{n-1}}_{c'}) \\ &= (\widehat{c}, c')\end{aligned}$$

τυχαία με την ελπίδα ότι στο $\widehat{c}$ δεν περιέχονται σφάλματα και με βάση την υπόθεση αυτή, να υπολογίσει το απλό κείμενο $\boldsymbol{m}$. Σε αυτή την περίπτωση, εφόσον το διάνυσμα σφάλματος $(\widehat{e} = \mathbf{0}_k)$ θα ισχύει το εξής:

$$\widehat{c} = m\widehat{G}. \tag{5.2}$$

Επιπλέον, εφαρμόζοντας την *απαλοιφή του Gauss* (Gaussian elimination), ο πίνακας $\boldsymbol{G}_{pub}$, μπορεί να αναλυθεί με παρόμοιο τρόπο ως εξής

$$\boldsymbol{G}_{pub} = \left[\widehat{G}, G'\right]$$

όπου $\widehat{G}$ και G', είναι $k \times k$ και $k \times n - k$ πίνακες αντίστοιχα.

Συνεπώς, η τάξη του πίνακα $\boldsymbol{G}_{pub}$, θα είναι k και κατ' επέκταση ο κρυπταναλυτής μπορεί με μοναδικό τρόπο να ανακτήσει το αρχικό κείμενο $\boldsymbol{m}$, σύμφωνα με την (5.2) ως εξής:

$$\boldsymbol{m} = \widehat{c}\widehat{G}^{-1}.$$

Ωστόσο, η επίθεση αυτή αποτελεί μια *ανά - μήνυμα επίθεση* (per - message attack), καθώς το ιδιωτικό κλειδί του κρυπτοσυστήματος, παραμένει άγνωστο στον κρυπταναλυτή.

Η πιθανότητα κατά την οποία το διάνυσμα σφάλματος $(\widehat{e} = \mathbf{0}_k)$, υπολογίζεται ως εξής:

$$\mathsf{Pr}(\widehat{e} = \mathbf{0}_k) = \prod_{i=0}^{k=1} \mathsf{Pr}(e_{i=0}) = \left(1 - \frac{t}{n}\right)^k.$$

Καθώς, η πιθανότητα να υπάρχουν σφάλματα κατά την μετάδοση του κρυπτοκειμένου $\boldsymbol{c}$, είναι:

$$q = \mathsf{Pr}(e_i = 1) = \frac{t}{n}. \tag{5.3}$$

Κατ΄επέκταση, ο κρυπταναλυτής εκτελεί κατά μέσο όρο $1/q$ προσπάθειες, προκειμένου να επιτύχει τον στόχο του. Επιπλέον, για κάθε μία από αυτές τις προσπάθειες, ο πίνακας $\widehat{G}_{k\times k}$ πρέπει να είναι αντιστρέψιμος. Κατά συνέπεια, η πολυπλοκότητα που απαιτείται, έως ότου ο κρυπταναλυτής ανακτήσει το αρχικό κείμενο $\boldsymbol{m}$, υπολογίζεται ως εξής:

$$W = b^3 \cdot \left(1 - \frac{t}{n}\right)^{-k}.$$

Λήμμα 5.1. *Το b^3 προκύπτει από τις πράξεις που απαιτούνται, προκειμένου να εκτελεστεί η απαλοιφή του Gauss και να προκύψει ένας αντιστρέψιμος πίνακας.*

Απόδειξη. Έστω ένας πίνακας $x \times n$, όπου $x \leq n$. Αρχικά πραγματοποιούνται x συγκρίσεις, προκειμένου να εξαλειφθούν οι μη-μηδενικές συνιστώσες που περιέχονται σε κάθε στήλη του πίνακα και να παραμείνουν μόνο οι *οδηγοί* (pivots). Ωστόσο, φτάνοντας περίπου στην μέση των γραμμών του πίνακα $(\frac{x}{2})$, ας υποθέσουμε σε κάποιο σημείο i, εξασφαλίζεται ότι για τις προηγούμενες $i-1$ στήλες του πίνακα, έχουν απαλειφθεί οι μη-μηδενικές συνιστώσες και έχουν μείνει μόνο οι οδηγοί. Κατά συνέπεια, δεν είναι απαραίτητο να διατρέξουμε όλο τον πίνακα, παρά μόνο το κομμάτι της διαγωνίου που υπολείπεται. Συνεπώς:

$$\sum_{i=1}^{x} \frac{x}{2} \cdot (n - i - 1) = \frac{1}{2} \cdot x^2 \cdot (n - \frac{x-1}{2}) \approx x^3$$

όπου στην γενική περίπτωση $x = k$ ή $x = n - k$, αναλόγως αν η ανάλυση γίνεται στον πίνακα γεννήτορα, ή στον πίνακα ελέγχου ισοτιμίας, αντίστοιχα. ■

Σύμφωνα με τον McEliece, οι βέλτιστες τιμές των παραμέτρων, για τις οποίες διατηρείται η ασφάλεια του κρυπτοσυστήματος, είναι οι εξής:

- μήκος του κώδικα: $n = 1024 = 2^{10}$
- διορθωτική ικανότητα: $t = 50$

- διάσταση του κώδικα: $k = 1024 - 50 \cdot 10 = 524$

Συγκεκριμένα, για αυτές τις τιμές υπάρχουν 10^{149} πιθανά Goppa πολυώνυμα και ένας ιδιαίτερα μεγάλος αριθμός επιλογών για τους πίνακες $\boldsymbol{S}$ και $\boldsymbol{P}$.

Συνεπώς, στην παραπάνω επίθεση ο κρυπταναλυτής καλείται να μαντέψει σωστά 524 αναλλοίωτες στήλες (διάνυσμα σφάλματος $\boldsymbol{e}_{1\times 524} = \mathbf{0}$) του κρυπτοκειμένου $\boldsymbol{c}$, από τις πιθανές $974 = 1024 - 50$, καθώς υπάρχουν 50 σφάλματα ενσωματωμένα σε 1024 στήλες του κρυπτοκειμένου $\boldsymbol{c}$. Πιο συγκεκριμένα:

$$\binom{524}{1024} / \binom{524}{974} \approx 1.37 \times 10^{16}. \tag{5.4}$$

Απαιτούνται δηλαδή, περίπου 1.37×10^{16} εικασίες, ώστε να εκτελέσει ο κρυπταναλυτής μια επιτυχημένη επίθεση. Επιπλέον, η πολυπλοκότητα της επίθεσης, με βάση τις βέλτιστες παραμέτρους που πρότεινε ο McEliece θα είναι περίπου:

$$W = b^3 \cdot \left(1 - \frac{t}{n}\right)^{-k} = 10^{19} \approx 2^{65} \tag{5.5}$$

5.3.1 Βελτιστοποίηση παραμέτρων

Ωστόσο, το 1989 οι Adams και Melter [23] πρότειναν κάποιες βελτιστοποιήσεις, όσον αφορά τις τιμές των παραμέτρων που εισήγαγε ο McEliece, προκειμένου να δυσχεράνουν ακόμα περισσότερο μια ενδεχόμενη κρυπτανάλυση. Συγκεκριμένα, διαφοροποίησαν τον τρόπο υπολογισμού της πιθανότητας, κατά την οποία το διάνυσμα σφάλματος $\boldsymbol{e} = \mathbf{0}_k$, εμφανίζεται με πιθανότητα

$$\mathsf{Pr}(\boldsymbol{e} = \mathbf{0}_k) = \begin{pmatrix} n-t \\ k \end{pmatrix} / \begin{pmatrix} n \\ k \end{pmatrix}$$

όπου t οι μη-μηδενικές συνιστώσες του διανύσματος σφάλματος $\boldsymbol{e}$ ($\mathsf{wt}(\boldsymbol{e}) = t$) και κατ' επέκταση $n - t$ οι μηδενικές συνιστώσες του $\boldsymbol{e}$.

Έστω ότι το b εκφράζει τα βήματα που απαιτούνται, προκειμένου να εκτελεστεί η απαλοιφή του Gauss και να προκύψει ένας αντιστρέψιμος πίνακας. Κατά συνέπεια, η πολυπλοκότητα που απαιτείται έως ότου ο κρυπταναλυτής ανακτήσει το αρχικό κείμενο $\boldsymbol{m}$, υπολογίζεται ως εξής:

$$W = b \begin{pmatrix} n \\ k \end{pmatrix} / \begin{pmatrix} n-t \\ k \end{pmatrix}. \tag{5.6}$$

Εντούτοις, για $n = 2^i$, οι παράμετροι n, k, t συνδέονται με την παρακάτω σχέση $k \geq 2^i - it$ [47]. Συνεπώς, η μέγιστη τιμή της πολυπλοκότητας σύμφωνα με την (5.6), για $2 < a \leq 3$, προκύπτει όταν t =37. Συγκεκριμένα, για $a = 3$ και $t = 37$

$$W \approx 2^{84.1}$$

ενώ για $a = 3$ και $t = 50$

$$W \approx 2^{80.7}.$$

Επιπλέον, για $t = 37$ αυξάνεται η μικρότερη τιμή του k από 524 που πρότεινε ο McEliece, σε 654 και κατά συνέπεια περιορίζεται η επέκταση των δεδομένων στο κρυπτοσύστημα.

5.4 Επιθέσεις στο κρυπτοσύστημα McEliece

Στο παρελθόν, υπήρξαν πολλοί ερευνητές που επιχείρησαν να σπάσουν το κρυπτοσύστημα McEliece. Ωστόσο, καμία απόπειρα δεν κατέστη επιτυχής, στην γενική περίπτωση. Ανάμεσα σε αυτούς, οι Korzhik και Turkin [25] οι οποίοι ισχυρίστηκαν ότι είχαν σπάσει το κρυπτοσύστημα McEliece. Γεγονός όμως, που δεν έγινε αποδεκτό από τους περισσότερους κρυπτογράφους, λόγω της αδυναμίας τους να επιβεβαιώσουν με εμφανείς αποδείξεις, το χρονικό όριο που ισχυρίζονταν κατά την ανάλυσή τους. Επιπλέον, ο Berson [32] ο οποίος έδειξε ότι το κρυπτοσύστημα McEliece υποφέρει από δύο αδυναμίες:

1. Αδυναμία να προστατευθεί οποιοδήποτε μήνυμα, κρυπτογραφείται παραπάνω από μια φορά.

2. Αδυναμία να προστατευθούν οποιαδήποτε μηνύματα, ικανοποιούν μια γραμμική σχέση μεταξύ τους.

Πράγματι, ένας επιτιθέμενος στο κρυπτοσύστημα, μπορεί με την συμπεριφορά του να προκαλέσει κάποια από τις παραπάνω αδυναμίες. Εισάγοντας για παράδειγμα, κάποια σφάλματα στο κρυπτοκείμενο που αποστέλλεται στον παραλήπτη, έτσι ώστε να μην μπορεί η αποκρυπτογράφηση να εκτελεστεί σωστά. Αυτό έχει ως αποτέλεσμα, να θεωρηθεί από τον παραλήπτη ότι προκλήθηκε κάποιο σφάλμα κατά την διαδικασία της κρυπτογράφησης και να ζητήσει από τον αποστολέα αναμετάδοση του μηνύματος (το μήνυμα κρυπτογραφείται και το κρυπτοκείμενο αποστέλλεται ξανά). Κατ' επέκταση, προκύπτει η πρώτη αδυναμία.

Ωστόσο, λαμβάνοντας υπόψη τις αδυναμίες αυτές, ο Berson δεν κατάφερε να σπάσει εντελώς το κρυπτοσύστημα McEliece, καθώς ακόμα και σε αυτή την περίπτωση, δεν κατέστη εφικτό, να ανακτηθεί το ιδιωτικό κλειδί του κρυπτοσυστήματος. Πιο συγκεκριμένα, ο Berson βασιζόμενος στις παραπάνω αδυναμίες, πρότεινε δύο επιθέσεις στο κρυπτοσύστημα McEliece. Την *επίθεση επανεκπομπής μηνύματος* (message-resend attack) και την *επίθεση σχετιζόμενου μηνύματος* (related-message attack), που αναλύονται στην συνέχεια.

5.4.1 Επίθεση επανεκπομπής μηνύματος

Υποθέτουμε ότι ένα αρχικό κείμενο $\boldsymbol{m}$ κρυπτογραφείται δύο φορές, είτε εξαιτίας κάποιου ατυχήματος που προκλήθηκε κατά την μετάδοση, είτε κάποιας κακόβουλης ενέργειας που εκτελείται από τον κρυπταναλυτή. Συνεπώς, ο κρυπταναλυτής γνωρίζει:

$$\boldsymbol{c}_1 = \boldsymbol{m}\boldsymbol{G}_{pub} + \boldsymbol{e}_1$$
$$\boldsymbol{c}_2 = \boldsymbol{m}\boldsymbol{G}_{pub} + \boldsymbol{e}_2$$

όπου $\boldsymbol{e}_1 = (e_{1,1}, \cdots, e_{1,n})$, $\boldsymbol{e}_2 = (e_{2,1}, \cdots, e_{2,n})$ και επιπλέον $\boldsymbol{e}_1 \neq \boldsymbol{e}_2$.

Η κατάσταση αυτή είναι γνωστή, ως *κατάσταση επανεκπομπής μηνύματος* (message resend condition). Καθίσταται λοιπόν, εύκολο για τον κρυπταναλυτή να ανακτήσει το αρχικό κείμενο $\boldsymbol{m}$, λαμβάνοντας υπόψη το παραπάνω σύστημα εξισώσεων του κρυπτοκειμένου $\boldsymbol{c}_i$. Ωστόσο, εξετάζεται μόνο η περίπτωση κατά την οποία το πλήθος επανεκπομπών είναι δύο, δηλαδή υπάρχουν δύο διαφορετικά κρυπτοκείμενα $\boldsymbol{c}_1, \boldsymbol{c}_2$ που αντιστοιχούν στο ίδιο αρχικό κείμενο $\boldsymbol{m}$. Προφανώς, όσο μεγαλύτερο είναι το πλήθος επανεκπομπών, τόσο ευκολότερη καθίσταται η επίθεση. Παρατηρούμε ότι,

$$\boldsymbol{c}_1 + \boldsymbol{c}_2 = \boldsymbol{e}_1 + \boldsymbol{e}_2. \tag{5.7}$$

Άρα ο κρυπταναλυτής μπορεί εύκολα να ανιχνεύει μια κατάσταση επανεκπομπής μηνύματος, παρατηρώντας το βάρος Hamming του αθροίσματος των δύο κρυπτοκειμένων. Ακριβέστερα, στην περίπτωση που τα καθαρά κείμενα είναι διαφορετικά μεταξύ τους, το αναμενόμενο βάρος του αθροίσματος είναι περίπου $\mathsf{wt}(\boldsymbol{c}_1 + \boldsymbol{c}_2) = \frac{1024}{2} = 512$. Αντιθέτως, στην περίπτωση που ταυτίζονται $\boldsymbol{m}_1 = \boldsymbol{m}_2 = \boldsymbol{m}$, το βάρος του αθροίσματος των κρυπτοκειμένων, είναι

$$\mathsf{wt}(\boldsymbol{c}_1 + \boldsymbol{c}_2) \leq 100. \tag{5.8}$$

Εφόσον το βάρος κάθε διανύσματος σφάλματος, που προστίθεται στο κρυπτοσύστημα McEliece είναι $\mathsf{wt}(\boldsymbol{e}) \leq 50$, προκύπτει ότι

$$\mathsf{wt}(\boldsymbol{e}_1) = 50,$$
$$\mathsf{wt}(\boldsymbol{e}_2) = 50.$$

Συνεπώς, $\mathsf{wt}(\boldsymbol{e}_1 + \boldsymbol{e}_2) \leq 100$ και βάσει της (5.7), προκύπτει η (5.8).

Εφόσον λοιπόν, $\mathsf{wt}(\boldsymbol{e}_1) = \mathsf{wt}(\boldsymbol{e}_2)$ σύμφωνα με την παραπάνω σχέση, προκύπτει ότι $\mathsf{wt}(\boldsymbol{e}_1 + \boldsymbol{e}_2) = \mathsf{wt}(\boldsymbol{e}_1) + \mathsf{wt}(\boldsymbol{e}_2) - 2\,\mathsf{wt}(\boldsymbol{e}_1\boldsymbol{e}_2)$ συνεπώς,

$$\mathsf{wt}(\boldsymbol{e}_1\boldsymbol{e}_2) = 50 - 0.5\,\mathsf{wt}(\boldsymbol{e}_1 + \boldsymbol{e}_2) \Leftrightarrow \mathsf{wt}(\boldsymbol{e}_1\boldsymbol{e}_2) = 50 - 0.5\,\mathsf{wt}(\boldsymbol{c}_1 + \boldsymbol{c}_2) \tag{5.9}$$

όπου $\boldsymbol{e}_1\boldsymbol{e}_2$ είναι ένα διάνυσμα $(e_{1,1}e_{2,1}, \cdots e_{1,n}e_{2,n})$, μήκους n. Δηλαδή, κάθε του συνιστώσα, είναι το γινόμενο των αντίστοιχων συνιστωσών των $\boldsymbol{e}_1$ και $\boldsymbol{e}_2$. Σύμφωνα με την ανάλυση του Berson το αναμενόμενο βάρος Hamming $\mathsf{wt}(\boldsymbol{e}_1 + \boldsymbol{e}_2) = 95.1$.

Σημείωση 5.2. Αν υποθέσουμε ότι, το $\mathsf{wt}(\boldsymbol{e}_1 + \boldsymbol{e}_2) = \mathsf{wt}(\boldsymbol{c}_1 + \boldsymbol{c}_2) = 94$. Σύμφωνα λοιπόν με την (5.9), προκύπτει ότι: $\mathsf{wt}(\boldsymbol{e}_1\boldsymbol{e}_2) = 3$. Ακριβέστερα ο κρυπταναλυτής είναι σε θέση να γνωρίζει ότι από τις 1024 θέσεις του διανύσματος $\boldsymbol{c}_1$, στις $1024 - 94 = 930$ θέσεις, θα περιέχονται μόνο 3 σφάλματα.

Προκειμένου λοιπόν, να μπορέσει ο κρυπταναλυτής να εφαρμόσει επιθέσεις συνόλου πληροφορίας, για λεπτομέρειες (βλ. Κεφ. 6) (όπου πρέπει να βρει 524 θέσεις στις οποίες δεν περιέχονται σφάλματα, προκειμένου να είναι σε θέση να αντιστρέψει το αρχικό κείμενο $\boldsymbol{m} = \boldsymbol{m}_1 = \boldsymbol{m}_2$), πρέπει να διαπιστώσει σε πόσους από τους $\binom{930}{524}$ συνδυασμούς δεν περιέχεται κανένα σφάλμα. Συγκεκριμένα, το πλήθος τους θα είναι ακριβώς $\binom{927}{524}$.

Κατά συνέπεια, η πιθανότητα σε έναν τυχαίο συνδυασμό να μην περιέχονται σφάλματα, υπολογίζεται ως εξής:

$$\mathsf{Pr} = \binom{927}{524} / \binom{930}{524} \approx 0.0828.$$

Συμπεραίνουμε λοιπόν, ότι απαιτούνται περίπου $\frac{1}{\mathsf{Pr}}$ επαναλαμβανόμενες εικασίες προκειμένου να εκτελεστεί μια επιτυχημένη επίθεση. Ακριβέστερα, στο παράδειγμα που προηγήθηκε $\frac{1}{\mathsf{Pr}} = 12.077 \approx 12$ εικασίες. Άρα, απαιτείται πολύ λίγη προσπάθεια από την μεριά του κρυπταναλυτή, προκειμένου να ανακτήσει το αρχικό κείμενο $\boldsymbol{m}$. Επιπλέον παρατηρούμε ότι, η επίθεση αυτή είναι βελτιωμένη κατά $\approx 10^{15}$, συγκριτικά με το αποτελέσμα της επίθεσης που προτάθηκε από τον McEliece (5.4).

Επιπροσθέτως, αν υποθέσουμε ότι $\mathsf{wt}(\boldsymbol{e}_1 + \boldsymbol{e}_2) = 96$, μπορούμε να διαπιστώσουμε ότι, απαιτούνται 5 εικασίες. Συνεπώς, στην γενική περίπτωση προκύπτει ο Αλγόριθμος 5.1.

Αλγ. 5.1 Επίθεση επανεκπομπής μηνύματος

είσοδος: c_1, c_2, G_{pub}

1: $J = \mathrm{supp}(c_1 + c_2)$ » $\mathrm{supp}(x) := \{i \in \{1, \cdots, n\} : x_i = 1\}$
2: $l =\mid J \mid$ » $\mid J \mid$, εκφράζει την πληθικότητα του J
3: **if** $\mathrm{l} > 2\mathrm{t}$ **then**
4: **break**
5: **end**
6: $I = \{1, 2, \cdots, n\} \setminus J$
7: $\mu =\mid I \mid$ » όπου μ είναι η πληθικότητα του I
8: $q = (t - l/2)/(n - l)$ » δείχνει πόσα σφάλματα περιέχονται στο c_1
9: $p = \begin{pmatrix} n - t - l/2 \\ k \end{pmatrix} \cdot \begin{pmatrix} n - l \\ k \end{pmatrix}^{-1}$ » p πιθανότητα επιτυχίας
10: $\widetilde{c} \leftarrow c_1 \mid I \mid$
11: $\widetilde{G} \leftarrow G_{pub}(*, I)$
12: $m = ISD(\widetilde{c}, \widetilde{G}, q, p)$ » εκτελείται έως ότου βρεθούν k θέσεις χωρίς σφάλματα

έξοδος: διάνυσμα m

Με βάση λοιπόν, τον παραπάνω αλγόριθμο, συμπεραίνουμε ότι, δημιουργείται ένας νέος κώδικας. Και συγκεκριμένα

$$\widetilde{c}_{1\times\mu} = m_{1\times k}\widetilde{G}_{k\times\mu} + \widetilde{e}_{1\times\mu}.$$

5.4.2 Επίθεση σχετιζόμενου μηνύματος

Υποθέτουμε ότι κρυπτογραφούνται δύο μηνύματα m_1, m_2 και ότι ο κρυπταναλυτής γνωρίζει μια γραμμική σχέση μεταξύ τους, π.χ (m_1+m_2). Κατ' επέκταση γνωρίζει ότι:

$$c_1 = m_1 G_{pub} + e_1$$
$$c_2 = m_2 G_{pub} + e_2$$

όπου $m_1 \neq m_2$ και $e_1 \neq e_2$. Η κατάσταση αυτή, ονομάζεται *κατάσταση σχετιζόμενου μηνύματος* (related-message condition). Συγκεκριμένα, σε αυτή την μορφή επίθεσης ο κρυπταναλυτής καλείται να ανακτήσει το αρχικό κείμενο m_i, από ένα σύνολο κρυπτοκειμένων c_i, εκτελώντας μια κωδικοποίηση και επανεκτελώντας την πρώτη μορφή επίθεσης. Ακριβέστερα, συνδυάζοντας τις δύο παραπάνω σχέσεις προκύπτει το εξής:

$$c_1 + c_2 = m_1 G_{pub} + e_1 + m_2 G_{pub} + e_2 = (m_1 + m_2) G_{pub} + (e_1 + e_2).$$

Εφόσον, η γραμμική σχέση $m_1 + m_2$ είναι εκ των προτέρων γνωστή στον κρυπταναλυτή, όπως επίσης και ο πίνακας G_{pub} εφόσον αποτελεί το δημόσιο κλειδί του κρυπτοσυστήματος, προκύπτει ότι:

$$c_1 + c_2 + (m_1 + m_2) G_{pub} = (e_1 + e_2). \tag{5.10}$$

Συνεπώς, ο κρυπταναλυτής και σε αυτή την περίπτωση, εκτελεί επίθεση επανεκπομπής μηνύματος, με την μόνη διαφορά ότι αντικαθιστά το αριστερό μέλος της (5.7), με το αριστερό μέλος της (5.10).

Συμπεραίνουμε λοιπόν, ότι η επίθεση επανεκπομπής μηνύματος αποτελεί, ουσιαστικά ειδική περίπτωση της επίθεσης σχετιζόμενου μηνύματος, όπου η γραμμική σχέση μεταξύ των καθαρών κειμένων, είναι $\boldsymbol{m}_1 + \boldsymbol{m}_2 = \mathbf{0}$.

Αλγ. 5.2 Επίθεση σχετιζόμενου μηνύματος

είσοδος: $\boldsymbol{c}_1$, $\boldsymbol{c}_2$, $\boldsymbol{G}_{pub}, \Delta\boldsymbol{m}$

$J = supp(\boldsymbol{c}_1 + \boldsymbol{c}_2 + \Delta\boldsymbol{m}\boldsymbol{G}_{pub})$ » $\Delta\boldsymbol{m} = \boldsymbol{m}_1 + \boldsymbol{m}_2$
$l =\mid J \mid$ » $\mid J \mid$, εκφράζει την πληθικότητα του J
if $l > 2t$ **then**
  break
end
$I = \{1, 2, \cdots, n\} \setminus J$
$\mu =\mid I \mid$
$q = (t - l/2)/(n - l)$ » δείχνει πόσα σφάλματα περιέχονται στο $\boldsymbol{c}_1$
$p = \begin{pmatrix} n-t-l/2 \\ k \end{pmatrix} \cdot \begin{pmatrix} n-l \\ k \end{pmatrix}^{-1}$ » p πιθανότητα επιτυχίας
$\widetilde{\boldsymbol{c}} \leftarrow \boldsymbol{c}_1 \mid I \mid$
$\widetilde{\boldsymbol{G}} \leftarrow \boldsymbol{G}_{pub}(*, I)$
$\boldsymbol{m} = ISD(\widetilde{\boldsymbol{c}}, \widetilde{\boldsymbol{G}}, q, p)$ » εκτελείται έως ότου βρεθούν k θέσεις χωρίς σφάλματα

έξοδος: διάνυσμα $\boldsymbol{m}$

5.4.3 Βελτιστοποιήσεις ασφάλειας στο κρυπτοσύστημα McEliece

Το 1998 ο Sun [34] πρότεινε κάποιες αλλαγές στο κρυπτοσύστημα McEliece, ώστε να μπορεί να προστατευθεί ενάντια στις επιθέσεις που πρότεινε ο Berson. Πρόκειται μάλιστα, για βελτιστοποιήσεις, οι οποίες δεν περιορίζουν τον ρυθμό πληροφορίας του κρυπτοσυστήματος. Επιπλέον, τόσο το δημόσιο, όσο και το ιδιωτικό κλειδί του κρυπτοσυστήματος παραμένουν ίδια, όπως αρχικά προτάθηκαν από τον McEliece.

Παραλλαγή I

Έστω μια μονόδρομη συνάρτηση κατακερματισμού (βλ. ορισμό 2.3), με είσοδο το τυχαίο διάνυσμα σφάλματος $\boldsymbol{e}$ και έξοδο ένα διάνυσμα μήκους k. Ακριβέστερα το κρυπτοκείμενο $\boldsymbol{c}$, κρυπτογραφείται ως εξής:

$$\boldsymbol{c} = (\boldsymbol{m} + h(\boldsymbol{e}))\boldsymbol{G}_{pub} + \boldsymbol{e}.$$

Στην συνέχεια, ο παραλήπτης χρησιμοποιεί τον αλγόριθμο αποκρυπτογράφησης του McEliece στην αρχική του περιγραφή, ώστε να ανακτήσει το αρχικό κείμενο

$\boldsymbol{m}$. Και συγκεκριμένα:

$$\boldsymbol{m} = (\boldsymbol{m} + h(\boldsymbol{e})) + h(\boldsymbol{e}).$$

Συνεπώς, στην περίπτωση αυτή το κρυπτοσύστημα McEliece καθίσταται ασφαλές έναντι των επιθέσεων του Berson, διότι:

α. Αν υποθέσουμε ότι, έχουμε δύο μηνύματα $\boldsymbol{m}_1, \boldsymbol{m}_2$. Αν $\boldsymbol{m}_1 = \boldsymbol{m}_2$, τότε προκύπτει το εξής:

$$\boldsymbol{c}_1 + \boldsymbol{c}_2 = (h(\boldsymbol{e}_1) + h(\boldsymbol{e}_2))\boldsymbol{G}_{pub} + \boldsymbol{e}_1 + \boldsymbol{e}_2.$$

Δεδομένου λοιπόν, ότι $h(\boldsymbol{e}_1), h(\boldsymbol{e}_2)$ είναι άγνωστα, η ποσότητα $(h(\boldsymbol{e}_1)+h(\boldsymbol{e}_2))\boldsymbol{G}_{pub}$ δεν είναι δυνατόν να υπολογιστεί. Κατ' επέκταση δεν μπορεί να προκύψει καμία πληροφορία, σχετικά με τις θέσεις στις οποίες εντοπίζονται τα σφάλματα. Επομένως, η επίθεση επανεκπομπής μηνύματος αποτυγχάνει.

β. Επιπλέον, ακόμα και στην περίπτωση που είναι γνωστή η γραμμική σχέση $\boldsymbol{m}_1 + \boldsymbol{m}_2$, δηλαδή:

$$\boldsymbol{c}_1 + \boldsymbol{c}_2 = (\boldsymbol{m}_1 + \boldsymbol{m}_2 + h(\boldsymbol{e}_1) + h(\boldsymbol{e}_2))\boldsymbol{G}_{pub} + \boldsymbol{e}_1\boldsymbol{e}_2$$

επίσης, λόγω έλλειψης γνώσης των $h(\boldsymbol{e}_1), h(\boldsymbol{e}_2)$, δεν είναι δυνατός ο υπολογισμός της ποσότητας $(\boldsymbol{m}_1 + \boldsymbol{m}_2 + h(\boldsymbol{e}_1) + h(\boldsymbol{e}_2))\boldsymbol{G}_{pub}$. Κατ' επέκταση, αποτυγχάνει και η επίθεση σχετιζόμενου μηνύματος.

Παραλλαγή ΙΙ

Έστω μια μονόδρομη συνάρτηση καταπακτής (βλ. ορισμό 2.2) δύο εισόδων ($\boldsymbol{m}$ και $\boldsymbol{e}$), με έξοδο ένα διάνυσμα μήκους k. Συγκεκριμένα, δοθείσας της συνάρτησης $f(\boldsymbol{m}, \boldsymbol{e})$, είναι υπολογιστικά ανέφικτο, να υπολογιστούν τα $\boldsymbol{m}, \boldsymbol{e}$. Αντιθέτως, δοθέντων των $f(\boldsymbol{m}, \boldsymbol{e})$ και $\boldsymbol{e}$, είναι υπολογιστικά εφικτό να υπολογιστεί το αρχικό κείμενο $\boldsymbol{m}$. Ακριβέστερα, το κρυπτοκείμενο $\boldsymbol{c}$ κρυπτογραφείται ως εξής:

$$\boldsymbol{c} = f(\boldsymbol{m}, \boldsymbol{e})\boldsymbol{G}_{pub} + \boldsymbol{e}$$

όπου $\boldsymbol{e} \in \mathbb{F}_2^n$, βάρους $\mathsf{wt}(\boldsymbol{e}) = t$.

Στην συνέχεια, ο παραλήπτης χρησιμοποιεί τον αλγόριθμο αποκρυπτογράφησης που χρησιμοποίησε και ο McEliece στην αρχική του περιγραφή, ώστε να ανακτήσει την συνάρτηση καταπακτής $f(\boldsymbol{m}, \boldsymbol{e})$. Ύστερα αντιστρέφοντας την συνάρτηση $f(\boldsymbol{m}, \boldsymbol{e})$, υπολογίζει το αρχικό κείμενο $\boldsymbol{m}$. Συνεπώς, στην περίπτωση αυτή το κρυπτοσύστημα McEliece καθίσταται ασφαλές έναντι των επιθέσεων του Berson, διότι:

(α) Αν υποθέσουμε ότι, έχουμε δύο μηνύματα $\boldsymbol{m}_1, \boldsymbol{m}_2$. Αν $\boldsymbol{m}_1 = \boldsymbol{m}_2$, τότε προκύπτει το εξής:

$$\boldsymbol{c}_1 + \boldsymbol{c}_2 = (f(\boldsymbol{m}_1, \boldsymbol{e}_1) + f(\boldsymbol{m}_2, \boldsymbol{e}_2))\boldsymbol{G}_{pub} + \boldsymbol{e}_1 + \boldsymbol{e}_2.$$

Δεδομένου λοιπόν, ότι οι $f(\boldsymbol{m}_1, \boldsymbol{e}_1), f(\boldsymbol{m}_2, \boldsymbol{e}_2)$ είναι άγνωστες, η ποσότητα $(f(\boldsymbol{m}_1, \boldsymbol{e}_1) + f(\boldsymbol{m}_2, \boldsymbol{e}_2))\boldsymbol{G}_{pub}$ δεν είναι δυνατόν να υπολογιστεί. Κατ' επέκταση δεν μπορεί να προκύψει καμία πληροφορία, σχετικά με τις θέσεις στις οποίες εντοπίζονται τα σφάλματα. Επομένως, η επίθεση επανεκπομπής μηνύματος αποτυγχάνει. Επιπλέον, ακόμα και στην περίπτωση που είναι γνωστή η σχέση $\boldsymbol{m}_1 + \boldsymbol{m}_2$, δηλαδή:

$$\boldsymbol{c}_1 + \boldsymbol{c}_2 = (\boldsymbol{m}_1 + \boldsymbol{m}_2 + (f(\boldsymbol{m}_1, \boldsymbol{e}_1) + f(\boldsymbol{m}_2, \boldsymbol{e}_2)))\boldsymbol{G}_{pub} + \boldsymbol{e}_1 + \boldsymbol{e}_2$$

επίσης, λόγω έλλειψης γνώσης των $f(\boldsymbol{m}_1, \boldsymbol{e}_1), f(\boldsymbol{m}_2, \boldsymbol{e}_2)$, δεν είναι δυνατός ο υπολογισμός της ποσότητας $(\boldsymbol{m}_1 + \boldsymbol{m}_2 + (f(\boldsymbol{m}_1, \boldsymbol{e}_1) + f(\boldsymbol{m}_2, \boldsymbol{e}_2)))\boldsymbol{G}_{pub}$. Συνεπώς, αποτυγχάνει και η επίθεση σχετιζόμενου μηνύματος.

Παραλλαγή III

Έστω ότι το αρχικό κείμενο είναι: $\boldsymbol{m} = (\boldsymbol{m}_a, \boldsymbol{m}_b)$. Το κρυπτοκείμενο $\boldsymbol{c}$ δίνεται από τον τύπο:

$$\boldsymbol{c} = \boldsymbol{m}_a\boldsymbol{G}_{pub} + \boldsymbol{e}$$

όπου $\boldsymbol{e} = g(\boldsymbol{m}_b)$ η g είναι μια αντιστρέψιμη συνάρτηση, η οποία αποτελεί απεικόνιση του διανύσματος σφάλματος $\boldsymbol{e} \in \mathbb{F}_2^n$, βάρους $\mathsf{wt}(\boldsymbol{e}) = t$.

Στην συνέχεια, ο παραλήπτης χρησιμοποιεί τον αλγόριθμο αποκρυπτογράφησης (βλ. ενότητα 5.1), ώστε να ανακτήσει το $\boldsymbol{m}_a$. Ύστερα αντιστρέφοντας την συνάρτηση $g(\boldsymbol{m}_b)$, υπολογίζει το αρχικό κείμενο $\boldsymbol{m}$ από τη σχέση

$$\boldsymbol{m}_b = g^{-1}(g(\boldsymbol{m}_b)).$$

Ωστόσο, σύμφωνα με τον Sun [34], χρησιμοποιώντας αυτή την μέθοδο είναι δυνατόν να βελτιωθεί ο ρυθμός πληροφορίας του κώδικα από $R \approx 0.51$, σε $R \approx 0.79$, (μεταφέρονται επιπλέον 284 ψηφία πληροφορίας) για τις τιμές των παραμέτρων που πρότεινε ο McEliece [9] $(k = 24, n = 1024, t = 50)$, και από $R \approx 0.63$ σε $R \approx 0.87$ (μεταφέρονται επιπλέον 225 ψηφία πληροφορίας) για τις τιμές που πρότειναν οι Adams και Melter [23] $(k = 654, n = 1024, t = 37)$.

Η ουσιαστική διαφορά της παραλλαγής αυτής, από την αρχική εκδοχή του κρυπτοσυστήματος McEliece, έγκειται στην τυχαιότητα του διανύσματος σφάλματος, καθώς στην περίπτωση αυτή το διάνυσμα σφάλματος, εξαρτάται από το $\boldsymbol{m}_b$ και η κρυπτογράφηση που πραγματοποιείται είναι ντετερμινιστική [10, 7] (σε αντίθεση με την Παραλλαγή I και την Παραλλαγή II, στις οποίες η κρυπτογράφηση που πραγματοποιείται, είναι πιθανοτική [15]). Υπάρχουν δηλαδή περιπτώσεις κατά τις οποίες, εύκολα μπορεί να διαρρεύσει τμήμα της πληροφορίας σε έναν ενδεχόμενο κρυπταναλυτή.

Πιο συγκεκριμένα, έστω $\boldsymbol{m}_1 = (\boldsymbol{m}_{1a}, \boldsymbol{m}_{1b})$ και $\boldsymbol{m}_2 = (\boldsymbol{m}_{2a}, \boldsymbol{m}_{2b})$ τα μηνύματα που κρυπτογραφούνται. Δεδομένου ότι κάθε μήνυμα στην παραλλαγή αυτή αποτε-

Πίνακας 5.1: Οι πιθανές αδυναμίες της παραλλαγής III

Δοθείσα Πληροφορία	**Πληροφορία που ενδέχεται να διαρρεύσει**
$\boldsymbol{m}_{1a}$ (ή $\boldsymbol{m}_{2a}$)	$\boldsymbol{m}_{1b}$ (ή $\boldsymbol{m}_{2b}$)
$\boldsymbol{m}_{1b}$ (ή $\boldsymbol{m}_{2b}$)	$\boldsymbol{m}_{1a}$ (ή $\boldsymbol{m}_{2a}$)
$\boldsymbol{m}_{1a} = \boldsymbol{m}_{2a}$, $\boldsymbol{m}_{1b} = \boldsymbol{m}_{2b}$	Καμία πληροφορία
$\boldsymbol{m}_{1a} = \boldsymbol{m}_{2a}$, $\boldsymbol{m}_{1b} \neq \boldsymbol{m}_{2b}$	$\boldsymbol{m}_{1a}, \boldsymbol{m}_{1b}, \boldsymbol{m}_{2a}, \boldsymbol{m}_{2b}$
$\boldsymbol{m}_{1a} \neq \boldsymbol{m}_{2a}$, $\boldsymbol{m}_{1b} = \boldsymbol{m}_{2b}$	$\boldsymbol{m}_{1a} + \boldsymbol{m}_{2a}$
$\boldsymbol{m}_{1a} + \boldsymbol{m}_{2a}$, $\boldsymbol{m}_{1b} = \boldsymbol{m}_{2b}$	Καμία πληροφορία
$\boldsymbol{m}_{1a} + \boldsymbol{m}_{2a}$, $\boldsymbol{m}_{1b} \neq \boldsymbol{m}_{2b}$	$\boldsymbol{m}_{1a}, \boldsymbol{m}_{1b}, \boldsymbol{m}_{2a}, \boldsymbol{m}_{2b}$

λείται από δύο μέρη, η γραμμική σχέση μεταξύ των μηνυμάτων $\boldsymbol{m}_1$ και $\boldsymbol{m}_2$ επεκτείνεται σε πολλαπλές περιπτώσεις. Στον πίνακα που ακολουθεί, συνοψίζονται οι αδυναμίες που μπορεί να προκύψουν ανά περίπτωση και συγκεκριμένα οι πληροφορίες που ενδέχεται να διαρρεύσουν σε έναν πιθανό κρυπταναλυτή. Προκειμένου να παρακαμφθούν οι παραπάνω αδυναμίες και να βελτιωθεί ο ρυθμός πληροφορίας στο κρυπτοσύστημα McEliece, ο Sun [34] πρότεινε δύο επιπλέον παραλλαγές.

Παραλλαγή IV

Έστω ότι το αρχικό κείμενο είναι: $\boldsymbol{m} = (\boldsymbol{m}_a, \boldsymbol{m}_b)$. Και το κρυπτοκείμενο $\boldsymbol{c}$ δίνεται από τον τύπο

$$\boldsymbol{c} = (\boldsymbol{m}_a + h(\boldsymbol{e}))\boldsymbol{G}_{pub} + \boldsymbol{e}$$

όπου $\boldsymbol{e} = g(\boldsymbol{r}||\boldsymbol{m}_b)$ $\boldsymbol{r}$ είναι ένα τυχαίο διάνυσμα μήκους q και g είναι μια αντιστρέψιμη συνάρτηση, η οποία απεικονίζει το $\boldsymbol{m}_b$ στο διάνυσμα σφάλματος $\boldsymbol{e} \in \mathbb{F}_2^n$ βάρους $\mathsf{wt}(\boldsymbol{e}) = t$. Επιπλέον, η h είναι μια μονόδρομη συνάρτηση κατακερματισμού, με είσοδο το διάνυσμα σφάλματος $\boldsymbol{e}$ και έξοδο ένα διάνυσμα μήκους k. Στην συνέχεια, ο παραλήπτης χρησιμοποιεί τον αλγόριθμο αποκρυπτογράφησης που χρησιμοποίησε και ο McEliece στην αρχική του περιγραφή, ώστε να ανακτήσει το $\boldsymbol{m}'_a$, το οποίο υπολογίζεται ως εξής:

$$\boldsymbol{m}'_a = \boldsymbol{m}_a + h(\boldsymbol{e})$$

καθώς επίσης και το διάνυσμα σφάλματος $\boldsymbol{e}$. Έπειτα, υπολογίζει το:

$$\boldsymbol{r}||\boldsymbol{m}_b = g^{-1}(\boldsymbol{e})$$

όπου $g^{-1}(\boldsymbol{e})$ η αντίστροφη συνάρτηση της g. Αποβάλλοντας το διάνυσμα $\boldsymbol{r}$, προκύπτει το $\boldsymbol{m}_b$. Τελικά, ο παραλήπτης υπολογίζει το $\boldsymbol{m}_a$, ως εξής:

$$m_a = m'_a + h(e).$$

Ωστόσο, σύμφωνα με τον Sun [34], χρησιμοποιώντας αυτή την μέθοδο είναι δυνατόν να βελτιωθεί ο ρυθμός πληροφορίας του κώδικα από $R \approx 0.51$, σε $R \approx 0.79$, για τις τιμές των παραμέτρων που πρότεινε ο McEliece [9] ($k = 24, n = 1024, t =$

Πίνακας 5.2: Οι πιθανές αδυναμίες της παραλλαγής IV, όταν $q = 0$

Δοθείσα Πληροφορία	**Πληροφορία που ενδέχεται να διαρρεύσει**
$\boldsymbol{m}_{1a}$ (ή $\boldsymbol{m}_{2a}$)	Καμία πληροφορία
$\boldsymbol{m}_{1b}$ (ή $\boldsymbol{m}_{2b}$)	$\boldsymbol{m}_{1a}$ (ή $\boldsymbol{m}_{2a}$)
$\boldsymbol{m}_{1a} = \boldsymbol{m}_{2a}$, $\boldsymbol{m}_{1b} = \boldsymbol{m}_{2b}$	Καμία πληροφορία
$\boldsymbol{m}_{1a} = \boldsymbol{m}_{2a}$, $\boldsymbol{m}_{1b} \neq \boldsymbol{m}_{2b}$	Καμία πληροφορία
$\boldsymbol{m}_{1a} \neq \boldsymbol{m}_{2a}$, $\boldsymbol{m}_{1b} = \boldsymbol{m}_{2b}$	$\boldsymbol{m}_{1a} + \boldsymbol{m}_{1b}$
$\boldsymbol{m}_{1a} + \boldsymbol{m}_{2a}$, $\boldsymbol{m}_{1b} = \boldsymbol{m}_{2b}$	Καμία πληροφορία
$\boldsymbol{m}_{1a} + \boldsymbol{m}_{2a}$, $\boldsymbol{m}_{1b} \neq \boldsymbol{m}_{2b}$	Καμία πληροφορία

50), καθώς και για $q = 0$ και από $R \approx 0.51$, σε $R \approx 0.73$, για $k = 24, n = 1024, t = 50$ και $q = 64$. Επιπλέον, από $R \approx 0.63$ σε $R \approx 0.87$ για τις τιμές που πρότειναν οι Adams και Melter [23] ($k = 654, n = 1024, t = 37$), καθώς επίσης και για $q = 0$, ενώ από $R \approx 0.63$ σε $R \approx 0.8$ για $k = 654, n = 1024, t = 37$ και $q = 64$.

Κατ' επέκταση, η ασφάλεια αυτής της μεθόδου εκτιμάται για $q = 0$ (ντετερμινιστική κρυπτογράφηση) και $q = 64$ (πιθανοτική κρυπτογράφηση) αντίστοιχα και τα αποτελέσματα απεικονίζονται στους πίνακες 5.2 και 5.3.

Πίνακας 5.3: Οι πιθανές αδυναμίες της παραλλαγής IV, όταν $q = 64$

Δοθείσα Πληροφορία	**Πληροφορία που ενδέχεται να διαρρεύσει**
$\boldsymbol{m}_{1a}$ (ή $\boldsymbol{m}_{2a}$)	Καμία πληροφορία
$\boldsymbol{m}_{1b}$ (ή $\boldsymbol{m}_{2b}$)	Καμία πληροφορία
$\boldsymbol{m}_{1a} = \boldsymbol{m}_{2a}$, $\boldsymbol{m}_{1b} = \boldsymbol{m}_{2b}$	Καμία πληροφορία
$\boldsymbol{m}_{1a} = \boldsymbol{m}_{2a}$, $\boldsymbol{m}_{1b} \neq \boldsymbol{m}_{2b}$	Καμία πληροφορία
$\boldsymbol{m}_{1a} \neq \boldsymbol{m}_{2a}$, $\boldsymbol{m}_{1b} = \boldsymbol{m}_{2b}$	Καμία πληροφορία
$\boldsymbol{m}_{1a} + \boldsymbol{m}_{2a}$, $\boldsymbol{m}_{1b} = \boldsymbol{m}_{2b}$	Καμία πληροφορία
$\boldsymbol{m}_{1a} + \boldsymbol{m}_{2a}$, $\boldsymbol{m}_{1b} \neq \boldsymbol{m}_{2b}$	Καμία πληροφορία

Ακριβέστερα, για $q = 64$, δεν είναι δυνατόν να διαρρεύσει καμία πληροφορία, λόγω έλλειψης γνώσης των $h(g(\boldsymbol{r}_1 \| \boldsymbol{m}_{1b})) + h(g(\boldsymbol{r}_2 \| \boldsymbol{m}_{2b}))$ και $g(\boldsymbol{r}_1 \| \boldsymbol{m}_{1b}) + g(\boldsymbol{r}_2 \| \boldsymbol{m}_{2b})$, αντίστοιχα ανά περίπτωση.

Παραλλαγή V

Έστω ότι το αρχικό κείμενο είναι: $\boldsymbol{m} = (\boldsymbol{m}_a, \boldsymbol{m}_b)$. Και το κρυπτοκείμενο $\boldsymbol{c}$ δίνεται από τον τύπο

$$\boldsymbol{c} = f(\boldsymbol{m}_a, \boldsymbol{e})\boldsymbol{G}_{pub} + \boldsymbol{e}$$

όπου $\boldsymbol{e} = g(\boldsymbol{r} \| \boldsymbol{m}_b)$, g μια αντιστρέψιμη συνάρτηση η οποία απεικονίζει το $\boldsymbol{r} \| \boldsymbol{m}_b$ στο διάνυσμα σφάλματος $\boldsymbol{e} \in \mathbb{F}_2^n$, βάρους $\mathrm{wt}(\boldsymbol{e}) = t$. Επιπλέον, η f είναι μια μονόδρομη συνάρτηση καταπακτής δύο εισόδων $(\boldsymbol{m}_a, \boldsymbol{e})$ και έξοδο ένα διάνυσμα μήκους k.

Πίνακας 5.4: Οι πιθανές αδυναμίες της παραλλαγής V, όταν $q = 0$

Δοθείσα Πληροφορία	**Πληροφορία που ενδέχεται να διαρρεύσει**
$\boldsymbol{m}_{1a}$ (ή $\boldsymbol{m}_{2a}$)	Καμία πληροφορία
$\boldsymbol{m}_{1b}$ (ή $\boldsymbol{m}_{2b}$)	$\boldsymbol{m}_{1a}$ (ή $\boldsymbol{m}_{2a}$)
$\boldsymbol{m}_{1a} = \boldsymbol{m}_{2a}$, $\boldsymbol{m}_{1b} = \boldsymbol{m}_{2b}$	Καμία πληροφορία
$\boldsymbol{m}_{1a} = \boldsymbol{m}_{2a}$, $\boldsymbol{m}_{1b} \neq \boldsymbol{m}_{2b}$	Καμία πληροφορία
$\boldsymbol{m}_{1a} \neq \boldsymbol{m}_{2a}$, $\boldsymbol{m}_{1b} = \boldsymbol{m}_{2b}$	Καμία πληροφορία
$\boldsymbol{m}_{1a} + \boldsymbol{m}_{2a}$, $\boldsymbol{m}_{1b} = \boldsymbol{m}_{2b}$	Καμία πληροφορία
$\boldsymbol{m}_{1a} + \boldsymbol{m}_{2a}$, $\boldsymbol{m}_{1b} \neq \boldsymbol{m}_{2b}$	Καμία πληροφορία

Πίνακας 5.5: Οι πιθανές αδυναμίες της παραλλαγής V, όταν $q = 64$

Δοθείσα Πληροφορία	**Πληροφορία που ενδέχεται να διαρρεύσει**
$\boldsymbol{m}_{1a}$ (ή $\boldsymbol{m}_{2a}$)	Καμία πληροφορία
$\boldsymbol{m}_{1b}$ (ή $\boldsymbol{m}_{2b}$)	Καμία πληροφορία
$\boldsymbol{m}_{1a} = \boldsymbol{m}_{2a}$, $\boldsymbol{m}_{1b} = \boldsymbol{m}_{2b}$	Καμία πληροφορία
$\boldsymbol{m}_{1a} = \boldsymbol{m}_{2a}$, $\boldsymbol{m}_{1b} \neq \boldsymbol{m}_{2b}$	Καμία πληροφορία
$\boldsymbol{m}_{1a} \neq \boldsymbol{m}_{2a}$, $\boldsymbol{m}_{1b} = \boldsymbol{m}_{2b}$	Καμία πληροφορία
$\boldsymbol{m}_{1a} + \boldsymbol{m}_{2a}$, $\boldsymbol{m}_{1b} = \boldsymbol{m}_{2b}$	Καμία πληροφορία
$\boldsymbol{m}_{1a} + \boldsymbol{m}_{2a}$, $\boldsymbol{m}_{1b} \neq \boldsymbol{m}_{2b}$	Καμία πληροφορία

Στην συνέχεια, ο παραλήπτης χρησιμοποιεί τον αλγόριθμο αποκρυπτογράφησης (βλ. ενότητα 5.1), ώστε να ανακτήσει το $\boldsymbol{m}'_a$, από τη σχέση

$$\boldsymbol{m}'_a = f(\boldsymbol{m}_a, \boldsymbol{e})$$

καθώς επίσης και το διάνυσμα σφάλματος $\boldsymbol{e}$. Έπειτα, υπολογίζει το:

$$\boldsymbol{m}_b = g^{-1}(\boldsymbol{e})$$

όπου $g^{-1}(\boldsymbol{e})$ η αντίστροφη συνάρτηση της g. Τελικά, ο παραλήπτης υπολογίζει το $\boldsymbol{m}_a$, ως εξής:

$$\boldsymbol{m}_a = f^{-1}(\boldsymbol{m}'_a, \boldsymbol{e})$$

όπου f^{-1} η αντίστροφη συνάρτηση της f. Ωστόσο, ο ρυθμός πληροφορίας του κώδικα για την μέθοδο αυτή, διατηρεί τις ίδιες τιμές με την Παραλλαγή IV. Καθώς επίσης και η ασφάλεια της μεθόδου αυτής εκτιμάται, για $q = 0$ (ντετερμινιστική κρυπτογράφηση) και $q = 64$ (πιθανοτική κρυπτογράφηση) αντίστοιχα και τα αποτελέσματα απεικονίζονται στους πίνακες 5.4 και 5.5.

ΚΕΦΑΛΑΙΟ 6

Αποκωδικοποίηση συνόλου πληροφορίας

Οι αλγόριθμοι που βασίζονται στην *αποκωδικοποίηση του συνόλου πληροφορίας* (Information set decoding ή ISD) αποτελούν την πιο γνωστή κατηγορία γενικής φύσεως αλγορίθμων αποκωδικοποίησης τυχαίων γραμμικών κωδίκων. Συνεπώς, οι αλγόριθμοι αυτοί επιλύουν (αν και σε εκθετικό χρόνο) το πρόβλημα CSD, στο οποίο βασίζεται και η ασφάλεια των κρυπτοσυστημάτων McEliece και Niederreiter.

Για την ακρίβεια, δεν υπάρχει αξιοποιήσιμη δομή, πλην εκείνης του διανυσματικού χώρου για γραμμικούς κώδικες (βάση της οποίας εφαρμόζονται στοιχειώδεις μετασχηματισμοί). Η γνώση του (ακριβούς) βάρους ω του διανύσματος σφάλματος, επιτρέπει τον προσδιορισμό του χώρου εξαντλητικής αναζήτησης [24], περιορίζοντας την αναζήτηση σε συγκεκριμένα διανύσματα σφάλματος. Κατ' επέκταση, η αποκωδικοποίηση συνόλου πληροφορίας σε όλες τις μορφές της, λειτουργεί αξιοποιώντας τον πλεονασμό του κώδικα.

Στη συνέχεια, θα εξετάσουμε πλήθος ISD αλγορίθμων που εισήχθησαν από τους Prange [1], Lee-Brickell [19], Leon [20], Stern [22], Finiasz και Sendrier [55], Bernstein *et al.* [59], Johansson και Löndahl [64], May *et al.* [63] και Becker *et al.* [65].

6.1 Αλγόριθμος του Prange

Το 1962 ο Prange [1] εισήγαγε τον πρώτο ISD αλγόριθμο, ο οποίος συχνά αποκαλείται *απλή αποκωδικοποίηση συνόλου πληροφορίας* (plain information set decoding ή plain ISD), και αποτέλεσε την βάση για όλες τις επακόλουθες βελτιστοποιήσεις. Στη συνέχεια, αναλύεται η κεντρική ιδέα του αλγορίθμου.

Σε έναν (n, k) κώδικα αν υπάρχει ένα σύνολο από k ψηφία (από το λαμβανόμενο διάνυσμα), που αντιστοιχούν σε γραμμικά ανεξάρτητες στήλες του πίνακα γεννήτορα, τότε μπορεί να κατασκευαστεί η μοναδική κωδική λέξη του κώδικα, η οποία συμφωνεί με το λαμβανόμενο διάνυσμα στα ψηφία αυτά. Συνεπώς,

πρόκειται για ένα σύνολο, που καθορίζει με μοναδικό τρόπο την κωδική λέξη, το οποίο αποκαλείται σύνολο πληροφορίας[1].

Συγκεκριμένα, αν το λαμβανόμενο διάνυσμα δεν περιέχει σφάλματα στο σύνολο πληροφορίας, μπορεί να ανακτηθεί το διάνυσμα σφάλματος, βρίσκοντας τη μοναδική κωδική λέξη, που ταυτίζεται με το λαμβανόμενο διάνυσμα στο σύνολο πληροφορίας. Συνεπώς, τα σφάλματα *παγιδεύονται*. Κατ' επέκταση κάθε διορθώσιμο διάνυσμα σφάλματος, που βρίσκεται εκτός του συνόλου πληροφορίας μπορεί να διορθωθεί με αυτή την μέθοδο.

Ισοδύναμη περιγραφή μπορεί να αποδοθεί με βάση το σύνδρομο. Για την ακρίβεια, η διαδικασία που εκτελείται είναι η ακόλουθη. Αρχικά, εφαρμόζεται τυχαία μετάθεση των στηλών του πίνακα ελέγχου ισοτιμίας $\boldsymbol{H}$, από όπου προκύπτει ο πίνακας $\widetilde{\boldsymbol{H}} = \boldsymbol{HP}$ (πολλαπλασιάζοντας τον $\boldsymbol{H}$ με έναν τυχαίο πίνακα μετάθεσης $\boldsymbol{P} \in \mathbb{F}_2^{n\times n}$). Αντιστοίχως, εκτελείται μετάθεση των συνιστωσών του διανύσματος σφάλματος $\boldsymbol{e}$, δηλαδή $\widetilde{\boldsymbol{e}} = \boldsymbol{P}^{-1}\boldsymbol{e}$. Εν συνέχεια, εφαρμόζεται απαλοιφή του Gauss [18] στο δεξιό τμήμα του υποπίνακα[2] $\widetilde{\boldsymbol{H}}_I$, $I = \{k+1, \cdots, n\}$. Αν είναι επιτυχής, προκύπτει ένας αντιστρέψιμος πίνακας $\boldsymbol{Q}' \in \mathbb{F}_2^{(n-k)\times(n-k)}$, ο οποίος πολλαπλασιάζεται με τον πίνακα $\boldsymbol{HP}$, καθώς επίσης και με το αντίστοιχο σύνδρομο $\boldsymbol{s}$, δηλαδή $\widetilde{\boldsymbol{s}} = \boldsymbol{Q}'\boldsymbol{s}$. Κατά συνέπεια, ο πίνακας $\widetilde{\boldsymbol{H}}$ που τελικά προκύπτει είναι σε συστηματική μορφή

$$\widetilde{\boldsymbol{H}} = \boldsymbol{Q}'\boldsymbol{HP} = \begin{pmatrix} \boldsymbol{Q} & \boldsymbol{I}_{n-k} \end{pmatrix}. \tag{6.1}$$

Για την ακρίβεια, $\mathsf{wt}(\boldsymbol{e}) = \mathsf{wt}(\widetilde{\boldsymbol{e}})$, συνεπώς

$$\boldsymbol{He} = \boldsymbol{s} \Leftrightarrow \boldsymbol{Q}'\boldsymbol{HPP}^{-1}\boldsymbol{e} = \boldsymbol{Q}'\boldsymbol{s} \Leftrightarrow \widetilde{\boldsymbol{H}}\widetilde{\boldsymbol{e}} = \widetilde{\boldsymbol{s}}. \tag{6.2}$$

Κατ' επέκταση, το διάνυσμα σφάλματος $\boldsymbol{e}$ αποτελεί λύση του $\mathrm{CSD}(\boldsymbol{H}, \boldsymbol{s}, \omega)$ αν και μόνο αν το $\widetilde{\boldsymbol{e}}$ αποτελεί λύση του $\mathrm{CSD}(\widetilde{\boldsymbol{H}}, \widetilde{\boldsymbol{s}}, \omega)$. Κατά συνέπεια, αν θεωρήσουμε ότι όλες οι μη-μηδενικές θέσεις του διανύσματος σφάλματος $\widetilde{\boldsymbol{e}}$ περιέχονται αποκλειστικά στις τελευταίες $n-k$ συνιστώσες (η τυχαία μετάθεση που εφαρμόστηκε ήταν καλή [[66], Ενότητα 4.1.2]), προκύπτει ότι

$$\widetilde{\boldsymbol{s}} = \widetilde{\boldsymbol{H}}\widetilde{\boldsymbol{e}} = \begin{pmatrix} \boldsymbol{Q} & \boldsymbol{I}_{n-k} \end{pmatrix} \begin{pmatrix} \widetilde{\boldsymbol{e}}_L \\ \widetilde{\boldsymbol{e}}_R \end{pmatrix} = \boldsymbol{Q}\widetilde{\boldsymbol{e}}_L + \widetilde{\boldsymbol{e}}_R = \widetilde{\boldsymbol{e}}_R$$

όπου $\widetilde{\boldsymbol{e}}_L = \widetilde{\boldsymbol{e}}_{[1,k]}$ και $\widetilde{\boldsymbol{e}}_R = \widetilde{\boldsymbol{e}}_{[k+1,n]}$. Συνεπώς, το σύνδρομο $\widetilde{\boldsymbol{s}}$ αποκαλύπτει ότι δεν υπάρχουν σφάλματα στις πρώτες k συνιστώσες του διανύσματος σφάλματος $\boldsymbol{e}$, καθώς $\widetilde{\boldsymbol{e}} = (\boldsymbol{0}, \widetilde{\boldsymbol{s}})$ όπου $\boldsymbol{0}$ το μηδενικό διάνυσμα μήκους k. Η διαδικασία εκτελείται επαναληπτικά, έως ότου βρεθεί το κατάλληλο διάνυσμα σφάλματος $\boldsymbol{e}$ βάρους

[1] Ένα σύνολο πληροφορίας Z για τον πίνακα ελέγχου ισοτιμίας $\boldsymbol{H}$, είναι ένα σύνολο από k ακεραίους στο $\{1, 2, \cdots, n\}$, για το οποίο οι $n-k$ στήλες του $\boldsymbol{H}$ (που δεν περιέχονται στο Z) είναι γραμμικά ανεξάρτητες. Συνεπώς, οι κωδικές λέξεις προσδιορίζονται (με μοναδικό τρόπο) από τις συνιστώσες που περιέχονται στο σύνολο Z.

[2] όπου $\widetilde{\boldsymbol{H}}_I$ εκφράζει το τμήμα του πίνακα $\widetilde{\boldsymbol{H}}$, οι στήλες του οποίου ανήκουν στο σύνολο $I = \{k+1, \cdots, n\}$.

$\mathsf{wt}(\boldsymbol{e}) = \omega$, που θα επιλύει το CSD$(\boldsymbol{H}, \boldsymbol{s}, \omega)$. Η ακριβής περιγραφή δίνεται από τον Αλγ. 6.1.

Αλγ. 6.1 Ο αλγόριθμος του Prange

είσοδος: $\boldsymbol{H}$, $\boldsymbol{e}$, βάρος ω

$\boldsymbol{s} \leftarrow \boldsymbol{H}\boldsymbol{e}$
repeat
  Επέλεξε $\boldsymbol{P}$ » τυχαίος πίνακας αντιμετάθεσης
  $\widetilde{\boldsymbol{H}} \leftarrow \boldsymbol{Q}'\boldsymbol{H}\boldsymbol{P} = \begin{pmatrix} \boldsymbol{Q} & \boldsymbol{I}_{n-k} \end{pmatrix}$ » με απαλοιφή του Gauss στον $\boldsymbol{HP}$
  $\widetilde{\boldsymbol{s}} \leftarrow \boldsymbol{Q}'\boldsymbol{s}$
  $\widetilde{\boldsymbol{e}}_R \leftarrow \widetilde{\boldsymbol{s}}$
  $\widetilde{\boldsymbol{e}} \leftarrow \begin{pmatrix} \boldsymbol{0} & \widetilde{\boldsymbol{e}}_R \end{pmatrix}$
until $\mathsf{wt}(\widetilde{\boldsymbol{e}}) = \omega$
$\boldsymbol{e} \leftarrow \boldsymbol{P}\widetilde{\boldsymbol{e}}$

έξοδος: διάνυσμα $\boldsymbol{e}$

Υπάρχει ωστόσο και η περίπτωση, κατά την οποία οι τυχαίες μεταθέσεις που εφαρμόζονται κατά την εκτέλεση του Αλγ. 6.1 είναι *ελαττωματικές* (defective permutations), δηλαδή ο υποπίνακας $(\boldsymbol{HP})_{[k+1,n]}$ έχει τάξη $< n - k$. Εντούτοις, ακόμα και στην περίπτωση αυτή, μπορεί να υπολογιστεί η συστηματική μορφή του πίνακα $\boldsymbol{H}$, καθώς και το διάνυσμα σφάλματος $\widetilde{\boldsymbol{e}} = \boldsymbol{P}^{-1}\boldsymbol{e}$, με πιθανότητα τουλάχιστον $1 - e^{-1}$, όπως αποδεικνύεται στο Λήμμα 4.14, [66].

Πολλοί εξωραϊσμοί της βασικής ιδέας η οποία εισήχθη από τον Prange, προτάθηκαν στην συνέχεια. Για την ακρίβεια, μια παραλλαγή της βασικής ιδέας εισήχθη το 1964 από τον Kasami [2]. Πρόκειται για την μέθοδο των *πολυωνύμων κάλυψης* (covering polynomials), η ιδέα της οποίας στηρίζεται στην κατάρριψη της συνθήκης με βάση την οποία το σύνολο πληροφορίας είναι απαλλαγμένο από σφάλματα, καθώς και την εν συνεχεία συστηματική αναζήτηση με στόχο τον εντοπισμό σφαλμάτων στο σύνολο πληροφορίας.

Άλλη μια γνωστή μέθοδος που προτάθηκε στην συνέχεια, αναφέρεται ως *γενικευμένη αποκωδικοποίηση συνόλου πληροφορίας* [24]. Πρόκειται για μια μέθοδο που επιλέγει ένα πλήθος k ψηφίων, τα οποία δεν συνιστούν απαραιτήτως ένα σύνολο πληροφορίας. Για την ακρίβεια, αν το επιλεγμένο πλήθος διαθέτει λιγότερα από k ανεξάρτητα ψηφία, αυξάνεται με την προσθήκη επιπρόσθετων ψηφίων, έως ότου εν τέλει να περιέχει k γραμμικά ανεξάρτητα. Έπειτα εκτελείται αναζήτηση σε ολόκληρο το σύνολο. Η μέθοδος αυτή είναι ισοδύναμη με την *αποκωδικοποίηση με πολλαπλασιαστές* (decoding with multipliers) που προτάθηκε το 1976 από τους Baumert *et al.* [4] και ουσιαστικά πρόκειται για έναν συνδυασμό των μεθόδων συνόλου πληροφορίας και πολυωνύμων κάλυψης.

6.1.1 Πολυπλοκότητα

Παρά το ιδιαίτερο ενδιαφέρον για τους ISD αλγορίθμους, δεν υπήρχαν αρχικά σαφείς εκτιμήσεις σχετικά με την πολυπλοκότητα που απαιτείται κατά την αποκωδικοποίηση. Η πρώτη εκτίμηση, εισήχθη το 1990 από τους Coffey και Goodman [24], οι οποίοι στα πλαίσια μιας εκτεταμένης μελέτης πρότειναν μια λύση, η οποία αποδείχθηκε λογαριθμικά ακριβής για όλους σχεδόν τους γραμμικούς κώδικες [24].

Ειδικότερα, οι Coffey και Goodman εξετάζοντας την δυσκολότερη περίπτωση κατά την οποία $W = D_{GV}(R)$, ανέλυσαν την *πολυπλοκότητα της εξαντλητικής αναζήτησης* (brute-force complexity), για την επίλυση του CSD προβλήματος και συγκεκριμένα για έναν κώδικα μεγάλου μήκους και ρυθμού πληροφορίας R. Η εν λόγω ανάλυση παρουσίασε ιδιαίτερο ενδιαφέρον, καθώς αποτέλεσε σημείο αναφοράς για όλους τους ISD αλγορίθμους. Συγκεκριμένα, για $W = D_{GV}(R)$ ένας αλγόριθμος εξαντλητικής αναζήτησης είτε:

- απαριθμεί όλα τα διανύσματα σφάλματος $\boldsymbol{e}$ βάρους ω, γεγονός που απαιτεί $2^{(1-R)n+o(n)}$ πράξεις,
- είτε αναζητά έναν αντιπρόσωπο $\boldsymbol{e}^* = \boldsymbol{x}+\boldsymbol{c}^*$ του ομοσυνόλου $\boldsymbol{x}+C$ τέτοιο ώστε $\mathsf{wt}(\boldsymbol{e}^*) = \min_{\boldsymbol{c}\in C} \mathsf{wt}(\boldsymbol{x}+\boldsymbol{c}) = \mathsf{d}(\boldsymbol{x}, C)$, γεγονός που απαιτεί $2^{Rn+o(n)}$ πράξεις.

Συνεπώς, ο συντελεστής χρονικής πολυπλοκότητας $T^*_{BF}(R) := T^*_{BF}(R, D_{GV}(R))$ με βάση την (3.30), προκύπτει ως εξής:

$$T^*_{BF}(R) = \begin{cases} R, & 0 \le R \le \frac{1}{2}, \\ 1-R, & \frac{1}{2} < R \le 1. \end{cases} . \tag{6.3}$$

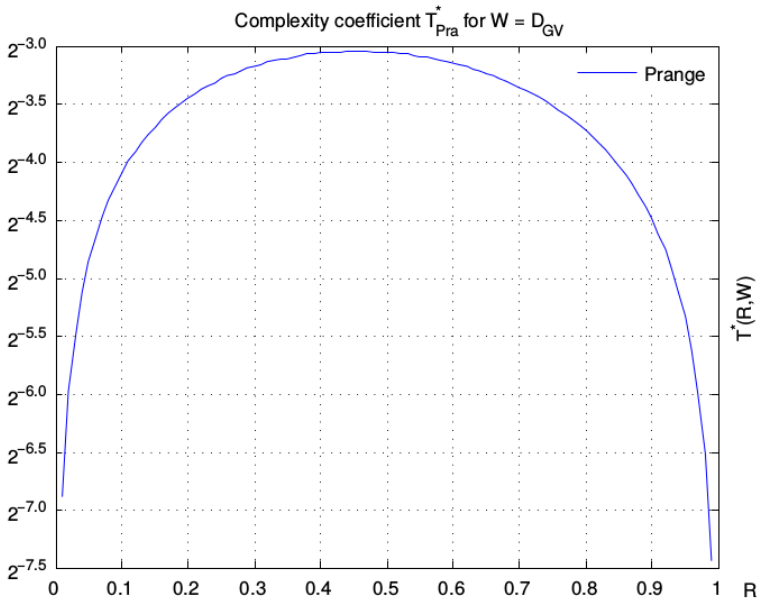
Σημείωση 6.1. Σε αντίθεση με τους ISD αλγορίθμους, ο αλγόριθμος εξαντλητικής αναζήτησης είναι *ντετερμινιστικός* (deterministic) [46] και αποδίδει πάντοτε λύση, ύστερα από $2^{T^*_{BF}(R)n+o(n)}$ το πολύ βήματα.

Επιπλέον, εξετάστηκε η συμπεριφορά των ISD αλγορίθμων για την επίλυση του CSD προβλήματος (για σταθερά n, R και μεταβλητό W) και ειδικότερα για τον Αλγ. 6.1. Η πιθανότητα επιτυχίας του αλγορίθμου, δίνεται από τη σχέση

$$\mathsf{Pr}_{success} = \frac{\binom{n-k}{\omega}}{\binom{n}{\omega}} \tag{6.4}$$

καθώς το διάνυσμα $\widetilde{\boldsymbol{e}}$ αποτελεί μια τυχαία λέξη μήκους n και βάρους ω και μεταξύ αυτών υπάρχουν ακριβώς $\binom{n-k}{\omega}$ επιθυμητές, εκ των δυνατών $\binom{n}{\omega}$ περιπτώσεων. Από την (6.4) παίρνουμε την ακόλουθη εκτίμηση σχετικά με τον αριθμό των επαναλήψεων που απαιτούνται

$$\begin{aligned} N_{Pra}(n,k,\omega) := \frac{1}{\mathsf{Pr}_{success}} &= \binom{n}{\omega}\binom{n-k}{\omega}^{-1} \\ &\simeq 2^{n\cdot H_2\left(\frac{\omega}{n}\right)-(n-k)\cdot H_2\left(\frac{\omega}{n-k}\right)}. \end{aligned} \tag{6.5}$$



Σχήμα 6.1: Συντελεστής χρονικής πολυπλοκότητας για τον αλγόριθμο του Prange, όπου $0 \leq R \leq 1$ και $W = D_{GV}$.

Οι Coffey και Goodman υπολόγισαν τον συντελεστή πολυπλοκότητας T^*_{Pra} από τις σχέσεις (6.4) και (6.5), ορίζοντας αρχικά την ποσότητα

$$\begin{aligned} N^*_{Pra} &= \frac{1}{n}\log_2 N_{Pra} = H_2\left(\frac{\omega}{n}\right) - \left(1-\frac{k}{n}\right)\cdot H_2\left(\frac{\omega}{n-k}\right) \\ &= H_2\left(\frac{\omega}{n}\right) - \left(1-\frac{k}{n}\right)\cdot H_2\left(\frac{\frac{\omega}{n}}{1-\frac{k}{n}}\right) \end{aligned}$$

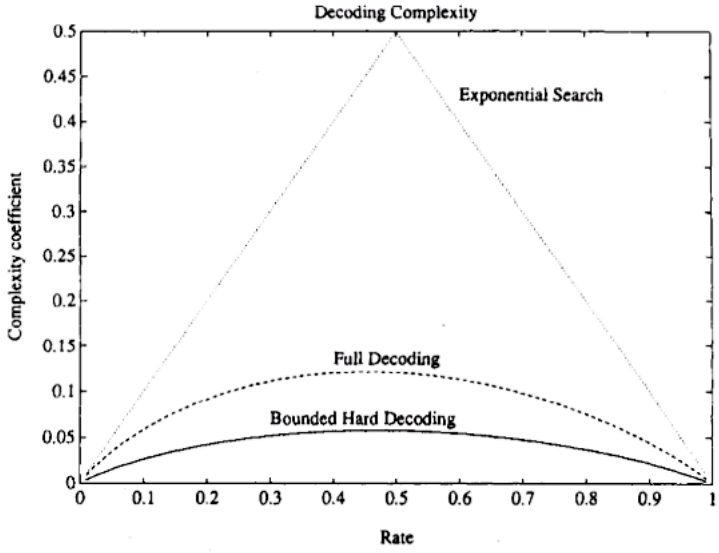
και παίρνοντας τελικά το όριο

$$\begin{aligned} T^*_{Pra}(R,W) &= \lim_{n\to\infty} N_{Pra}(n, \lfloor Rn \rfloor, \lfloor Wn \rfloor) \\ &= H_2(W) - (1-R)H_2\left(\frac{W}{1-R}\right) \end{aligned} \tag{6.6}$$

όπως απεικονίζεται στο Σχ. 6.1. Για την ακρίβεια, απέδειξαν ότι για σχεδόν όλους τους κώδικες, ο συντελεστής χρονικής πολυπλοκότητας για MDD και για $W = D_{GV}(R)$, χρησιμοποιώντας τον Αλγ. 6.1, προκύπτει από την (6.6) θέτοντας $H_2(D_{GV}) = 1 - R$, καταλήγοντας στη σχέση

$$T^*_{Pra}(R,W) = H_2(H_2^{-1}(1-R)) - (1-R)H_2\left(\frac{H_2^{-1}(1-R)}{1-R}\right) + o(1). \tag{6.7}$$

Επιπλέον, απέδειξαν ότι για σχεδόν όλους τους κώδικες ο συντελεστής χρονικής πολυπλοκότητας για *αποκωδικοποίηση φραγμένης απόστασης* (bounded



Σχήμα 6.2: Συντελεστές πολυπλοκότητας εξαντλητικής αναζήτησης T^*_{BF} και του Αλγ. 6.1, όπου $W = D_{GV}(R)$.

distance decoding ή BDD), κατά την οποία αποκωδικοποιούνται όλα τα διανύσματα σφάλματος βάρους μέχρι ω (όπου $\omega = \lfloor (d-1)/2 \rfloor$) και για $W = D_{GV}(R)$, χρησιμοποιώντας τον Αλγ. 6.1, προκύπτει από την (6.6) ως εξής

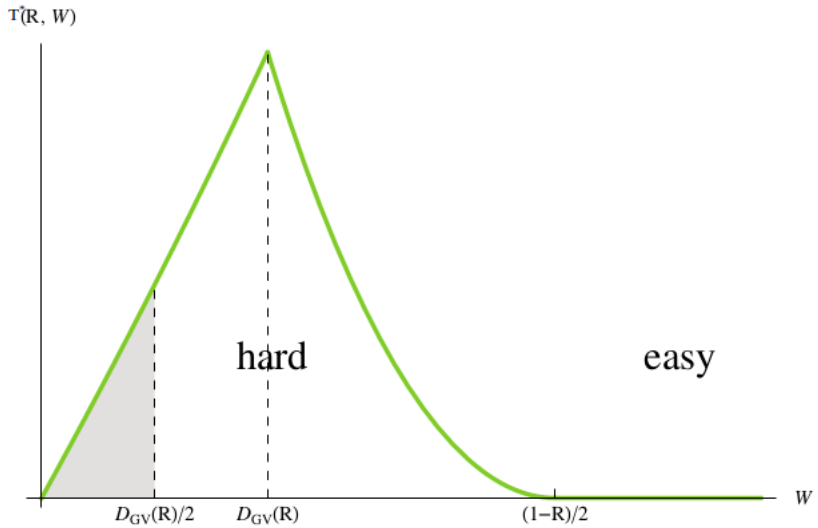
$$T^*_{Pra}(R, W) = H_2(H_2^{-1}(1-R)/2) - (1-R)H_2\left(\frac{H_2^{-1}(1-R)}{2(1-R)}\right) + o(1). \qquad (6.8)$$

Η παραπάνω ανάλυση, απεικονίζεται στο Σχ. 6.2. Συγκεκριμένα, στην περίπτωση της πλήρους αποκωδικοποίησης διακρίνεται ιδιαίτερη βελτίωση έναντι της διαδικασίας εξαντλητικής αναζήτησης, για οποιοδήποτε καθορισμένο ρυθμό πληροφορίας. Ειδικότερα, για $R = 1/2$ ο Αλγ. 6.1 απαιτεί λιγότερο από την τέταρτη ρίζα του αριθμού των υπολογισμών που απαιτούνται για την αναζήτηση μεταξύ όλων των κωδικών λέξεων. Επιπλέον, στην αποκωδικοποίηση φραγμένης απόστασης ο αριθμός των υπολογισμών που απαιτούνται είναι σημαντικά μικρότερος, τόσο συγκριτικά με την εξαντλητική αναζήτηση όσο και με την πλήρη αποκωδικοποίηση, καθώς για $R = 1/2$ οι υπολογισμοί είναι λιγότεροι από την τετραγωνική ρίζα εκείνων που απαιτούνται για την πλήρη αποκωδικοποίηση.

Εν συνεχεία, επεκτείνοντας τη μελέτη για όλες τις δυνατές τιμές του W και συγκεκριμένα για $W \leq D_{GV}(R)$ και $W > D_{GV}(R)$, ο συντελεστής χρονικής πολυπλοκότητας διαμορφώνεται ως εξής

$$T^*_{Pra}(R, W) = \begin{cases} H_2(W) - (1-R)H_2\left(\frac{W}{1-R}\right), & W \leq D_{GV}(R) \\ 1 - R - (1-R)H_2\left(\frac{W}{1-R}\right), & D_{GV} < W < \frac{1-R}{2} \end{cases}. \qquad (6.9)$$

Ως εκ τούτου, παρατηρούμε ότι διακρίνονται δύο περιπτώσεις "μία λύση" έναντι "πολλών λύσεων". Για την ακρίβεια όλοι οι ISD αλγόριθμοι προσπαθούν με



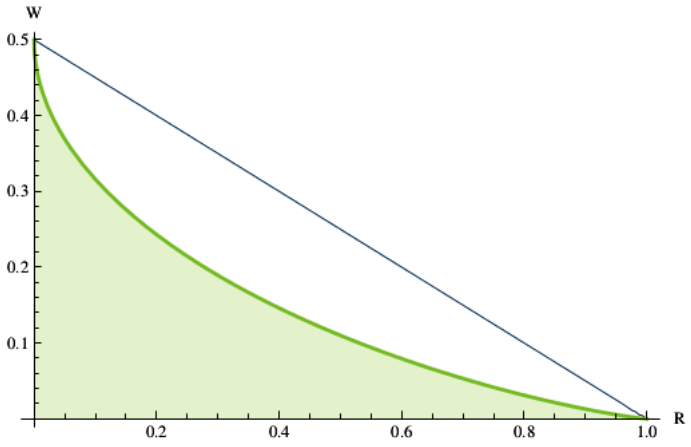
Σχήμα 6.3: Πολυπλοκότητα των ISD αλγορίθμων για $0 < W < \frac{1}{2}$. Η σκιασμένη περιοχή εκφράζει την περίπτωση κατά την οποία, το W βρίσκεται εντός της διορθωτικής ικανότητας του κώδικα - BDD.

επαναληπτικό τρόπο να ανακτήσουν τα διανύσματα σφάλματος $\boldsymbol{e}$ με μια συγκεκριμένη *κατανομή βάρους* (weight distribution). Ειδικότερα, όπως αναλύσαμε στον Αλγ. 6.1 το διάνυσμα σφάλματος $\boldsymbol{e}$ θα πρέπει να έχει μηδενικό βάρος Hamming στις πρώτες k συνιστώσες. Για μια καθορισμένη λύση $\boldsymbol{e}$ η πιθανότητα επιτυχίας (για μια μόνο επανάληψη) μπορεί εύκολα να υπολογιστεί με βάση την πρώτη περίπτωση της (6.9), η οποία αναπαριστά ακριβώς τον αναμενόμενο αριθμό επαναλήψεων, έως ότου βρεθεί μια λύση. Ωστόσο, εάν υπάρχουν (εκθετικά) πολλές λύσεις ο αναμενόμενος αριθμός των επαναλήψεων μπορεί να διαιρεθεί από τον αριθμό των λύσεων. Δηλαδή, η δεύτερη περίπτωση της (6.9) προκύπτει απλά από την πρώτη αφαιρώντας τον αριθμό των λύσεων, οι οποίες δίνονται από $H_2(W) - (1 - R)$, με βάση το πόρισμα 3.8.

Εντούτοις, συνοψίζοντας τα παραπάνω (για όλες τις δυνατές τιμές του W) και με βάση το Σχ. 6.3 προκύπτει ότι ο συντελεστής χρονικής πολυπλοκότητας $T^*_{Pra}(R, W)$ επιτυγχάνει την μέγιστη δυνατή τιμή του για $W = D_{GV}(R)$ και επιπλέον η διαδικασία απλοποιείται για $W = \frac{1-R}{2}$ (βλ. Σχ. 6.4). Συνεπώς, το CSD πρόβλημα παρουσιάζει ιδιαίτερο ενδιαφέρον κυρίως για $0 < W \leq D_{GV}(R)$.

6.2 Αλγόριθμος των Lee–Brickell

Το 1988 οι Lee και Brickell [19] εισήγαγαν έναν νέο ISD αλγόριθμο, επεκτείνοντας τον αλγόριθμο του Prange. Ακολούθησαν την ίδια διαδικασία μετασχηματισμού του πίνακα ελέγχου ισοτιμίας $\boldsymbol{H}$ στην συστηματική του μορφή



Σχήμα 6.4: Εύρος τιμών W υπό το GV όριο σε σχέση με το R (σκιασμένη περιοχή) και το άνω φράγμα $\frac{1-R}{2}$.

$\widetilde{\boldsymbol{H}} = \boldsymbol{Q}'\boldsymbol{H}\boldsymbol{P} = \begin{pmatrix}\boldsymbol{Q} & \boldsymbol{I}_{n-k}\end{pmatrix}$. Παρατήρησαν ωστόσο, ότι ήταν εξαιρετικά δύσκολο όλες οι μη-μηδενικές θέσεις του διανύσματος σφάλματος $\widetilde{\boldsymbol{e}}$, να περιέχονται (αποκλειστικά) στις τελευταίες $n-k$ συνιστώσες. Διαφοροποίησαν έτσι την αρχική ιδέα, θεωρώντας ότι p μη-μηδενικές θέσεις περιέχονται στις πρώτες k συνιστώσες. Συγκεκριμένα, πρότειναν σε κάθε επανάληψη του Αλγ. 6.1 να ελέγχεται κάθε γραμμικός συνδυασμός ακριβώς p στηλών του πίνακα $\boldsymbol{Q}$ και εν συνεχεία να υπολογίζεται η απόσταση Hamming από το σύνδρομο $\widetilde{\boldsymbol{s}}$. Αν η απόσταση είναι ακριβώς $\omega - p$, τότε προστίθενται στις p στήλες του πίνακα $\boldsymbol{Q}$, $\omega - p$ μοναδιαία διανύσματα από τον μοναδιαίο $\boldsymbol{I}_{n-k}$ πίνακα που παράγουν το σύνδρομο $\widetilde{\boldsymbol{s}}$. Δηλαδή

$$\widetilde{\boldsymbol{s}} = \widetilde{\boldsymbol{H}}\widetilde{\boldsymbol{e}} = \begin{pmatrix}\boldsymbol{Q} & \boldsymbol{I}_{n-k}\end{pmatrix}\begin{pmatrix}\widetilde{\boldsymbol{e}}_L \\ \widetilde{\boldsymbol{e}}_R\end{pmatrix} = \boldsymbol{Q}\widetilde{\boldsymbol{e}}_L + \widetilde{\boldsymbol{e}}_R \tag{6.10}$$

όπου $\widetilde{\boldsymbol{e}}_L = \widetilde{\boldsymbol{e}}_{[1,k]}$ και $\widetilde{\boldsymbol{e}}_R = \widetilde{\boldsymbol{e}}_{[k+1,n]}$. Η διαδικασία εκτελείται επαναληπτικά, έως ότου βρεθεί το κατάλληλο διάνυσμα σφάλματος $\boldsymbol{e}$ βάρους $\mathsf{wt}(\boldsymbol{e}) = \omega$, που θα επιλύει το $\mathrm{CSD}(\boldsymbol{H}, \boldsymbol{s}, \omega)$. Η ακριβής περιγραφή δίνεται από τον Αλγ. 6.2.

Προκειμένου να παραμείνει η πολυπλοκότητα του αλγορίθμου σε φυσιολογικά επίπεδα, το βάρος p επιλέγεται ώστε να είναι ακέραιος μικρού μεγέθους.

6.2.1 Πολυπλοκότητα

Η πολυπλοκότητα του Αλγ. 6.2 εξετάζεται ως προς τις ακόλουθες τρεις ποσότητες.

Αλγ. 6.2 Ο αλγόριθμος των Lee–Brickell

είσοδος: $\boldsymbol{H}$, $\boldsymbol{s}$, βάρος ω, παράμετρος p

$\boldsymbol{s} \leftarrow \boldsymbol{He}$
repeat
  Επέλεξε $\boldsymbol{P}$ » τυχαίος πίνακας αντιμετάθεσης
  $\widetilde{\boldsymbol{H}} \leftarrow \boldsymbol{Q'HP} = \begin{pmatrix} \boldsymbol{Q} & \boldsymbol{I}_{n-k} \end{pmatrix}$ » με απαλοιφή του Gauss στον $\boldsymbol{HP}$
  $\widetilde{\boldsymbol{s}} \leftarrow \boldsymbol{Q's}$
  for all $\widetilde{\boldsymbol{e}}_L \in \mathrm{W}_{k,p}$ **do** » όπου $\mathrm{W}_{k,p}$ Hamming Ball
    $\widetilde{\boldsymbol{e}}_R \leftarrow \widetilde{\boldsymbol{s}} + \boldsymbol{Q}\widetilde{\boldsymbol{e}}_L$
    if $\mathrm{wt}(\widetilde{\boldsymbol{e}}_R) = \omega - p$ **then**
      $\boldsymbol{e} \leftarrow \begin{pmatrix} \widetilde{\boldsymbol{e}}_L & \widetilde{\boldsymbol{e}}_R \end{pmatrix}$
      break
    end
  end
until $\mathrm{wt}(\widetilde{\boldsymbol{e}}) = \omega$
$\boldsymbol{e} \leftarrow \boldsymbol{P}\widetilde{\boldsymbol{e}}$

έξοδος: διάνυσμα $\boldsymbol{e}$

Πιθανότητα επιτυχίας

$$\mathsf{Pr}_{success} = \frac{\binom{k}{p}\binom{n-k}{\omega-p}}{\binom{n}{\omega}}. \tag{6.11}$$

Για περισσότερες λεπτομέρειες, σχετικά με την ασυμπτωτική έκφραση της πιθανότητας επιτυχίας (βλ. Ενότητα 8.1).

Ασυμπτωτικός συντελεστής χωρικής πολυπλοκότητας[3]

$$S^*_{LB}(P,R) = H_2\left(\frac{P}{R}\right)\cdot R. \tag{6.12}$$

Ασυμπτωτικός συντελεστής χρονικής πολυπλοκότητας[4]

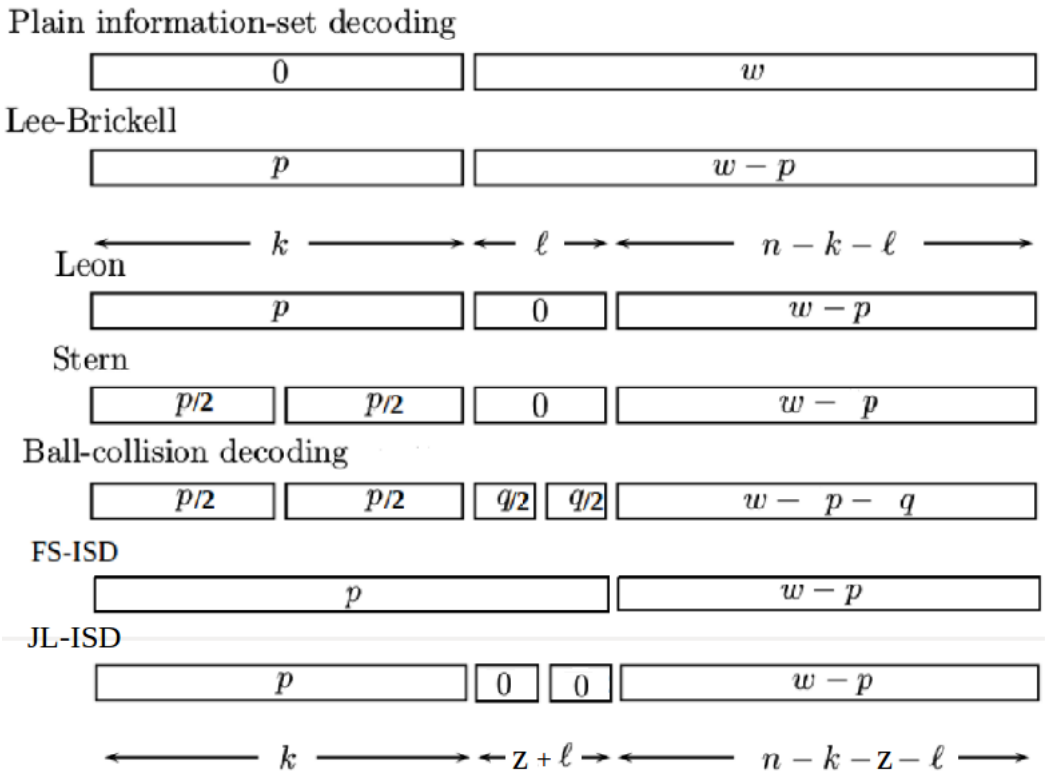
$$T^*_{LB}(R,P,W) = H_2(W) - H_2\left(\frac{W-P}{1-R}\right)\cdot(1-R). \tag{6.13}$$

6.3 Αλγόριθμος του Leon

Το 1988 ο Leon [20] εισήγαγε έναν νέο βελτιωμένο ISD αλγόριθμο, επεκτείνοντας τον Αλγ. 6.2. Η διαφοροποίηση των δύο αλγορίθμων έγκειται στην εισαγωγή ενός παραθύρου μηδενικών συνιστωσών, μήκους ℓ στις τελευταίες $n-k$ συνιστώσες του διανύσματος σφάλματος $\boldsymbol{e}$. Για την ακρίβεια, το παράθυρο αυτό εισάγεται στις $k+1,\cdots,k+\ell$ θέσεις του διανύσματος σφάλματος. Η

[3] Η χωρική πολυπλοκότητα εκφράζει το μέγεθος του αποθηκευτικού χώρου που απαιτείται κατά την εκτέλεση ενός αλγορίθμου.

[4] Η χρονική πολυπλοκότητα εκφράζει τον χρόνο που απαιτείται για την εκτέλεση ενός αλγορίθμου.



Σχήμα 6.5: Συγκριτικός πίνακας κατανομής βάρους του διανύσματος σφάλματος για τους διάφορους ISD αλγορίθμους.

σχέση του αλγορίθμου του Leon με τους προηγούμενους δύο (plain ISD και Lee-Brickell), καθώς και με τους επόμενους που θα εξετάσουμε, απεικονίζεται στο Σχ. 6.5. Αρχικά, εκτελείται η διαδικασία μετασχηματισμού του πίνακα ελέγχου ισοτιμίας $\boldsymbol{H}$ στην συστηματική του μορφή, έτσι ώστε

$$\widetilde{\boldsymbol{H}} := \boldsymbol{Q}'\boldsymbol{H}\boldsymbol{P} = \left(\begin{array}{c|cc} \boldsymbol{Q}^{[\ell]} & \boldsymbol{I}_{\ell} & \\ \boldsymbol{Q} & & \boldsymbol{I}_{n-k-\ell} \end{array}\right) \tag{6.14}$$

όπου $\boldsymbol{Q}^{[\ell]}$ εκφράζει την προβολή των στηλών του πίνακα $\boldsymbol{Q}$ στις ℓ πρώτες γραμμές του. Στη συνέχεια, θεωρώντας ότι p μη-μηδενικές θέσεις περιέχονται στις πρώτες k συνιστώσες, ελέγχεται κάθε γραμμικός συνδυασμός ακριβώς p στηλών του πίνακα $\boldsymbol{Q}^{[\ell]}$, ώστε να υπάρχει ακριβής ταύτιση με το τμήμα εκείνο του συνδρόμου που αντιστοιχεί στις ℓ πρώτες θέσεις, δηλαδή $\widetilde{\boldsymbol{s}}_L = \widetilde{\boldsymbol{s}}_{[1,\ell]}$. Έπειτα υπολογίζεται η απόσταση Hamming $\boldsymbol{Q}\widetilde{\boldsymbol{e}}_L + \widetilde{\boldsymbol{e}}_R$, από το αντίστοιχο σύνδρομο $\widetilde{\boldsymbol{s}}_R = \widetilde{\boldsymbol{s}}_{[\ell+1,n-k]}$ και ελέγχεται αν είναι ακριβώς $\omega - p$. Δηλαδή

$$\widetilde{\boldsymbol{s}} = \widetilde{\boldsymbol{H}}\widetilde{\boldsymbol{e}} \Leftrightarrow \begin{pmatrix} \widetilde{\boldsymbol{s}}_L \\ \widetilde{\boldsymbol{s}}_R \end{pmatrix} = \begin{pmatrix} \boldsymbol{Q}^{[\ell]} & \boldsymbol{I}_{\ell} & \\ \boldsymbol{Q} & & \boldsymbol{I}_{n-k-\ell} \end{pmatrix} \begin{pmatrix} \widetilde{\boldsymbol{e}}_L \\ \widetilde{\boldsymbol{e}}_M \\ \widetilde{\boldsymbol{e}}_R \end{pmatrix}$$

$$\Leftrightarrow \begin{cases} \widetilde{s}_L = Q^{[\ell]}\widetilde{e}_L + \widetilde{e}_M, \\ \widetilde{s}_R = Q\widetilde{e}_L + \widetilde{e}_R \end{cases}$$

όπου $\widetilde{e}_L = \widetilde{e}_{[1,k]}$, $\widetilde{e}_M = \widetilde{e}_{[k+1,k+\ell]} = \mathbf{0}$ και $\widetilde{e}_R = \widetilde{e}_{[k+\ell+1,n]}$. Η διαδικασία εκτελείται επαναληπτικά, έως ότου βρεθεί το κατάλληλο διάνυσμα σφάλματος $\boldsymbol{e}$ βάρους ω, που θα επιλύει το CSD$(\boldsymbol{H}, \boldsymbol{s}, \omega)$. Η ακριβής περιγραφή δίνεται από τον Αλγ. 6.3.

Αλγ. 6.3 Ο αλγόριθμος του Leon

είσοδος: $\boldsymbol{H}$, $\boldsymbol{s}$, βάρος ω, παράμετροι p, ℓ

1: $\boldsymbol{s} \leftarrow \boldsymbol{He}$
2: **repeat**
3: Επέλεξε $\boldsymbol{P}$ » τυχαίος πίνακας αντιμετάθεσης
4: $\widetilde{\boldsymbol{H}} \leftarrow \boldsymbol{Q'HP} = \left(\begin{array}{c|c} Q^{[\ell]} & I_\ell \\ Q & \\ & I_{n-k-\ell} \end{array}\right)$ » με απαλοιφή του Gauss στον $\boldsymbol{HP}$
5: $\widetilde{\boldsymbol{s}} \leftarrow \boldsymbol{Q's}$ » $\begin{pmatrix} \widetilde{s}_L \\ \widetilde{s}_R \end{pmatrix}$
6: **for all** $\widetilde{e}_L \in \mathrm{W}_{k,p}$ **do**
7: **if** $\widetilde{s}_\ell = \widetilde{e}_L Q^{[\ell]}$ **then** » $\widetilde{e}_M = \mathbf{0}$
8: $\widetilde{e}_R \leftarrow s_R + \widetilde{e}_L Q$
9: **if** $\mathrm{wt}(\widetilde{e}_R) = \omega - p$ **then**
10: $\widetilde{e} \leftarrow \begin{pmatrix} \widetilde{e}_L & \mathbf{0} & \widetilde{e}_R \end{pmatrix}$
11: **break**
12: **end**
13: **end**
14: **end**
15: **until** $\mathrm{wt}(\widetilde{e}) = \omega$
16: $\boldsymbol{e} \leftarrow \boldsymbol{P}\widetilde{\boldsymbol{e}}$

έξοδος: διάνυσμα $\boldsymbol{e}$

6.3.1 Πολυπλοκότητα

Η πολυπλοκότητα του Αλγ. 6.3 εξετάζεται ως προς τις ακόλουθες τρεις ποσότητες.

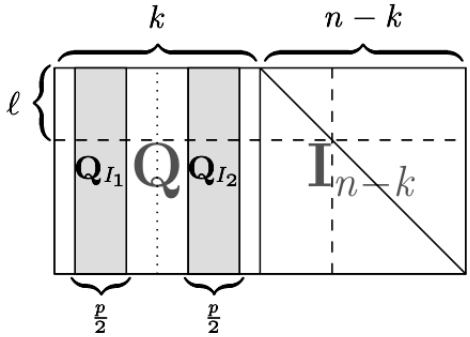
Πιθανότητα επιτυχίας

$$\mathrm{Pr}_{success} = \frac{\binom{k}{p}\binom{n-k-\ell}{\omega-p}}{\binom{n}{\omega}}. \tag{6.15}$$

Για περισσότερες λεπτομέρειες, σχετικά με την ασυμπτωτική έκφραση της πιθανότητας επιτυχίας (βλ. Ενότητα 8.2).

Ασυμπτωτικός συντελεστής χωρικής πολυπλοκότητας

$$S^*_{Leon}(P,R) = H_2\left(\frac{P}{R}\right) \cdot R. \tag{6.16}$$



Σχήμα 6.6: Η δομή του πίνακα ελέγχου ισοτιμίας στον αλγόριθμο αναζήτησης συγκρούσεων του Stern.

Ασυμπτωτικός συντελεστής χρονικής πολυπλοκότητας

$$T^*_{Leon}(R,P,L,W) = H_2(W) - H_2\left(\frac{W-P}{1-R-L}\right)\cdot(1-R-L). \tag{6.17}$$

6.4 Αλγόριθμος του Stern

Το 1989 ο Stern [22] επεκτείνοντας τον Αλγ. 6.3, πρότεινε έναν νέο βελτιωμένο ISD αλγόριθμο που αποτέλεσε την βάση για όλες τις μετέπειτα βελτιστοποιήσεις, καθώς εισήγαγε πρώτος την ιδέα της *αναζήτησης συγκρούσεων* (collision search), που υιοθετήθηκε στη συνέχεια από πολλούς ακόμη ερευνητές.

Αρχικά, εκτελείται η διαδικασία μετασχηματισμού του πίνακα ελέγχου ισοτιμίας $\boldsymbol{H}$ στην συστηματική του μορφή $\widetilde{\boldsymbol{H}} = \boldsymbol{Q}'\boldsymbol{H}\boldsymbol{P} = \begin{pmatrix}\boldsymbol{Q} & \boldsymbol{I}_{n-k}\end{pmatrix}$. Κατόπιν σε κάθε επανάληψη, ο αλγόριθμος καλείται να επιλύσει το *πρόβλημα ταύτισης υποπίνακα* (submatrix matching problem ή SMP), όπως επικράτησε ως ορολογία μετέπειτα στη βιβλιογραφία. Το πρόβλημα ταύτισης υποπίνακα με παραμέτρους ℓ, k και $p \leq k$ ορίζεται ως εξής:

Ορισμός 6.2. Δοθέντος ενός τυχαίου πίνακα $\boldsymbol{Q} = [\boldsymbol{q}_1 \cdots \boldsymbol{q}_k] \in_R \mathbb{F}_2^{\ell\times k}$ και του αντίστοιχου συνδρόμου $\widetilde{\boldsymbol{s}}_L \in \mathbb{F}_2^{\ell}$, αναζητείται ένα *σύνολο δεικτών* (index set) I μεγέθους το πολύ p, έτσι ώστε οι αντίστοιχες στήλες του πίνακα $\boldsymbol{Q}$ να αθροίζουν στο σύνδρομο $\widetilde{\boldsymbol{s}}_L$. Για την ακρίβεια, αναζητείται $I \subset [k]$, με $|I| \leq p$, τέτοιο ώστε[5]

$$\pi_{[\ell]}(\boldsymbol{Q}_I) = \sum_{i\in I} \boldsymbol{q}_i = \widetilde{\boldsymbol{s}}_L \in \mathbb{F}_2^{\ell}. \tag{6.18}$$

[5]όπου με $\pi(\boldsymbol{Q}_I)$ συμβολίζεται το άθροισμα των στηλών του πίνακα $\boldsymbol{Q}$. Επιπλέον, ο συμβολισμός $\pi_L(\boldsymbol{Q}_I)$ εκφράζει την προβολή (του αθροίσματος των στηλών του πίνακα $\boldsymbol{Q}$) στις ℓ γραμμές του. Καθώς επίσης, η σχέση του συνδρόμου $\widetilde{\boldsymbol{s}}_L \in \mathbb{F}_2^{|\mathbb{L}|}$, εκφράζει την προβολή του $\widetilde{\boldsymbol{s}}_L$ στις $|\ell|$ συντεταγμένες, όπου $|\mathbb{L}| = \ell$.

Ειδικότερα, για[6] $I = I_1 \dot{\cup} I_2$ με $I_1 \subseteq \left[1, \frac{k}{2}\right]$ και $I_2 \subseteq \left[\frac{k}{2}+1, k\right]$ (βλ. Σχ. 6.6) έχουμε

$$\pi_{[\ell]}(\boldsymbol{Q}_{I_1}) = \pi_{[\ell]}(\boldsymbol{Q}_{I_2}) + \widetilde{\boldsymbol{s}}_L \in \mathbb{F}_2^{\ell}. \tag{6.19}$$

Επιπλέον, το πρόβλημα ταύτισης υποπίνακα μπορεί να θεωρηθεί ως ακόμα μια περίπτωση αποκωδικοποίησης συνδρόμου, με πίνακα ελέγχου ισοτιμίας $\boldsymbol{Q}$, σύνδρομο $\widetilde{\boldsymbol{s}}$ και παραμέτρους $[k, \ell, p]$. Για την ακρίβεια, κατασκευάζονται δύο λίστες:

$$\mathcal{L}_1 = \left\{ (I_1, \pi_{[\ell]}(\boldsymbol{Q}_{I_1})) : I_1 \subseteq \left[1, \frac{k}{2}\right], |I_1| = \frac{p}{2} \right\},$$
$$\mathcal{L}_2 = \left\{ (I_2, \pi_{[\ell]}(\boldsymbol{Q}_{I_2}) + \widetilde{\boldsymbol{s}}_L) : I_2 \subseteq \left[\frac{k}{2}+1, k\right], |I_2| = \frac{p}{2} \right\}.$$

Κατόπιν, ταξινομείται η λίστα $\mathcal{L}_2$ σύμφωνα με τις *ετικέτες* (labels) $\pi_{\ell}(\boldsymbol{Q}_{I_2}) + \widetilde{\boldsymbol{s}}_L$ και εκτελείται αναζήτηση στα στοιχεία της λίστας $\mathcal{L}_1$ ως προς το $\pi_{\ell}(\boldsymbol{Q}_{I_1})$, προκειμένου να εντοπιστεί ένα στοιχείο της λίστας $\mathcal{L}_2$ με το οποίο θα υπάρξει ταύτιση, πραγματοποιείται δηλαδή η αναζήτηση συγκρούσεων. Κατ' επέκταση, κάθε ζεύγος (I_1, I_2) το οποίο ικανοποιεί την εξίσωση (6.19), αποτελεί λύση του προβλήματος ταύτισης υποπίνακα. Έπειτα, για όλες εκείνες τις υποψήφιες λύσεις, ελέγχεται αν υπάρχει κάποια που να ικανοποιεί την συνθήκη $\mathsf{wt}(\pi(\boldsymbol{Q}_{I_1}) + \pi(\boldsymbol{Q}_{I_2}) + \widetilde{\boldsymbol{s}}_R) = \omega - p$. Όταν εντοπιστεί το κατάλληλο ζεύγος (I_1, I_2) ο αλγόριθμος τερματίζει, όπως απεικονίζεται στον Αλγ. 6.4.

6.4.1 Πολυπλοκότητα

Η πολυπλοκότητα του Αλγ. 6.4 εξετάζεται ως προς τις ακόλουθες ποσότητες.

Πιθανότητα επιτυχίας

$$\mathsf{Pr}_{success} = \frac{\binom{k/2}{p/2}^2 \binom{n-k-\ell}{\omega-p}}{\binom{n}{\omega}}. \tag{6.20}$$

Ασυμπτωτικός εκθέτης αριθμού επαναλήψεων[7]

$$N^*_{Stern}(R, W, P, L) = H_2(W) - H_2\left(\frac{P}{R}\right) \cdot R - H_2\left(\frac{W-P}{1-R-L}\right) \cdot (1-R-L). \tag{6.21}$$

Ασυμπτωτικός συντελεστής χωρικής πολυπλοκότητας

$$S^*_{Stern}(P, R) = H_2\left(\frac{P}{R}\right) \cdot \frac{R}{2}. \tag{6.22}$$

[6] το σύμβολο $\dot{\cup}$ μεταξύ των συνόλων I_1, I_2 εκφράζει ότι: $I_1 \cap I_2 = \varnothing$ (δηλαδή είναι μεταξύ τους ξένα).

[7] ο αριθμός των επαναλήψεων που απαιτούνται προκειμένου ο Αλγ. 6.4 να συγκλίνει στην επιθυμητή λύση, ορίζεται ως $\mathsf{Pr}^{-1}_{success} = N_{Stern}$, ενώ αντίστοιχα ο ασυμπτωτικός εκθέτης συμβολίζεται με N^*_{Stern}.

Αλγ. 6.4 Ο αλγόριθμος του Stern

είσοδος: $\boldsymbol{H}$, $\boldsymbol{s}$, βάρος ω, παράμετροι p, ℓ
αρχικοποίηση: $\max\{0, k+\ell+\omega-n\} \leq p \leq \min\{k,\omega\}$, $0 \leq \ell \leq n-k$.

repeat
  Επέλεξε $\boldsymbol{P}$ » τυχαίος πίνακας αντιμετάθεσης
  $\widetilde{\boldsymbol{H}} \leftarrow \boldsymbol{Q}'\boldsymbol{H}\boldsymbol{P}$ » με απαλοιφή του Gauss στον $\boldsymbol{HP}$
  $\widetilde{\boldsymbol{s}} \leftarrow \boldsymbol{Q}'\boldsymbol{s}$ » $\begin{pmatrix} \widetilde{s}_L \\ \widetilde{s}_R \end{pmatrix}$
  Κατασκεύασε $\mathcal{L}_1, \mathcal{L}_2$
  Ταξινόμησε $\mathcal{L}_2$ σύμφωνα με την ετικέτα $\pi_{[\ell]}(\boldsymbol{Q}_{I_2}) + \widetilde{s}_L$.
  for all $(I_1, \pi_{[\ell]}(\boldsymbol{Q}_{I_1})) \in \mathcal{L}_1$ **do**
    for all $(I_2, \pi_{[\ell]}(\boldsymbol{Q}_{I_2}) + \widetilde{s}_L) \in \mathcal{L}_2$ with $\pi_{[\ell]}(\boldsymbol{Q}_{I_1}) = \pi_{[\ell]}(\boldsymbol{Q}_{I_2}) + \widetilde{s}_L$ **do**
      if $\mathrm{wt}(\pi(\boldsymbol{Q}_{I_1}) + \pi(\boldsymbol{Q}_{I_2}) + \widetilde{s}_R) = \omega - p$ **then**
        Υπολόγισε $\widetilde{\boldsymbol{e}} \in \mathbb{F}_2^n$ θέτοντας
        $\widetilde{e}_{1_i} = 1 \forall i \in I_1$
        $\widetilde{e}_{2_i} = 1 \forall i \in I_2$
        $\widetilde{e}_{k+j} = 1 \forall j \in \mathrm{supp}(\pi_{[n-k]\setminus[\ell]}(\pi(\boldsymbol{Q}_{I_1}) + \pi(\boldsymbol{Q}_{I_2}) + \widetilde{s}_R))$
      end
    end
  end
$\boldsymbol{e} \leftarrow \boldsymbol{P}\widetilde{\boldsymbol{e}}$

έξοδος: διάνυσμα $\boldsymbol{e}$

Ασυμπτωτικός συντελεστής χρονικής πολυπλοκότητας

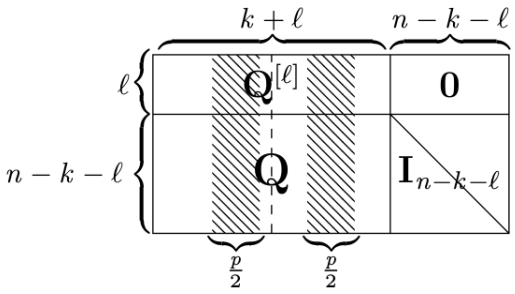
$$T^*_{Stern}(R,W) = \min_{P,L}\left\{N^*_{Stern}(R,W,P,L) + \max\left\{S^*_{Stern}(P,R), 2S^*_{Stern}(P,R) - L\right\}\right\}. \quad (6.23)$$

Για περισσότερες λεπτομέρειες, σχετικά με την ασυμπτωτική έκφραση της πιθανότητας επιτυχίας και την πλήρη ασυμπτωτική ανάλυση του αλγορίθμου (βλ. Ενότητα 8.3).

6.5 Αλγόριθμος FS-ISD

Όπως σε όλους τους ISD αλγορίθμους, ομοίως και στην περίπτωση του ISD αλγορίθμου των Finiasz - Sendrier (FS-ISD) [55] που εισήχθη το 2009, αρχικά εφαρμόζεται τυχαία μετάθεση των στηλών του πίνακα ελέγχου ισοτιμίας $\boldsymbol{H}$, από όπου προκύπτει ο πίνακας $\widetilde{\boldsymbol{H}} = \boldsymbol{H}\boldsymbol{P}$ πολλαπλασιάζοντας τον $\boldsymbol{H}$ με έναν τυχαίο πίνακα μετάθεσης $\boldsymbol{P} \in \mathbb{F}_2^{n\times n}$. Εν συνεχεία, εφαρμόζεται *μερική απαλοιφή του Gauss* (partial Gaussian Elimination) [18] στο δεξιό τμήμα του υποπίνακα[8] $\widetilde{\boldsymbol{H}}^I_J \in \mathbb{F}_2^{(n-k-\ell)\times(n-k-\ell)}$, με *σύνολα δεικτών* τα $I = \{\ell+1, \cdots, n-k\}$ και $J = \{k+\ell+1, \cdots, n\}$. Επιπλέον, προσθέτοντας γραμμές από τον μοναδιαίο πίνακα $\boldsymbol{I}_{n-k-\ell}$, προκύπτει το $\ell \times (n-k-\ell)$ μηδενικό τμήμα των στηλών που υπολείπεται, στις ℓ πρώτες

[8] όπου $\boldsymbol{A}^I_J := (a_{i,j})_{i\in I, j\in J}$ ένας πίνακας, που αποτελείται από $|I|$ γραμμές (όπου $|I|$ εκφράζει την *πληθικότητα* (cardinality) ενός συνόλου I) και $|J|$ στήλες, αντίστοιχα.



Σχήμα 6.7: Η δομή του πίνακα ελέγχου ισοτιμίας στον αλγόριθμο FS-ISD

γραμμές του υποπίνακα $\widetilde{\boldsymbol{H}_J}$. Κατά συνέπεια, προκειμένου ο πίνακας $\widetilde{\boldsymbol{H}}$ που τελικά προκύπτει, να είναι σε συστηματική μορφή, πολλαπλασιάζεται ο πίνακας $\boldsymbol{HP}$ με έναν αντιστρέψιμο πίνακα $\boldsymbol{Q'} \in \mathbb{F}_2^{(n-k)\times(n-k)}$, δηλαδή:

$$\left(\begin{array}{c|c} \boldsymbol{Q}^{[\ell]} & \mathbf{0} \\ \boldsymbol{Q} & \boldsymbol{I}_{n-k-\ell} \end{array}\right) = \boldsymbol{Q'HP}. \tag{6.24}$$

Όπως παρατηρούμε στο Σχ. 6.7, ο αλγόριθμος FS-ISD ακολουθεί την ίδια στρατηγική *αναζήτησης συγκρούσεων* με τον αλγόριθμο του Stern, με την μόνη διαφορά ότι ο υποπίνακας $\boldsymbol{Q}^{[\ell]}$ αποτελείται από $k+\ell$ στήλες, έναντι k στηλών. Συγκεκριμένα, σε κάθε επανάληψη ο FS-ISD καλείται επίσης να επιλύσει το πρόβλημα ταύτισης υποπίνακα, με παραμέτρους ℓ, k και $p \leq k+\ell$. Δοθέντος δηλαδή ενός τυχαίου πίνακα $\boldsymbol{Q}^{[\ell]} = [\boldsymbol{q}_1 \cdots \boldsymbol{q}_{k+\ell}] \in_R \mathbb{F}_2^{\ell\times(k+\ell)}$ και του αντίστοιχου συνδρόμου $\widetilde{\boldsymbol{s}}_L \in \mathbb{F}_2^{\ell}$, αναζητείται ένα σύνολο δεικτών $I \subset [k+\ell]$, $|I| \leq p$, έτσι ώστε

$$\pi_{[\ell]}(\boldsymbol{Q}_{\boldsymbol{I}}) = \sum_{i\in I} \boldsymbol{q}_i = \widetilde{\boldsymbol{s}}_L \in \mathbb{F}_2^{\ell}.$$

Ειδικότερα, για $I = I_1 \dot\cup I_2$ με $I_1 \subseteq \left[1, \frac{k+\ell}{2}\right]$ και $I_2 \subseteq \left[\frac{k+\ell}{2}+1, k+\ell\right]$ έχουμε

$$\pi_{[\ell]}(\boldsymbol{Q}_{I_1}) = \pi_{[\ell]}(\boldsymbol{Q}_{I_2}) + \widetilde{\boldsymbol{s}}_L \in \mathbb{F}_2^{\ell}. \tag{6.25}$$

Για την αναζήτηση συγκρούσεων σύμφωνα με την (6.25), κατασκευάζονται δύο λίστες:

$$\mathcal{L}_1 := \left\{(I_1, \pi_{[\ell]}(\boldsymbol{Q}_{I_1})) : I_1 \subseteq \left[1, \frac{k+\ell}{2}\right], |I_1| = \frac{p}{2}\right\},$$
$$\mathcal{L}_2 := \left\{(I_2, \pi_{[\ell]}(\boldsymbol{Q}_{I_2}) + \widetilde{\boldsymbol{s}}_L) : I_2 \subseteq \left[\frac{k+\ell}{2}+1, k+\ell\right], |I_2| = \frac{p}{2}\right\}.$$

Κατόπιν, ταξινομείται η λίστα $\mathcal{L}_2$ σύμφωνα με τις ετικέτες $\pi_\ell(\boldsymbol{Q}_{I_2}) + \widetilde{\boldsymbol{s}}_L$ και εκτελείται αναζήτηση σε όλα τα στοιχεία $\pi_\ell(\boldsymbol{Q}_{I_1})$ που περιέχονται στην λίστα $\mathcal{L}_1$, προκειμένου να εντοπιστεί ένα στοιχείο της λίστας $\mathcal{L}_2$ με το οποίο θα υπάρξει ταύτιση. Κατ' επέκταση, κάθε ζεύγος (I_1, I_2) το οποίο ικανοποιεί την εξίσωση

(6.27), αποτελεί λύση του προβλήματος ταύτισης υποπίνακα. Έπειτα, για όλες εκείνες τις υποψήφιες λύσεις, ελέγχεται αν υπάρχει κάποια που να ικανοποιεί την συνθήκη $\mathsf{wt}(\pi(\boldsymbol{Q}_{I_1})+\pi(\boldsymbol{Q}_{I_2})+\widetilde{\boldsymbol{s}}_R) = \omega - p$. Όταν εντοπιστεί το κατάλληλο ζεύγος (I_1, I_2) ο αλγόριθμος τερματίζει, λεπτομερής περιγραφή δίνεται στον Αλγ. 6.5 που ακολουθεί.

Αλγ. 6.5 Ο αλγόριθμος FS-ISD

είσοδος: $\boldsymbol{H}$, $\boldsymbol{s}$, βάρος ω.
αρχικοποίηση: $\max\{0, k+\ell+\omega-n\} \le p \le \min\{k+\ell, \omega\}$, $0 \le \ell \le n-k$.

repeat
  Επέλεξε $\boldsymbol{P}$ » τυχαίος πίνακας αντιμετάθεσης
  $\widetilde{\boldsymbol{H}} \leftarrow \boldsymbol{Q}'\boldsymbol{H}\boldsymbol{P} = \left(\begin{array}{c|c} \boldsymbol{Q}^{[\ell]} & \boldsymbol{0} \\ \boldsymbol{Q} & \boldsymbol{I}_{n-k-\ell} \end{array}\right)$ » με απαλοιφή του Gauss στον $\boldsymbol{HP}$
  $\widetilde{\boldsymbol{s}} \leftarrow \boldsymbol{Q}'\boldsymbol{s}$ » $\begin{pmatrix} \widetilde{\boldsymbol{s}}_L \\ \widetilde{\boldsymbol{s}}_R \end{pmatrix}$
  Κατασκεύασε $\mathcal{L}_1, \mathcal{L}_2$
  Ταξινόμησε $\mathcal{L}_2$ σύμφωνα με την ετικέτα $\pi_{[\ell]}(\boldsymbol{Q}_{I_2}) + \widetilde{\boldsymbol{s}}_L$.
  for all $(I_1, \pi_{[\ell]}(\boldsymbol{Q}_{I_1})) \in \mathcal{L}_1$ **do**
    for all $(I_2, \pi_{[\ell]}(\boldsymbol{Q}_{I_2}) + \widetilde{\boldsymbol{s}}_L) \in \mathcal{L}_2$ with $\pi_{[\ell]}(\boldsymbol{Q}_{I_1}) = \pi_{[\ell]}(\boldsymbol{Q}_{I_2}) + \widetilde{\boldsymbol{s}}_L$ **do**
      if $\mathsf{wt}(\pi(\boldsymbol{Q}_{I_1}) + \pi(\boldsymbol{Q}_{I_2}) + \widetilde{\boldsymbol{s}}_R) = \omega - p$ **then**
        Υπολόγισε $\widetilde{\boldsymbol{e}} \in \mathbb{F}_2^n$ θέτοντας
        $\widetilde{\boldsymbol{e}}_{1_i} = 1 \forall i \in I_1$
        $\widetilde{\boldsymbol{e}}_{2_i} = 1 \forall i \in I_2$
        $\widetilde{\boldsymbol{e}}_{k+\ell+j} = 1 \forall j \in \mathsf{supp}(\pi_{[n-k]\setminus[\ell]}(\pi(\boldsymbol{Q}_{I_1}) + \pi(\boldsymbol{Q}_{I_2}) + \widetilde{\boldsymbol{s}}_R))$
      end
    end
  end
$\boldsymbol{e} \leftarrow \boldsymbol{P}\widetilde{\boldsymbol{e}}$

έξοδος: διάνυσμα $\boldsymbol{e}$

6.5.1 Πολυπλοκότητα

Η πολυπλοκότητα του Αλγ. 6.5 εξετάζεται ως προς τις ακόλουθες ποσότητες.

Πιθανότητα επιτυχίας

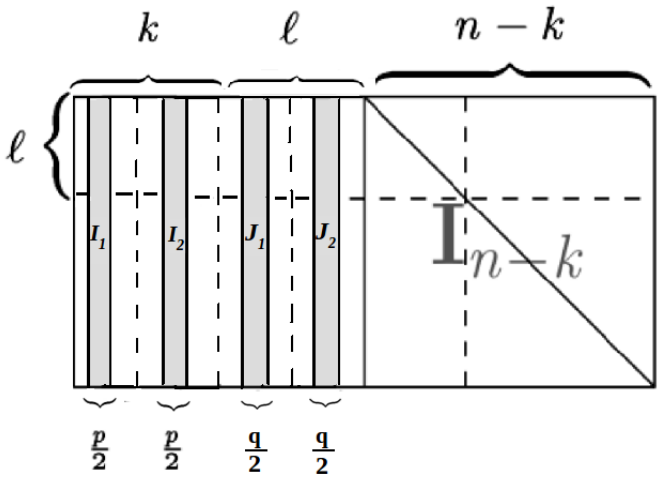
$$\mathsf{Pr}_{success} = \frac{\binom{k+\ell/2}{p/2}^2 \binom{n-k-\ell}{\omega-p}}{\binom{n}{\omega}}. \tag{6.26}$$

Ασυμπτωτικός εκθέτης αριθμού επαναλήψεων

$$N^*_{FS-ISD}(R,W,P,L) = H_2(W) - H_2\left(\frac{P}{R+L}\right)\cdot(R+L) - H_2\left(\frac{W-P}{1-R-L}\right)\cdot(1-R-L). \tag{6.27}$$

Ασυμπτωτικός συντελεστής χωρικής πολυπλοκότητας

$$S^*_{FS-ISD}(P,R,L) = H_2\left(\frac{P}{R+L}\right)\cdot\frac{R+L}{2}. \tag{6.28}$$



Σχήμα 6.8: Η δομή του πίνακα ελέγχου ισοτιμίας στον αλγόριθμο BCD.

Ασυμπτωτικός συντελεστής χρονικής πολυπλοκότητας

$$T^*_{FS-ISD}(R,W) = \min_{P,L}\left\{N^*_{FS-ISD}(R,W,P,L) + \max\left\{S^*_{FS-ISD}(P,R,L), 2S^*_{FS-ISD}(P,R,L) - L\right\}\right\}. \tag{6.29}$$

Για περισσότερες λεπτομέρειες, σχετικά με την ασυμπτωτική έκφραση της πιθανότητας επιτυχίας και την πλήρη ασυμπτωτική ανάλυση του αλγορίθμου (βλ. Ενότητα 8.4).

6.6 Αλγόριθμος BCD

Το 2011 οι Bernstein, Lange και Peters [59] εισήγαγαν έναν νέο ISD αλγόριθμο, που ονομάζεται *αποκωδικοποίηση σύγκρουσης μπάλας* (Ball-collision decoding ή BCD). Η ιδέα τους βασίστηκε στην επέκταση του Αλγ. 6.4, με την προσθήκη ενός επιπρόσθετου αριθμού q μη-μηδενικών συνιστωσών, εντός του καθορισμένου 0-παραθύρου (μήκους ℓ) (βλ. Σχ. 6.8). Επιπλέον, πρότειναν μια σειρά από τεχνικές βελτιστοποίησης, προκειμένου να μειώσουν τις πράξεις και το κόστος που απαιτείται σε κάθε επανάληψη και να επιταχύνουν την εκτέλεση του BCD συγκριτικά με προγενέστερους ISD αλγορίθμους (βλ. [54],[59]). Μετά τη διαδικασία μετασχηματισμού του πίνακα ελέγχου ισοτιμίας $\boldsymbol{H}$ στην συστηματική του μορφή

$$\left(\begin{array}{c|ccc} \boldsymbol{Q}^{[\ell]} & \boldsymbol{I}_{\ell_1} & & \\ \boldsymbol{Q} & & \boldsymbol{I}_{\ell_2} & \\ & & & \boldsymbol{I}_{n-k-\ell} \end{array}\right) = \boldsymbol{Q'HP} \tag{6.30}$$

ο αλγόριθμος BCD καλείται να επιλύσει το πρόβλημα ταύτισης υποπίνακα, με παραμέτρους ℓ, k, $p \leq k$ και $q \leq \ell$, σε κάθε επανάληψη. Δοθέντος δηλαδή ενός

τυχαίου πίνακα

$$\boldsymbol{Q}^{[\ell]} = \begin{pmatrix} \boldsymbol{Q}_1^{[\ell]} & \boldsymbol{Q}_2^{[\ell]} \\ \boldsymbol{Q}_3^{[\ell]} & \boldsymbol{Q}_4^{[\ell]} \end{pmatrix} \in_R \mathbb{F}_2^{\ell \times k}, \tag{6.31}$$

του μοναδιαίου πίνακα

$$\boldsymbol{I}_\ell = \begin{pmatrix} \boldsymbol{I}_{\ell_1} & 0 \\ 0 & \boldsymbol{I}_{\ell_2} \end{pmatrix} \in_R \mathbb{F}_2^{\ell \times \ell} \tag{6.32}$$

και του συνδρόμου

$$\widetilde{\boldsymbol{s}}_L = \begin{pmatrix} \widetilde{\boldsymbol{s}}_{L_1} \\ \widetilde{\boldsymbol{s}}_{L_2} \end{pmatrix} \tag{6.33}$$

αναζητούνται τα σύνολα δεικτών $I_1 \subseteq \left[1, \frac{k}{2}\right]$, $I_2 \subseteq \left[\frac{k}{2}+1, k\right]$ και $J_1 \subseteq \left[k+1, k+\frac{\ell}{2}\right]$, $J_2 \subseteq \left[k+\frac{\ell}{2}+1, k+\ell\right]$ αντίστοιχα, έτσι ώστε να ισχύει

$$\pi_{[\ell]}(\boldsymbol{Q}_{1 I_1}) + \pi_{[\ell]}(\boldsymbol{Q}_{2 I_2}) + \pi_{[\ell]}(\boldsymbol{I}_{J_1}) = \widetilde{\boldsymbol{s}}_{L_1} \in \mathbb{F}_2^\ell, \tag{6.34}$$

$$\pi_{[\ell]}(\boldsymbol{Q}_{3 I_1}) + \pi_{[\ell]}(\boldsymbol{Q}_{4 I_2}) + \pi_{[\ell]}(\boldsymbol{I}_{J_2}) = \widetilde{\boldsymbol{s}}_{L_2} \in \mathbb{F}_2^\ell. \tag{6.35}$$

Θα πρέπει δηλαδή να ισχύει

$$\pi_{[\ell]}(\boldsymbol{Q}_I) + \pi_{[\ell]}(\boldsymbol{I}_J) = \widetilde{\boldsymbol{s}}_L \in \mathbb{F}_2^\ell \tag{6.36}$$

όπου $I = I_1 \dot\cup I_2$ και $J = J_1 \dot\cup J_2$. Για την ακρίβεια, κατασκευάζονται δύο λίστες:

$$\mathcal{L}_1 := \left\{ (I_1, J_1, \pi_{[\ell]}(\boldsymbol{Q}_I)) : I_1 \subseteq \left[1, \frac{k}{2}\right], |I_1| = \frac{p}{2}, J_1 \subseteq \left[k+1, k+\frac{\ell}{2}\right], |J_1| = \frac{q}{2} \right\},$$
$$\mathcal{L}_2 := \left\{ (I_2, J_2, \pi_{[\ell]}(\boldsymbol{I}_J) + \widetilde{\boldsymbol{s}}_L) : I_2 \subseteq \left[\frac{k}{2}+1, k\right], |I_2| = \frac{p}{2}, J_2 \subseteq \left[k+\frac{\ell}{2}+1, k+\ell\right], |J_2| = \frac{q}{2} \right\}.$$

Κατόπιν, ταξινομείται η λίστα $\mathcal{L}_2$ σύμφωνα με τις *ετικέτες* (labels) $\pi_{[\ell]}(\boldsymbol{I}_J) + \widetilde{\boldsymbol{s}}_L$ και εκτελείται αναζήτηση σε όλα τα στοιχεία $\pi_\ell(\boldsymbol{Q}_{I_1})$ που περιέχονται στην λίστα $\mathcal{L}_1$, προκειμένου να εντοπιστούν συγκρούσεις με τη λίστα $\mathcal{L}_2$. Κατ' επέκταση, κάθε ζεύγος (I, J) το οποίο ικανοποιεί την εξίσωση (6.36), αποτελεί λύση του προβλήματος ταύτισης υποπίνακα. Έπειτα, για όλες εκείνες τις υποψήφιες λύσεις, ελέγχεται αν υπάρχει κάποια που να ικανοποιεί την συνθήκη $\mathsf{wt}(\pi(\boldsymbol{Q}_I) + \pi(\boldsymbol{I}_J) + \widetilde{\boldsymbol{s}}_R) = \omega - p - q$. Όταν εντοπιστεί το κατάλληλο ζεύγος (I, J) ο αλγόριθμος τερματίζει, λεπτομερής περιγραφή δίνεται στον Αλγ. 6.6.

6.6.1 Πολυπλοκότητα

Η πολυπλοκότητα του Αλγ. 6.6 εξετάζεται ως προς τις ακόλουθες ποσότητες.

Αλγ. 6.6 Ο αλγόριθμος BCD

είσοδος: $\boldsymbol{H}$, $\boldsymbol{s}$, βάρος ω, παράμετροι p, ℓ, q

αρχικοποίηση: $0 \leq \ell \leq n-k$, $0 \leq q \leq \min\{\ell, \omega\}$, $\max\{0, k+\ell+\omega-n\} \leq p \leq \min\{k, \omega\}$.

repeat
  Επέλεξε $\boldsymbol{P}$ » τυχαίος πίνακας αντιμετάθεσης
  $\widetilde{\boldsymbol{H}} \leftarrow \boldsymbol{Q}'\boldsymbol{H}\boldsymbol{P} = \left(\begin{array}{c|cc} \boldsymbol{Q}^{[\ell]} & \boldsymbol{I}_\ell & \\ \boldsymbol{Q} & & \boldsymbol{I}_{n-k-\ell} \end{array}\right)$, όπου $\boldsymbol{I}_\ell = \begin{pmatrix} \boldsymbol{I}_{\ell_1} & 0 \\ 0 & \boldsymbol{I}_{\ell_2} \end{pmatrix}$
  $\widetilde{\boldsymbol{s}} \leftarrow \boldsymbol{Q}'\boldsymbol{s}$ » $\begin{pmatrix} \widetilde{\boldsymbol{s}}_L \\ \widetilde{\boldsymbol{s}}_R \end{pmatrix}$, όπου $\widetilde{\boldsymbol{s}}_L = \begin{pmatrix} \widetilde{\boldsymbol{s}}_{L_1} \\ \widetilde{\boldsymbol{s}}_{L_2} \end{pmatrix}$
  Κατασκεύασε $\mathcal{L}_1, \mathcal{L}_2$
  Ταξινόμησε $\mathcal{L}_2$ σύμφωνα με την ετικέτα $\pi_{[\ell]}(\boldsymbol{I}_J) + \widetilde{\boldsymbol{s}}_L$.
  for all $(I_1, J_1, \pi_{[\ell]}(\boldsymbol{Q}_I))$ **do**
    for all $(I_2, J_2, \pi_{[\ell]}(\boldsymbol{I}_J) + \widetilde{\boldsymbol{s}}_L)$ with $\pi_{[\ell]}(\boldsymbol{Q}_I) + \pi_{[\ell]}(\boldsymbol{I}_J) = \widetilde{\boldsymbol{s}}_L$ **do**
      if $\mathrm{wt}(\pi(\boldsymbol{Q}_I) + \pi(\boldsymbol{I}_J) + \widetilde{\boldsymbol{s}}_R) = \omega - p - q$ **then**
        Υπολόγισε $\widetilde{\boldsymbol{e}} \in \mathbb{F}_2^n$ θέτοντας
        $\widetilde{\boldsymbol{e}}_{1,1_i} = 1 \forall i \in I_1$
        $\widetilde{\boldsymbol{e}}_{1,2_i} = 1 \forall i \in I_2$
        $\widetilde{\boldsymbol{e}}_{2,1_j} = 1 \forall j \in J_1$
        $\widetilde{\boldsymbol{e}}_{2,2_j} = 1 \forall j \in J_2$
        $\widetilde{\boldsymbol{e}}_{k+\ell+z} = 1 \forall z \in \mathrm{supp}(\pi_{[n-k]\setminus[\ell]}(\pi(\boldsymbol{Q}_I) + \pi(\boldsymbol{I}_J) + \widetilde{\boldsymbol{s}}_R))$
      end
    end
  end
$\boldsymbol{e} \leftarrow \boldsymbol{P}\widetilde{\boldsymbol{e}}$

έξοδος: διάνυσμα $\boldsymbol{e}$

Πιθανότητα επιτυχίας

$$\mathsf{Pr}_{success} = \frac{\left(\binom{k/2}{p/2}\binom{\ell/2}{q/2}\right)^2 \binom{n-k-\ell}{\omega-p-q}}{\binom{n}{\omega}}. \tag{6.37}$$

Ασυμπτωτικός εκθέτης αριθμού επαναλήψεων

$$N^*_{BCD}(R, W, P, Q, L) = H_2(W) - H_2\left(\frac{P}{R}\right)\cdot R - H_2\left(\frac{Q}{L}\right)\cdot L - H_2\left(\frac{W-P-Q}{1-R-L}\right)\cdot(1-R-L). \tag{6.38}$$

Ασυμπτωτικός συντελεστής χωρικής πολυπλοκότητας

$$S^*_{BCD}(R, L, P, Q) = H_2\left(\frac{P}{R}\right)\cdot\frac{R}{2} + H_2\left(\frac{Q}{L}\right)\cdot\frac{L}{2}. \tag{6.39}$$

Ασυμπτωτικός συντελεστής χρονικής πολυπλοκότητας

$$T^*_{BCD}(R, W) = \min_{P,L}\left\{N^*_{BCD}(R, W, P, Q, L) + \max\left\{S^*_{BCD}(R, L, P, Q), 2S^*_{BCD}(R, L, P, Q) - L\right\}\right\}. \tag{6.40}$$

Για περισσότερες λεπτομέρειες, σχετικά με την ασυμπτωτική έκφραση της πιθανότητας επιτυχίας και την πλήρη ασυμπτωτική ανάλυση του αλγορίθμου (βλ. Ενότητα 8.5).

6.7 Βελτιωμένοι ISD αλγόριθμοι

6.7.1 Αλγόριθμος MMT-ISD

Το 2011 οι May *et al.* [63] επινόησαν ένα νέο ISD αλγόριθμο, που αποκαλείται MMT-ISD, η ιδέα του οποίου προήλθε από τον Αλγ. 6.5 και βασίζεται στην *τεχνική αναπαράστασης* (representation technique). Η εν λόγω τεχνική εισήχθη αρχικά το 2010 από τους Howgrave-Graham και Joux [56] και βελτιώθηκε ακολούθως το 2011 από τους Becker *et al.* [61].

Παρομοίως με τον Αλγ. 6.5, ο αλγόριθμος MMT-ISD καλείται να επιλύσει το πρόβλημα ταύτισης υποπίνακα (βλ. ορισμό 6.2) με παραμέτρους ℓ, k και $p \leq k + \ell$. Ωστόσο, η τεχνική επίλυσης που ακολουθείται είναι πιο αποδοτική, καθώς χρησιμοποιούνται περισσότερες αναπαραστάσεις της λύσης I. Συγκεκριμένα, για τυχαίο πίνακα $\boldsymbol{Q} = [\boldsymbol{q}_1 \cdots \boldsymbol{q}_{k+\ell}] \in_R \mathbb{F}_2^{\ell \times (k+\ell)}$ και σύνδρομο $\widetilde{\boldsymbol{s}}_L \in \mathbb{F}_2^{\ell}$, το σύνολο I (μεγέθους ακριβώς p), θεωρείται ότι αποτελεί λύση του προβλήματος εάν προκύπτει από δύο σύνολα I_1 και I_2 μεγέθους $\frac{p}{2}$ το καθένα. Η επιλογή των συνόλων I_1 και I_2 είναι παρόμοια με την τεχνική αναπαράστασης [56] και εν αντιθέσει με τον FS-ISD, πραγματοποιείται από ολόκληρο το σύνολο $[k + \ell]$ και όχι *ξένα* (disjoint) σύνολα μήκους $\frac{k+\ell}{2}$ [42]. Επιπλέον, η επιλογή αυτή έχει σαν αποτέλεσμα $\binom{p}{p/2} \approx 2^p$ διαφορετικές διαμερίσεις της λύσης $I = I_1 \dot{\cup} I_2$ με

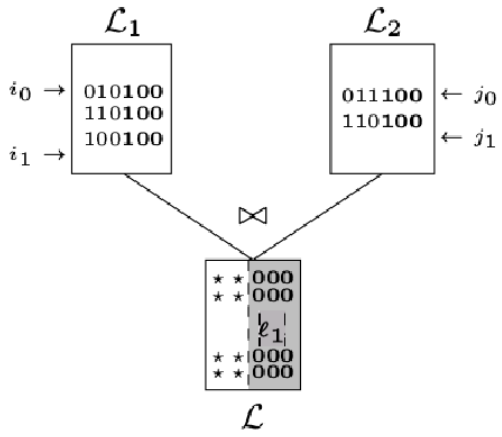
$$\sum_{i \in I_1} \boldsymbol{q}_i = \sum_{i \in I_2} \boldsymbol{q}_i + \widetilde{\boldsymbol{s}}_L \in \mathbb{F}_2^{\ell}. \tag{6.41}$$

Επομένως, σκοπός του MMT-ISD είναι η εύρεση μιας εκ των 2^p διαφορετικών αναπαραστάσεων της λύσης I, που ικανοποιούν την εξίσωση (6.41), με σταθερή πιθανότητα επιτυχίας. Ωστόσο, δεν συνυπολογίζονται όλα τα πιθανά αθροίσματα των στοιχείων, που περιέχονται στα σύνολα I_1 και I_2, παρά μόνο εκείνα που ικανοποιούν επιπρόσθετους περιορισμούς. Προκειμένου, να διαμορφωθούν οι εν λόγω περιορισμοί, εισάγονται οι παράμετροι ℓ_1 και ℓ_2, με $\ell_1 + \ell_2 = \ell$ οι οποίες αντιστοιχούν στα ξένα υποσύνολα $L_1, L_2 \subset [\ell]$ (μεγέθους ℓ_1 και ℓ_2, αντίστοιχα).

Στο πρώτο βήμα του αλγορίθμου κατασκευάζονται επιμέρους λύσεις που ήδη αθροίζουν στο επιθυμητό σύνδρομο $\widetilde{\boldsymbol{s}}$ στις ℓ_2 θέσεις, του συνόλου δεικτών L_2. Για την εύρεση των λύσεων αυτών, κατασκευάζονται οι λίστες:

$$\mathcal{L}_1 := \left\{ (I_1, \pi_{[\ell_1]}(\boldsymbol{Q}_{I_1})) : I_1 \subset [k+\ell], \quad |I_1| = \tfrac{p}{2} \quad \text{και} \quad \pi_{[\ell_2]}(\boldsymbol{Q}_{\boldsymbol{I_1}}) = \boldsymbol{0} \in \mathbb{F}_2^{\ell_2} \right\}$$
$$\mathcal{L}_2 := \left\{ (I_2, \pi_{[\ell_1]}(\boldsymbol{Q}_{I_2}) + \widetilde{\boldsymbol{s}}_{L_1}) : I_2 \subset [k+\ell], \quad |I_2| = \tfrac{p}{2} \quad \text{και} \quad \pi_{[\ell_2]}(\boldsymbol{Q}_{\boldsymbol{I_2}}) = \widetilde{\boldsymbol{s}}_{L_2} \in \mathbb{F}_2^{\ell_2} \right\}.$$

Παρατηρούμε ότι, από τις 2^p πιθανές αναπαραστάσεις της λύσης που ικανοποιούν την (6.41) εξετάζονται μόνο εκείνες, των οποίων το άθροισμα των στηλών είναι ίσο με μηδέν στο σύνολο δεικτών L_2, δηλαδή ικανοποιούν τον περιορισμό



Σχήμα 6.9: Ο αλγόριθμος MERGE-JOIN παίρνει ως είσοδο 2 ταξινομημένες λίστες $\mathcal{L}_1$, $\mathcal{L}_2$ και με την βοήθεια τεσσάρων βοηθητικών μετρητών, εξετάζει προοδευτικά από ποια λίστα θα αντλήσει στοιχεία για να τα βάλει στην συγχωνευμένη λίστα $\mathcal{L}$.

$\pi_{[\ell_2]}(\boldsymbol{Q}_{I_1}) = \mathbf{0} \in \mathbb{F}_2^{\ell_2}$. Συνεπώς, ο αναμενόμενος αριθμός των λύσεων που εν τέλει διατηρούνται, είναι $2^{p-\ell_2}$.

Στη συνέχεια ταξινομείται η λίστα $\mathcal{L}_2$ σύμφωνα με τις ετικέτες $\pi_{[\ell_1]}(\boldsymbol{Q}_{I_2}) + \widetilde{\boldsymbol{s}}_{L_1}$ και εκτελείται αναζήτηση στα στοιχεία $\pi_{[\ell_1]}(\boldsymbol{Q}_{I_1})$ που περιέχονται στην λίστα $\mathcal{L}_1$, προκειμένου να εντοπιστούν συγκρούσεις με τη λίστα $\mathcal{L}_2$.

Η διαδικασία ταύτισης μπορεί να υλοποιηθεί πιο αποδοτικά, αν οι λίστες $\mathcal{L}_1, \mathcal{L}_2$ κατασκευαστούν με τέτοιο τρόπο, ώστε να περιέχουν δυαδικά διανύσματα $\boldsymbol{x}_1 \cdots \boldsymbol{x}_{|\mathcal{L}_1|}$ και $\boldsymbol{y}_1 \cdots \boldsymbol{y}_{|\mathcal{L}_2|}$ μήκους $k + \ell$ η καθεμία, αντίστοιχα. Έτσι, κάθε ζεύγος $(\boldsymbol{x}_i, \boldsymbol{y}_j)$, με $\mathrm{wt}(\boldsymbol{x}_i + \boldsymbol{y}_j) = p$ για το οποίο ικανοποιείται η εξίσωση (6.41), αποτελεί λύση του προβλήματος ταύτισης υποπίνακα. Μετά, από την κατασκευή τους, ταξινομείται λεξικογραφικά η λίστα $\mathcal{L}_1$ σύμφωνα με την ετικέτα $L_1(\boldsymbol{x}_i) := (\boldsymbol{Q}\boldsymbol{x}_i)_{[\ell_1]}$ και η λίστα $\mathcal{L}_2$ σύμφωνα με τις ετικέτες $L_2(\boldsymbol{y}_j) := (\boldsymbol{Q}\boldsymbol{y}_j)_{[\ell_1]} + \widetilde{\boldsymbol{s}}_{L_1}$ και κατόπιν εκτελείται η διαδικασία αναζήτησης συγκρούσεων. Η παραπάνω διαδικασία ταύτισης, υλοποιείται μέσω ενός αλγορίθμου που ονομάζεται Merge-Join, η ιδέα του οποίου προήλθε από τον κλασσικό αλγόριθμο που παρουσίασε το 1998 ο Knuth [35]. Συγκεκριμένα, πρόκειται για έναν αλγόριθμο που χρησιμοποιείται για την επίλυση του προβλήματος ταύτισης υποπίνακα, η ακριβής περιγραφή του οποίου δίνεται στον Αλγ. 6.7.

Για την ανίχνευση των συγκρούσεων, αρχικοποιούνται δύο μετρητές i και j (ξεκινώντας από την αρχή των λιστών $\mathcal{L}_1$ και $\mathcal{L}_2$) οι οποίοι υποδεικνύουν τα στοιχεία $\boldsymbol{x}_i$ και $\boldsymbol{y}_j$, όπως απεικονίζεται στο Σχ. 6.9. Εφόσον, τα στοιχεία αυτά δεν οδηγούν σε κάποια σύγκρουση, αυξάνεται είτε ο δείκτης i, είτε ο δείκτης j ανάλογα με την σχετική σειρά των ετικετών $L_1(\boldsymbol{x}_i)$ και $L_2(\boldsymbol{y}_j)$. Στην περίπτωση σύγκρουσης, δηλαδή $L_1(\boldsymbol{x}_i) = L_2(\boldsymbol{y}_j)$, αρχικοποιούνται τέσσερις βοηθητικοί μετρη-

Αλγ. 6.7 Ο αλγόριθμος MERGE-JOIN

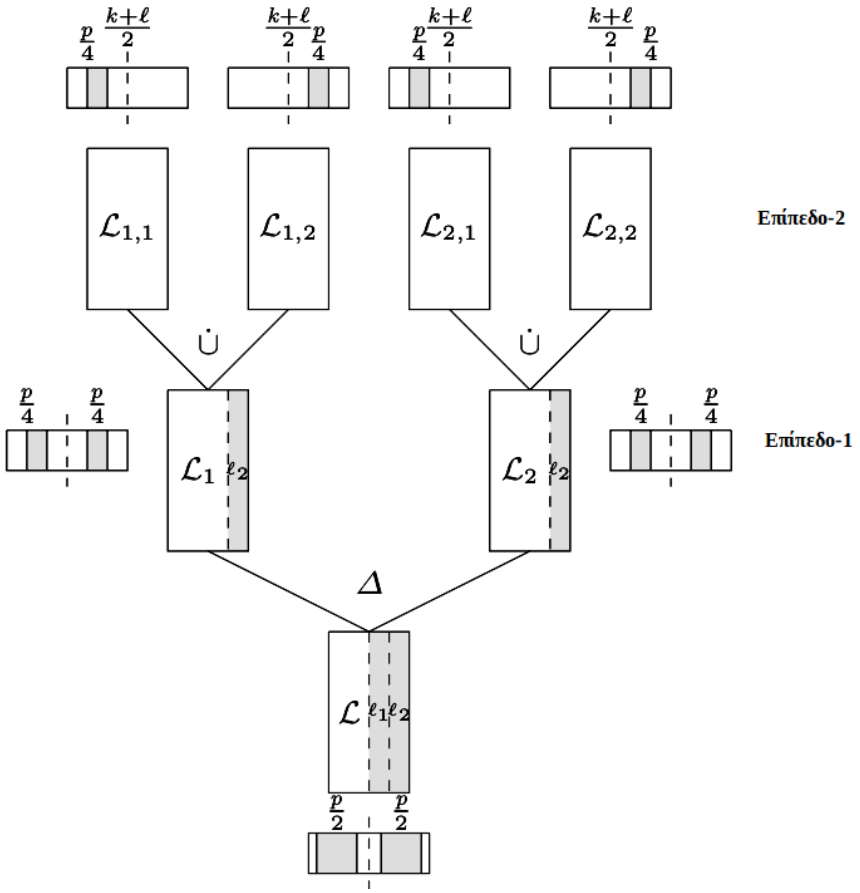
είσοδος: $\mathcal{L}_1$, $\mathcal{L}_2$, ℓ_1 p, $\widetilde{s}_{L_1} \in \mathbb{F}_2^{\ell_1}$

1: Ταξινόμησε λεξικογραφικά $\mathcal{L}_1$ και $\mathcal{L}_2$ σύμφωνα με τις ετικέτες $L_1(\boldsymbol{x}_i) := (\boldsymbol{Q}\boldsymbol{x}_i)_{[\ell_1]}$ and $L_2(\boldsymbol{y}_j) := (\boldsymbol{Q}\boldsymbol{y}_j)_{[\ell_1]} + \widetilde{s}_{L_1}$.
2: Θέσε μετρητή συγκρούσεων $C \leftarrow 0$ και $\mathcal{L} = \varnothing$. Έστω $i \leftarrow 0$ και $j \leftarrow 0$
3: **while** $i < |\mathcal{L}_1|$ and $j < |\mathcal{L}_2|$ **do**
4: **if** $L_1(\boldsymbol{x}_i) <_{lex} L_2(\boldsymbol{y}_j)$ **then** $i = i + 1$
5: **if** $L_1(\boldsymbol{x}_i) >_{lex} L_2(\boldsymbol{y}_j)$ **then** $j = j + 1$
6: **if** $L_1(\boldsymbol{x}_i) = L_2(\boldsymbol{y}_j)$ **then**
7: Let $i_0, i_1 \leftarrow i$ and $j_0, j_1 \leftarrow j$
8: **while** $i_1 < |\mathcal{L}_1|$ and $L_1(\boldsymbol{x}_{i_1}) = L_1(\boldsymbol{x}_{i_0})$ **do** $i_1 = i_1 + 1$
9: **while** $j_1 < |\mathcal{L}_2|$ and $L_2(\boldsymbol{y}_{j_1}) = L_1(\boldsymbol{y}_{j_0})$ **do** $j_1 = j_1 + 1$
10: **for** $i \leftarrow i_0$ to $i_1 - 1$ **do**
11: **for** $j \leftarrow j_0$ to $j_1 - 1$ **do**
12: $C = C + 1$
13: Εισήγαγε την σύγκρουση $\boldsymbol{x}_i + \boldsymbol{y}_j$ στην λίστα L (δίχως φιλτράρισμα)
14: Let $i \leftarrow i_1, j \leftarrow j_1$

έξοδος: $\mathcal{L}, C$ » $\mathcal{L} = \mathcal{L}_1 \bowtie \mathcal{L}_2$

τές i_0, i_1 και j_0, j_1, στις τιμές i και j αντίστοιχα. Ωστόσο, οι μετρητές i_1 και j_1 μπορούν να αυξηθούν περαιτέρω, εφόσον τα στοιχεία των λιστών διατηρούν τις ίδιες ετικέτες, ενώ οι μετρητές i_0 και j_0 καταγράφουν την πρώτη σύγκρουση (i, j) μεταξύ των ετικετών $L_1(\boldsymbol{x}_i)$ και $L_2(\boldsymbol{y}_j)$. Μέσω της διαδικασίας αυτής ορίζονται δύο σύνολα $C_1 = \left\{\boldsymbol{x}_{i_0}, \cdots, \boldsymbol{x}_{i_1}\right\}$ και $C_2 = \left\{\boldsymbol{y}_{i_0}, \cdots, \boldsymbol{y}_{i_1}\right\}$, τέτοια ώστε όλοι οι πιθανοί συνδυασμοί οδηγούν σε μία σύγκρουση, δηλαδή το σύνολο $C_1 \times C_2$ μπορεί να προστεθεί στην λίστα εξόδου $\mathcal{L}$. Εντούτοις, η διαδικασία συνεχίζεται στη γρ. 14 του Αλγ. 6.7, έως ότου κάποιος από τους μετρητές i, j φτάσει στο τέλος της αντίστοιχης λίστας. Επιπλέον, οι *ασυνεπείς λύσεις* (incosistent solutions), δηλαδή όλα τα διανύσματα ταύτισης, με $\mathsf{wt}(\boldsymbol{x}_i + \boldsymbol{y}_j) \neq p$) σε συνδυασμό με τα *διπλότυπα* (duplicates), δηλαδή διανύσματα ταύτισης που υπάρχουν ήδη, $\boldsymbol{x}_i + \boldsymbol{y}_j = \boldsymbol{x}_k + \boldsymbol{y}_\ell$, εξαιρούνται από την διαδικασία. Ιδιαίτερα χρήσιμος είναι ο μετρητής συγκρούσεων C, καθώς καταδεικνύει τον χρόνο που απαιτείται, προκειμένου να αφαιρεθούν οι ασταθείς λύσεις και τα διπλότυπα από την διαδικασία ταύτισης.

Εν συνεχεία, στο δεύτερο βήμα του αλγορίθμου εκτελείται επιπλέον διαμέριση των συνόλων δεικτών $I_1 = I_{1,1} \dot{\cup} I_{1,2}$, με $|I_{1,1}| = |I_{1,2}| = \frac{p}{4}$, όπου $I_{1,1} \subset [1, \frac{k+\ell}{2}]$, $I_{1,2} \subset [\frac{k+\ell}{2} + 1, k + \ell]$, καθώς επίσης και $I_2 = I_{2,1} \dot{\cup} I_{2,2}$, με $|I_{2,1}| = |I_{2,2}| = \frac{p}{4}$, όπου $I_{2,1} \subset [1, \frac{k+\ell}{2}]$ και $I_{2,2} \subset [\frac{k+\ell}{2} + 1, k + \ell]$, αντίστοιχα. Ακολουθώντας την ίδια διαδι-



Σχήμα 6.10: Η βασική ιδέα του αλγορίθμου MMT-ISD

κασία, οι λίστες που προκύπτουν είναι

$$\mathcal{L}_{1,1} := \left\{(I_{1,1}, \pi_{[\ell_2]}(\boldsymbol{Q}_{I_{1,1}})) : I_{1,1} \subset [1, \tfrac{k+\ell}{2}],\ \ |I_{1,1}| = \tfrac{p}{4}\right\}$$
$$\mathcal{L}_{1,2} := \left\{(I_{1,2}, \pi_{[\ell_2]}(\boldsymbol{Q}_{I_{1,2}})) : I_{1,2} \subset [\tfrac{k+\ell}{2}+1, k+\ell],\ \ |I_{1,2}| = \tfrac{p}{4}\right\}$$
$$\mathcal{L}_{2,1} := \left\{(I_{2,1}, \pi_{[\ell_2]}(\boldsymbol{Q}_{I_{2,1}})) : I_{2,1} \subset [1, \tfrac{k+\ell}{2}],\ \ |I_{2,1}| = \tfrac{p}{4}\right\}$$
$$\mathcal{L}_{2,2} := \left\{(I_{2,2}, \pi_{[\ell_2]}(\boldsymbol{Q}_{I_{2,2}}) + \widetilde{\boldsymbol{s}}_{L_2}) : I_{2,2} \subset [\tfrac{k+\ell}{2}+1, k+\ell],\ \ |I_{2,2}| = \tfrac{p}{4}\right\}.$$

Έπειτα, ταξινομείται η λίστα $\mathcal{L}_{1,2}$ σύμφωνα με την ετικέτα $\pi_{[\ell_2]}(\boldsymbol{Q}_{I_{1,2}})$ και εκτελείται αναζήτηση σε όλα τα στοιχεία $\pi_{[\ell_2]}(\boldsymbol{Q}_{I_{1,1}})$ που περιέχονται στην $\mathcal{L}_{1,1}$, ώστε να εντοπιστεί ένα στοιχείο της λίστας $\mathcal{L}_{1,2}$, με το οποίο θα υπάρξει ταύτιση. Όμοια είναι και η διαδικασία αναζήτησης συγκρούσεων στις λίστες $\mathcal{L}_{2,1}$ και $\mathcal{L}_{2,2}$, στις οποίες όμως χρησιμοποιούνται τα $\pi_{[\ell_2]}(\boldsymbol{Q}_{I_{2,1}})$ και $\pi_{[\ell_2]}(\boldsymbol{Q}_{I_{2,2}}) + \widetilde{\boldsymbol{s}}_{L_2}$ ως ετικέτες αντίστοιχα.

Η κατασκευή των λιστών σε κάθε επίπεδο απεικονίζεται γραφικά στο Σχ. 6.10. Τα οριζόντια ορθογώνια που βρίσκονται άνω και πλευρικά των λιστών, αναπαριστούν την δομή των συνόλων δεικτών $I_{i,j}$, τα οποία περιέχονται σε ξε-

χωριστές λίστες. Για παράδειγμα, η λίστα $\mathcal{L}_{1,1}$ που βρίσκεται στο επίπεδο-2, περιλαμβάνει το σύνολο δεικτών $I_{1,1}$ του οποίου οι $\frac{p}{4}$ μη-μηδενικές συντεταγμένες, εκτείνονται στο πρώτο μισό του πλήρους συνόλου $[k+\ell]$, κάτι που δηλώνεται από τη γκρίζα σκίαση.

Παρατήρηση 6.3. Η κατασκευή των λιστών $\mathcal{L}_1$ και $\mathcal{L}_2$, μέσω διαμερίσεων $I_1 = I_{1,1} \dot\cup I_{1,2}$ και $I_2 = I_{2,1} \dot\cup I_{2,2}$, μειώνει τον αριθμό των αναπαραστάσεων. Παρότι, αυτό έχει σαν αποτέλεσμα την απώλεια κάποιων λύσεων, εντούτοις επωφελείται ο αλγόριθμος ως προς την πολυπλοκότητα. Συγκεκριμένα, αντί να εξετάζεται κάθε υποσύνολο $I_1 \subset I$ μεγέθους $\frac{p}{2}$, λαμβάνεται κάθε υποσύνολο I_1 με ίσο αριθμό $\frac{p}{4}$ δεικτών, προερχόμενο είτε από το σύνολο $[1, \frac{k+\ell}{2}]$, είτε από $[\frac{k+\ell}{2}+1, k+\ell]$, αντίστοιχα. Άρα, για κάθε λύση που ικανοποιεί την (6.41) ο αλγόριθμος μπορεί να αξιοποιήσει το πολύ $\binom{p/2}{p/4}^2$ αντί $\binom{p}{p/2}$ διαφορετικών αναπαραστάσεων. Ασυμπτωτικά, αυτή η διαφορά είναι αμελητέα, εφόσον και οι δύο όροι είναι ίσοι με $2^{p(1-o(1))}$, κάτι που αποδεικνύεται εύκολα από το πόρισμα 2.33. Ακριβής περιγραφή του MMT-ISD δίνεται από τον Αλγ. 6.8.

Αλγ. 6.8 Ο αλγόριθμος MMT-ISD

είσοδος: $\boldsymbol{H}$, $\boldsymbol{s}$, βάρος ω, παράμετροι p, ℓ, ℓ_1, ℓ_2
αρχικοποίηση: $\ell = \ell_1 + \ell_2$

repeat
  Επέλεξε $\boldsymbol{P}$ » τυχαίος πίνακας αντιμετάθεσης
  $\widetilde{\boldsymbol{H}} = \boldsymbol{Q}'\boldsymbol{H}\boldsymbol{P} = \left(\boldsymbol{Q} \;\middle|\; \begin{matrix}\boldsymbol{0}\\ \boldsymbol{I}_{n-k-\ell}\end{matrix}\right)$ » με απαλοιφή του Gauss στον HP
  $\widetilde{\boldsymbol{s}} \leftarrow \boldsymbol{Q}'\boldsymbol{s}$
  Υπολόγισε $\mathcal{L}_1, \mathcal{L}_2$
  $\mathcal{L} \leftarrow MERGE - JOIN(\mathcal{L}_1, \mathcal{L}_2, p, \boldsymbol{Q}, \widetilde{\boldsymbol{s}}_L)$
  for all $(I_1, I_2) \in \mathcal{L}$ **do**
    if $\mathrm{wt}(\pi(\boldsymbol{Q}_I) + \widetilde{\boldsymbol{s}}_R) = \omega - |I|$ **then**
      Υπολόγισε $\widetilde{\boldsymbol{e}} \in \mathbb{F}_2^n$ θέτοντας
      $\widetilde{\boldsymbol{e}}_i = 1 \forall i \in I$
      $\widetilde{\boldsymbol{e}}_{k+\ell+j} = 1 \forall j \in \mathrm{supp}(\pi_{[n-k]\setminus[\ell]}(\pi(\boldsymbol{Q}_I) + \widetilde{\boldsymbol{s}}_R))$
    end
  end
$\boldsymbol{e} \leftarrow \boldsymbol{P}\widetilde{\boldsymbol{e}}$

έξοδος: διάνυσμα $\boldsymbol{e}$

6.7.2 Πολυπλοκότητα

Ο Αλγ. 6.8 επιτυγχάνει την εύρεση μιας αναπαράστασης της λύσης I, με πιθανότητα τουλάχιστον $\frac{1}{2}$, σύμφωνα με το [63, Θεώρημα 2]. Η πολυπλοκότητα του αλγορίθμου εξετάζεται ανά επίπεδο, ως προς τις ακόλουθες ποσότητες.

Πιθανότητα επιτυχίας

$$\mathsf{Pr}_{success} = \frac{1}{2} \cdot \left(\frac{\binom{(k+\ell)/2}{p/2}^2 \binom{n-k-\ell}{\omega-p}}{\binom{n}{\omega}} \right). \tag{6.42}$$

Ασυμπτωτικός εκθέτης αριθμού επαναλήψεων

$$N^*_{MMT-ISD}(R,W,P,L) = H_2(W) - H_2\left(\frac{P}{R+L}\right)\cdot(R+L) - H_2\left(\frac{W-P}{1-R-L}\right)\cdot(1-R-L). \tag{6.43}$$

Συνολικός ασυμπτωτικός συντελεστής χωρικής πολυπλοκότητας

$$S^*_{MMT-ISD}(P,R,L,L_2) = \max\left\{S^*_{MMT-ISD(2)}(R,P,L), S^*_{MMT-ISD(1)}(R,P,L,L_2)\right\} \tag{6.44}$$

όπου $S^*_{MMT-ISD(1)}$ και $S^*_{MMT-ISD(2)}$ εκφράζουν τους συντελεστές χωρικής πολυπλοκότητας για τα επίπεδα 1 και 2, αντίστοιχα.

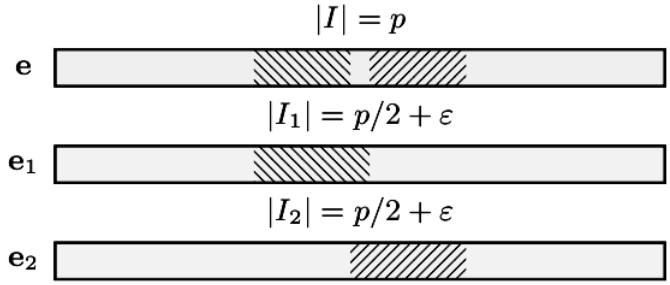
Συνολικός ασυμπτωτικός συντελεστής χρονικής πολυπλοκότητας

$$T^*_{MMT-ISD}(R,W) = \min_{P,L}\Big\{N^*_{MMT-ISD}(1)(R,W,P,L) + \max\Big\{S^*_{MMT-ISD(2)}(R,P,L), S^*_{MMT-ISD(1)}(R,P,L,L_2), 2S^*_{MMT-ISD(1)}(R,P,L,L_2) - L_1\Big\}\Big\}. \tag{6.45}$$

Για περισσότερες λεπτομέρειες, σχετικά με την πλήρη ασυμπτωτική ανάλυση του αλγορίθμου (βλ. Ενότητα 8.7).

6.7.3 Αλγόριθμος BJMM-ISD

Το 2012 οι Becker *et al.* [65] εισήγαγαν έναν νέο βελτιωμένο ISD αλγόριθμο που ονομάζεται BJMM-ISD, ως επέκταση του Αλγ. 6.8. Όπως αναλύσαμε ο Αλγ. 6.8, επιλέγει τα σύνολα I_1, I_2 από το σύνολο $[k+\ell]$, έτσι ώστε να είναι μεταξύ τους ξένα, δηλαδή $I_1 \cap I_2 = \varnothing$. Αυτό σημαίνει ότι, κάθε ένα από τα p στοιχεία του συνόλου $I = I_1 \dot\cup I_2$, μπορεί να εμφανιστεί σαν στοιχείο είτε του I_1, είτε του I_2. Αντιθέτως, ο αλγόριθμος BJMM-ISD επιλέγει τεμνόμενα σύνολα, δηλαδή $I_1 \cap I_2 \neq \varnothing$ από το σύνολο $[k+\ell]$. Επιτρέπει δηλαδή, ακόμα και στα $k+\ell-p$ στοιχεία που βρίσκονται εκτός της ένωσης $I = I_1 \cup I_2$ να μπορούν να εμφανιστούν στα I_1, I_2 και εφόσον εμφανίζονται και στα δύο σύνολα, να απαλείφονται. Κατ' ατό τον τρόπο, αυξάνεται σημαντικά ο αριθμός των αναπαραστάσεων, εφόσον για περιπτώσεις τυχαίων κωδίκων, ο αριθμός των μηδενικών σε ένα διάνυσμα σφάλματος $\boldsymbol{e}$ είναι μεγαλύτερος από τον αριθμό των μη-μηδενικών του στοιχείων. Για την ακρίβεια, ο αλγόριθμος MMT-ISD επιτρέπει τον διαχωρισμό κάθε στοιχείου-1 του $\boldsymbol{e}$ σε δύο τμήματα, είτε $1 = 0+1$, είτε $1 = 1+0$. Αντιθέτως ο BJMM-ISD επιτρέπει τον διαχωρισμό του στοιχείου-0 σε δύο τμήματα είτε $0 = 0+0$, είτε $0 = 1+1$, αυξάνοντας τον αριθμό των αναπαραστάσεων ανά λύση I σε $\binom{p}{p/2}\cdot\binom{k+\ell-p}{\epsilon}$, όπου ϵ ένας καθορισμένος αριθμός συντεταγμένων βλ. Σχ. 6.11. Συνεπώς, το σύνολο



Σχήμα 6.11: Αποσύνθεση του συνόλου δεικτών I σε δύο αλληλοεπικαλυπτόμενα σύνολα δεικτών.

δεικτών I που τελικά προκύπτει, αναπαρίσταται ως $I_1 \Delta I_2 := (I_1 \cup I_2)\ (I_1 \cap I_2)$, δηλαδή ένα σύνολο δεικτών με πληθικότητα p, εφόσον τα σύνολα I_1, I_2 τέμνονται ακριβώς σε ϵ θέσεις.

Ο αλγόριθμος απεικονίζεται ως ένα δέντρο υπολογισμού βάθους-3 το οποίο απεικονίζεται στο Σχ. 6.12. Το επίπεδο-3 ταυτίζεται με τον αρχικό υπολογισμό των ξένων λιστών βάσης B_1 και B_2 και το επίπεδο-0 με την έξοδο της τελικής λίστας $\mathcal{L}$. Οι παράμετροι ϵ_1 και ϵ_2 αναπαριστούν τον αριθμό των επιπρόσθετων μη-μηδενικών στοιχείων που επιτρέπονται στα επίπεδα 1 και 2 αντίστοιχα και επιπλέον, οι άνω δείκτες σε κάθε λίστα, εκφράζουν το επίπεδο στο οποίο εμπίπτει. Σκοπός όπως και στον αλγόριθμο MMT-ISD, είναι η εύρεση ενός συνόλου δεικτών $|I| = p$, ώστε να ικανοποιείται η εξίσωση (6.41).

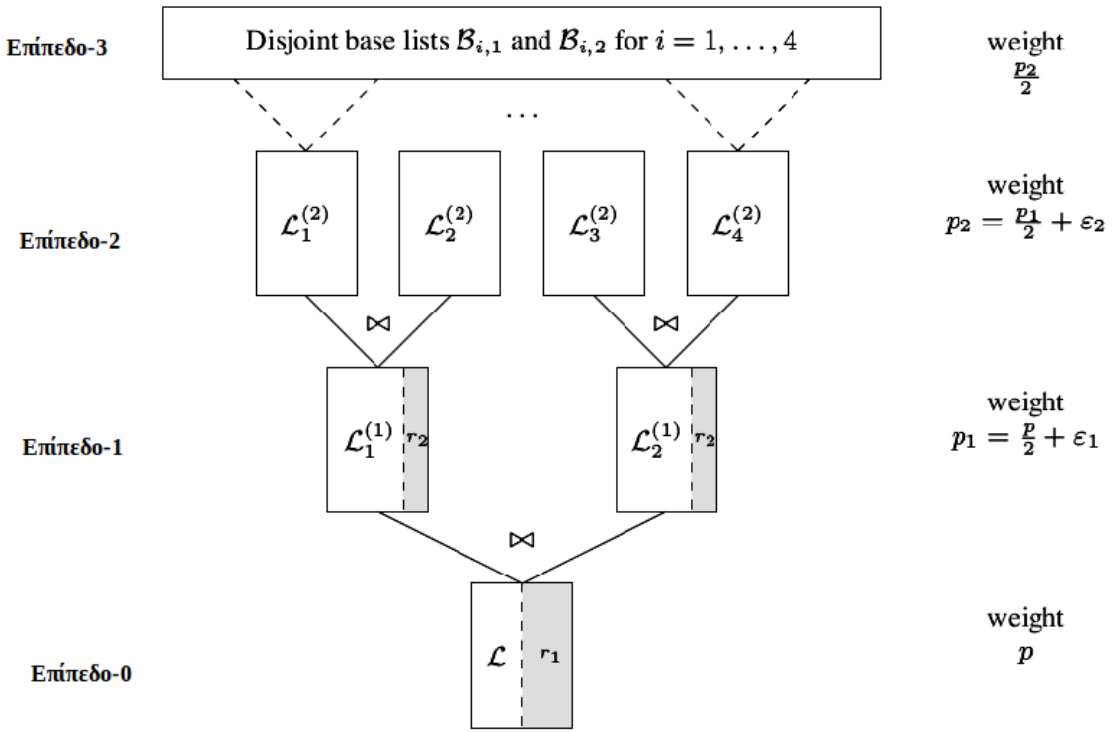
Αρχικά, στο επίπεδο-1 αναζητούνται τα σύνολα δεικτών $I_1^{(1)}$ και $I_2^{(1)}$ στο σύνολο $[k+\ell]$, πληθικότητας $p_1 = \frac{p}{2} + \epsilon_1$ που τέμνονται σε ακριβώς ϵ_1 συντεταγμένες, έτσι ώστε $I = I_1^{(1)} \Delta I_2^{(1)}$. Με άλλα λόγια, δημιουργούνται λίστες δυαδικών διανυσμάτων $\boldsymbol{e}_1^{(1)}$ και $\boldsymbol{e}_2^{(1)}$, βάρους p_1 και αναζητούνται εν συνεχεία τα ζεύγη $(\boldsymbol{e_1}^{(1)}, \boldsymbol{e_2}^{(1)})$ για τα οποία $\mathsf{wt}(\boldsymbol{e_1}^{(1)} + \boldsymbol{e_2}^{(1)}) = p$ και $\boldsymbol{Q}(\boldsymbol{e_1}^{(1)} + \boldsymbol{e_2}^{(1)}) = \widetilde{\boldsymbol{s}}_L \in \mathbb{F}_2^{\ell}$. Ο αριθμός των ζευγών $(\boldsymbol{e_1}^{(1)}, \boldsymbol{e_2}^{(1)})$ που αναπαριστούν μια μοναδική λύση $\boldsymbol{e}$ είναι

$$R_1(p, \ell; \epsilon_1) := \binom{p}{\frac{p}{2}} \binom{k+\ell-p}{\epsilon_1}. \tag{6.46}$$

Ωστόσο, επιβάλλεται ένας περιορισμός σε $r_1 \approx \log_2 R_1$ συντεταγμένες των αντίστοιχων διανυσμάτων $\boldsymbol{Q}\boldsymbol{e}_i^{(1)}$, ώστε να είναι ευκολότερο να βρεθεί μια αναπαράσταση της επιθυμητής λύσης $\boldsymbol{e}$, συμβάλλοντας έτσι στην βελτιστοποίηση του χρόνου εκτέλεσης.

Για την ακρίβεια, καθορίζοντας ένα τυχαίο διάνυσμα $\boldsymbol{s}_1^{(1)} \in_R \mathbb{F}_2^{r_1}$ [9]και υπολο-

[9]όπου $\in_R$ συμβολίζει ότι το $\boldsymbol{s}_1^{(1)}$ λαμβάνεται ομοιόμορφα, με τυχαίο τρόπο από το $\mathbb{F}_2^{r_1}$.



Σχήμα 6.12: Η βασική ιδέα του αλγορίθμου BJMM-ISD

γίζονται οι λίστες $\mathcal{L}_1^{(1)}$, $\mathcal{L}_2^{(1)}$, ως εξής

$$\mathcal{L}_i^{(1)} = \left\{ \boldsymbol{e}_i^{(1)} \in \mathbb{F}_2^{k+\ell} | \ \mathsf{wt}(\boldsymbol{e}_i) = p_1 \quad \text{και} \quad (\boldsymbol{Q}\boldsymbol{e}_i^{(1)})_{[r_1]} = \boldsymbol{s}_i^{(1)} \right\}, \forall i = 1, 2.$$

Παρατήρηση 6.4. Κάθε διάνυσμα $\boldsymbol{e}_i^{(1)} \in \mathcal{L}_i^{(1)}$, $\forall i = 1, 2$, εκ κατασκευής ικανοποιεί την εξίσωση $(\boldsymbol{Q}(\boldsymbol{e}_1^{(1)} + \boldsymbol{e}_2^{(1)}))_{[r_1]} = \widetilde{\boldsymbol{s}}_{[r_1]}$, δηλαδή ταυτίζεται ήδη με το σύνδρομο $\widetilde{\boldsymbol{s}}$ στις r_1 συντεταγμένες.

Εν συνεχεία, κατασκευάζονται οι λίστες $\mathcal{L}_1^{(2)}$, $\mathcal{L}_2^{(2)}$, $\mathcal{L}_3^{(2)}$, $\mathcal{L}_4^{(2)}$. Τα διανύσματα $\boldsymbol{e}_i^{(1)}$ στο επίπεδο-2 αναπαριστώνται, σαν άθροισμα δύο αλληλοεπικαλυπτόμενων διανυσμάτων $\boldsymbol{e}_{2i-1}^{(2)}$, $\boldsymbol{e}_{2i}^{(2)}$, $\forall i = 1, 2, 3, 4$, βάρους $p_2 := \frac{p_1}{2} + \epsilon_2$, αναζητούνται δηλαδή δύο διανύσματα που έχουν ακριβώς ϵ_2 κοινές συντεταγμένες. Ως εκ τούτου, η τελική λύση $\boldsymbol{e}$ αναπαρίσταται ως εξής

$$\boldsymbol{e} = \boldsymbol{e}_1^{(1)} + \boldsymbol{e}_2^{(1)} = \ \boldsymbol{e}_1^{(2)} + \boldsymbol{e}_2^{(2)} + \boldsymbol{e}_3^{(2)} + \boldsymbol{e}_4^{(2)}. \tag{6.47}$$

Παρομοίως με το επίπεδο-1, ο αριθμός των αναπαραστάσεων στο επίπεδο-2 είναι

$$R_2(p, \ell; \epsilon_1, \epsilon_2) = \binom{p_1}{\frac{p_1}{2}} \cdot \binom{k + \ell - p_1}{\epsilon_2} \tag{6.48}$$

όπου $p_1 = \frac{p}{2} + \epsilon_1$, επιβάλλοντας έναν περιορισμό σε $r_2 \approx \log_2 R_2$ συντεταγμένες των αντίστοιχων διανυσμάτων $\boldsymbol{Q}\boldsymbol{e}_i^{(2)}$. Για την ακρίβεια, καθορίζοντας δύο τυχαία

διανύσματα $\boldsymbol{s}_1^{(2)} \in_R \mathbb{F}_2^{r_1}$, $\boldsymbol{s}_3^{(2)} \in_R \mathbb{F}_2^{r_1}$ και θέτοντας $\boldsymbol{s}_{2j}^{(2)} := \boldsymbol{s}_{j_{[r_2]}}^{(1)} + \boldsymbol{s}_{2j-1}^{(2)}$, $\forall j = 1,2$ υπολογίζονται οι λίστες $\mathcal{L}_1^{(2)}$, $\mathcal{L}_2^{(2)}$, $\mathcal{L}_3^{(2)}$, $\mathcal{L}_4^{(2)}$ από τις σχέσεις

$$\mathcal{L}_i^{(2)} = \left\{ \boldsymbol{e}_i^{(2)} \in \mathbb{F}_2^{k+\ell} | \ \mathsf{wt}(\boldsymbol{e}_i^{(2)}) = p_2 \quad \text{και} \quad (\boldsymbol{Q}\boldsymbol{e}_i^{(2)})_{[r_2]} = \boldsymbol{s}_i^{(2)} \right\}, \forall i = 1,2,3,4.$$

Παρατήρηση 6.5. Κάθε διάνυσμα $\boldsymbol{e}_i^{(2)} \in \mathcal{L}_i^{(2)}$, $\forall i = 1,2,3,4$, εκ κατασκευής ικανοποιεί την εξίσωση $(\boldsymbol{Q}(\boldsymbol{e}_1^{(2)} + \boldsymbol{e}_2^{(2)} + \boldsymbol{e}_3^{(2)} + \boldsymbol{e}_4^{(2)}))_{[r_2]} = \boldsymbol{s}_{j_{[r_2]}}^{(1)}$, δηλαδή ταυτίζεται ήδη με το σύνδρομο $\boldsymbol{s}_{j_{[r_2]}}^{(1)}$ στις r_2 συντεταγμένες.

Αναλόγως κατασκευάζονται και οι λίστες βάσης $\mathcal{B}_{i,1}, \mathcal{B}_{i,2}$, $\forall i = 1,2,3,4$ στο επίπεδο-3. Έστω για παράδειγμα, το διάνυσμα $\boldsymbol{e}_1^{(2)}$ το οποίο αναπαρίσταται σαν άθροισμα δύο ξένων διανυσμάτων $\boldsymbol{y}, \boldsymbol{z}$, μήκους $k+\ell$ και βάρους $\mathsf{wt}(\boldsymbol{y}) = \mathsf{wt}(\boldsymbol{z}) = \frac{p_2}{2}$. Επιλέγεται δηλαδή, τυχαία διαμέριση του συνόλου $[k+\ell]$ σε δύο ίσου μεγέθους σύνολα J_1, J_2 πληθικότητας $|J_1| = |J_2| = \frac{k+\ell}{2}$, άρα $[k+\ell] = J_1 \cup J_2$. Συνεπώς, οι λίστες $\mathcal{B}_1, \mathcal{B}_2$ κατασκευάζονται ως εξής

$$\mathcal{B}_1 := \left\{ \boldsymbol{y} \in \mathbb{F}_2^{k+\ell} | \ \mathsf{wt}(\boldsymbol{y}) = \frac{p_2}{2} \quad \text{και} \quad \mathsf{supp}(\boldsymbol{y}) \subset J_1 \right\},$$
$$\mathcal{B}_2 := \left\{ \boldsymbol{z} \in \mathbb{F}_2^{k+\ell} | \ \mathsf{wt}(\boldsymbol{z}) = \frac{p_2}{2} \quad \text{και} \quad \mathsf{supp}(\boldsymbol{z}) \subset J_2 \right\}.$$

Παρομοίως υπολογίζονται και οι υπόλοιπες λίστες $\forall \boldsymbol{e}_i^{(2)}$, $\forall i = 1,2,3,4$. Έπειτα πραγματοποιείται η διαδικασία ταύτισης από τον Αλγ. 6.7, στη γρ. 7 του Αλγ. 6.9. Στη συνέχεια, εφαρμόζονται δύο ακόμα επικλήσεις του Αλγ. 6.7 για τον υπολογισμό των λιστών $\mathcal{L}_j^{(1)}, \forall j = 1,2$ και μια τελευταία για τον υπολογισμό της τελικής λίστας $\mathcal{L}$, οι καταχωρήσεις της οποίας αποτελούν λύσεις του προβλήματος ταύτισης υποπίνακα. Ακριβής περιγραφή του αλγορίθμου BJMM-ISD δίνεται από τον Αλγ. 6.9.

6.7.4 Πολυπλοκότητα

Ο Αλγ. 6.9 επιτυγχάνει την εύρεση μιας λύσης, με πιθανότητα τουλάχιστον $1 - \frac{3}{e^2}$, σύμφωνα με το [65, Θεώρημα 1]. Η πολυπλοκότητα του αλγορίθμου εξετάζεται ανά επίπεδο, ως προς τις ακόλουθες τρεις ποσότητες.

Πιθανότητα επιτυχίας

$$\mathsf{Pr}_{success} = \left(1 - \frac{3}{e^2}\right) \cdot \left(\frac{\binom{(k+\ell)/2}{p2/2}^2}{\binom{k+\ell}{p_2}} \right). \tag{6.49}$$

Ασυμπτωτικός εκθέτης αριθμού επαναλήψεων

$$N^*_{BJMM-ISD}(R,W,P,L) = H_2(W) - H_2\left(\frac{P}{R+L}\right) \cdot (R+L) - H_2\left(\frac{W-P}{1-R-L}\right) \cdot (1-R-L). \tag{6.50}$$

Αλγ. 6.9 Ο αλγόριθμος BJMM-ISD

είσοδος: $Q^{\ell\times k+\ell}$, $\widetilde{s}_L \in \mathbb{F}_2^{\ell}$, παράμετροι $p \leq k+\ell$, ϵ_1, ϵ_2, p_1, p_2
αρχικοποίηση: $p_1 = \frac{p}{2} + \epsilon_1$, $p_2 = \frac{p_1}{2} + \epsilon_2$

repeat
  Επέλεξε $s_1^{(1)} \in_R \mathbb{F}_2^{r_1}$ και θέσε $s_2^{(1)} = \widetilde{s}_{[r_1]} + s_1^{(1)}$
  Επέλεξε $s_1^{(2)} \in_R \mathbb{F}_2^{r_1}$, $s_3^{(2)} \in_R \mathbb{F}_2^{r_1}$ και θέσε $s_2^{(2)} = s_{1_{[r_2]}}^{(1)} + s_1^{(2)}$ και $s_4^{(2)} = s_{2_{[r_2]}}^{(1)} + s_3^{(2)}$
  for all $i = 1$ to 4 **do**
    Επέλεξε τυχαία διαμέριση $J_{i,1} \dot\cup J_{i,2} = [k+\ell]$
    Υπολόγισε τις λίστες βάσης $\mathcal{B}_{i,1}, \mathcal{B}_{i,2}$
    $\mathcal{L}_i^{(2)} \leftarrow MERGE - JOIN(\mathcal{B}_{i,1}, \mathcal{B}_{i,2}, r_1, p_1, s_i^{(1)})$
  end
  for all $i = 1$ to 2 **do**
    $\mathcal{L}_i^{(1)} \leftarrow MERGE - JOIN(\mathcal{L}_{2i-1}^{(2)}, \mathcal{L}_{2i}^{(2)}, r_2, p_2, s_i^{(2)})$
  end
  $\mathcal{L} \leftarrow MERGE - JOIN(\mathcal{L}_1^{(1)}, \mathcal{L}_2^{(1)}, \ell, p, \widetilde{s}_L)$
  return $\mathcal{L}$ » εμπεριέχει τα διανύσματα $e \in \mathbb{F}_2^{k+\ell}$, $\mathrm{wt}(e) = p$ και $Qe = \widetilde{s}_L$

έξοδος: $\mathcal{L}$

Συνολικός ασυμπτωτικός συντελεστής χωρικής πολυπλοκότητας

$$S^*_{BJMM-ISD}(P,R,L,R_1,R_2) = \max\Big\{S^*_{BMMJ_ISD(3)}(R,P_2,L), S^*_{BJMM-ISD(2)}(P_2,R,L,R_2), S^*_{BJMM-ISD(1)}(P_1,R,L,R_1)\Big\}. \tag{6.51}$$

Συνολικός ασυμπτωτικός συντελεστής χρονικής πολυπλοκότητας

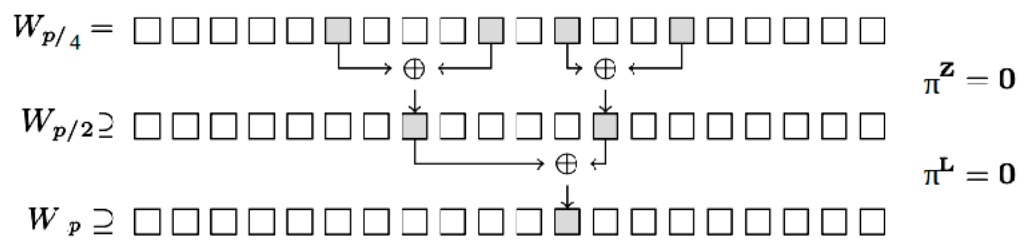
$$T^*_{BJMM-ISD}(R,W) = \min_{P,L,R_1,R_2}\Big\{N^*_{BJMM-ISD}(R,W,P,L) + \max\Big\{C^*_{BJMM-ISD(1)}(P_2,R,L,R_1), C^*_{BJMM-ISD(2)}(P_2,R,L,R_1,R_2),\ C^*_{BJMM-ISD(3)}(P_2,R,L,R_2)\Big\}\Big\} \tag{6.52}$$

όπου $C^*_{BMMJ-ISD}$ είναι το κόστος που απαιτείται ανά επίπεδο, σε κάθε επανάληψη ου αλγορίθμου. Για περισσότερες λεπτομέρειες, σχετικά με την πλήρη ασυμπτωτική ανάλυση του αλγορίθμου (βλ. Ενότητα 8.8).

6.8 Αλγόριθμος JL-ISD

Το 2011 οι Johansson και Löndahl [64] εισήγαγαν έναν νέο ISD αλγόριθμο ως βελτιστοποίηση του Αλγ. 6.4. Η ανάλυσή τους διαφοροποιήθηκε από την αρχική περιγραφή του αλγορίθμου του Stern [22], επικεντρώνοντας τη μελέτη τους στον πίνακα γεννήτορα G. Ακολουθώντας ωστόσο, την ίδια διαδικασία μετασχηματισμού του G στην συστηματική του μορφή

$$\widetilde{G} = [IZLJ] \tag{6.53}$$



Σχήμα 6.13: Κατασκευή των λιστών ανά επίπεδο, στον αλγόριθμο JL-ISD.

όπου $\boldsymbol{I} \in \mathbb{F}_2^{k\times k}$ ο μοναδιαίος πίνακας, $\boldsymbol{Z} \in \mathbb{F}_2^{k\times z}$, $\boldsymbol{L} \in \mathbb{F}_2^{k\times \ell}$ και $\boldsymbol{J} \in \mathbb{F}_2^{k\times n-k-z-\ell}$. Μια επιπλέον διαφοροποίηση του JL-ISD έγκειται στην κατανομή των μη-μηδενικών θέσεων του διανύσματος σφάλματος, συγκριτικά με τον Αλγ. 6.4 (βλ. Σχ. 6.5). Υποθέτοντας ότι, ικανοποιείται η ανωτέρω συνθήκη ως προς την κατανομή του βάρους, οι Johansson και Löndahl εισήγαγαν μια επιπλέον συνθήκη, με σκοπό να περιορίσουν την υπολογιστική πολυπλοκότητα που απαιτείται σε κάθε επανάληψη του αλγορίθμου του Stern (όπως αυτός εισήχθη στην αρχική του περιγραφή). Συγκεκριμένα, απέδειξαν ότι μπορεί να βρεθεί επιτυχώς, η επιθυμητή κωδική λέξη $\boldsymbol{c}$ αν υπάρχουν δύο διανύσματα $\boldsymbol{x}, \boldsymbol{x}' \in W_{p/2}$, έτσι ώστε $\boldsymbol{c} = \boldsymbol{x} + \boldsymbol{x}'$ και επιπλέον $\pi_{\boldsymbol{Z}}(\boldsymbol{x}) = [00\cdots 0]$[10]. Εφόσον, από υπόθεση και με βάση το Σχ. 6.5 ισχύει ότι $\pi_{\boldsymbol{Z}}(\boldsymbol{c}) = [0, 0, \cdots, 0]$, συνεπώς $\pi_{\boldsymbol{Z}}(\boldsymbol{x}') = [0, 0, \cdots, 0]$. Η εν λόγω συνθήκη γράφεται

$$\exists \boldsymbol{x}, \boldsymbol{x}' \in W_{p/2} : \pi_{\boldsymbol{Z}}(\boldsymbol{x}) = \boldsymbol{0}. \tag{6.54}$$

Αναλυτική περιγραφή του αλγορίθμου JL-ISD δίνεται από τον Αλγ. 6.10.

Παρατήρηση 6.6. Οι Johansson και Löndahl υιοθέτησαν μια παρόμοια προσέγγιση διαχωρισμού επιπέδων για την κατασκευή των λιστών, όπως παρουσιάστηκε στον Αλγ. 6.9 και απεικονίζεται στο Σχ.6.13.

6.8.1 Πολυπλοκότητα

Η πολυπλοκότητα που απαιτείται για την κατασκευή της λίστας $\mathcal{L}_0$, συμβολίζεται με $S^*_{JL-ISD(1)}$ και υπολογίζεται ως εξής

$$S_{JL-ISD(1)} = \binom{k}{p/4} c_1 \tag{6.55}$$

όπου $c_1 = pz/4$, απαιτείται δηλαδή το άθροισμα διανυσμάτων, βάρους p/4 σε z ψηφία για την εύρεση του $\pi_{\boldsymbol{Z}}(\boldsymbol{x})$. Αντιστοίχως, η πολυπλοκότητα που απαιτείται για την κατασκευή της λίστας $\mathcal{L}_1$, συμβολίζεται με $S^*_{JL-ISD(2)}$ και υπολογίζεται

[10] όπου $\pi_{\boldsymbol{Z}}(\boldsymbol{x}) \in \mathbb{F}_2^z$ εκφράζει την τιμή του $\boldsymbol{x}$ στις θέσεις που αντιστοιχούν στον πίνακα $\boldsymbol{Z}$, δηλαδή $\pi_{\boldsymbol{Z}}(\boldsymbol{x}) = [x_{k+1}\, x_{k+2} \cdots x_{k+z}]$, αντιστοίχως $\pi_{\boldsymbol{L}}(\boldsymbol{x}) \in \mathbb{F}_2^\ell$ εκφράζει την τιμή του $\boldsymbol{x}$ στις θέσεις που αντιστοιχούν στον πίνακα $\boldsymbol{L}$, δηλαδή $\pi_{\boldsymbol{L}}(\boldsymbol{x}) = [x_{k+z+1}\, x_{k+z+2} \cdots x_{k+z+\ell}]$.

Αλγ. 6.10 Ο αλγόριθμος JL-ISD

είσοδος: $\boldsymbol{G}$, παράμετροι p, z, ℓ

1: Επιλογή $\boldsymbol{P}$ » τυχαίος πίνακας μετάθεσης
2: Μετέτρεψε τον πίνακα $\boldsymbol{G}$ στην συστηματική του μορφή $\widetilde{\boldsymbol{G}} = [\boldsymbol{IZLJ}]$ » όπου $\boldsymbol{I} \in \mathbb{F}_2^{k\times k}$, $\boldsymbol{Z} \in \mathbb{F}_2^{k\times z}$, $\boldsymbol{L} \in \mathbb{F}_2^{k\times \ell}$ και $\boldsymbol{J} \in \mathbb{F}_2^{k\times n-k-z-\ell}$
3: Έστω $\boldsymbol{m} \in W_{p/2}$. Αποθήκευσε όλα τα διανύσματα $\boldsymbol{x} = \boldsymbol{m}\widetilde{\boldsymbol{G}}$ με $\pi_{\boldsymbol{Z}}(\boldsymbol{x}) = [0, 0, \cdots, 0]$ σε μια λίστα $\mathcal{L}_1$, που ταξινομείται σύμφωνα με την ετικέτα $\pi_{\boldsymbol{L}}(\boldsymbol{x})$. Κατασκεύασε μια λίστα $\mathcal{L}_0$ που περιλαμβάνει όλα τα διανύσματα $\boldsymbol{x} = \boldsymbol{m}\widetilde{\boldsymbol{G}}$, όπου $\boldsymbol{m} \in W_{p/4}$. Τότε πρόσθεσε όλα τα ζεύγη διανυσμάτων $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{L}_0$ στην λίστα έτσι ώστε $\pi_{\boldsymbol{Z}}(\boldsymbol{x}) = \pi_{\boldsymbol{Z}}(\boldsymbol{x}')$. » τα διανύσματα $\boldsymbol{x}$, $\boldsymbol{x}'$ δεν αλληλοεπικαλύπτονται
4: Παρομοίως, αποθήκευσε όλα τα διανύσματα $\boldsymbol{x} = \boldsymbol{m}\widetilde{\boldsymbol{G}}$ με $\pi_{\boldsymbol{L}}(\boldsymbol{x}) = [0, 0, \cdots, 0]$ σε μια λίστα $\mathcal{L}_2$, προσθέτοντας όλα τα ζεύγη διανυσμάτων $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{L}_1$ στην λίστα με $\pi_{\boldsymbol{L}}(\boldsymbol{x}) = \pi_{\boldsymbol{L}}(\boldsymbol{x}')$.
5: Για κάθε $\boldsymbol{x} \in \mathcal{L}_2$, έλεγξε αν $\mathsf{wt}(\boldsymbol{x}) = \omega - p$. Αν δεν βρεθεί κάποια κωδική λέξη, επέστρεψε στο βήμα 1.

ως εξής

$$S_{JL-ISD(2)} = \binom{k}{p/2} c_2 \tag{6.56}$$

όπου $c_2 = p\ell/2$, απαιτείται δηλαδή το άθροισμα διανυσμάτων, βάρους p/2 σε ℓ ψηφία για την εύρεση του $\pi_{\boldsymbol{L}}(\boldsymbol{x})$. Παρομοίως, η πολυπλοκότητα που απαιτείται για την κατασκευή της λίστας $\mathcal{L}_2$, συμβολίζεται με $S^*_{JL-ISD(3)}$ και υπολογίζεται ως εξής

$$S_{JL-ISD}(3) = \left(\binom{k}{p/2} \cdot 2^{-z}\right)^2 \cdot 2^{-\ell} \cdot c_3 \tag{6.57}$$

όπου $c_3 = p(n-k)$ εκφράζει το κόστος που απαιτείται για τον έλεγχο εύρεσης του επιθυμητού διανύσματος, ώστε $\mathsf{wt}(\boldsymbol{e}) = \omega - p$.

ΚΕΦΑΛΑΙΟ 7

Εναλλακτικές μέθοδοι αποκωδικοποίησης

Εκτός των αλγορίθμων αποκωδικοποίησης συνόλου πληροφορίας, θα εξετάσουμε πρόσθετες μεθόδους αποκωδικοποίησης για την επίλυση του CSD προβλήματος. Συγκεκριμένα, θα μελετήσουμε την στατιστική αποκωδικοποίηση που εισήχθη από τον Al Jabri [43], την αποκωδικοποίηση ελαχίστων διανυσμάτων που προτάθηκε από τον Hwang [11], την αποκωδικοποίηση γειτόνων μηδενικής απόστασης από τους Levitin και Hartmann [17], τον αλγόριθμο επιδίωξης ταύτισης συνδρόμου από τους Kalouptsidis και Kolokotronis [62], τον αλγόριθμο αποκωδικοποίησης διαχωρισμού συνδρόμου που εισήχθη από τον Dumer [21], τον αλγόριθμο αποκωδικοποίησης υπερκώδικα που προτάθηκε από τους Barg *et al.* [40] και τον αλγόριθμο BKW που εισήχθη από τους Blum *et al.* [44]. Οι τρεις πρώτες εξ αυτών, χρησιμοποιούν *σύνολα δοκιμής, τα οποία υπολογίζονται και αποθηκεύονται εκ των προτέρων*. Παρά το γεγονός ότι σε πολλές περιπτώσεις ο προϋπολογισμός του συνόλου δοκιμής αποτελεί ιδιαίτερα δαπανηρή επιλογή συγκριτικά με την άμεση εφαρμογή των ISD αλγορίθμων, επιταχύνει τη διαδικασία αποκωδικοποίησης, κυρίως στην περίπτωση που απαιτείται η επαναλαμβανόμενη αποκωδικοποίηση του ίδιου κώδικα.

7.1 Στατιστική αποκωδικοποίηση

Έστω C και $\mathcal{H}$ τα σύνολα όλων των κωδικών λέξεων, έτσι ώστε:

$$\boldsymbol{h}^t \boldsymbol{c} = 0 \tag{7.1}$$

$\forall$ κωδική λέξη $\boldsymbol{c} \in C$ και $\forall$ *εξίσωση ελέγχου ισοτιμίας* (parity check equation) $\boldsymbol{h} \in \mathcal{H}$, όπου το $\mathcal{H} = \{\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots\}$ αποτελεί ένα σύνολο δοκιμής.

Στην *στατιστική αποκωδικοποίηση* (statistical decoding) διακρίνουμε δύο εκδοχές αποκωδικοποίησης σφαλμάτων, την *περιττή ανίχνευση σφαλμάτων* (odd error detection) και την *άρτια ανίχνευση σφαλμάτων* (even error detection).

Έστω x το λαμβανόμενο διάνυσμα στην έξοδο του καναλιού, τότε αν $\boldsymbol{h}^t\boldsymbol{x} = 1$, τότε η συγκεκριμένη εξίσωση ελέγχου ισοτιμίας $\boldsymbol{h}$, θα παρέχει περιττή ανίχνευση σφαλμάτων για το λαμβανόμενο διάνυσμα $\boldsymbol{x}$. Ισοδύναμα θα ισχύει ότι $\boldsymbol{h}^t\boldsymbol{e} = 1$, καθώς σύμφωνα με τις (3.6) και (7.1) οι μη-μηδενικές θέσεις του $\boldsymbol{h}$, αποκαλύπτουν πληροφορίες σχετικά με τις θέσεις των σφαλμάτων (περιττός αριθμός θέσεων σφαλμάτων, συμπίπτει με τις μη-μηδενικές θέσεις του $\boldsymbol{h}$). Ομοίως, αν $\boldsymbol{h}^t\boldsymbol{x} = 0$, τότε η $\boldsymbol{h}$ θα παρέχει άρτια ανίχνευση σφαλμάτων για το λαμβανόμενο διάνυσμα $\boldsymbol{x}$, όπου συμπεριλαμβάνεται και η περίπτωση κατά την οποία, δεν εντοπίζεται κανένα σφάλμα κατά την αποκωδικοποίηση. Συνεπώς, με βάση τις πληροφορίες που συλλέγονται από τα διαφορετικά $\boldsymbol{h}$, καθίσταται δυνατή η ανάκτηση του διανύσματος σφάλματος $\boldsymbol{e}$.

Ο Al Jabri [43] επικεντρώθηκε στην περιττή ανίχνευση σφαλμάτων κατά την ανάλυσή του και πρότεινε την χρήση ενός υποσυνόλου $\mathcal{H}_\omega$, του αρχικού συνόλου δοκιμής $\mathcal{H}$, που αποτελείται από έναν μεγάλο αριθμό εξισώσεων ελέγχου ισοτιμίας, καθορισμένου βάρους ω. Ο υπολογισμός του $\mathcal{H}_\omega$, που παράγεται από τον πίνακα ελέγχου ισοτιμίας $\boldsymbol{H}$, απαιτεί τον υπολογισμό όλων των κωδικών λέξεων καθορισμένου βάρους στον δυϊκό κώδικα $C^\perp$. Σύμφωνα με το [43], το *διάνυσμα εντοπισμού σφαλμάτων* (error - locating vector), ορίζεται ως εξής

$$v = \sum_{h \in \mathcal{H}_\omega} (\boldsymbol{h}^t\boldsymbol{x})\boldsymbol{h}.$$

Η περίπτωση $\boldsymbol{h}^t\boldsymbol{e} = 1$, χωρίζει το $\mathcal{H}_\omega$ σε δύο υποσύνολα, βάσει των οποίων παρέχονται πληροφορίες, σχετικά με τις θέσεις των σφαλμάτων. Συγκεκριμένα, αν t η διορθωτική ικανότητα του κώδικα, το ένα υποσύνολο παρέχει πληροφορίες, μέσω των μεγαλύτερων τιμών u του διανύσματος v για κάποιο $u \geq t$, ενώ αντίστοιχα το άλλο υποσύνολο, παρέχει πληροφορίες μέσω των μικρότερων τιμών u του διανύσματος v για κάποιο u, με $t \leq u \leq n - k$. Εντούτοις, η επιλογή της μεγαλύτερης ή μικρότερης τιμής του u, εξαρτάται από το αν το $\mathsf{wt}(\boldsymbol{e})$ είναι άρτιος ή περιττός αριθμός. Συγκεκριμένα, το u χρησιμοποιείται, προκειμένου να περιορίσει την αβεβαιότητα σχετικά με τις θέσεις των σφαλμάτων, καθώς αποτελεί ένα *κατώφλι* (threshold), που εγγυάται ότι όλα τα διανύσματα σφάλματος είναι διορθώσιμα.

Εφόσον, όμως το $\mathsf{wt}(\boldsymbol{e})$ δεν είναι εκ των προτέρων γνωστό, ο αποκωδικοποιητής υποθέτει ότι υπάρχουν δύο εκδοχές. Συγκεκριμένα, με βάση την πρώτη εκδοχή, υποθέτει ότι το $\mathsf{wt}(\boldsymbol{e})$ είναι περιττός αριθμός και συνεπώς τα σφάλματα εντοπίζονται στις θέσεις που αντιστοιχούν στις μεγαλύτερες τιμές u του διανύσματος v, ενώ αντίστοιχα σύμφωνα με την δεύτερη εκδοχή, το $\mathsf{wt}(\boldsymbol{e})$ είναι άρτιος αριθμός και τα σφάλματα εντοπίζονται στις θέσεις που αντιστοιχούν στις μικρότερες τιμές u του διανύσματος v. Στην συνέχεια, εφόσον καθοριστούν οι u υποψήφιες θέσεις σφαλμάτων, επιλέγεται ένα υποδιάνυσμα $\boldsymbol{x}_k$, που αποτελείται από τις θέσεις του λαμβανόμενου διανύσματος $\boldsymbol{x}$ στις οποίες δεν περιέχονται

Αλγ. 7.1 Ο αλγόριθμος του Al Jabri

είσοδος: $\boldsymbol{H}_\omega$, $\boldsymbol{x}$ » Το $\mathcal{H}_\omega$ υπολογίζεται και αποθηκεύεται εκ των προτέρων

1: **repeat**
2: $\quad \boldsymbol{v} \leftarrow \sum_{\boldsymbol{h}\in\mathcal{H}_\omega}(\boldsymbol{h}\boldsymbol{x}^t)\boldsymbol{h}$
3: $\quad \boldsymbol{m}_i \leftarrow \boldsymbol{x}_{ki}^t \boldsymbol{G}_{ki}^{-1}$ » $i = 1, 2$
4: **until** $\mathrm{wt}(\boldsymbol{m}_i^t\boldsymbol{G} + \boldsymbol{x}^t) \le t$ » $\boldsymbol{m}$ διάνυσμα πληροφορίας

έξοδος: διάνυσμα $\boldsymbol{m}$

σφάλματα, καθώς επίσης και ένας υποπίνακας $\boldsymbol{G}_k$ του πίνακα γεννήτορα $\boldsymbol{G}$, προκειμένου να υπολογιστεί, αν υπάρχει κάποιο διάνυσμα:

$$\boldsymbol{m}_i^t = \boldsymbol{x}_{ki}^t \boldsymbol{G}_{ki}^{-1}, \quad i = 1, 2.$$

Τελικά, ο αποκωδικοποιητής βρίσκει την σωστή λύση, ελέγχοντας το $\mathrm{wt}(\boldsymbol{m}_1^t\boldsymbol{G}+\boldsymbol{x}^t)$ και το $\mathrm{wt}(\boldsymbol{m}_2^t\boldsymbol{G} + \boldsymbol{x}^t)$, και επιλέγει εκείνο το διάνυσμα $\boldsymbol{m}$, για το οποίο ισχύει $\mathrm{wt}(\boldsymbol{m}) \le t$, σύμφωνα με τον Αλγ. 7.1. Σύμφωνα με την ασυμπτωτική ανάλυση του [43], η πιθανότητα p υπολογίζεται ως

$$p = \frac{\sum \binom{n-t}{w-i}\binom{t-1}{i-1}}{\sum \binom{t}{i}\binom{n-t}{w-i}} \tag{7.2}$$

και εκφράζει τις λανθασμένες θέσεις του διανύσματος εντοπισμού σφαλμάτων $\boldsymbol{v}$. Επιπλέον έστω q η πιθανότητα

$$q = \frac{\sum \binom{n-t-1}{w-i-1}\binom{t}{i}}{\sum \binom{t}{i}\binom{n-t}{w-i}} \tag{7.3}$$

η οποία εκφράζει τις θέσεις του διανύσματος $\boldsymbol{v}$, στις οποίες δεν περιέχονται σφάλματα.

Καθώς παρέμενε ανοιχτό το ζήτημα της βέλτιστης επιλογής του ω για τον αποδοτικό υπολογισμό του $\mathcal{H}_\omega$, ο Overbeck [48] πραγματοποίησε κάποιες βελτιστοποιήσεις στον αλγόριθμο του Al Jabri και πραγματοποίησε μια συγκεκριμένη ανάλυση (βάσει παραμέτρων του McEliece), συμπεραίνοντας ότι δεν υπάρχει επιλογή του ω που να επιτυγχάνει σωστή αποκωδικοποίηση, με καλή πιθανότητα και ανταγωνιστική απόδοση.

7.2 Αποκωδικοποίηση κλίσης

Η *αποκωδικοποίηση κλίσης* (gradient decoding), αποτελεί μια επιπλέον MDD μέθοδο αποκωδικοποίησης, που βασίζεται στον εκ των προτέρων υπολογισμό και αποθήκευση ενός συνόλου δοκιμής. Θα εξετάσουμε δύο χαρακτηριστικά παραδείγματα αυτής: την *αποκωδικοποίηση ελαχίστων διανυσμάτων* (minimal vectors decoding) και την *αποκωδικοποίηση γειτόνων μηδενικής απόστασης* (zero-neighbors decoding).

7.2.1 Αποκωδικοποίηση ελαχίστων διανυσμάτων

Ο Hwang [11] το 1979 πρότεινε έναν μη-εξαντλητικό αλγόριθμο αποκωδικοποίησης, ο οποίος αποτελεί παραλλαγή της βασικής μεθόδου *αποκωδικοποίησης συσχέτισης*[1] (correlation decoding) και επιτυγχάνει αποκωδικοποίηση μεγίστης πιθανοφάνειας.

Έστω C ένας (n,k) γραμμικός τμηματικός κώδικας στο $\mathbb{F}_2^n$ και $\boldsymbol{c} = (c_1, c_2, \cdots, c_n) \in C$ η μεταδιδόμενη κωδική λέξη. Επιπλέον, έστω C^* ένας γραμμικός τμηματικός κώδικας στο $\{+1, -1\}$ και $\boldsymbol{c}^* = (c_1^*, c_2^*, \cdots, c_n^*) \in C^*$ η αντίστοιχη μεταδιδόμενη κωδική λέξη, έτσι ώστε η ομάδα $\{C, \oplus\}$ να είναι ισομορφική (βλ. ορισμό 2.6) με την ομάδα $\{C^*, \times\}$[2]. Η (5.5) για τον κώδικα C^* διαμορφώνεται ως εξής (βλ. ενότητα 3.4.1):

$$\mathsf{Pr}(\boldsymbol{c}^* \mid \boldsymbol{x}) = \frac{\mathsf{Pr}(\boldsymbol{x} \mid \boldsymbol{c}^*)\,\mathsf{Pr}(\boldsymbol{c}^*)}{\mathsf{Pr}(\boldsymbol{x})} \tag{7.4}$$

όπου $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ το λαμβανόμενο διάνυσμα στην έξοδο του καναλιού. Εφόσον, οι κωδικές λέξεις του κώδικα C^* θεωρούνται ισοπίθανες, θα ισχύει ότι:

$$\mathsf{Pr}(\boldsymbol{c}^*) = 2^{-k}. \tag{7.5}$$

Συνδυάζοντας τις (7.4) και (7.5), προκύπτει τελικά

$$\mathsf{Pr}(\boldsymbol{c}^* \mid \boldsymbol{x}) = \frac{\mathsf{Pr}(\boldsymbol{x} \mid \boldsymbol{c}^*)2^{-k}}{\mathsf{Pr}(\boldsymbol{x})}.$$

Θεωρούμε ότι υπάρχει μια και μόνο μια κωδική λέξη $\boldsymbol{c}_m^*$ που μεγιστοποιεί την παραπάνω πιθανότητα, συνεπώς η $\mathsf{Pr}(\boldsymbol{c}_m^* \mid \boldsymbol{x})$ είναι η μέγιστη μεταξύ όλων των $\mathsf{Pr}(\boldsymbol{c}_j^* \mid \boldsymbol{x})$, $j = 1, 2, \cdots, 2^k$ και κατ' επέκταση:

$$\mathsf{Pr}(\boldsymbol{x} \mid \boldsymbol{c}_m^*) > \mathsf{Pr}(\boldsymbol{x} \mid \boldsymbol{c}_j^*), \qquad \forall\, j \neq m, 1 \leq j \leq 2^k. \tag{7.6}$$

Εφόσον, σύμφωνα με τον Hwang, το κανάλι μετάδοσης είναι διακριτού χρόνου και χωρίς μνήμη όπως το BSC, προκύπτει ότι:

$$\mathsf{Pr}(\boldsymbol{x} \mid \boldsymbol{c}^*) = \prod_{i=1}^{n} \mathsf{Pr}(\boldsymbol{x}_i \mid \boldsymbol{c}_i^*).$$

Συνεπώς, η (7.6) αναδιατυπώνεται ως εξής:

$$\ln\left(\frac{\prod_i \mathsf{Pr}(\boldsymbol{x}_i \mid \boldsymbol{c}_{mi}^*)}{\prod_i \mathsf{Pr}(\boldsymbol{x}_i \mid \boldsymbol{c}_{ji}^*)}\right) > 0$$

[1] Αποτελεί μια μέθοδο εξαντλητικής αναζήτησης, καθώς η λαμβανόμενη κωδική λέξη, συγκρίνεται με κάθε λέξη του κώδικα. Για τον λόγο αυτό στην πράξη χρησιμοποιείται μόνο σε κώδικες που διαθέτουν μικρό αριθμό κωδικών λέξεων (κώδικες χαμηλού ρυθμού πληροφορίας ή κώδικες μεσαίου προς υψηλού ρυθμού πληροφορίας, με μικρό μήκος τμημάτων.)

[2] όπου η πράξη $\times$ εκφράζει τον πολλαπλασιασμό συνιστώσας προς συνιστώσα, πχ. $\boldsymbol{c}^* \times \boldsymbol{d}^* = (c_1^* d_1^*, c_2^* d_2^*, \cdots, c_n^* d_n^*)$

$$\Leftrightarrow \sum_{i=1}^{n} \ln \frac{\Pr(\boldsymbol{x}_i \mid \boldsymbol{c}_{mi})}{\Pr(\boldsymbol{x}_i \mid \boldsymbol{c}_{ji})} > 0. \tag{7.7}$$

Ορίζουμε τον *λογάριθμο του λόγου πιθανοφάνειας* (log-likelihood ratio) (βλ. [14] κεφ. 10, ενότητα 10.2) του $\boldsymbol{x}_i$, ως εξής:

$$\phi_i = \ln \frac{\Pr(\boldsymbol{x}_i \mid 1)}{\Pr(\boldsymbol{x}_i \mid -1)}, \qquad i = 1, 2, \cdots, n.$$

Συνεπώς, $\forall i$ για το οποίο $\boldsymbol{c}^*_{mi} \neq \boldsymbol{c}^*_{ji}$, θα έχουμε ότι:

$$\ln \frac{\Pr(\boldsymbol{x}_i \mid \boldsymbol{c}^*_{mi})}{\Pr(\boldsymbol{x}_i \mid \boldsymbol{c}^*_{ji})} = \phi_i \cdot \boldsymbol{c}^*_{mi}. \tag{7.8}$$

Επιπλέον, $\boldsymbol{c}_{mi} + \boldsymbol{c}_{ji} = 1$ εάν $\boldsymbol{c}_{mi} \neq \boldsymbol{c}_{ji}$ και 0 διαφορετικά. Άρα σύμφωνα με την (7.8), θα έχουμε ότι:

$$\sum_{\boldsymbol{c}_{mi} \neq \boldsymbol{c}_{ji}, i=1}^{n} 1 \cdot \phi_i \cdot \boldsymbol{c}^*_{mi} + \sum_{\boldsymbol{c}_{mi} = \boldsymbol{c}_{ji}, i=0}^{n} 0 \cdot \phi_i \cdot \boldsymbol{c}^*_{mi} = \sum_{i=1}^{n} (\boldsymbol{c}_{mi} + \boldsymbol{c}_{ji}) \cdot \phi_i \cdot \boldsymbol{c}^*_{mi}. \tag{7.9}$$

Συνδυάζοντας, τις (7.7), (7.8) και (7.9), τελικά προκύπτει ότι:

$$(\boldsymbol{c}_m + \boldsymbol{c}_j)(\boldsymbol{\phi} \times \boldsymbol{c}^*_m) > 0, \quad \forall j \neq m, 1 \leq j \leq 2^k$$

όπου $\boldsymbol{\phi} = (\phi_1, \phi_2, \cdots, \phi_n)$. Εφόσον λοιπόν, $\boldsymbol{c}_m + \boldsymbol{c}_j$ είναι μια μη-μηδενική κωδική λέξη, προκύπτει το ακόλουθο θεώρημα.

Θεώρημα 7.1. *$\boldsymbol{c}^*_m$ είναι μια κωδική λέξη η οποία μεγιστοποιεί την πιθανότητα $\Pr(\boldsymbol{c}^* | \boldsymbol{r})$ αν και μόνο αν:*

$$\boldsymbol{c}_j \cdot (\boldsymbol{\phi} \times \boldsymbol{c}^*_m) > 0 \tag{7.10}$$

$\forall$ μη-μηδενική κωδική λέξη $\boldsymbol{c}_j \in C$.

Επιπλέον, από τις (7.7) και (7.8), προκύπτει ότι $\sum_{i=1}^{n} \phi_i \cdot (\boldsymbol{c}^*_{mi} - \boldsymbol{c}^*_{ji}) > 0$, όπου $\boldsymbol{c}^*_{mi} = (-1)^{\boldsymbol{c}_{mi}}$ και $\boldsymbol{c}^*_{ji} = (-1)^{\boldsymbol{c}_{ji}}$. Και συγκεκριμένα,

$$\sum_{i=1}^{n} \phi_i \cdot \boldsymbol{c}^*_{mi} > \sum_{i=1}^{n} \phi_i \cdot \boldsymbol{c}^*_{ji}$$

οδηγώντας στο ακόλουθο αποτέλεσμα

Πόρισμα 7.2. *Η μη-μηδενική κωδική λέξη $\boldsymbol{c}^*_m$ μεγιστοποιεί την πιθανότητα $\Pr(\boldsymbol{c}^* | \boldsymbol{x})$, αν και μόνο αν μεγιστοποιεί το $\sum_{i=1}^{n} \phi_i \cdot \boldsymbol{c}^*_i$.*

Συνεπώς, το Θεώρημα 7.1 αποτελεί ικανή και αναγκαία συνθήκη, έτσι ώστε το $\boldsymbol{c}^*_m$ να είναι η επιθυμητή έξοδος, στον αποκωδικοποιητή μεγίστης πιθανοφάνειας. Για την ακρίβεια, ο Hwang [11] εισάγοντας την έννοια της *προβολής* (projection), καθώς και την δυϊκή της έννοια, *επικάλυψης* (covering), εισήγαγε έναν μη-εξαντλητικό βέλτιστο αλγόριθμο αποκωδικοποίησης, αντικαθιστώντας τον αλγόριθμο αποκωδικοποίησης συσχέτισης. Πριν διατυπώσουμε το βασικό αποτέλεσμα, χρειαζόμαστε τους ακόλουθους ορισμούς.

Ορισμός 7.3. Μια μη-μηδενική κωδική λέξη $c_2 \in C$ προβάλλεται από μια μη-μηδενική κωδική λέξη $c_1 \in C$, αν και μόνο αν $c_2 \times c_1 = c_1$, δηλαδή το c_2 καλύπτει το c_1 και $\mathsf{supp}(c_1) \subseteq \mathsf{supp}(c_2)$.

Ορισμός 7.4. Το C_s αποτελεί το μικρότερο υποσύνολο μη-μηδενικών κωδικών λέξεων του κώδικα C, έτσι ώστε $\forall$ μη-μηδενική κωδική λέξη $c_j \in C$ και $c_j \notin C_s$, υπάρχει μια κωδική λέξη $c_s \in C_s$, έτσι ώστε $\mathsf{supp}(c_s) \subseteq \mathsf{supp}(c_j)$.

Θεώρημα 7.5. *Η μη-μηδενική κωδική λέξη c_m^* μεγιστοποιεί την πιθανότητα $\mathsf{Pr}(c^*|x)$, αν και μόνο αν $c_s \cdot (\phi \times c_m^*) > 0, \quad \forall c_s \in C_s$.*

Με βάση λοιπόν, το θεώρημα 7.5, μια κωδική λέξη c_1^* δεν αποτελεί επιθυμητή λύση, αν και μόνο αν υπάρχει μια μη-μηδενική κωδική λέξη $c_s \in C_s$, έτσι ώστε $c_s \cdot (\phi \times c_1^*) \leq 0$. Από τα παραπάνω, προκύπτει το ακόλουθο θεώρημα.

Θεώρημα 7.6. *Αν $c_s \cdot (\phi \times c_1^*) \leq 0$, αυτό συνεπάγεται ότι $\mathsf{Pr}(c_1^*)|x) \leq \mathsf{Pr}(c_1^* \times c_s^*|x)$.*

Αλγ. 7.2 Ο αλγόριθμος του Hwang

είσοδος: x » όπου $x = (x_1, x_2, \cdots, x_n)$ η λαμβανόμενο κωδική λέξη

1: $\phi_i \leftarrow \ln\left[\frac{\mathsf{Pr}(x_i|1)}{\mathsf{Pr}(x_i|-1)}\right]$ » $\phi = (\phi_1, \phi_2, \cdots, \phi_n)$ ο λογάριθμος του λόγου πιθανοφάνειας
2: Επέλεξε $c_1^* \in C^*$
3: **for all** $c_s \in C_s$ **do**
4: **if** $c_s \cdot (\phi \times c_1^*) > 0$ **then**
5: **break**
6: **else**
7: $c_1^* \leftarrow c_1^* \times c_j^*$ » $c_j \leftarrow \sum_{i=1}^{u} c_{s_i}, \quad \forall i = 1, 2, \cdots u$
8: **end**
9: **end**

έξοδος: η επιθυμητή κωδική λέξη c_1^*

Ο αλγόριθμος του Hwang, επιτυγχάνει αποκωδικοποίηση μεγίστης πιθανοφάνειας, καθώς αναζητά την κωδική λέξη c_m^*, που μεγιστοποιεί την πιθανότητα $\mathsf{Pr}(c^*|x)$. Πρόκειται για έναν επαναληπτικό αλγόριθμο, που αρχικά επιλέγει, μια κωδική λέξη c_1^* και κατόπιν ελέγχει αν ικανοποιείται το Θεώρημα 7.5. Στην περίπτωση που η c_1^* δεν αποτελεί την επιθυμητή λύση, υπάρχει μια μη-μηδενική κωδική λέξη $c_{s_1} \in C_s$, έτσι ώστε: $c_{s_1} \cdot (\phi \times c_1^*) < 0$. Άρα, από το Θεώρημα 7.6, προκύπτει $\mathsf{Pr}(c_1^*|x) < \mathsf{Pr}(c_1^* \times c_{s_1}^*|x)$. Η διαδικασία αποκωδικοποίησης επαναλαμβάνεται, όπως φαίνεται και στον Αλγ. 7.2, ελέγχοντας αν η κωδική λέξη $c_1^* \times c_{s_1}^*$ αποτελεί την επιθυμητή λύση. Κατά συνέπεια, ο αλγόριθμος τερματίζει, εφόσον βρεθεί η κωδική λέξη $c_m^* = c_1^* \times c_{s_1}^* \times \cdots$, για την οποία ισχύει ότι $c_s \cdot (\phi \times c_m^*) > 0, \forall c_s \in Cs$ και συνεπώς μεγιστοποιείται η πιθανότητα $\mathsf{Pr}(c^*|x)$. Ωστόσο, ανοιχτό παραμένει το ερώτημα του πλήθους των επαναλήψεων που απαιτούνται, προκειμένου να τερματιστεί επιτυχώς η διαδικασία αποκωδικοποίησης, καθώς ο Hwang δεν προτείνει κάποιο κατώφλι κατά την ανάλυσή του (βλ. [11] για περισσότερες λεπτομέρειες).

7.2.2 Αποκωδικοποίηση γειτόνων μηδενικής απόστασης

Ο Berlekamp *et al.* [8] το 1978, απέδειξαν ότι ένα πρόβλημα που συνδέεται με το MDD πρόβλημα, ανήκει στην κλάση NP - πλήρων προβλημάτων. Αυτό σημαίνει ότι, είναι εξαιρετικά απίθανο να υπάρχει MDD αλγόριθμος, του οποίου η πολυπλοκότητα να αυξάνεται πολυωνυμικά και όχι εκθετικά, με το μήκος του χρησιμοποιούμενου κώδικα. Κατ' επέκταση δεν μπορεί να υπάρξει αλγόριθμος με σημαντικά μικρότερη πολυπλοκότητα, από τον αλγόριθμο *εξαντλητικής αναζήτησης* (exhaustive search). Ωστόσο, οι Levitin και Hartmann [17] το 1985 εισήγαγαν έναν νέο αλγόριθμο, που ονομάζεται ZNA (*αλγόριθμος γειτόνων μηδενικής απόστασης* (zero neighbors algorithm)) και βασίζεται σε ένα ειδικό σύνολο κωδικών λέξεων, τους γείτονες μηδενικής απόστασης. Η χρονική πολυπλοκότητα του αλγορίθμου αυτού, αυξάνεται πολυωνυμικά ως προς το μήκος του κώδικα, ενώ η χωρική πολυπλοκότητα, παρότι εξακολουθεί να είναι εκθετική, αυξάνεται πολύ πιο αργά από το $\min(2^k, 2^{n-k})$, για ένα ευρύ φάσμα ρυθμών πληροφορίας. Για την ακρίβεια, για $R = \frac{k}{n} = 0.5$ η χωρική πολυπλοκότητα αυξάνεται περίπου ως προς την τετραγωνική ρίζα του συνολικού αριθμού των κωδικών λέξεων, σύμφωνα με τους Levitin και Hartmann [17]. Για την περιγραφή του αλγορίθμου χρειάζεται να ορίσουμε χρήσιμες για τη συνέχεια έννοιες.

Ορισμός 7.7 ([17]). Το πεδίο $D(\boldsymbol{c})$ μιας κωδικής λέξης $\boldsymbol{c} \in C$, είναι το σύνολο όλων των διανυσμάτων $\boldsymbol{x} \in \mathbb{F}_2^n$, για τα οποία

$$\mathsf{d}(\boldsymbol{x}, \boldsymbol{c}) \leq \mathsf{d}(\boldsymbol{x}, \boldsymbol{c}'), \quad \forall \boldsymbol{c}' \in C, \boldsymbol{c}' \neq \boldsymbol{c}. \tag{7.11}$$

Ορισμός 7.8 ([17]). Η *γειτνίαση* (vicinity) $B(\boldsymbol{x})$, του $\boldsymbol{x} \in \mathbb{F}_2^n$ είναι το σύνολο όλων των διανυσμάτων $\boldsymbol{y} \in \mathbb{F}_2^n$, τέτοια ώστε $\mathsf{d}(\boldsymbol{x}, \boldsymbol{y}) = 1$. Το *πλαίσιο πεδίου* (domain frame) $G(\boldsymbol{c})$ μιας κωδικής λέξης $\boldsymbol{c} \in C$, είναι το σύνολο $G(\boldsymbol{c}) = \bigcup_{\boldsymbol{x} \in D(\boldsymbol{c})} B(\boldsymbol{x}) - D(\boldsymbol{c})$.

Ορισμός 7.9 ([17]). Το σύνολο των γειτόνων μηδενικής απόστασης, είναι ένα σύνολο $N_0 \subseteq C$ κωδικών λέξεων, για το οποίο ισχύουν

$$G(0) \subset \bigcup_{\boldsymbol{c} \in N_0} D(\boldsymbol{c}) \tag{7.12}$$

$$| N_0 | = \min_N \left\{ | N |, N \subseteq C, \quad G(0) \subset \bigcup_{\boldsymbol{c} \in N} D(\boldsymbol{c}) \right\}. \tag{7.13}$$

Συγκεκριμένα, το σύνολο των πεδίων των γειτόνων μηδενικής απόστασης, σχηματίζουν ένα *ελάχιστο επικάλυμμα* (minimum covering) του πλαισίου πεδίου της μηδενικής κωδικής λέξης.

Βάσει των ορισμών 7.7, 7.8 και 7.9, προκύπτουν τα εξής:

Λήμμα 7.10 ([17]). *Εάν υπάρχει διάνυσμα* $\boldsymbol{x} \in \mathbb{F}_2^n$ *έτσι ώστε*

$$\begin{aligned} \boldsymbol{x} \in G(0), \qquad & \boldsymbol{x} \in D(\boldsymbol{c}), \\ \boldsymbol{x} \notin D(\boldsymbol{c}'), \qquad & \boldsymbol{c}' \in C, \boldsymbol{c}' \neq \boldsymbol{c} \end{aligned} \tag{7.14}$$

τότε $\boldsymbol{c} \in N_0$.

Λήμμα 7.11 ([17]). *Εάν υπάρχει διάνυσμα* $\boldsymbol{x} \in \mathbb{F}_2^n$, *έτσι ώστε:*

$$\begin{aligned} \boldsymbol{x} \in G(\boldsymbol{c}), \qquad \boldsymbol{x} \in D(0), \\ \boldsymbol{x} \notin D(\boldsymbol{c}'), \qquad \boldsymbol{c}' \in C, \boldsymbol{c}' \neq 0 \end{aligned} \tag{7.15}$$

τότε $\boldsymbol{c} \in N_0$.

Λήμμα 7.12 ([17]). *Αν* $\boldsymbol{x} \in D(0)$, *τότε οποιοσδήποτε απόγονος*[3] *(descendant)* $\boldsymbol{u}$ *του διανύσματος* $\boldsymbol{x}$, *ανήκει επίσης στο πεδίο* $D(0)$. *Επιπλέον, αν το* $\boldsymbol{x} \notin D(\boldsymbol{c})$, $\boldsymbol{c} \in C, \boldsymbol{c} \neq 0$, *τότε* $\boldsymbol{u} \notin D(\boldsymbol{c})$. *Συνεπώς:*

1. *Αν* $\boldsymbol{x} \in D(\boldsymbol{c})$, *τότε* $\boldsymbol{u} \in D(\boldsymbol{c})$,
2. *Αν* $\boldsymbol{x} \in D(0), x \in G(\boldsymbol{c})$, *τότε* $\boldsymbol{u} \in D(0)$, $\boldsymbol{u} \in G(\boldsymbol{c})$.

Ειδικότερα, ο αλγόριθμος ZNA βασίζεται σε μια θεμελιώδη ιδιότητα του συνόλου των γειτόνων μηδενικής απόστασης, η οποία προκύπτει από το ακόλουθο θεώρημα

Θεώρημα 7.13. *Αν* $\boldsymbol{x} \notin D(0)$, *τότε υπάρχει* $\boldsymbol{c} \in N_0$, *έτσι ώστε* $\mathsf{wt}(\boldsymbol{x} + \boldsymbol{c}) < \mathsf{wt}(\boldsymbol{x})$.

Απόδειξη. Θεωρούμε μια ακολουθία από απογόνους του διανύσματος

$$\boldsymbol{x}_0 = 0, \qquad \boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_{\mathsf{wt}(\boldsymbol{x}-1)}, \qquad \boldsymbol{x}_{\mathsf{wt}(\boldsymbol{x})} = \boldsymbol{x}$$

έτσι ώστε το $\boldsymbol{x}_{i-1}$ να αποτελεί έναν ενδιάμεσο απόγονο του $\boldsymbol{x}_{i,1 \leq i \leq \mathsf{wt}(\boldsymbol{x})}$. Αν το διάνυσμα $\boldsymbol{x} \notin D(0)$ τότε υπάρχει $\boldsymbol{x}_{i-1}$ και $\boldsymbol{x}_i$, έτσι ώστε $\boldsymbol{x}_{i-1} \in D(0)$ και $\boldsymbol{x}_i \notin D(0)$. Συνεπώς, $\boldsymbol{x}_i \in G(0)$ και επιπλέον $\exists \boldsymbol{c} \in N_0$, έτσι ώστε $\boldsymbol{x}_i \in D(\boldsymbol{c})$. Ως εκ τούτου,

$$\mathsf{d}(\boldsymbol{x}, \boldsymbol{c}) \leq \mathsf{d}(\boldsymbol{x}, \boldsymbol{x}_i) + \mathsf{d}(\boldsymbol{x}_i, \boldsymbol{c}) < \mathsf{d}(\boldsymbol{x}, \boldsymbol{x}_i) + \mathsf{d}(\boldsymbol{x}_i, 0) = \mathsf{d}(\boldsymbol{x}, 0)$$

και $\mathsf{wt}(\boldsymbol{x} + \boldsymbol{c}) = \mathsf{d}(\boldsymbol{x}, \boldsymbol{c}) < \mathsf{d}(\boldsymbol{x}, 0) = \mathsf{wt}(\boldsymbol{x})$. ■

Πόρισμα 7.14. *Αν όλες οι κωδικές λέξεις έχουν άρτιο βάρος και επιπλέον* $\boldsymbol{x} \notin D(0)$, *τότε* $\exists c \in N_0$, *έτσι ώστε* $\mathsf{wt}(\boldsymbol{x} + \boldsymbol{c}) \leq \mathsf{wt}(\boldsymbol{x}) - 2$.

Βάσει του θεωρήματος 7.13, μπορούμε να διατυπώσουμε τον Αλγ. 7.3, ο οποίος παρουσιάζεται στο [17].

[3] Δηλαδή οποιοδήποτε διάνυσμα για το οποίο ισχύει ότι $\mathsf{supp}(\boldsymbol{u}) \subseteq \mathsf{supp}(\boldsymbol{x})$.

Αλγ. 7.3 Ο αλγόριθμος του Levitin

είσοδος: λαμβανόμενο διάνυσμα $\boldsymbol{x}$, $\mathsf{wt}(\boldsymbol{x})$, $\boldsymbol{c} \in N_0$

αρχικοποίηση: $i = 1$

1: **while** $(\mathsf{wt}(\boldsymbol{x}_{i-1} + \boldsymbol{c}_i) < \mathsf{wt}(\boldsymbol{x}_{i-1})) \wedge (i \leq \mathsf{wt}(\boldsymbol{x}))$ **do** » $\forall \boldsymbol{c}_i \in N_0$, θεώρημα 7.13
2: $\quad \boldsymbol{x}_i = \boldsymbol{x}_{i-1} + \boldsymbol{c}_i$
3: $\quad i = i + 1$
4: **end**
5: $\boldsymbol{x}_m \leftarrow \boldsymbol{x} + \sum_{i=1}^{m} \boldsymbol{c}_i \in D(0)$ » x_m αντιπρόσωπος ομοσυνόλου, m το βήμα τερματισμού
6: $\boldsymbol{c} \leftarrow \sum_{i-1}^{m} \boldsymbol{c}_i \in C$ » η κοντινότερη κωδική λέξη στο διάνυσμα x

έξοδος: διάνυσμα $\boldsymbol{x}_m$

Σύμφωνα με την περιγραφή του Αλγ. 7.3, θεωρούμε το λαμβανόμενο διάνυσμα $\boldsymbol{x} = \boldsymbol{x}_0 \in Z$. Ξεκινώντας από ένα διάνυσμα x_{i-1} (x_0 για $i = 1$) και για όσο ικανοποιείται η συνθήκη του Θεωρήματος 7.13, επαναλαμβάνονται τα βήματα 2-3, δημιουργώντας με αυτόν τον τρόπο μια ακολουθία απογόνων $\boldsymbol{x}_i$, προσθέτοντας εκείνες τις κωδικές λέξεις $\boldsymbol{c}_i \in N_0$, οι οποίες είναι ικανές να μειώνουν το βάρος του διανύσματος $\boldsymbol{x}_i$ έως ότου φτάσουμε την ελάχιστη απόσταση του κώδικα, δηλαδή καταλήξουμε σε έναν αντιπρόσωπο ομοσυνόλου $\boldsymbol{x}_m \in D(0)$.

Μπορούμε να παρατηρήσουμε ότι, για κώδικες κατά τους οποίους όλες οι κωδικές λέξεις έχουν άρτιο βάρος, σε κάθε βήμα του ZNA μειώνεται το βάρος του $\boldsymbol{x}_i$ τουλάχιστον κατά 2, οδηγώντας στη σχέση

$$m \leq \left\lfloor \frac{\mathsf{wt}(\boldsymbol{x})}{2} \right\rfloor \leq \left\lfloor \frac{n}{2} \right\rfloor$$

όπου $\lfloor a \rfloor$ συμβολίζει το ακέραιο μέρος του a και n το μήκος του χρησιμοποιούμενου κώδικα αντίστοιχα.

Επιπλέον οι Levitin και Hartmann, προκειμένου να μειώσουν τον αριθμό των βημάτων του ZNA, περίπου στα μισά, πρότειναν την εκ των προτέρων αποθήκευση των κωδικών λέξεων μεγίστου βάρους (π.χ την κωδική λέξη βάρους n, εφόσον αυτή ανήκει στον κώδικα), οι οποίες εν συνεχεία προστίθενται στο λαμβανόμενο διάνυσμα $\boldsymbol{x}$, ώστε να προκύψει ένα διάνυσμα $\boldsymbol{x}'$ με $\mathsf{wt}(\boldsymbol{x}') \leq \frac{n}{2}$ και έπειτα να εφαρμόσουν τον ZNA. Ωστόσο, μια διαφορετική υλοποίηση βασιζόμενη στην ίδια ιδέα με τον αλγόριθμο των Levitin και Hartmann, αποτελεί ο αλγόριθμος της επόμενης ενότητας.

7.3 Επιδίωξη ταύτισης συνδρόμου

Οι Kalouptsidis και Kolokotronis [62] το 2011, πρότειναν έναν *άπληστο αλγόριθμο*[4] (greedy algorithm), τον *αλγόριθμο επιδίωξης ταύτισης συνδρόμου*

[4] Αλγόριθμοι που χρησιμοποιούνται κυρίως σε προβλήματα βελτιστοποίησης. Επιλύουν προβλήματα επιλέγοντας κάθε φορά την τοπικά βέλτιστη λύση, προσδοκώντας την συνολικά βέλτιστη.

(syndrome matching pursuit algorithm ή SMP), στα πλαίσια της αποκωδικοποίησης γραμμικών τμηματικών κωδίκων. Η ιδέα του αλγορίθμου αυτού, βασίστηκε στον αλγόριθμο *επιδίωξης ταύτισης* (matching pursuit) ή MP), ο οποίος αρχικά προτάθηκε από τους Mallat και Zhang [27] το 1993. Συγκεκριμένα, ο MP αποτελεί έναν άπληστο αλγόριθμο, ο οποίος αρχικά αναλύει το σήμα ως ένα γραμμικό συνδυασμό διανυσμάτων τα οποία επιλέγονται ένα προς ένα από κάποιο λεξικό. Για την ακρίβεια, η επιλογή των διανυσμάτων εκτελείται μέσω μιας επαναληπτικής διαδικασίας, έτσι ώστε να ταιριάζουν με τον καλύτερο δυνατό τρόπο με τις δομές του σήματος, παράγοντας τελικά ένα σχεδόν τέλειο συνδυασμό, ενώ παράλληλα, σε κάθε βήμα βελτιστοποιείται η προσέγγιση του σήματος.

Ακολουθώντας την ίδια στρατηγική στον αλγόριθμο SMP, επιλέγονται στοιχεία τα οποία σχετίζονται στενά με το υπόλοιπο, δημιουργώντας με επαναληπτικό τρόπο την ακόλουθη προσέγγιση

$$\widehat{\boldsymbol{s}_v} = \sum_{i=1}^{\sigma} \boldsymbol{h}_{\lambda_i}, \qquad \boldsymbol{h}_{\lambda_i} \in \mathbb{F}_2^{n-k}.$$

Το διάνυσμα $\boldsymbol{h}_{\lambda_i}$, επιλέγεται με ομοιόμορφα τυχαίο τρόπο από τις στήλες του πίνακα ελέγχου ισοτιμίας και το $\widehat{\boldsymbol{s}}$ εκφράζει το υπολειπόμενο τμήμα του συνδρόμου σε κάθε επανάληψη του αλγορίθμου. Ο SMP υπολογίζει αρχικά, το σύνδρομο με βάση την λαμβανόμενη λέξη και στην συνέχεια, επιλέγει κατάλληλα γραμμικό συνδυασμό των στηλών του πίνακα ελέγχου ισοτιμίας $\boldsymbol{H}$, στοχεύοντας στη συνεχή μείωση του υπολειπόμενου τμήματος του συνδρόμου, έως ότου $\boldsymbol{s} = \boldsymbol{0}$. Ο αλγόριθμος παρουσιάζεται στον Αλγ. 7.4. Υπάρχουν ωστόσο, περιπτώσεις που δεν είναι υπολογιστικά εφικτό να βρεθεί η βέλτιστη λύση, αντί αυτού υπολογίζεται μια λύση κοντά στην βέλτιστη.

Αλγ. 7.4 Ο αλγόριθμος SMP

είσοδος: πίνακας ελέγχου ισοτιμίας $\boldsymbol{H}$, λαμβανόμενο διάνυσμα $\boldsymbol{x}$, μέγιστος αριθμός επαναλήψεων v.

αρχικοποίηση: $\Lambda = \varnothing$, $i = 0$

1: $\boldsymbol{s} = \boldsymbol{H}\boldsymbol{x}$ » σύνδρομο
2: **while** $(\boldsymbol{s} \neq 0) \wedge (i < v)$ **do**
3: $\quad \lambda \in_R \arg\max \{\langle \boldsymbol{s}, \boldsymbol{h}_\omega \rangle : \omega \notin \Lambda\}$ » επιλέγουμε τυχαία
4: $\quad \boldsymbol{s} = \boldsymbol{s} + \boldsymbol{h}_\lambda$
5: $\quad \Lambda = \Lambda \cup \{\lambda\}$ » Λ το σύνολο των θέσεων που επιλέγονται εκ των στηλών του $\boldsymbol{H}$
6: $\quad i = i + 1$
7: **end**

έξοδος: υπόλοιπο $\boldsymbol{s}$, θέσεις σφαλμάτων Λ

7.4 Αποκωδικοποίηση διαχωρισμού συνδρόμου

Το 1989 ο Dumer [21] εισήγαγε τον αλγόριθμο *αποκωδικοποίησης διαχωρισμού συνδρόμου* (split syndrome decoding ή SSD). Η κεντρική ιδέα του SSD βασίζεται στην γραμμικότητα της εξίσωσης $\boldsymbol{He} = \boldsymbol{s}$ και συγκεκριμένα, τον διαχωρισμό του πίνακα ελέγχου ισοτιμίας $\boldsymbol{H} = \begin{pmatrix} \boldsymbol{H}_L & \boldsymbol{H}_R \end{pmatrix}$ σε δύο υποπίνακες ιδίου (σχεδόν) μεγέθους. Οι υποπίνακες αυτοί προκύπτουν, επιλέγοντας την κατάλληλη παράμετρο m, που καθορίζει το σημείο διαχωρισμού των στηλών του πίνακα $\boldsymbol{H}$. Κατ' αντιστοιχία, διαχωρίζονται και οι συνιστώσες του διανύσματος σφάλματος $\boldsymbol{e} = \begin{pmatrix} \boldsymbol{e}_L & \boldsymbol{e}_R \end{pmatrix}$, έτσι ώστε

$$\boldsymbol{H}_L \boldsymbol{e}_L = \boldsymbol{s} + \boldsymbol{e}_R \boldsymbol{H}_R \tag{7.16}$$

όπου $s_L = \boldsymbol{H}_L \boldsymbol{e}_L$ και $s_R = \boldsymbol{H}_R \boldsymbol{e}_R$. Κατόπιν, κατασκευάζεται μια λίστα που εμπεριέχει όλα τα διανύσματα $\boldsymbol{e}_L$, ορισμένου βάρους p στην οποία αποθηκεύονται οι αντίστοιχες τιμές $\boldsymbol{H}_L \boldsymbol{e}_L$. Παρομοίως, κατασκευάζεται και μια δεύτερη λίστα που εμπεριέχει όλα τα διανύσματα $\boldsymbol{e}_R$, ορισμένου βάρους $\omega - p$ στην οποία αποθηκεύονται οι αντίστοιχες τιμές $\boldsymbol{H}_R \boldsymbol{e}_R$ και έπειτα αναζητείται η εύρεση συγκρούσεων στις δύο λίστες. Η ακριβής περιγραφή δίνεται στον Αλγ. 7.5 που ακολουθεί.

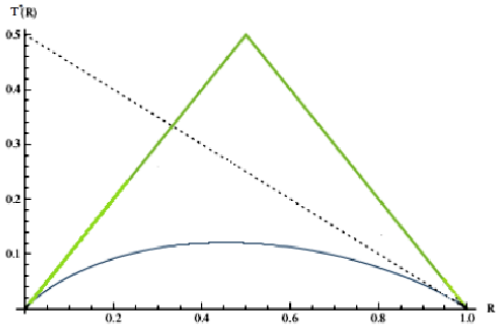
Αλγ. 7.5 Ο αλγόριθμος SSD

είσοδος: $\boldsymbol{H}$, $\boldsymbol{e}$, βάρος ω, παράμετροι p, m

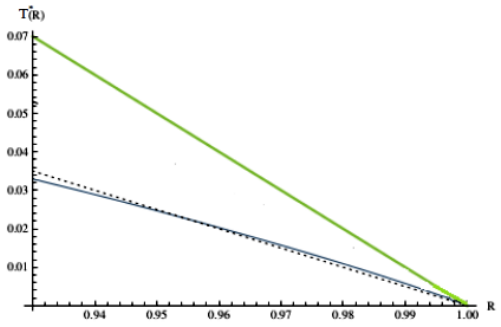
$\boldsymbol{H} = \begin{pmatrix} \boldsymbol{H}_L & \boldsymbol{H}_R \end{pmatrix}$ » υπολογίζονται εκ των προτέρων κι έχουν το ίδιο μέγεθος.
$\boldsymbol{s} \leftarrow \boldsymbol{He}$
repeat
  $\mathcal{L}_1 \leftarrow \{(\boldsymbol{H}_L \boldsymbol{e}_L,\ \boldsymbol{e}_L) : \boldsymbol{e}_L \in \mathrm{W}_{m,p}\}$
  $\mathcal{L}_2 \leftarrow \{(\boldsymbol{H}_R \boldsymbol{e}_R,\ \boldsymbol{e}_R) : \boldsymbol{e}_R \in \mathrm{W}_{n-m,\omega-p}\}$
  for all $(v,\ \boldsymbol{e}_L) \in \mathcal{L}_1$ **do**
    for all $(v,\ \boldsymbol{e}_R) \in \mathcal{L}_2$ **do**
      if $\boldsymbol{H}_L \boldsymbol{e}_L = \boldsymbol{s} + \boldsymbol{H}_R \boldsymbol{e}_R$ **then**
        return $\boldsymbol{e}$ » $\boldsymbol{e} = (\boldsymbol{e}_L \| \boldsymbol{e}_R)$
      end
    end
  end

έξοδος: διάνυσμα $\boldsymbol{e}$

Μια παρόμοια ιδέα υιοθετήθηκε από πολλούς ISD αλγορίθμους όπως είδαμε στο Κεφ. 6, με την διαφορά ωστόσο ότι ο αλγόριθμος SSD δεν μετασχηματίζει τον πίνακα ελέγχου ισοτιμίας $\boldsymbol{H}$ στην συστηματική του μορφή. Ο Dumer απέδειξε ότι εξισορροπώντας το μέγεθος των δύο λιστών, επιτυγχάνεται η καλύτερη δυνατή απόδοση του αλγορίθμου και επιπλέον μειώνεται η πολυπλοκότητα της απλής αποκωδικοποίησης συνδρόμου με έναν συντελεστή τετραγωνικής ρίζας, όπως επιβεβαίωσαν και οι Barg *et al.* το 1999 [40, Θεώρημα 6], με τον αντίστοιχο



(α′) Σύγκριση μεταξύ F_{SSD} (διακεκομμένη γραμμή), εξαντλητικής αναζήτησης (παχιά γραμμή) και απλού ISD (λεπτή γραμμή).



(β′) Σε διευρυμένη άποψη για υψηλούς ρυθμούς πληροφορίας $R \geq 0.93$.

Σχήμα 7.1: SSD vs plain ISD

συντελεστή πολυπλοκότητας

$$F_{SSD}(R) = \frac{1-R}{2}. \tag{7.17}$$

Συγκεκριμένα, για ρυθμό πληροφορίας $R \geq 0.954$, ο αλγόριθμος SSD υπερτερεί έναντι του απλού ISD, όπως απεικονίζεται στο Σχ. 7.1.

7.4.1 Διάτρητη αποκωδικοποίηση διαχωρισμού συνδρόμου

Η παραπάνω ιδιότητα παρακίνησε τον Dumer και το 1991 [26] εισήγαγε μια βελτιωμένη παραλλαγή του αλγορίθμου SSD, που ονομάστηκε *διάτρητη αποκωδικοποίηση διαχωρισμού συνδρόμου* (punctured split syndrome decoding ή PSSD). Πρότεινε την διάτρηση του δοθέντος κώδικα C, με τέτοιο τρόπο ώστε ο ρυθμός πληροφορίας να γίνει αρκετά μεγάλος, καθώς εφαρμόζοντας τον Αλγ. 7.5 για κώδικες υψηλού ρυθμού, υπάρχει κέρδος ως προς στην πολυπλοκότητα. Η κεντρική ιδέα του αλγορίθμου PSSD αναλύεται ως ακολούθως.

Δοθέντος ενός κώδικα C, μήκους n και ρυθμού πληροφορίας R και ενός διανύσματος $\boldsymbol{x} = \boldsymbol{m}^t\boldsymbol{G} + \boldsymbol{e}$, όπου $\mathsf{wt}(\boldsymbol{e}) = Wn$, επιλέγονται δύο παράμετροι a, β όπου $R \leq \beta \leq 1$ και $\max\{0, W+\beta-1\} \leq a \leq \min\{W, \beta\}$. Συγκεκριμένα, η παράμετρος a εκφράζει το ασυμπτωτικό βάρος διάτρητης λύσης και η $1-\beta$ εκφράζει τον ρυθμό διάτρησης. Ο διάτρητος κώδικας C' θεωρείται ότι προκύπτει από τις πρώτες βn συντεταγμένες. Αν ο πίνακας γεννήτορας του κώδικα C είναι $\boldsymbol{G} = (\boldsymbol{G}', \boldsymbol{G}'')$, όπου $\boldsymbol{G}'$ αποτελείται από τις πρώτες βn στήλες του πίνακα $\boldsymbol{G}$, τότε ο κώδικας C' παράγεται από τον πίνακα $\boldsymbol{G}'$. Συνεπώς, ο διάτρητος κώδικας C' έχει μήκος $n' = \beta n$ και ρυθμό πληροφορίας $R' = \frac{k}{n'} = \frac{R}{\beta}$.

Αρχικά, αναζητούνται όλες οι κωδικές λέξεις $\boldsymbol{c}' \in C'$, οι οποίες απέχουν απόσταση an από το λαμβανόμενο διάνυσμα $\boldsymbol{x}'$. Συνεπώς, υπολογίζονται όλες

οι λύσεις $\boldsymbol{e}'$ βάρους $\mathsf{wt}(\boldsymbol{e}') = an$, που ικανοποιούν την εξίσωση

$$\boldsymbol{H}'\boldsymbol{e}' = \boldsymbol{s}' \tag{7.18}$$

όπου $\boldsymbol{s}' = \boldsymbol{H}'\boldsymbol{x}'$ είναι το σύνδρομο του διάτρητου λαμβανόμενου διανύσματος $\boldsymbol{x}'$ και $\boldsymbol{H}'$ είναι ο πίνακας ελέγχου ισοτιμίας του κώδικα C'. Κατόπιν, διαχωρίζεται ο πίνακας $\boldsymbol{H}'$ και το διάνυσμα σφάλματος $\boldsymbol{e}'$ σε δύο ξένα τμήματα, όπως στον αλγόριθμο SSD. Εάν η επέκταση $\boldsymbol{c}$ σε όλες τις n συντεταγμένες μιας εκ των κωδικών λέξεων $\boldsymbol{c}'$, απέχει Wn από το λαμβανόμενο διάνυσμα $\boldsymbol{x}$, λαμβάνεται ως έξοδος. Για την ακρίβεια, η επέκταση αυτή μπορεί να υπολογιστεί εύκολα, επιλύοντας $\boldsymbol{m}^t\boldsymbol{G}' = \boldsymbol{c}'$ για τη εύρεση του $\boldsymbol{m}$ και υπολογίζοντας στη συνέχεια $\boldsymbol{c} = \boldsymbol{m}^t\boldsymbol{G}$. Ωστόσο, αν η παραπάνω διαδικασία αποτύχει, επαναλαμβάνεται πραγματοποιώντας μετάθεση των συντεταγμένων του κώδικα C.

7.4.2 Πολυπλοκότητα

Η πολυπλοκότητα του αλγορίθμου PSSD εξετάζεται ως προς τις ακόλουθες ποσότητες

Πιθανότητα επιτυχίας

$$\mathsf{Pr}_{success} = \left(\frac{\binom{\beta n}{an}\binom{(1-\beta)n}{\omega - an}}{\binom{n}{\omega}} \right). \tag{7.19}$$

Ασυμπτωτικός εκθέτης αριθμού επαναλήψεων

$$N^*_{PSSD}(W, a, \beta) = H_2(W) - H_2\left(\frac{a}{\beta}\right)\cdot\beta - H_2\left(\frac{W-a}{1-\beta}\right)\cdot(1-\beta). \tag{7.20}$$

Ασυμπτωτικός συντελεστής χωρικής πολυπλοκότητας

$$S^*_{PSSD}(a, \beta) = H_2\left(\frac{a}{\beta}\right)\cdot\frac{\beta}{2}. \tag{7.21}$$

Ασυμπτωτικός συντελεστής χρονικής πολυπλοκότητας

$$T^*_{PSSD}(R, W) = \min_{a,\beta}\left\{N^*_{PSSD}(W, a, \beta) + \max\left\{S^*_{PSSD}(a, \beta), 2S^*_{PSSD}(a, \beta) - (\beta - R)\right\}\right\}. \tag{7.22}$$

Για περισσότερες λεπτομέρειες για την πλήρη ασυμπτωτική ανάλυση του αλγορίθμου (βλ. Ενότητα 8.9).

7.5 Αλγόριθμος αποκωδικοποίησης υπερκώδικα

Το 1999 οι Barg *et al.* [40] εισήγαγαν τον αλγόριθμο *αποκωδικοποίησης υπερκώδικα*[5] (supercode decoding ή SCD). Παρομοίως, με τους ISD αλγορίθμους

[5]Ορίζεται ως ένας κώδικας που περιλαμβάνει όλα τα στοιχεία ενός μικρότερου κώδικα. Αν C είναι ένας υποκώδικας του C', τότε C' είναι ένας υπερκώδικας του C, δηλαδή $C' \supset C$.

ο SCD μετασχηματίζει τον πίνακα ελέγχου ισοτιμίας $\boldsymbol{H}$ στην συστηματική του μορφή, με την επιπλέον διαμέριση του $\boldsymbol{H}$ σε σ ίσου μεγέθους τμήματα, κάθε ένα εκ των οποίων αποτελείται από $\rho = \frac{n-k}{\sigma}$ γραμμές. Συγκεκριμένα, για κάθε τμήμα του πίνακα ελέγχου ισοτιμίας, το διάνυσμα σφάλματος $\boldsymbol{e}$ ταυτίζεται με το αντίστοιχο τμήμα του συνδρόμου. Κάθε τμήμα i του $\boldsymbol{H}$ έχει την ακόλουθη μορφή

$$\begin{pmatrix} \boldsymbol{Q}_i & \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{I}_\rho & \mathbf{0} & \cdots \mathbf{0} \end{pmatrix} \tag{7.23}$$

όπου $\boldsymbol{Q}_i \in \mathbb{F}_2^{\rho \times k}$ ένας κατάλληλος υποπίνακας του $\boldsymbol{H}$ και $\boldsymbol{I}_\rho \in \mathbb{F}_2^{\rho \times \rho}$ ο μοναδιαίος πίνακας μεταξύ των συντεταγμένων $k + i\rho$ και $k + (i+1)\rho - 1$. Κατ' αντιστοιχία, εκτελείται και ο διαχωρισμός του διανύσματος σφάλματος $\boldsymbol{e}$ σε $(\boldsymbol{e}_0 \ \boldsymbol{e}_1 \cdots \boldsymbol{e}_\sigma)$, όπου $\mathsf{wt}(\boldsymbol{e}_0) = \omega_1$ και $\mathsf{wt}(\boldsymbol{e_1} \cdots \boldsymbol{e}_\sigma) = \omega - \omega_1$. Άρα,

$$\boldsymbol{s} = \boldsymbol{H}\boldsymbol{e} \Leftrightarrow \boldsymbol{s}_i = \boldsymbol{Q}_i \boldsymbol{e}_0 + \boldsymbol{e}_i, \quad \forall i = 1, \cdots \sigma \tag{7.24}$$

όπου $\boldsymbol{e}_0 = \boldsymbol{e}_{[1,k]}$ και $\boldsymbol{e}_i = \boldsymbol{e}_{[k+1,n]}$, $\forall i = 1, \cdots, \sigma$. Στην ειδική περίπτωση, που το πλήθος των σφαλμάτων των διανυσμάτων $\boldsymbol{e}_i$ κατανέμεται ομοιόμορφα, ισχύει ότι $\omega_2 := \lceil \frac{\omega - \omega_1}{\sigma} \rceil$. Ωστόσο στην γενική περίπτωση, πρέπει να υπάρχει μια τουλάχιστον επιλογή b τμημάτων $\boldsymbol{e}_i$, ώστε κάθε τμήμα να περιέχει το πολύ $\omega_2 := \lceil \frac{\omega - \omega_1}{\sigma - b + 1} \rceil$ σφάλματα, όπου b σταθερή ακέραια παράμετρος του SCD.

Λόγω της (7.24), για κάθε τμήμα i ο αλγόριθμος SCD υπολογίζει μια λίστα K_i από υποψήφια ζεύγη $(\boldsymbol{e}_0, \boldsymbol{e}_i)$, $\forall i = 1, \cdots, \sigma$, με $\mathsf{wt}(\boldsymbol{e}_0) = \omega_1$ και $\mathsf{wt}(\boldsymbol{e}_i) \leq \omega_2$, που ταυτίζονται με το αντίστοιχο τμήμα του συνδρόμου. Εν συνεχεία, έχοντας βρει μια ταύτιση, υπολογίζουμε τις τομές b υποψηφίων λιστών $\bigcap_{k=1}^{b} K_{i_k}$, ως προς την ετικέτα $\boldsymbol{e}_0$, προκειμένου να επεκτείνει την ταύτιση σε b τμήματα του συνδρόμου. Τουλάχιστον μια από τις τομές θα περιλαμβάνει τις πρώτες k συντεταγμένες του πραγματικού διανύσματος σφάλματος $\boldsymbol{e}$. Μέσω γραμμικών μετασχηματισμών είναι στη συνέχεια δυνατή η ανάκτηση της κωδικής λέξης $\boldsymbol{c}$, από τις πρώτες k θέσεις σφαλμάτων του λαμβανόμενου διανύσματος $\boldsymbol{x}$ σύμφωνα με την (3.6).

Οι Barg *et al.* ισχυρίστηκαν ότι ο αλγόριθμος SCD παρέχει καλύτερη ασυμπτωτική πολυπλοκότητα, συγκριτικά με όλους τους γνωστούς αλγορίθμους αποκωδικοποίησης (συμπεριλαμβανομένων και των πιο πρόσφατων έως τότε ISD αλγορίθμων). Ωστόσο, οι Bernstein *et al.* [60, Ενότητα 4] κατέρριψαν τον ισχυρισμό αυτό. Για την ακρίβεια, οι Barg *et al.* [40, Πόρισμα 12], υπολόγισαν το μέγεθος των τομών $\bigcap_{k=1}^{b} K_{i_k}$, ως εξής

$$\binom{k}{\omega_1}\binom{\rho}{\omega_2} 2^{-b\rho} \tag{7.25}$$

διότι κάθε λίστα K_i έχει αναμενόμενο μέγεθος $S = \binom{k}{\omega_1}\binom{\rho}{\omega_2} 2^{-\rho}$. Εφόσον κάθε ζεύγος $(\boldsymbol{e}_0, \boldsymbol{e}_i)$ ταυτίζεται με το αντίστοιχο τμήμα του συνδρόμου, με πιθανότητα $2^{-\rho}$. Εντούτοις, οι Bernstein *et al.* απέδειξαν ότι το μέγεθος των τομών $\bigcap_{k=1}^{b} K_{i_k}$,

προκύπτει ως εξής

$$\binom{k}{\omega_1}\left(S\binom{k}{\omega_1}^{-1}\right)^b = \binom{k}{\omega_1}\binom{\rho}{\omega_2}^b 2^{-b\rho} \tag{7.26}$$

που οδηγεί σε απόκλιση παράγοντα $\binom{\rho}{\omega_2}^{b-1}$ από την (7.25). Ο αριθμός των επαναλήψεων που απαιτούνται, προκειμένου ο αλγόριθμος να συγκλίνει στην επιθυμητή λύση, υπολογίζεται ως

$$N_{SCD} = (n \log n)\left[\frac{\binom{n}{\omega}}{\binom{k}{\omega_1}\binom{n-k}{\omega-\omega_1}}\right] \tag{7.27}$$

Αναλυτική περιγραφή του SCD δίνεται από τον Αλγ. 7.6.

Αλγ. 7.6 Ο αλγόριθμος SCD

είσοδος: $\boldsymbol{H}$, $\boldsymbol{s}$, $\boldsymbol{x}$, βάρος ω, παράμετρος b

1: Υπολόγισε το σύνδρομο $\boldsymbol{s} = \boldsymbol{H}\boldsymbol{x}$. Θέσε $\boldsymbol{c} = \boldsymbol{0}$.
2: Τα βήματα 2-11 εκτελούνται N_{SCD} φορές. » βλ. (7.27)
3: Επέλεξε ένο τυχαίο υποσύνολο $W \subset [n]$, $|W| = k$. »
ώστε W σύνολο πληροφορίας, από $[n] : \{1, 2, \cdots, n\}$
4: Μετέτρεψε τον πίνακα $\boldsymbol{H}$ στην συστηματική του μορφή $\widetilde{\boldsymbol{H}} = (\boldsymbol{Q} \;\; \boldsymbol{I}_{n-k})$.
5: Διαμέρισε τις στήλες $n-k$ σε σ τμήματα μεγέθους $\rho = \frac{n-k}{\sigma}$.
6: Για κάθε i, $1 \le i \le \sigma$ εκτέλεσε τα ακόλουθα δύο βήματα:
7: Σχημάτισε τον $\rho \times (k+\rho)$ πίνακα $\boldsymbol{H}_i = (\boldsymbol{Q}_i \;\; \boldsymbol{I}_\rho)$, απομονώνοντας τις γραμμές $\rho(i-1)+j$, $1 \le j \le \rho$ του πίνακα $\widetilde{\boldsymbol{H}}$, ώστε $\boldsymbol{s}_i := (\boldsymbol{s}_{p(i-1)+j}, 1 \le j \le \rho)$. »
s_i τμήμα του συνδρόμου s
8: Εφάρμοσε τον Αλγ. 7.5 για να κατασκευάσεις την λίστα K_i των διανυσμάτων $(\boldsymbol{e}_0, \boldsymbol{e}_i)$, βάρους $\mathsf{wt}(\boldsymbol{e}_0) = \omega_1$ και $\mathsf{wt}(\boldsymbol{e}_i) \le \omega_2$, που ικανοποιούν την εξίσωση (7.24). » $\omega_2 := \lceil \frac{\omega-\omega_1}{\sigma} \rceil$
9: Υπολόγισε τις τομές b υποψηφίων λιστών $\bigcap_{k=1}^{b} K_{i_k}(W)$.
10: Για κάθε διάνυσμα $\boldsymbol{e}_0 \in \bigcap_{k=1}^{b} K_{i_k}(W)$ παράγονται επιτυχώς όλες οι κωδικές λέξεις $\widetilde{\boldsymbol{c}} \in C$.
11: Αν $\mathsf{d}(\boldsymbol{x}, \widetilde{\boldsymbol{c}}) < \mathsf{d}(\boldsymbol{x}, \boldsymbol{c})$ τότε θέσε $\boldsymbol{c} \leftarrow \widetilde{\boldsymbol{c}}$.

έξοδος: διάνυσμα $\boldsymbol{c}$ » $\boldsymbol{c}$ επιθυμητή κωδική λέξη

7.6 Αλγόριθμος BKW

Οι Blum, Kalai και Wasserman [44] το 2003 εισήγαγαν τον πιο γνωστό αλγόριθμο επίλυσης του *προβλήματος εκμάθησης θορυβώδους ισοτιμίας* (Learning Parities with Noise Problem ή LPN) τον αλγόριθμο BKW. Θα περιγράψουμε αρχικά, το LPN πρόβλημα και εν συνεχεία θα αναλύσουμε την βασική ιδέα του αλγορίθμου.

7.6.1 Το πρόβλημα LPN

Ορίζουμε αρχικά την παράμετρο θορύβου η, όπου $\eta \in (0, \frac{1}{2})$ η οποία ακολουθεί την κατανομή Bernoulli [42] και συμβολίζεται με Ber_η. Για τη ακρίβεια, $e \leftarrow Ber_\eta$ σημαίνει ότι το ψηφίο $e \in \mathbb{F}_2$ είναι τυχαία μεταβλητή με πιθανότητα $\Pr[e = 1] = \eta$ και $\Pr[e = 0] = 1 - \eta$. Παρομοίως, με $Bin_\eta(n)$ συμβολίζεται η *διωνυμική κατανομή* (binomial distribution) [53] με παραμέτρους $n \in \mathbb{N}$ και $\eta \in (0, \frac{1}{2})$ και $\boldsymbol{e} \leftarrow Bin_\eta(n)$ περιγράφει ένα διάνυσμα μήκους n του οποίου οι συντεταγμένες προκύπτουν (ανεξάρτητα) από την κατανομή Ber_η.

Σκοπός του προβλήματος, είναι η ανάκτηση ενός *μυστικού* διανύσματος $\boldsymbol{m}$, έχοντας πρόσβαση στο LPN *μαντείο* (oracle)[6] $\Pi_{\boldsymbol{m},\eta}$. Συγκεκριμένα, το LPN μαντείο $\Pi_{\boldsymbol{m},\eta}$ με μυστικό $\boldsymbol{m}$ και παράμετρο θορύβου η, επιστρέφει (ανεξάρτητα) εσωτερικά γινόμενα του $\boldsymbol{m}$, στα οποία υπεισέρχεται θόρυβος, δηλαδή επιστρέφει ζεύγη $(\boldsymbol{g}, \boldsymbol{m}^t \cdot \boldsymbol{g} + e)$, όπου $\boldsymbol{g} \in_R \mathbb{F}_2^k$ και $e \leftarrow Ber_\eta$, τα οποία επιλέγονται ανεξάρτητα για κάθε ερώτημα που λαμβάνει το μαντείο. Εναλλακτικά, πραγματοποιώντας n ερωτήματα στο LPN μαντείο και ορίζοντας τα μεμονωμένα δείγματα $\boldsymbol{g}_i$ ως διανύσματα στήλες σε έναν $k \times n$ πίνακα $\boldsymbol{G}$ προκύπτει

$$(\boldsymbol{G}, \boldsymbol{m}^t \boldsymbol{G} + \boldsymbol{e}) \tag{7.28}$$

με $\boldsymbol{e} \leftarrow Bin_\eta(n)$. Συνεπώς, με βάση την (7.28) παρατηρούμε ότι το πρόβλημα LPN είναι ισοδύναμο με το πρόβλημα αποκωδικοποίησης τυχαίων γραμμικών κωδίκων, καθώς η ανάκτηση του μυστικού διανύσματος $\boldsymbol{m}$ είναι ισοδύναμη με την αποκωδικοποίηση ενός τυχαίου κώδικα με πίνακα γεννήτορα $\boldsymbol{G}$, όπου το διάνυσμα σφάλματος $\boldsymbol{e}$ έχει βάρος Hamming $\mathsf{wt}(\boldsymbol{e}) = \eta n$.

Ορισμός 7.15 (Πρόβλημα LPN). Ένας αλγόριθμος $\mathcal{A}$ $(\boldsymbol{q}, \boldsymbol{t}, \boldsymbol{\mu}, \theta)$ επιλύει το LPN πρόβλημα με παράμετρο θορύβου $\eta \in (0, \frac{1}{2})$ αν ο $\mathcal{A}$ έχει πολυπλοκότητα το πολύ $\boldsymbol{t}$, απαιτεί το πολύ $\boldsymbol{\mu}$ ψηφία μνήμης, θέτει το πολύ $\boldsymbol{q}$ ερωτήματα στο μαντείο $\Pi_{\boldsymbol{m},\eta}$ και επιτυγχάνει με πιθανότητα τουλάχιστον θ, δηλαδή

$$\Pr[\mathcal{A}^{\Pi_{\boldsymbol{m},\eta}}(1^k) = \boldsymbol{\mu}] \geq \theta, \quad \forall \mu \in \mathbb{F}_2^k$$

με καθορισμένο $k \in \mathbb{N}$.

Συνεπώς, οι αλγόριθμοι επίλυσης του προβλήματος LPN και συγκεκριμένα ο αλγόριθμος BKW, εξετάζονται ως προς τα ακόλουθα κριτήρια.

- *Χρονική Πολυπλοκότητα*: Ο αλγόριθμος BKW επιτυγχάνει *υποεκθετική χρονική πολυπλοκότητα* (subexponential time complexity) $2^{O(k/\ln k)}$ για κάθε παράμετρο θορύβου η.
- *Πολυπλοκότητα Μαντείου*: Ο αλγόριθμος BKW βασίζεται στην διαθεσιμότητα $2^{O(k/\ln k)}$ δειγμάτων, δηλαδή θέτει $2^{O(k/\ln k)}$ ερωτήματα στο μαντείο.

[6] Μια συνάρτηση που παράγει μια πραγματικά τυχαία έξοδο για κάθε *ερώτημα* (query) που λαμβάνει.

- *Πολυπλοκότητα Μνήμης*: Όλοι οι γνωστοί αλγόριθμοι απαιτούν $2^{O(k/\ln k)}$ μνήμη.

Παρότι, η χρονική πολυπλοκότητα $2^{O(k)}$ που επιτυγχάνουν οι αλγόριθμοι ISD είναι υποδεέστερη συγκριτικά με τον αλγόριθμο BKW, μελετώντας την δυσκολία του προβλήματος LPN από την άποψη του προβλήματος αποκωδικοποίησης, παρουσιάζεται ιδιαίτερο ενδιαφέρον καθώς

- Για μικρούς ρυθμούς θορύβου η, οι αλγόριθμοι ISD έχουν πραγματικό χρόνο εκτέλεσης $2^{c(\eta)k}$, όπου $c(\eta) \ll 1$. Συνεπώς, υπερτερούν έναντι του BKW για μέτριες διαστάσεις k.
- Οι αλγόριθμοι ISD χρειάζονται μικρότερο αριθμό ερωτημάτων, καθώς η μετάθεση των συνιστωσών του διανύσματος σφάλματος, μπορεί να θεωρηθεί ως τρόπος ανακύκλωσης των ερωτημάτων στο μαντείο.
- Οι αλγόριθμοι ISD απαιτούν μειωμένη κατανάλωση μνήμης.

Ωστόσο, για μεγαλύτερους ρυθμούς θορύβου η, ο αλγόριθμος BKW υπερτερεί σημαντικά έναντι των ISD αλγορίθμων.

7.6.2 Περιγραφή του Αλγορίθμου

Έστω $(\boldsymbol{g}_1, \ell_1), \cdots (\boldsymbol{g}_n, \ell_n)$ τα ζεύγη που επιστρέφει το μαντείο $\Pi_{\boldsymbol{m},\eta}$. Από το *λήμμα συσσωρεύσεως* (pilling-up lemma) [28] έχουμε ότι $\ell_1+\ell_2+\cdots+\ell_n$ εκφράζει την σωστή τιμή του $(\boldsymbol{g}_1+\boldsymbol{g}_2+\cdots+\boldsymbol{g}_n)\cdot\boldsymbol{m}$, με πιθανότητα $\frac{1}{2}+\frac{1}{2}(1-2\eta)^n$. Η ιδέα των Blum, Kalai, Wasserman βασίστηκε στην διαμέριση του πίνακα $\boldsymbol{G}$ (βλ. (7.28)) σε a *τμήματα*, μήκους b το καθένα, δηλαδή $k = ab$, όπου κάθε τμήμα ανήκει σε μια από τις 2^b κλάσεις[7]. Σε κάθε κλάση αντιστοιχεί μια μοναδική ετικέτα, η τιμή της οποίας προκύπτει από μια 1-1 απεικόνιση, ανάμεσα στην τιμή ενός b-ψηφίου τμήματος και του συνόλου των 2^b ετικετών. Συγκεκριμένα, για κάθε μη-κενή κλάση ρ επιλέγεται τυχαία ένα διάνυσμα $\boldsymbol{g}_{j_\rho}$ από ένα σύνολο ομοιόμορφων και ανεξάρτητα κατανεμημένων δειγμάτων $\boldsymbol{g}_1, \cdots, \boldsymbol{g}_n \in \mathbb{F}_2^{a\times n}$, το οποίο προστίθεται σε όλα τα άλλα διανύσματα $\boldsymbol{g}_{k_\rho}$ της ίδιας κλάσης (οι συνιστώσες των οποίων ταυτίζονται) και εν συνεχεία το διάνυσμα $\boldsymbol{g}_{j_\rho}$ απορρίπτεται. Συνεπώς, το αντίστοιχο τμήμα μηδενίζεται, ενώ τα διανύσματα που συνεισφέρουν στο επόμενο τμήμα είναι $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{n'}$, όπου $n' \geq n - 2^b$, (εφόσον απορρίπτεται το πολύ ένα διάνυσμα ανά κλάση). Η ίδια διαδικασία εκτελείται επαναληπτικά για τα υπόλοιπα $a-2$ τμήματα, καθώς σκοπός του αλγορίθμου είναι ο μηδενισμός όλων των $a-j+1, \cdots, a$, $\forall j = 1, 2, \cdots, a-1$ τμημάτων. Άρα, κάθε διάνυσμα σε κάποιο $a-j$ τμήμα του πίνακα $\boldsymbol{G}$, μπορεί να γραφεί σαν άθροισμα δύο διανυσμάτων του $a-j-1$ τμήματος και συγκεκριμένα σαν άθροισμα το πολύ 2^j στηλών του αρχικού πίνακα. Ωστόσο, στο τελευταίο τμήμα του πίνακα $\boldsymbol{G}$ εκτελείται εξαντλητική αναζήτηση, ώστε να βρεθούν μοναδιαία διανύσματα $\boldsymbol{u}_i$ που συνθέτουν

[7] Σύνολα λιστών των οποίων το i-στό b-ψηφίο τμήμα έχει την ίδια τιμή.

Αλγ. 7.7 Ο αλγόριθμος BKW

είσοδος: $k \times n$ πίνακας γεννήτορας $\boldsymbol{G}$, ληφθείσα λέξη $\boldsymbol{\ell}$, πιθανότητα σφάλματος η

αρχικοποίηση: επιλογή $a, b \in \mathbb{Z} : ab \geq k$, $t = \mathsf{poly}\left((1-2\eta)^{-2^a}, b\right)$

for $i_1 \in [a]$ **do**
  θέσε $\boldsymbol{m}_{i_1} = (0 \ \cdots \ 0)$ » i_1–στό τμήμα του $\boldsymbol{m}$ μήκους b
  for $j_1 \in [t]$ **do**
    επιλογή πίνακα $\boldsymbol{A} \in_{\mathsf{R}} \boldsymbol{G}$ με $O\left(a2^b\right)$ στήλες
    διαγραφή στηλών στο $\boldsymbol{A}$ από τον $\boldsymbol{G}$
    for $i_2 \in [a] \setminus \{i_1\}$ **do**
      ταξινόμηση των στηλών του $\boldsymbol{A}$ ως προς το i_2–στό τμήμα
      γράψε τον $\boldsymbol{A}$ σε τμηματική μορφή $\left(\boldsymbol{A}_0 \ \cdots \ \boldsymbol{A}_{2^b-1}\right)$
      for $j_2 \in [2^b - 1]$ **do**
        επιλογή στήλης $\boldsymbol{g} \in_{\mathsf{R}} \boldsymbol{A}_{j_2}$
        πρόσθεση του $\boldsymbol{g}$ στις υπόλοιπες στήλες του $\boldsymbol{A}_{j_2}$
        διαγραφή του $\boldsymbol{g}$ από τον $\boldsymbol{A}_{j_2}$
      end
    end
    εύρεση στηλών του $\boldsymbol{A}$ που το i_1–στό τμήμα έχει βάρος 1
    παραγωγή της j_1–στής εκτίμησης του $\boldsymbol{m}_{i_1}$ » σωστό με πιθανότητα $\frac{1}{2} + \frac{1}{2}(1-2\eta)^{-2^{a-1}}$
  end
  υπολογισμός του $\boldsymbol{m}_{i_1}$ μέσω αποκωδικοποίησης MLG
end

έξοδος: άγνωστο διάνυσμα $\boldsymbol{m}^t = (\boldsymbol{m}_1 \ \cdots \ \boldsymbol{m}_a) : \boldsymbol{m}^t \boldsymbol{G} = \boldsymbol{\ell}$

μια ορθοκανονική βάση. Αν αυτό δεν καταστεί εφικτό, η διαδικασία εκτελείται εκ νέου, με βάση τα νέα ζεύγη που επιστρέφει το μαντείο $\Pi_{\boldsymbol{m},\eta}$. Κατ' αυτό τον τρόπο, θα προκύψουν εκτιμήσεις για κάθε ψηφίο του μυστικού $\boldsymbol{m}$, καθώς το i-οστό μοναδιαίο διάνυσμα $\boldsymbol{u}_i$, διαρρέει πληροφορίες για το i-οστό ψηφίο του μυστικού $\boldsymbol{m}$, εφόσον $\boldsymbol{m}^t \cdot \boldsymbol{u}_i = m_i$. Καθώς όμως υπεισέρχεται θόρυβος, εφαρμόζεται η διαδικασία *αποκωδικοποίησης πλειοψηφίας* (majority-logic decoding) [14], προκειμένου να καθοριστεί αν το αντίστοιχο ψηφίο ℓ_i ταυτίζεται με την πραγματική τιμή του m_i. Αν δεν υπάρξει ταύτιση, η διαδικασία εκτελείται εκ νέου, ώστε να προκύψει κάποια άλλη εκτίμηση για το ίδιο ψηφίο. Συνεπώς, χρησιμοποιώντας, κάθε φορά τα νέα ζεύγη που επιστρέφει το μαντείο $\Pi_{\boldsymbol{m},\eta}$ και συγκεκριμένα $poly((\frac{1}{1-2\eta})^{2a}, b)$ φορές, μπορεί να καθοριστεί με πολύ υψηλή πιθανότητα η πραγματική τιμή του ψηφίου m_i. Εκτελώντας επαναληπτικά την ίδια διαδικασία, μπορεί με παρόμοιο τρόπο, να προσδιοριστεί κάθε ψηφίο του μυστικού $\boldsymbol{m}$, με συνολικό χρόνο υπολογισμού $poly((\frac{1}{1-2\eta})^{2a}, b)$. Επιπλέον, οι Blum, Kalai και Wasserman απέδειξαν ότι για $a = \frac{1}{2}\log k$ και $b = \frac{2k}{\log k}$, επιτυγχάνεται η μικρότερη δυνατή πολυπλοκότητα. Η ακριβής περιγραφή δίνεται στον Αλγ. 7.7.

ΚΕΦΑΛΑΙΟ 8

Ασυμπτωτική ανάλυση και συγκριτικά αποτελέσματα

Στο κεφάλαιο αυτό, παρατίθεται λεπτομερώς η πλήρης ασυμπτωτική ανάλυση των ISD αλγορίθμων που εξετάστηκαν στο Κεφ. 6, της εναλλακτικής μεθόδου του Dumer [21]. καθώς επίσης και η ασυμπτωτική συσχέτιση κάποιων εξ' αυτών. Ακολουθεί, συγκριτική μελέτη των συντελεστών χωρικής και χρονικής πολυπλοκότητας, για τις διάφορες εξεταζόμενες μεθόδους επιλέγοντας τις βέλτιστες τιμές των εμπλεκόμενων παραμέτρων, ώστε να διασφαλίζονται τα απαιτούμενα επίπεδα ασφαλείας.

8.1 Ασυμπτωτική ανάλυση

8.1.1 Επισκόπηση ασυμπτωτικών εκφράσεων

Για την βέλτιστη αναζήτηση των τελικών ασυμπτωτικών συντελεστών χρονικής και χωρικής πολυπλοκότητας ανά εξεταζόμενη μέθοδο, ακολουθούν οι Πίνακες 8.1, 8.2, αντίστοιχα.

8.1.2 Ασυμπτωτική ανάλυση του αλγορίθμου του Prange

Η πιθανότητα επιτυχίας $\mathsf{Pr}_{success}$ για τον Αλγ. 6.1 δίνεται από τη σχέση (6.4). Λαμβάνοντας υπόψη και τη (6.5), έχουμε

$$\begin{aligned}\mathsf{Pr}^{*}_{success} &= \frac{1}{n}\log_2 \mathsf{Pr}_{success} \\ &= H_2\left(\frac{\omega}{n-k}\right)\cdot\left(1-\frac{k}{n}\right) - H_2\left(\frac{\omega}{n}\right) \\ &= H_2\left(\frac{\frac{\omega}{n}}{1-\frac{k}{n}}\right)\cdot\left(1-\frac{k}{n}\right) - H_2\left(\frac{\omega}{n}\right). \end{aligned} \tag{8.1}$$

Πίνακας 8.1: Συγκεντρωτικός πίνακας συντελεστών χρονικής πολυπλοκότητας.

Αλγόριθμοι	**Συντελεστές χρονικής πολυπλοκότητας**
Prange	$T^*_{Pra}(R,W) = H_2(W) - (1-R)H_2\left(\frac{W}{1-R}\right)$
Lee-Brickell	$T^*_{LB}(R,P,W) = H_2(W) - H_2\left(\frac{W-P}{1-R}\right)\cdot(1-R)$
Leon	$T^*_{Leon}(R,P,L,W) = H_2(W) - H_2\left(\frac{W-P}{1-R-L}\right)\cdot(1-R-L)$
Stern	$T^*_{Stern}(R,W) = \min_{P,L}\left\{N^*_{Stern}(R,W,P,L) + \max\left\{S^*_{Stern}, 2S^*_{Stern} - L\right\}\right\}$
BCD	$T^*_{BCD}(R,W) = \min_{P,L}\left\{N^*_{BCD}(R,W,P,Q,L) + \max\left\{S^*_{BCD}, 2S^*_{BCD} - L\right\}\right\}$
FS-ISD	$T^*_{FS-ISD}(R,W) = \min_{P,L}\left\{N^*_{FS-ISD}(R,W,P,L) + \max\left\{S^*_{FS-ISD}, 2S^*_{FS-ISD} - L\right\}\right\}$
MMT-ISD	$T^*_{MMT-ISD}(R,W) = \min_{P,L}\left\{N^*_{MMT-ISD}(1)(R,W,P,L) + \max\left\{S^*_{MMT-ISD(2)}(R,P,L), S^*_{MMT-ISD(1)}(R,P,L,L_2), 2S^*_{MMT-ISD(1)}(R,P,L,L_2) - L_1\right\}\right\}$
BJMM-ISD	$T^*_{BJMM-ISD}(R,W) = \min_{P,L,R_1,R_2}\left\{N^*_{BJMM-ISD}(R,W,P,L) + \max\left\{C^*_{BJMM-ISD(1)}, C^*_{BJMM-ISD(2)}, C^*_{BJMM-ISD(3)}\right\}\right\}$
PSSD	$T^*_{PSSD}(R,W) = \min_{a,\beta}\left\{N^*_{PSSD}(a,\beta) + \max\left\{S^*_{PSSD}, 2S^*_{PSSD} - (\beta - R)\right\}\right\}$

Πίνακας 8.2: Συγκεντρωτικός πίνακας συντελεστών χωρικής πολυπλοκότητας.

Αλγόριθμοι	**Συντελεστές χωρικής πολυπλοκότητας**
Prange	$o(1)$
Lee-Brickell	$S^*_{LB}(P,R) = H_2\left(\frac{P}{R}\right)\cdot R$
Leon	$S^*_{Leon}(P,R) = H_2\left(\frac{P}{R}\right)\cdot R$
Stern	$S^*_{Stern}(P,R) = H_2\left(\frac{P}{R}\right)\cdot\frac{R}{2}$
BCD	$S^*_{BCD}(P,R,Q,L) = H_2\left(\frac{P}{R}\right)\cdot\frac{R}{2} + H_2\left(\frac{Q}{L}\right)\cdot\frac{L}{2}$
FS-ISD	$S^*_{FS-ISD}(P,R,L) = H_2\left(\frac{P}{R+L}\right)\cdot\frac{R+L}{2}$
MMT-ISD	$S^*_{MMT-ISD}(P,R,L,L_2) = \max\left\{S^*_{MMT-ISD(2)}(R,P,L), S^*_{MMT-ISD(1)}(R,P,L,L_2)\right\}$
BJMM-ISD	$S^*_{BJMM-ISD}(P,R,L,R_1,R_2) = \max\left\{S^*_{BMMJ_1SD(3)}(R,P_2,L), S^*_{BJMM-ISD(2)}(P_2,R,L,R_2), S^*_{BJMM-ISD(1)}(P_1,R,L,R_1)\right\}$
PSSD	$S^*_{PSSD}(a,\beta) = H_2\left(\frac{a}{\beta}\right)\cdot\frac{\beta}{2}$

Υπολογίζοντας το όριο της $\mathsf{Pr}^*_{success}$, προκύπτει ο ασυμπτωτικός εκθέτης της πιθανότητας επιτυχίας $B^* = \lim_{n\to\infty}\mathsf{Pr}^*_{success}$ ως ακολούθως

$$B^*(R,W) = H_2\left(\frac{W}{1-R}\right)\cdot(1-R) - H_2(W). \tag{8.2}$$

Τελικά, η ασυμπτωτική έκφραση της πιθανότητας επιτυχίας (για μια επανάληψη του αλγορίθμου), είναι $\mathsf{Pr}_{success} = 2^{nB^*(R,W)+o(1)}$.

8.1.3 Ασυμπτωτική ανάλυση του αλγορίθμου των Lee-Brickell

Ασυμπτωτική έκφραση πιθανότητας επιτυχίας Σύμφωνα με την (6.11) και τον ορισμό 2.32, υπολογίζεται ο ασυμπτωτικός εκθέτης της $\mathsf{Pr}_{success}$, ως εξής

$$\mathsf{Pr}_{success} \simeq 2^{H_2\left(\frac{p}{k}\right)\cdot k + H_2\left(\frac{\omega-p}{n-k}\right)\cdot(n-k) - H_2\left(\frac{\omega}{n}\right)\cdot n}.$$

Ορίζοντας την $\mathsf{Pr}^*_{success} = \frac{1}{n}\log_2 \mathsf{Pr}_{success}$ όπως στην (8.1)

$$\begin{aligned}\mathsf{Pr}^*_{success} &= H_2\left(\frac{p}{k}\right)\cdot\frac{k}{n} + H_2\left(\frac{\omega-p}{n-k}\right)\cdot\left(1-\frac{k}{n}\right) - H_2\left(\frac{\omega}{n}\right)\\ &= H_2\left(\frac{\frac{p}{n}}{\frac{k}{n}}\right)\cdot\frac{k}{n} + H_2\left(\frac{\frac{\omega}{n}-\frac{p}{n}}{1-\frac{k}{n}}\right)\cdot\left(1-\frac{k}{n}\right) - H_2\left(\frac{\omega}{n}\right)\end{aligned}$$

και υπολογίζοντας στη συνέχεια το όριο της $\mathsf{Pr}^*_{success}$, προκύπτει ο ασυμπτωτικός εκθέτης της πιθανότητας επιτυχίας $B(P,R,W)$, ως ακολούθως

$$B^*(P,R,W) = H_2\left(\frac{P}{R}\right)\cdot R + H_2\left(\frac{W-P}{1-R}\right)\cdot(1-R) - H_2(W). \tag{8.3}$$

Τελικά, η ασυμπτωτική έκφραση της πιθανότητας επιτυχίας (για μια επανάληψη του αλγορίθμου), είναι $\mathsf{Pr}_{success} = 2^{nB^*(P,R,W)+o(1)}$.

8.1.4 Ασυμπτωτική ανάλυση του αλγορίθμου του Leon

Ασυμπτωτική έκφραση πιθανότητας επιτυχίας Σύμφωνα με την (6.15) και τον ορισμό 2.32, υπολογίζεται ο ασυμπτωτικός εκθέτης της $\mathsf{Pr}_{success}$, ως εξής

$$\mathsf{Pr}_{success} \simeq 2^{H_2\left(\frac{p}{k}\right)\cdot k + H_2\left(\frac{\omega-p}{n-k-\ell}\right)\cdot(n-k-\ell) - H_2\left(\frac{\omega}{n}\right)\cdot n}.$$

Ορίζοντας την $\mathsf{Pr}^*_{success} = \frac{1}{n}\log_2 \mathsf{Pr}_{success}$ όπως στην (8.1)

$$\begin{aligned}\mathsf{Pr}^*_{success} &= H_2\left(\frac{p}{k}\right)\cdot\frac{k}{n} + H_2\left(\frac{\omega-p}{n-k-\ell}\right)\cdot\left(1-\frac{k}{n}-\frac{\ell}{n}\right) - H_2\left(\frac{\omega}{n}\right)\\ &= H_2\left(\frac{\frac{p}{n}}{\frac{k}{n}}\right)\cdot\frac{k}{n} + H_2\left(\frac{\frac{\omega}{n}-\frac{p}{n}}{1-\frac{k}{n}-\frac{\ell}{n}}\right)\cdot\left(1-\frac{k}{n}-\frac{\ell}{n}\right) - H_2\left(\frac{\omega}{n}\right)\end{aligned}$$

και υπολογίζοντας στη συνέχεια το όριο της $\mathsf{Pr}^*_{success}$, προκύπτει ο ασυμπτωτικός εκθέτης της πιθανότητας επιτυχίας $B^*(P,L,R,W)$, ως ακολούθως

$$B^*(P,L,R,W) = H_2\left(\frac{P}{R}\right)\cdot R + H_2\left(\frac{W-P}{1-R-L}\right)\cdot(1-R-L) - H_2(W) \tag{8.4}$$

Τελικά, η ασυμπτωτική έκφραση της πιθανότητας επιτυχίας (για μια επανάληψη του αλγορίθμου), είναι $\mathsf{Pr}_{success} = 2^{nB^*(P,L,R,W)+o(1)}$.

8.1.5 Ασυμπτωτική ανάλυση του αλγορίθμου του Stern

Ασυμπτωτική έκφραση πιθανότητας επιτυχίας Σύμφωνα με την (6.20) και τον ορισμό 2.32, υπολογίζεται ο ασυμπτωτικός εκθέτης της $\mathsf{Pr}_{success}$, ως εξής

$$\mathsf{Pr}_{success} \simeq 2^{2H_2\left(\frac{p/2}{k/2}\right)\cdot\frac{k}{2}+H_2\left(\frac{\omega-p}{n-k-\ell}\right)\cdot(n-k-\ell)-H_2\left(\frac{\omega}{n}\right)\cdot n}.$$

Ορίζοντας την $\mathsf{Pr}^*_{success} = \frac{1}{n}\log_2 \mathsf{Pr}_{success}$ όπως στην (8.1)

$$\begin{aligned}\mathsf{Pr}^*_{success} &= H_2\left(\frac{p}{k}\right)\cdot\frac{k}{n} + H_2\left(\frac{\omega-p}{n-k-\ell}\right)\cdot\left(1-\frac{k}{n}-\frac{\ell}{n}\right) - H_2\left(\frac{\omega}{n}\right)\\ &= H_2\left(\frac{\frac{p}{n}}{\frac{k}{n}}\right)\cdot\frac{k}{n} + H_2\left(\frac{\frac{\omega}{n}-\frac{p}{n}}{1-\frac{k}{n}-\frac{\ell}{n}}\right)\cdot\left(1-\frac{k}{n}-\frac{\ell}{n}\right) - H_2\left(\frac{\omega}{n}\right)\end{aligned}$$

και υπολογίζοντας στη συνέχεια το όριο της $\mathsf{Pr}^*_{success}$, προκύπτει ο ασυμπτωτικός εκθέτης της πιθανότητας επιτυχίας $B(P,L)$, ως ακολούθως

$$B^*(P,L,R,W) = H_2\left(\frac{P}{R}\right)\cdot R + H_2\left(\frac{W-P}{1-R-L}\right)\cdot(1-R-L) - H_2(W). \tag{8.5}$$

Τελικά, η ασυμπτωτική έκφραση της πιθανότητας επιτυχίας (για μια επανάληψη του αλγορίθμου), είναι $\mathsf{Pr}_{success} = 2^{nB^*(P,L,R,W)+o(1)}$.

Συντελεστής πολυπλοκότητας Ο αριθμός των επαναλήψεων που απαιτούνται προκειμένου ο Αλγ. 6.4 να συγκλίνει στην επιθυμητή λύση, συμβολίζεται με N_{Stern} και προκύπτει ως εξής

$$N_{Stern} = \mathsf{Pr}^{-1}_{success} = \left(\frac{\binom{k/2}{p/2}^2\binom{n-k-\ell}{\omega-p}}{\binom{n}{\omega}}\right)^{-1}. \tag{8.6}$$

Με βάση τον ορισμό 2.32, ο ασυμπτωτικός εκθέτης του N_{Stern} υπολογίζεται, ως εξής

$$N_{Stern} \simeq 2^{-2H_2\left(\frac{p/2}{k/2}\right)\cdot\frac{k}{2}-H_2\left(\frac{\omega-p}{n-k-\ell}\right)\cdot(n-k-\ell)+H_2\left(\frac{\omega}{n}\right)\cdot n}.$$

Ορίζοντας την $N^*_{Stern} = \frac{1}{n}\log_2 N_{Stern}$

$$\begin{aligned}N^*_{Stern} &= -H_2\left(\frac{p}{k}\right)\cdot\frac{k}{n} - H_2\left(\frac{\omega-p}{n-k-\ell}\right)\cdot\left(1-\frac{k}{n}-\frac{\ell}{n}\right) + H_2\left(\frac{\omega}{n}\right)\\ &= -H_2\left(\frac{\frac{p}{n}}{\frac{k}{n}}\right)\cdot\frac{k}{n} - H_2\left(\frac{\frac{\omega}{n}-\frac{p}{n}}{1-\frac{k}{n}-\frac{\ell}{n}}\right)\cdot\left(1-\frac{k}{n}-\frac{\ell}{n}\right) + H_2\left(\frac{\omega}{n}\right)\end{aligned}$$

και υπολογίζοντας στη συνέχεια το όριο της N^*_{Stern}, προκύπτει ότι

$$N^*_{Stern}(R,W,P,L) = H_2(W) - H_2\left(\frac{P}{R}\right)\cdot R - H_2\left(\frac{W-P}{1-R-L}\right)\cdot(1-R-L). \tag{8.7}$$

Στη συνέχεια, υπολογίζεται η πολυπλοκότητα που προκύπτει με βάση το μέγεθος των λιστών $\mathcal{L}_1$ και $\mathcal{L}_2$, η οποία συμβολίζεται με S^*_{Stern} και εκφράζει τον συντελεστή χωρικής πολυπλοκότητας ως εξής

$$|\mathcal{L}_i| = \binom{k/2}{p/2}, \quad \forall i = 1, 2 \tag{8.8}$$

όπου $|\mathcal{L}_i|$ εκφράζει την πληθικότητα της λίστας $\mathcal{L}_i$. Με βάση τον ορισμό 2.32, προκύπτει

$$|\mathcal{L}_i| = 2^{H_2\left(\frac{p}{k}\right)\cdot\frac{k}{2}}.$$

Ορίζοντας την $|\mathcal{M}_i| = \frac{1}{n}\log_2|\mathcal{L}_i|$, έχουμε

$$|\mathcal{M}_i| = H_2\left(\frac{\frac{p}{n}}{\frac{k}{n}}\right)\cdot\frac{k}{2n} \tag{8.9}$$

και υπολογίζοντας στη συνέχεια το όριο της (8.9)

$$S^*_{Stern}(P, R) = H_2\left(\frac{P}{R}\right)\cdot\frac{R}{2}. \tag{8.10}$$

Κατόπιν, υπολογίζεται η πολυπλοκότητα που απαιτείται για τον έλεγχο των συγκρούσεων ως εξής:

$$|\mathcal{L}_1|\cdot|\mathcal{L}_2|\cdot 2^{-\ell} = \binom{k/2}{p/2}^2\cdot 2^{-\ell}. \tag{8.11}$$

Με βάση τον ορισμό 2.32 προκύπτει

$$|\mathcal{L}_1|\cdot|\mathcal{L}_2|\cdot 2^{-\ell} = 2^{H_2\left(\frac{p}{k}\right)\cdot k-\ell} \tag{8.12}$$

τελικά υπολογίζοντας το όριο της (8.12), έχουμε

$$\lim_{n\to\infty}\frac{1}{n}\log(|\mathcal{L}_1|\cdot|\mathcal{L}_2|\cdot 2^{-\ell}) = H_2\left(\frac{P}{R}\right)\cdot R - L.$$

Επομένως, το κόστος που απαιτείται σε κάθε επανάληψη του αλγορίθμου είναι

$$\begin{aligned} C^*_{Stern}(P, R, L) &= \max\left\{H_2\left(\frac{P}{R}\right)\cdot\frac{R}{2}, H_2\left(\frac{P}{R}\right)\cdot R - L\right\} \\ &= \max\left\{S^*_{Stern}(P, R), 2S^*_{Stern}(P, R) - L\right\} \end{aligned} \tag{8.13}$$

συνεπώς, ο συνολικός συντελεστής χρονικής πολυπλοκότητας υπολογίζεται ως εξής

$$T^*_{Stern}(R, W) = \min_{P,L}\left\{N^*_{Stern}(R, W, P, L) + C^*_{Stern}(P, R, L)\right\}. \tag{8.14}$$

8.1.6 Ασυμπτωτική ανάλυση του αλγορίθμου FS-ISD

Ασυμπτωτική έκφραση πιθανότητας επιτυχίας Σύμφωνα με την (6.26) και τον ορισμό 2.32, υπολογίζεται ο ασυμπτωτικός εκθέτης της $\mathsf{Pr}_{success}$, ως εξής

$$\mathsf{Pr}_{success} \simeq 2^{2H_2\left(\frac{p/2}{k+\ell/2}\right)\cdot\frac{k+\ell}{2}+H_2\left(\frac{\omega-p}{n-k-\ell}\right)\cdot(n-k-\ell)-H_2\left(\frac{\omega}{n}\right)\cdot n}.$$

Ορίζοντας την $\mathsf{Pr}^*_{success} = \frac{1}{n}\log_2 \mathsf{Pr}_{success}$ όπως στην (8.1)

$$\begin{aligned}\mathsf{Pr}^*_{success} &= H_2\left(\frac{p}{k+\ell}\right)\cdot\frac{k+\ell}{n} + H_2\left(\frac{\omega-p}{n-k-\ell}\right)\cdot\left(1-\frac{k}{n}-\frac{\ell}{n}\right) - H_2\left(\frac{\omega}{n}\right)\\ &= H_2\left(\frac{\frac{p}{n}}{\frac{k+\ell}{n}}\right)\cdot\frac{k+\ell}{n} + H_2\left(\frac{\frac{\omega}{n}-\frac{p}{n}}{1-\frac{k}{n}-\frac{\ell}{n}}\right)\cdot\left(1-\frac{k}{n}-\frac{\ell}{n}\right) - H_2\left(\frac{\omega}{n}\right)\end{aligned}$$

τελικά υπολογίζοντας το όριο της $\mathsf{Pr}^*_{success}$, προκύπτει ο ασυμπτωτικός εκθέτης της πιθανότητας επιτυχίας $B(P,L)$, ως ακολούθως:

$$B^*(R,W,P,L) = H_2\left(\frac{P}{R+L}\right)\cdot(R+L) + H_2\left(\frac{W-P}{1-R-L}\right)\cdot(1-R-L) - H_2(W). \quad (8.15)$$

Τελικά, η ασυμπτωτική έκφραση της πιθανότητας επιτυχίας (για μια επανάληψη του αλγορίθμου), είναι $\mathsf{Pr}_{success} = 2^{nB^*(R,W,P,L)+o(1)}$.

Συντελεστής πολυπλοκότητας Ο αριθμός των επαναλήψεων που απαιτούνται προκειμένου ο αλγόριθμος FS-ISD να συγκλίνει στην επιθυμητή λύση, συμβολίζεται με N_{FS-ISD} και δίνεται από τη σχέση

$$N_{FS-ISD} = \mathsf{Pr}^{-1}_{success} = \left(\frac{\binom{k+\ell}{p}\binom{n-k-\ell}{\omega-p}}{\binom{n}{\omega}}\right)^{-1}. \quad (8.16)$$

Με βάση τον ορισμό 2.32 ο ασυμπτωτικός εκθέτης του N_{FS-ISD} υπολογίζεται, ως εξής

$$N_{FS-ISD} \simeq 2^{-H_2\left(\frac{p}{k+\ell}\right)\cdot(k+\ell)-H_2\left(\frac{\omega-p}{n-k-\ell}\right)\cdot(n-k-\ell)+H_2\left(\frac{\omega}{n}\right)\cdot n}.$$

Ορίζοντας την $N^*_{FS-ISD} = \frac{1}{n}\log_2 N_{FS-ISD}$

$$\begin{aligned}N^*_{FS-ISD} &= -H_2\left(\frac{p}{k+\ell}\right)\cdot\frac{k+\ell}{n} - H_2\left(\frac{\omega-p}{n-k-\ell}\right)\cdot\left(1-\frac{k}{n}-\frac{\ell}{n}\right) + H_2\left(\frac{\omega}{n}\right)\\ &= -H_2\left(\frac{\frac{p}{n}}{\frac{k+\ell}{n}}\right)\cdot\frac{k+\ell}{n} - H_2\left(\frac{\frac{\omega}{n}-\frac{p}{n}}{1-\frac{k}{n}-\frac{\ell}{n}}\right)\cdot\left(1-\frac{k}{n}-\frac{\ell}{n}\right) + H_2\left(\frac{\omega}{n}\right)\end{aligned}$$

και υπολογίζοντας στη συνέχεια το όριο της N^*_{FS-ISD}, προκύπτει ότι

$$N^*_{FS-ISD}(R,W,P,L) = H_2(W) - H_2\left(\frac{P}{R+L}\right)\cdot(R+L) - H_2\left(\frac{W-P}{1-R-L}\right)\cdot(1-R-L). \quad (8.17)$$

Στη συνέχεια, υπολογίζεται η πολυπλοκότητα που προκύπτει με βάση το μέγεθος των λιστών $\mathcal{L}_1$ και $\mathcal{L}_2$, η οποία συμβολίζεται με S^*_{FS-ISD} και εκφράζει τον συντελεστή χωρικής πολυπλοκότητας ως εξής

$$|\mathcal{L}_i| = \sqrt{\binom{k+\ell}{p}} \approx \binom{(k+\ell)/2}{p/2}, \quad \forall i = 1,2 \tag{8.18}$$

όπου $|\mathcal{L}_i|$ εκφράζει την πληθικότητα της λίστας $\mathcal{L}_i$. Με βάση τον ορισμό 2.32 προκύπτει

$$|\mathcal{L}_i| = 2^{H_2\left(\frac{\frac{p}{2}}{\frac{k+\ell}{2}}\right)\cdot\frac{k+\ell}{2}}.$$

Ορίζοντας την $|\mathcal{M}_i| = \frac{1}{n}\log_2|\mathcal{L}_i|$, έχουμε

$$|\mathcal{M}_i| = H_2\left(\frac{\frac{p}{n}}{\frac{k+\ell}{n}}\right)\cdot\frac{k+\ell}{2n} \tag{8.19}$$

και υπολογίζοντας στη συνέχεια το όριο της (8.19)

$$S^*_{FS-ISD}(P,R,L) = H_2\left(\frac{P}{R+L}\right)\cdot\frac{R+L}{2}. \tag{8.20}$$

Κατόπιν, υπολογίζεται η πολυπλοκότητα που απαιτείται για τον έλεγχο των συγκρούσεων ως εξής:

$$|\mathcal{L}_1|\cdot|\mathcal{L}_2|\cdot 2^{-\ell} = \binom{(k+\ell)/2}{p/2}^2\cdot 2^{-\ell}. \tag{8.21}$$

Με βάση τον ορισμό 2.32 προκύπτει

$$|\mathcal{L}_1|\cdot|\mathcal{L}_2|\cdot 2^{-\ell} = 2^{H_2\left(\frac{p}{k+\ell}\right)\cdot(k+\ell)-\ell} \tag{8.22}$$

τελικά υπολογίζοντας το όριο της (8.22), έχουμε

$$\lim_{n\to\infty}\frac{1}{n}\log(|\mathcal{L}_1|\cdot|\mathcal{L}_2|\cdot 2^{-\ell}) = H_2\left(\frac{P}{R+L}\right)\cdot(R+L)-L.$$

Επομένως, το κόστος που απαιτείται σε κάθε επανάληψη του αλγορίθμου είναι

$$\begin{aligned} C^*_{FS-ISD}(P,R,L) &= \max\left\{H_2\left(\frac{P}{R+L}\right)\cdot\frac{R+L}{2}, H_2\left(\frac{P}{R+L}\right)\cdot(R+L)-L\right\} \\ &= \max\left\{S^*_{FS-ISD}(P,R,L), 2S^*_{FS-ISD}(P,R,L)-L\right\} \end{aligned} \tag{8.23}$$

συνεπώς, ο συνολικός συντελεστής χρονικής πολυπλοκότητας υπολογίζεται ως εξής

$$T^*_{FS-ISD}(R,W) = \min_{P,L}\left\{N^*_{FS-ISD}(R,W,P,L) + C^*_{FS-ISD}(P,R,L)\right\}. \tag{8.24}$$

8.1.7 Ασυμπτωτική ανάλυση του αλγορίθμου BCD

Ασυμπτωτική ανάλυση πιθανότητας επιτυχίας Σύμφωνα με την (6.37) και τον ορισμό 2.32, υπολογίζεται ο ασυμπτωτικός εκθέτης της $\mathsf{Pr}_{success}$, ως εξής

$$\mathsf{Pr}_{success} \simeq 2^{2H_2\left(\frac{p/2}{k/2}\right)\cdot\frac{k}{2}+2H_2\left(\frac{q/2}{\ell/2}\right)\cdot\frac{\ell}{2}+H_2\left(\frac{\omega-p-q}{n-k-\ell}\right)\cdot(n-k-\ell)-H_2\left(\frac{\omega}{n}\right)\cdot n}.$$

Ορίζοντας την $\mathsf{Pr}^*_{success} = \frac{1}{n}\log_2 \mathsf{Pr}_{success}$ όπως στην (8.1)

$$\begin{aligned}\mathsf{Pr}^*_{success} &= H_2\left(\frac{p}{k}\right)\cdot\frac{k}{n} + H\left(\frac{q}{\ell}\right)\cdot\frac{\ell}{n} + H_2\left(\frac{\omega-p-q}{n-k-\ell}\right)\cdot\left(1-\frac{k}{n}-\frac{\ell}{n}\right) - H_2\left(\frac{\omega}{n}\right)\\ &= H_2\left(\frac{\frac{p}{n}}{\frac{k}{n}}\right)\cdot\frac{k}{n} + H_2\left(\frac{\frac{q}{n}}{\frac{\ell}{n}}\right)\cdot\frac{\ell}{n} + H_2\left(\frac{\frac{\omega}{n}-\frac{p}{n}-\frac{q}{n}}{1-\frac{k}{n}-\frac{\ell}{n}}\right)\cdot\left(1-\frac{k}{n}-\frac{\ell}{n}\right) - H_2\left(\frac{\omega}{n}\right)\end{aligned}$$

και υπολογίζοντας στη συνέχεια το όριο της $\mathsf{Pr}^*_{success}$, προκύπτει ο ασυμπτωτικός εκθέτης της πιθανότητας επιτυχίας $B(P,Q,L)$, ως ακολούθως

$$B^*(R,W,P,Q,L) = H_2\left(\frac{P}{R}\right)\cdot R + H_2\left(\frac{Q}{L}\right)\cdot L + H_2\left(\frac{W-P-Q}{1-R-L}\right)\cdot(1-R-L) - H_2(W). \tag{8.25}$$

Τελικά, η ασυμπτωτική έκφραση της πιθανότητας επιτυχίας (για μια επανάληψη του αλγορίθμου), είναι $\mathsf{Pr}_{success} = 2^{n(B^*(R,W,P,Q,L))+o(1)}$.

Συντελεστής πολυπλοκότητας Ο αριθμός των επαναλήψεων που απαιτούνται προκειμένου ο αλγόριθμος BCD να συγκλίνει στην επιθυμητή λύση, συμβολίζεται με N_{BCD} και ισούται με

$$N_{BCD} = \mathsf{Pr}^{-1}_{success} = \left(\frac{\binom{k/2}{p/2}^2\binom{\ell/2}{q/2}^2\binom{n-k-\ell}{\omega-p-q}}{\binom{n}{\omega}}\right)^{-1}. \tag{8.26}$$

Με βάση τον ορισμό 2.32 ο ασυμπτωτικός εκθέτης του N_{BCD} υπολογίζεται, ως εξής

$$N_{BCD} \simeq 2^{-2H_2\left(\frac{p/2}{k/2}\right)\cdot\frac{k}{2}-2H_2\left(\frac{q/2}{\ell/2}\right)\cdot\frac{\ell}{2}-H_2\left(\frac{\omega-p-q}{n-k-\ell}\right)\cdot(n-k-\ell)+H_2\left(\frac{\omega}{n}\right)\cdot n}.$$

Ορίζοντας την $N^*_{BCD} = \frac{1}{n}\log_2 N_{BCD}$

$$\begin{aligned}N^*_{BCD} &= -H_2\left(\frac{p}{k}\right)\cdot\frac{k}{n} - H_2\left(\frac{q}{\ell}\right)\cdot\frac{\ell}{n} - H_2\left(\frac{\omega-p-q}{n-k-\ell}\right)\cdot\left(1-\frac{k}{n}-\frac{\ell}{n}\right) + H_2\left(\frac{\omega}{n}\right)\\ &= -H_2\left(\frac{\frac{p}{n}}{\frac{k}{n}}\right)\cdot\frac{k}{n} - H_2\left(\frac{\frac{q}{n}}{\frac{\ell}{n}}\right)\cdot\frac{\ell}{n} - H_2\left(\frac{\frac{\omega}{n}-\frac{p}{n}-\frac{q}{n}}{1-\frac{k}{n}-\frac{\ell}{n}}\right)\cdot\left(1-\frac{k}{n}-\frac{\ell}{n}\right) + H_2\left(\frac{\omega}{n}\right)\end{aligned}$$

και υπολογίζοντας στη συνέχεια το όριο της N^*_{BCD}, προκύπτει ότι

$$\begin{aligned}N^*_{BCD}(R,W,P,Q,L) = H_2(W) &- H_2\left(\frac{P}{R}\right)\cdot R - H_2\left(\frac{Q}{L}\right)\cdot L\\ &- H_2\left(\frac{W-P-Q}{1-R-L}\right)\cdot(1-R-L).\end{aligned} \tag{8.27}$$

Στη συνέχεια, υπολογίζεται η πολυπλοκότητα που προκύπτει με βάση το μέγεθος των λιστών $\mathcal{L}_1$ και $\mathcal{L}_2$, η οποία συμβολίζεται με S^*_{BCD} και εκφράζει τον συντελεστή χωρικής πολυπλοκότητας ως εξής

$$|\mathcal{L}_i| = \binom{k/2}{p/2}\binom{\ell/2}{q/2}, \quad \forall i = 1, 2 \tag{8.28}$$

όπου $|\mathcal{L}_i|$ εκφράζει την πληθικότητα της λίστας $\mathcal{L}_i$. Με βάση τον ορισμό 2.32 προκύπτει

$$|\mathcal{L}_i| = 2^{H_2\left(\frac{p}{k}\right)\cdot\frac{k}{2}+H_2\left(\frac{q}{\ell}\right)\cdot\frac{\ell}{2}}.$$

Ορίζοντας την $|\mathcal{M}_i| = \frac{1}{n}\log_2|\mathcal{L}_i|$, έχουμε

$$|\mathcal{M}_i| = H_2\left(\frac{\frac{p}{n}}{\frac{k}{n}}\right)\cdot\frac{k}{2n} + H_2\left(\frac{\frac{q}{n}}{\frac{\ell}{n}}\right)\cdot\frac{\ell}{2n} \tag{8.29}$$

και υπολογίζοντας στη συνέχεια το όριο της (8.29)

$$S^*_{BCD}(R, L, P, Q) = H_2\left(\frac{P}{R}\right)\cdot\frac{R}{2} + H_2\left(\frac{Q}{L}\right)\cdot\frac{L}{2}. \tag{8.30}$$

Κατόπιν, υπολογίζεται η πολυπλοκότητα που απαιτείται για τον έλεγχο των συγκρούσεων ως εξής

$$|\mathcal{L}_1|\cdot|\mathcal{L}_2|\cdot 2^{-\ell} = \binom{k/2}{p/2}^2\cdot\binom{\ell/2}{q/2}^2\cdot 2^{-\ell}. \tag{8.31}$$

Με βάση τον ορισμό 2.32 προκύπτει

$$|\mathcal{L}_1|\cdot|\mathcal{L}_2|\cdot 2^{-\ell} = 2^{H_2\left(\frac{p}{k}\right)\cdot k+H_2\left(\frac{q}{\ell}\right)\cdot\ell-\ell} \tag{8.32}$$

τελικά υπολογίζοντας το όριο της (8.32), έχουμε

$$\lim_{n\to\infty}\frac{1}{n}\log(|\mathcal{L}_1|\cdot|\mathcal{L}_2|\cdot 2^{-\ell}) = H_2\left(\frac{P}{R}\right)\cdot R + H_2\left(\frac{Q}{L}\right)\cdot L - L \tag{8.33}$$

Επομένως, το κόστος που απαιτείται σε κάθε επανάληψη του αλγορίθμου είναι

$$\begin{aligned} C^*_{BCD}(P, R, Q, L) &= \max\left\{H_2\left(\frac{P}{R}\right)\cdot\frac{R}{2} + H_2\left(\frac{Q}{L}\right)\cdot\frac{L}{2}, H_2\left(\frac{P}{R}\right)\cdot R + H_2\left(\frac{Q}{L}\right)\cdot L - L\right\} \\ &= \max\left\{S^*_{BCD}(R, L, P, Q), 2S^*_{BCD}(R, L, P, Q) - L\right\} \end{aligned} \tag{8.34}$$

και τελικά ο συνολικός συντελεστής χρονικής πολυπλοκότητας υπολογίζεται ως εξής

$$T^*_{BCD}(R, W) = \min_{P,Q,L}\left\{N^*_{BCD}(R, W, P, Q, L) + C^*_{BCD}(P, Q, L)\right\}. \tag{8.35}$$

Ασυμπτωτική συσχέτιση μεταξύ των αλγορίθμων BCD και FS-ISD

Το 2011 οι May, Meurer και Thomae [63] χρησιμοποιώντας την ταυτότητα Vandermonde

Λήμμα 8.1 (Ταυτότητα Vandermonde). *Έστω $R, L > 0$. Για $0 \leq P \leq R$ και $0 \leq Q \leq L$, ισχύει ότι*

$$RH_2\left(\frac{P}{R}\right) + LH_2\left(\frac{Q}{L}\right) \leq (R+L)H_2\left(\frac{P+Q}{R+L}\right). \tag{8.36}$$

Επιπλέον, αν $\frac{P}{R} = \frac{Q}{L}$ τότε η (8.36) ισχύει με ισότητα.

απέδειξαν με βάση το θεώρημα 8.7, πως μετασχηματίζονται οι βέλτιστες παράμετροι (P, Q, L) του αλγορίθμου BCD, σε έγκυρες (όχι απαραίτητα βέλτιστες) παραμέτρους (P', L') για τον αλγόριθμο FS-ISD. Συγκεκριμένα

Θεώρημα 8.2. *Εάν (P, Q, L) ένα σύνολο παραμέτρων για τον αλγόριθμο BCD και $(P+Q, L)$ ένα σύνολο παραμέτρων για τον αλγόριθμο FS-ISD, τότε*

$$T^*_{BCD}(P, Q, L) \geq F_{FS-ISD}(P+Q, L),$$

δηλαδή ο αλγόριθμος FS-ISD είναι ασυμπτωτικά τουλάχιστον, τόσο αποδοτικός όσο ο αλγόριθμος BCD.

Απόδειξη. Έστω (P, Q, L) ένα έγκυρο σύνολο παραμέτρων για τον αλγόριθμο BCD, δηλαδή

$$\begin{aligned} &0 \leq L \leq 1 - R, \quad 0 \leq Q \leq \min\{L, W\}, \\ &\max\{0, R + L + W - Q - 1\} \leq P \leq \min\{R, W - Q\}. \end{aligned}$$

Τότε $(P', L') := (P+Q, L)$ είναι ένα έγκυρο σύνολο παραμέτρων για τον αλγόριθμο FS-ISD, εφόσον $0 \leq L' \leq 1 - R$ και συνεπώς

$$\begin{aligned} P' = P + Q &\leq \min\{R + L, W\} = \min\{R + L', W\}, \\ P' = P + Q &\geq \max\{0, R + W + L - 1\} = \max\{0, R + W + L' - 1\}. \end{aligned}$$

Συνδυάζοντας αρχικά τις (8.34) και (8.35), θεωρώντας ότι το $2S^*_{BCD} - L$ είναι το μέγιστο στην (8.34) (δηλαδή $L \leq S^*_{BCD}$) και με βάση το λήμμα 8.1 προκύπτει

$$S^*_{BCD}(P, Q, L) \leq \frac{R+L}{2} H_2\left(\frac{P+Q}{R+L}\right) = S^*_{FS-ISD}(P+Q, L) \tag{8.37}$$

σύμφωνα με την (8.20). Άρα $L \leq S^*_{FS-ISD}$ και συνεπώς στην (8.23) το $2S^*_{FS-ISD} - L$ θεωρείται επίσης μέγιστο. Επομένως, έχουμε

$$\begin{aligned} T^*_{FS-ISD}(P+Q, L) &= N^*_{FS-ISD}(P+Q, L) + 2S^*_{FS-ISD}(P+Q, L) - L \\ &= H_2(W) - (1 - R - L) \cdot H_2\left(\frac{W - (P+Q)}{1 - R - L}\right) - L \\ &= N^*_{BCD}(P, Q, L) + 2S^*_{BCD}(P, Q, L) - L \\ &= T^*_{BCD}(P, Q, L). \end{aligned} \tag{8.38}$$

Εν συνεχεία, εξετάζεται η περίπτωση κατά την οποία $L > S^*_{BCD}$, θεωρώντας αρχικά $L > S^*_{FS-ISD}$, δηλαδή S^*_{FS-ISD} είναι το μέγιστο, με βάση την (8.23). Συνεπώς, προκύπτει

$$\begin{aligned} T^*_{FS-ISD}(P+Q,L) &= N^*_{FS-ISD}(P+Q,L) + S^*_{FS-ISD}(P+Q,L) \\ &= H_2(W) - (1-R-L)\cdot H_2\left(\frac{W-(P+Q)}{1-R-L}\right) - \frac{R+L}{2}\cdot H_2\left(\frac{P+Q}{R+L}\right) \\ &\leq H_2(W) - (1-R-L)\cdot H_2\left(\frac{W-P-Q}{1-R-L}\right) - \frac{R}{2}\cdot H\left(\frac{P}{R}\right) - \frac{L}{2}\cdot H_2\left(\frac{Q}{L}\right) \\ &= N^*_{BCD}(P,Q,L) + S^*_{BCD}(P,Q,L) \\ &= T^*_{BCD}(P,Q,L) \end{aligned} \tag{8.39}$$

όπου για την ανισότητα χρησιμοποιείται το λήμμα 8.1. Ωστόσο, για $L < S^*_{FS-ISD}$ σύμφωνα με την (8.23) το μέγιστο είναι $2S^*_{FS-ISD} - L$, άρα

$$\begin{aligned} T^*_{FS-ISD}(P+Q,L) &= N^*_{FS-ISD}(P+Q,L) + 2S^*_{FS-ISD}(P+Q,L) - L \\ &= H_2(W) - (1-R-L)\cdot H_2\left(\frac{W-(P+Q)}{1-R-L}\right) - L \\ &< H_2(W) - (1-R-L)\cdot H_2\left(\frac{W-(P+Q)}{1-R-L}\right) - S^*_{BCD}(P,Q,L) \\ &= N^*_{BCD}(P,Q,L) + S^*_{BCD}(P,Q,L) \\ &= T^*_{BCD}(P,Q,L). \end{aligned} \tag{8.40}$$

Τελικά, σε όλες τις περιπτώσεις $T^*_{FS-ISD}(P+Q,L) \leq T^*_{BCD}(P,Q,L)$. ■

8.1.8 Ασυμπτωτική ανάλυση του αλγορίθμου MMT-ISD

Θα εξετάσουμε την πολυπλοκότητα που απαιτείται ανά επίπεδο. Υπολογίζοντας αρχικά, την πολυπλοκότητα που προκύπτει στο επίπεδο-2, με βάση το μέγεθος των λιστών $\mathcal{L}_{1,1}$, $\mathcal{L}_{1,2}$, $\mathcal{L}_{2,1}$ και $\mathcal{L}_{2,2}$, η οποία συμβολίζεται με $S^*_{MMT-ISD(2)}$ και εκφράζει τον συντελεστή χωρικής πολυπλοκότητας

$$|\mathcal{L}_{i,j}| = \binom{(k+\ell)/2}{p/4}, \quad \forall i = 1,2,\ j = 1,2 \tag{8.41}$$

όπου $|\mathcal{L}_{i,j}|$ εκφράζει την πληθικότητα της λίστας $\mathcal{L}_{i,j}$. Με βάση τον ορισμό 2.32 προκύπτει

$$|\mathcal{L}_{i,j}| = 2^{H_2\left(\frac{\frac{p}{4}}{\frac{k+\ell}{2}}\right)\cdot\frac{k+\ell}{2}}.$$

Ορίζοντας την $|\mathcal{M}_{i,j}| = \frac{1}{n}\log_2|\mathcal{L}_{i,j}|$, έχουμε

$$|\mathcal{M}_{i,j}| = H_2\left(\frac{\frac{p}{n}}{\frac{2(k+\ell)}{n}}\right)\cdot\frac{k+\ell}{2n} \tag{8.42}$$

και υπολογίζοντας στη συνέχεια το όριο της (8.42)

$$S^*_{MMT-ISD(2)}(P,R,L) = H_2\left(\frac{P}{2(R+L)}\right)\cdot\frac{R+L}{2}. \tag{8.43}$$

Κατόπιν, υπολογίζεται η πολυπλοκότητα που απαιτείται για τον έλεγχο συγκρούσεων μεταξύ των λιστών $\mathcal{L}_{1,1}$, $\mathcal{L}_{1,2}$, $\mathcal{L}_{2,1}$, $\mathcal{L}_{2,2}$, η οποία συμβολίζεται με $\Lambda_{MMT-ISD(2)}$ ως εξής

$$|\mathcal{L}_{1,1}|\cdot|\mathcal{L}_{1,2}|\cdot|\mathcal{L}_{2,1}|\cdot|\mathcal{L}_{2,2}|\cdot 2^{-\ell_2} = \binom{(k+\ell)/2}{p/4}^4\cdot 2^{-\ell_2}. \tag{8.44}$$

Με βάση τον ορισμό 2.32 προκύπτει

$$|\mathcal{L}_{1,1}|\cdot|\mathcal{L}_{1,2}|\cdot|\mathcal{L}_{2,1}|\cdot|\mathcal{L}_{2,2}|\cdot 2^{-\ell_2} = 2^{H_2\left(\frac{p}{2(k+\ell)}\right)\cdot 2(k+\ell)-\ell_2} \tag{8.45}$$

τελικά υπολογίζοντας το όριο της (8.45)

$$\Lambda_{MMT-ISD(2)}(P,R,L,L_2) = H_2\left(\frac{P}{2(R+L)}\right)\cdot 2(R+L) - L_2. \tag{8.46}$$

Επομένως, το κόστος που απαιτείται σε κάθε επανάληψη του αλγορίθμου στο επίπεδο-2 είναι

$$\begin{aligned} C^*_{MMT-ISD(2)}(P,R,L,L_2) &= \max\left\{H_2\left(\frac{P}{2(R+L)}\right)\cdot\frac{R+L}{2}, H_2\left(\frac{P}{2(R+L)}\right)\cdot 2(R+L) - L_2\right\} \\ &= \max\left\{S^*_{MMT-ISD(2)}(P,R,L), 2S^*_{MMT-ISD(2)}(P,R,L) - L_2\right\} \end{aligned} \tag{8.47}$$

Εν συνεχεία, υπολογίζεται ο αριθμός των επαναλήψεων που απαιτούνται προκειμένου ο αλγόριθμος MMT-ISD να συγκλίνει στην επιθυμητή λύση στο επίπεδο-1, ο οποίος συμβολίζεται με $N_{MMT-ISD(1)}$

$$N_{MMT-ISD(1)} = \mathsf{Pr}^{-1}_{success} = \left(\frac{\binom{(k+\ell)/2}{p/2}^2\binom{n-k-\ell}{\omega-p}}{\binom{n}{\omega}}\right)^{-1}. \tag{8.48}$$

Με βάση τον ορισμό 2.32 ο ασυμπτωτικός εκθέτης του $N_{MMT-ISD(1)}$ υπολογίζεται, ως εξής

$$N_{MMT-ISD(1)} \simeq 2^{-H_2\left(\frac{p}{k+\ell}\right)\cdot(k+\ell)-H_2\left(\frac{\omega-p}{n-k-\ell}\right)\cdot(n-k-\ell)+H_2\left(\frac{\omega}{n}\right)\cdot n}.$$

Ορίζοντας την $N^*_{MMT-ISD(1)} = \frac{1}{n}\log_2 N_{MMT-ISD(1)}$

$$\begin{aligned} N^*_{MMT-ISD(1)} &= -H_2\left(\frac{p}{k+\ell}\right)\cdot\frac{k+\ell}{n} - H_2\left(\frac{\omega-p}{n-k-\ell}\right)\cdot\left(1-\frac{k}{n}-\frac{\ell}{n}\right) + H_2\left(\frac{\omega}{n}\right) = \\ &= -H_2\left(\frac{\frac{p}{n}}{\frac{k+\ell}{n}}\right)\cdot\frac{k+\ell}{n} - H_2\left(\frac{\frac{\omega}{n}-\frac{p}{n}}{1-\frac{k}{n}-\frac{\ell}{n}}\right)\cdot\left(1-\frac{k}{n}-\frac{\ell}{n}\right) + H_2\left(\frac{\omega}{n}\right) \end{aligned}$$

και υπολογίζοντας στη συνέχεια το όριο της $N^*_{MMT-ISD(1)}$, προκύπτει ότι

$$N^*_{MMT-ISD(1)}(R,W,P,L) = H_2(W) - H_2\left(\frac{P}{R+L}\right)\cdot(R+L) - H_2\left(\frac{W-P}{1-R-L}\right)\cdot(1-R-L). \tag{8.49}$$

Στη συνέχεια, υπολογίζεται η πολυπλοκότητα που προκύπτει με βάση το μέγεθος των λιστών $\mathcal{L}_1$ και $\mathcal{L}_2$, η οποία συμβολίζεται με $S^*_{MMT-ISD(1)}$ ως εξής

$$|\mathcal{L}_i| = \binom{(k+\ell)/2}{p/4}^2 \cdot 2^{-\ell_2}, \quad \forall i = 1,2 \tag{8.50}$$

όπου $|\mathcal{L}_i|$ εκφράζει την πληθικότητα της λίστας $\mathcal{L}_{i,}$, η οποία προκύπτει ενώνοντας τις αντίστοιχες λίστες $\mathcal{L}_{i,j}$ (επίπεδο-2). Με βάση τον ορισμό 2.32 προκύπτει

$$|\mathcal{L}_i| = 2^{H_2\left(\frac{p}{2(k+\ell)}\right)\cdot k+\ell-\ell_2}.$$

Ορίζοντας την $|\mathcal{M}_i| = \frac{1}{n}\log_2|\mathcal{L}_i|$, έχουμε

$$|\mathcal{M}_i| = H_2\left(\frac{\frac{p}{n}}{\frac{2(k+\ell)}{n}}\right)\cdot\frac{k+\ell}{n} - \frac{\ell_2}{n} \tag{8.51}$$

και υπολογίζοντας στη συνέχεια το όριο της (8.51)

$$S^*_{MMT-ISD(1)}(P,R,L,L_2) = H_2\left(\frac{P}{2(R+L)}\right)\cdot(R+L) - L_2. \tag{8.52}$$

Κατόπιν, υπολογίζεται η πολυπλοκότητα που απαιτείται για τον έλεγχο των συγκρούσεων, η οποία συμβολίζεται με $\Lambda_{MMT-ISD(1)}$ ως εξής:

$$|\mathcal{L}_1|\cdot|\mathcal{L}_2|\cdot 2^{-\ell_1} = \binom{(k+\ell)/2}{p/4}^4 \cdot 2^{-2\ell_2-\ell_1}. \tag{8.53}$$

Με βάση τον ορισμό 2.32 προκύπτει

$$|\mathcal{L}_1|\cdot|\mathcal{L}_2|\cdot 2^{-\ell_1} = 2^{H_2\left(\frac{p}{2(k+\ell)}\right)\cdot 2(k+\ell)-2\ell_2-\ell_1} \tag{8.54}$$

τελικά υπολογίζοντας το όριο της (8.54), έχουμε

$$\Lambda_{MMT-ISD(1)}(P,R,L,L_1,L_2) = H_2\left(\frac{P}{2(R+L)}\right)\cdot 2(R+L) - 2L_2 - L_1. \tag{8.55}$$

Συνεπώς, το κόστος που απαιτείται σε κάθε επανάληψη του αλγορίθμου στο επίπεδο-1 είναι

$$\begin{aligned} C^*_{MMT-ISD(1)}(P,R,L,L_1,L_2) &= \max\left\{H_2\left(\frac{P}{2(R+L)}\right)\cdot(R+L) - L_2,\right. \\ &\qquad \left. H_2\left(\frac{P}{2(R+L)}\right)\cdot 2(R+L) - 2L_2 - L_1\right\} \\ &= \max\left\{S^*_{MMT-ISD(1)}(P,R,L,L_2), 2S^*_{MMT-ISD(1)}(P,R,L,L_2) - L_1\right\} \end{aligned} \tag{8.56}$$

άρα ο συντελεστής χρονικής πολυπλοκότητας για το επίπεδο-1 θα είναι

$$T^*_{MMT-ISD(1)}(R,W) = \min_{P,L}\left\{N^*_{MMT-ISD(1)}(R,W,P,L) + C^*_{MMT-ISD(1)}(P,R,L,L_1,L_2)\right\}. \tag{8.57}$$

Επομένως, οι συνολικοί συντελεστές χωρικής και χρονικής πολυπλοκότητας, υπολογίζονται ως εξής

$$S^*_{MMT-ISD}(P,R,L,L_2) = \max\left\{S^*_{MMT-ISD(2)}(R,P,L), S^*_{MMT-ISD(1)}(R,P,L,L_2)\right\}, \tag{8.58}$$

$$\begin{aligned} T^*_{MMT-ISD}(R,W) = \min_{P,L}\Big\{N^*_{MMT-ISD}(1)(R,W,P,L) + \max\Big\{S^*_{MMT-ISD(2)}(R,P,L),\\ S^*_{MMT-ISD(1)}(R,P,L,L_2), 2S^*_{MMT-ISD(1)}(R,P,L,L_2) - L_1\Big\}\Big\}. \end{aligned} \tag{8.59}$$

8.1.9 Ασυμπτωτική ανάλυση του αλγορίθμου BJMM-ISD

Θα εξετάσουμε την πολυπλοκότητα που απαιτείται ανά επίπεδο, υπολογίζοντας αρχικά τον απαιτούμενο αριθμό των επαναλήψεων στο επίπεδο-3, που συμβολίζεται με $N_{BJMM-ISD(3)}$ και προκύπτει ως εξής

$$N_{BJMM-ISD(3)} = \mathsf{Pr}^{-1}_{split} = \left(\frac{\binom{(k+\ell)/2}{p2/2}^2}{\binom{k+\ell}{p_2}}\right)^{-1}. \tag{8.60}$$

Με βάση τον ορισμό 2.32 ο ασυμπτωτικός εκθέτης του $N_{BJMM-ISD(3)}$, θα είναι

$$N_{BJMM-ISD(3)} \simeq 2^{-H_2\left(\frac{p_2}{(k+\ell)}\right)\cdot(k+\ell)+H_2\left(\frac{p_2}{k+\ell}\right)\cdot(k+\ell)} \approx 1.$$

Στη συνέχεια, υπολογίζεται η πολυπλοκότητα που προκύπτει με βάση το μέγεθος των λιστών $\mathcal{B}_{i,1}$, $\mathcal{B}_{i,2}$ $i = 1,2,3,4$ η οποία συμβολίζεται με $S^*_{BJMM-ISD(3)}$ και εκφράζει τον ασυμπτωτικό συντελεστή χωρικής πολυπλοκότητας ως εξής

$$|\mathcal{B}_{i,j}| = \binom{(k+\ell)/2}{p_2/2}, \quad \forall i = 1,2,3,4,\ j = 1,2 \tag{8.61}$$

όπου $|\mathcal{B}_{i,j}|$ εκφράζει την πληθικότητα της λίστας $\mathcal{B}_{i,j}$. Με βάση τον ορισμό 2.32 προκύπτει

$$|\mathcal{B}_{i,j}| = 2^{H_2\left(\frac{p_2}{k+\ell}\right)\cdot\frac{k+\ell}{2}}.$$

Ορίζοντας την $|\mathcal{M}_{3_{i,j}}| = \frac{1}{n}\log_2|\mathcal{B}_{i,j}|$, έχουμε

$$|\mathcal{M}_{3_{i,j}}| = H_2\left(\frac{\frac{p_2}{n}}{\frac{k+\ell}{n}}\right)\cdot\frac{k+\ell}{2n} \tag{8.62}$$

και υπολογίζοντας στη συνέχεια το όριο της (8.62)

$$S^*_{BJMM-ISD(3)}(P_2,R,L) = H_2\left(\frac{P_2}{R+L}\right)\cdot\frac{R+L}{2} \tag{8.63}$$

Κατόπιν, υπολογίζεται από την (8.63) η πολυπλοκότητα που απαιτείται για τον έλεγχο συγκρούσεων μεταξύ των λιστών $\mathcal{B}_{i,1}$, $\mathcal{B}_{i,2}$ $i = 1, 2, 3, 4$, η οποία συμβολίζεται με $\Lambda_{BJMM-ISD(3)}$ ως εξής

$$|\mathcal{B}_{i,1}| \cdot |\mathcal{B}_{i,2}| \cdot 2^{-r_2} = \binom{(k+\ell)/2}{p_2/2}^2 \cdot 2^{-r_2}, \forall i = 1, 2, 3, 4. \tag{8.64}$$

Με βάση τον ορισμό 2.32 προκύπτει

$$|\mathcal{B}_{i,1}| \cdot |\mathcal{B}_{i,2}| \cdot 2^{-r_2} = 2^{H_2(\frac{p_2}{k+\ell}) \cdot 2(k+\ell) - r_2} \tag{8.65}$$

τελικά υπολογίζοντας το όριο της (8.65)

$$\Lambda_{BJMM-ISD(3)}(P_2, R, L, R_2) = H_2\left(\frac{P_2}{R+L}\right) \cdot 2(R+L) - R_2. \tag{8.66}$$

Επομένως, το κόστος που απαιτείται σε κάθε επανάληψη του αλγορίθμου στο επίπεδο-3 είναι

$$\begin{aligned} C^*_{BJMM-ISD(3)}(P_2, R, L, R_2) &= \max\left\{H_2\left(\frac{P_2}{R+L}\right) \cdot \frac{R+L}{2}, H_2\left(\frac{P_2}{R+L}\right) \cdot 2(R+L) - R_2\right\} \\ &= \max\left\{S^*_{BJMM-ISD(3)}(P_2, R, L), 2S^*_{BJMM-ISD(3)}(P_2, R, L) - R_2\right\} \end{aligned} \tag{8.67}$$

Παρομοίως, υπολογίζεται η πολυπλοκότητα που προκύπτει με βάση το μέγεθος των λιστών $\mathcal{L}_i^{(2)}$, $\forall i = 1, 2, 3, 4$ η οποία συμβολίζεται με $S^*_{BJMM-ISD(2)}$ για το επίπεδο-2 ως εξής

$$|\mathcal{L}_i^{(2)}| = \binom{k+\ell}{p_2} \cdot 2^{-r_2}, \quad \forall i = 1, 2, 3, 4. \tag{8.68}$$

Με βάση τον ορισμό 2.32 προκύπτει

$$|\mathcal{L}_i^{(2)}| = 2^{H_2(\frac{p_2}{k+\ell}) \cdot (k+\ell) - r_2}.$$

Ορίζοντας την $|\mathcal{M}_i^{(2)}| = \frac{1}{n} \log_2 |\mathcal{L}_i^{(2)}|$, έχουμε

$$|\mathcal{M}_i^{(2)}| = H_2\left(\frac{\frac{p_2}{n}}{\frac{k+\ell}{n}}\right) \cdot \frac{k+\ell}{n} - \frac{r_2}{n} \tag{8.69}$$

και υπολογίζοντας στη συνέχεια το όριο της (8.69)

$$S^*_{BJMM-ISD(2)}(P_2, R, L, R_2) = H_2\left(\frac{P_2}{R+L}\right) \cdot (R+L) - R_2. \tag{8.70}$$

Κατόπιν, υπολογίζεται η πολυπλοκότητα που απαιτείται για τον έλεγχο των συγκρούσεων, η οποία συμβολίζεται με $\Lambda_{BJMM-ISD(2)}$ ως εξής:

$$|\mathcal{L}_1^{(2)}| \cdot |\mathcal{L}_2^{(2)}| \cdot |\mathcal{L}_3^{(2)}| \cdot |\mathcal{L}_4^{(2)}| \cdot 2^{r_2 - r_1} = \binom{(k+\ell)}{p_2}^2 \cdot 2^{-r_2 - r_1}. \tag{8.71}$$

Με βάση τον ορισμό 2.32 προκύπτει

$$|\mathcal{L}_1^{(2)}| \cdot |\mathcal{L}_2^{(2)}| \cdot |\mathcal{L}_3^{(2)}| \cdot |\mathcal{L}_4^{(2)}| \cdot 2^{r_2 - r_1} = 2^{H_2\left(\frac{p_2}{k+\ell}\right) \cdot 2(k+\ell) - r_2 - r_1} \tag{8.72}$$

τελικά υπολογίζοντας το όριο της (8.72), έχουμε

$$\Lambda_{BJMM-ISD(2)}(P_2, R, L, R_2, R_1) = H_2\left(\frac{P_2}{R+L}\right) \cdot 2(R+L) - R_2 - R_1. \tag{8.73}$$

Επομένως, το κόστος που απαιτείται σε κάθε επανάληψη του αλγορίθμου στο επίπεδο-2 είναι

$$\begin{aligned} C_{BJMM-ISD(2)}(P_2, R, L, R_1, R_2) = \max\Big\{ & H_2\left(\frac{P_2}{R+L}\right) \cdot (R+L) - R_2, \\ & H_2\left(\frac{P_2}{R+L}\right) \cdot 2(R+L) - 2R_2 + R_2 - R_1 \Big\} \\ = \max\Big\{ & S^*_{BJMM-ISD(2)}(P_2, R, L, R_2), \\ & 2S^*_{BJMM-ISD(2)}(P_2, R, L, R_2) - (R_1 - R_2) \Big\} \end{aligned} \tag{8.74}$$

Εν συνεχεία, υπολογίζεται η πολυπλοκότητα που προκύπτει με βάση το μέγεθος των λιστών $\mathcal{L}_1^{(1)}$, $\mathcal{L}_2^{(1)}$ η οποία συμβολίζεται με $S^*_{BJMM-ISD(1)}$ για το επίπεδο-1 ως εξής

$$|\mathcal{L}_i^{(1)}| = \binom{k+\ell}{p_1} \cdot 2^{-r_1}, \quad \forall i = 1, 2. \tag{8.75}$$

Με βάση τον ορισμό 2.32 προκύπτει

$$|\mathcal{L}_i^{(1)}| = 2^{H_2\left(\frac{p_1}{k+\ell}\right) \cdot (k+\ell) - r_1}.$$

Ορίζοντας την $|\mathcal{M}_i^{(1)}| = \frac{1}{n} \log_2 |\mathcal{L}_i^{(1)}|$, έχουμε

$$|\mathcal{M}_i^{(1)}| = H_2\left(\frac{\frac{p_1}{n}}{\frac{k+\ell}{n}}\right) \cdot \frac{k+\ell}{n} - \frac{r_1}{n} \tag{8.76}$$

και υπολογίζοντας στη συνέχεια το όριο της (8.76)

$$S^*_{BJMM-ISD(1)}(P_1, R, L, R_1) = H_2\left(\frac{P_1}{R+L}\right) \cdot (R+L) - R_1. \tag{8.77}$$

Κατόπιν, υπολογίζεται η πολυπλοκότητα που απαιτείται για τον έλεγχο των συγκρούσεων, η οποία συμβολίζεται με $\Lambda_{BJMM-ISD(1)}$ ως εξής:

$$|\mathcal{L}_1^{(1)}| \cdot |\mathcal{L}_2^{(1)}| \cdot 2^{r_1 - \ell} = \binom{k+\ell}{p_1}^2 \cdot 2^{-r_1 - \ell}. \tag{8.78}$$

Με βάση τον ορισμό 2.32 προκύπτει

$$|\mathcal{L}_1^{(1)}| \cdot |\mathcal{L}_2^{(1)}| \cdot 2^{r_1 - \ell} = 2^{H_2\left(\frac{p_1}{k+\ell}\right) \cdot 2(k+\ell) - r_1 - \ell} \tag{8.79}$$

τελικά υπολογίζοντας το όριο της (8.79), έχουμε

$$\Lambda_{BJMM-ISD(1)}(P_1, R, L, R_1) = H_2\left(\frac{P_1}{R+L}\right)\cdot 2(R+L) - R_1 - L. \tag{8.80}$$

Επομένως, το κόστος που απαιτείται σε κάθε επανάληψη του αλγορίθμου στο επίπεδο-1 είναι

$$\begin{aligned} C^*_{BJMM-ISD(1)}(P_2, R, L, R_1) &= \max\left\{H_2\left(\frac{P_1}{R+L}\right)\cdot(R+L) - R_1,\right. \\ &\qquad \left. H_2\left(\frac{P_1}{R+L}\right)\cdot 2(R+L) - 2R_1 + R_1 - L\right\} \\ &= \max\left\{S^*_{BJMM-ISD(1)}(P_1, R, L, R_1),\right. \\ &\qquad \left. 2S^*_{BJMM-ISD(1)}(P_1, R, L, L_1) - (L - R_1)\right\}. \end{aligned} \tag{8.81}$$

Επιπλέον, ο αριθμός των επαναλήψεων που απαιτούνται προκειμένου ο αλγόριθμος BJMM-ISD να συγκλίνει στην επιθυμητή λύση, συμβολίζεται με $N_{BJMM-ISD}$ και υπολογίζεται ως εξής

$$N_{BJMM-ISD} = \mathsf{Pr}^{-1}_{success} = \left(\frac{\binom{k+\ell}{p}\binom{n-k-\ell}{\omega-p}}{\binom{n}{\omega}}\right)^{-1}. \tag{8.82}$$

Με βάση τον ορισμό 2.32 ο ασυμπτωτικός εκθέτης του $N_{BJMM-ISD}$ υπολογίζεται, ως εξής

$$N_{BJMM-ISD} \simeq 2^{-H_2\left(\frac{p}{k+\ell}\right)\cdot(k+\ell) - H_2\left(\frac{\omega-p}{n-k-\ell}\right)\cdot(n-k-\ell) + H_2\left(\frac{\omega}{n}\right)\cdot n}.$$

Ορίζοντας την $N^*_{BJMM-ISD} = \frac{1}{n}\log_2 N_{BJMM-ISD}$

$$\begin{aligned} N^*_{BJMM-ISD} &= -H_2\left(\frac{p}{k+\ell}\right)\cdot\frac{k+\ell}{n} - H_2\left(\frac{\omega-p}{n-k-\ell}\right)\cdot\left(1-\frac{k}{n}-\frac{\ell}{n}\right) + H_2\left(\frac{\omega}{n}\right) = \\ &= -H_2\left(\frac{\frac{p}{n}}{\frac{k+\ell}{n}}\right)\cdot\frac{k+\ell}{n} - H_2\left(\frac{\frac{\omega}{n}-\frac{p}{n}}{1-\frac{k}{n}-\frac{\ell}{n}}\right)\cdot\left(1-\frac{k}{n}-\frac{\ell}{n}\right) + H_2\left(\frac{\omega}{n}\right) \end{aligned}$$

και υπολογίζοντας στη συνέχεια το όριο της $N^*_{BJMM-ISD}$, προκύπτει ότι

$$N^*_{BJMM-ISD}(R, W, P, L) = H_2(W) - H_2\left(\frac{P}{R+L}\right)\cdot(R+L) - H_2\left(\frac{W-P}{1-R-L}\right)\cdot(1-R-L). \tag{8.83}$$

Κατά συνέπεια, οι συνολικοί συντελεστές χωρικής και χρονικής πολυπλοκότητας υπολογίζονται ως εξής

$$\begin{aligned} S^*_{BJMM-ISD}(P, R, L, R_1, R_2) = \max\Big\{&S^*_{BMMJ_ISD(3)}(R, P_2, L), S^*_{BJMM-ISD(2)}(P_2, R, L, R_2) \\ &S^*_{BJMM-ISD(2)}(P_2, R, L, R_2), S^*_{BJMM-ISD(1)}(P_1, R, L, R_1)\Big\} \end{aligned} \tag{8.84}$$

$$\begin{aligned} T^*_{BJMM-ISD}(R, W) = \min_{P,L,R_1,R_2}\Big\{&N^*_{BJMM-ISD}(R, W, P, L) + \max\Big\{C^*_{BJMM-ISD(1)}(P_2, R, L, R_1), \\ &C^*_{BJMM-ISD(2)}(P_2, R, L, R_1, R_2),\ C^*_{BJMM-ISD(3)}(P_2, R, L, R_2)\Big\}\Big\} \end{aligned} \tag{8.85}$$

8.1.10 Ασυμπτωτική ανάλυση του αλγορίθμου JL-ISD

Η πιθανότητα επιτυχίας του JL-ISD δίνεται από τη σχέση

$$\mathsf{Pr}_{success} = \frac{\binom{k}{p}\binom{n-k-z-\ell}{\omega-p}}{\binom{n}{\omega}}. \tag{8.86}$$

Με βάση τον ορισμό 2.32 ο ασυμπτωτικός εκθέτης της $\mathsf{Pr}_{success}$ υπολογίζεται, ως εξής

$$\mathsf{Pr}_{success} \simeq 2^{H_2\left(\frac{p}{k}\right)\cdot k + H_2\left(\frac{\omega-p}{n-k-z-\ell}\right)\cdot(n-k-z-\ell) - H_2\left(\frac{\omega}{n}\right)\cdot n}.$$

Ορίζοντας την $\mathsf{Pr}^*_{success} = \frac{1}{n}\log_2 \mathsf{Pr}_{success}$ όπως στην (8.1)

$$\begin{aligned}\mathsf{Pr}^*_{success} &= H_2\left(\frac{p}{k}\right)\cdot\frac{k}{n} + H_2\left(\frac{\omega-p}{n-k-z-\ell}\right)\cdot\left(1-\frac{k}{n}-\frac{z}{n}-\frac{\ell}{n}\right) - H_2\left(\frac{\omega}{n}\right)\\ &= H_2\left(\frac{\frac{p}{n}}{\frac{k}{n}}\right)\cdot\frac{k}{n} + H_2\left(\frac{\frac{\omega}{n}-\frac{p}{n}}{1-\frac{k}{n}-\frac{z}{n}-\frac{\ell}{n}}\right)\cdot\left(1-\frac{k}{n}-\frac{z}{n}-\frac{\ell}{n}\right) - H_2\left(\frac{\omega}{n}\right)\end{aligned}$$

και υπολογίζοντας στη συνέχεια το όριο της $\mathsf{Pr}^*_{success}$, προκύπτει ο ασυμπτωτικός εκθέτης της πιθανότητας επιτυχίας $B^*(R,W,P,L,Z)$, ως ακολούθως

$$B^*(R,W,P,L,Z) = H_2\left(\frac{P}{R}\right)\cdot R + H_2\left(\frac{W-P}{1-R-Z-L}\right)\cdot(1-R-Z-L) - H_2(W). \tag{8.87}$$

8.1.11 Ασυμπτωτική ανάλυση του αλγορίθμου PSSD

Μια επανάληψη του αλγορίθμου είναι επιτυχής αν το διάνυσμα σφάλματος $\boldsymbol{e}$ έχει βάρος Hamming an στις πρώτες βn συντεταγμένες και $(W-a)n$ στις τελευταίες $(1-\beta)n$ συντεταγμένες. Άρα, ο αριθμός των επαναλήψεων που απαιτούνται προκειμένου ο αλγόριθμος PSSD να συγκλίνει στην επιθυμητή λύση, συμβολίζεται με N_{PSSD} και προκύπτει ως εξής

$$N_{PSSD} = \mathsf{Pr}^{-1}_{success} = \left(\frac{\binom{\beta n}{an}\binom{(1-\beta)n}{\omega-an}}{\binom{n}{\omega}}\right)^{-1}. \tag{8.88}$$

Με βάση τον ορισμό 2.32 ο ασυμπτωτικός εκθέτης του N_{PSSD} υπολογίζεται, ως εξής

$$N_{PSSD}(a,\beta) \simeq 2^{-H_2\left(\frac{a}{\beta}\right)\cdot\beta n - H_2\left(\frac{\frac{\omega}{n}-a}{1-\beta}\right)\cdot(1-\beta)n + H_2\left(\frac{\omega}{n}\right)\cdot n}. \tag{8.89}$$

Ορίζοντας την $N^*_{PSSD} = \frac{1}{n}\log_2 N_{PSSD}$

$$N^*_{PSSD}(a,\beta) = -H_2\left(\frac{a}{\beta}\right)\cdot\beta - H_2\left(\frac{\frac{\omega}{n}-a}{1-\beta}\right)\cdot(1-\beta) + H_2\left(\frac{\omega}{n}\right)$$

και υπολογίζοντας στη συνέχεια το όριο της N^*_{PSSD}, προκύπτει ότι

$$N^*_{PSSD}(a,\beta) = H_2(W) - H_2\left(\frac{a}{\beta}\right)\cdot\beta - H_2\left(\frac{W-a}{1-\beta}\right)\cdot(1-\beta). \tag{8.90}$$

Στη συνέχεια, υπολογίζεται η πολυπλοκότητα που προκύπτει με βάση το μέγεθος των λιστών $\mathcal{L}_1$ και $\mathcal{L}_2$, η οποία συμβολίζεται με S^*_{PSSD} και εκφράζει τον ασυμπτωτικό συντελεστή χωρικής πολυπλοκότητας ως εξής

$$|\mathcal{L}_i| = \binom{\frac{\beta}{2}n}{\frac{a}{2}n}, \quad \forall i = 1, 2 \tag{8.91}$$

όπου $|\mathcal{L}_i|$ εκφράζει την πληθικότητα της λίστας $\mathcal{L}_i$. Με βάση τον ορισμό 2.32, προκύπτει

$$|\mathcal{L}_i| = 2^{H_2\left(\frac{\frac{a}{2}n}{\frac{\beta}{2}n}\right)\cdot\frac{\beta}{2}n}$$

Ορίζοντας την $|\mathcal{M}_i| = \frac{1}{n}\log_2|\mathcal{L}_i|$, έχουμε

$$|\mathcal{M}_i| = H_2\left(\frac{a}{\beta}\right)\cdot\frac{\beta}{2} \tag{8.92}$$

και υπολογίζοντας στη συνέχεια το όριο της (8.92)

$$S^*_{PSSD}(a,\beta) = H_2\left(\frac{a}{\beta}\right)\cdot\frac{\beta}{2}. \tag{8.93}$$

Κατόπιν, υπολογίζεται η πολυπλοκότητα που απαιτείται για τον έλεγχο των συγκρούσεων ως εξής:

$$|\mathcal{L}_1|\cdot|\mathcal{L}_2|\cdot 2^{-(\beta-R)n} = \binom{\frac{\beta}{2}n}{\frac{a}{2}n}^2\cdot 2^{-(\beta-R)n}. \tag{8.94}$$

Καθώς, η πολυπλοκότητα μιας επανάληψης είναι η μέγιστη μεταξύ του αριθμού των διανυσμάτων μήκους $\frac{\beta}{2}n$ και βάρους $\frac{a}{2}n$ και του αριθμού των λύσεων της εξίσωσης $\boldsymbol{H}'\boldsymbol{e}' = \boldsymbol{s}'$. Εφόσον, ο κώδικας C' έχει ρυθμό πληροφορίας R/β και το σύνδρομο $\boldsymbol{s}'$ έχει μήκος $(1-\frac{R}{\beta})\beta n$, συνεπώς ο αριθμός των λύσεων εκτιμάται σύμφωνα με την (8.94) και με βάση τον ορισμό 2.32 και τον συμβολισμό Landau, προκύπτει

$$|\mathcal{L}_1|\cdot|\mathcal{L}_2|\cdot 2^{-(\beta-R)n} = 2^{H_2\left(\frac{an}{\beta n}\right)\cdot\beta n-(\beta-R)n} \tag{8.95}$$

τελικά υπολογίζοντας το όριο της (8.95), έχουμε

$$\lim_{n\to\infty}\frac{1}{n}\log(|\mathcal{L}_1|\cdot|\mathcal{L}_2|\cdot 2^{-(\beta-R)n}) = H_2\left(\frac{a}{\beta}\right)\cdot\beta-(\beta-R). \tag{8.96}$$

Επομένως, το κόστος που απαιτείται σε κάθε επανάληψη του αλγορίθμου είναι

$$\begin{aligned} C^*_{PSSD}(R,a,\beta) &= \max\left\{H_2\left(\frac{a}{\beta}\right)\cdot\frac{\beta}{2}, H_2\left(\frac{a}{\beta}\right)\cdot\beta-(\beta-R)\right\} \\ &= \max\left\{S^*_{PSSD}, 2S^*_{PSSD}-(\beta-R)\right\} \end{aligned} \tag{8.97}$$

συνεπώς, ο συνολικός συντελεστής πολυπλοκότητας υπολογίζεται ως εξής

$$T^*_{PSSD}(R,a,\beta) = \min\left\{N^*_{PSSD}(a,\beta) + C^*_{PSSD}(R,a,\beta)\right\}. \tag{8.98}$$

Ειδικότερα, για $W = D_{GV}(R)$ λαμβάνοντας υπόψη τους φυσικούς περιορισμούς $R < \beta < 1$ και $\max\{0, D_{GV}(R) + \beta - 1\} \le a \le \min\{D_{GV}(R), \beta\}$ και συνδυάζοντας τις (8.97) και (8.98), εξετάζουμε αρχικά την περίπτωση κατά την οποία το μέγιστο είναι $2S^*_{PSSD} - (\beta - R)$, συνεπώς

$$\begin{aligned} T^*_{PSSD}(R) &= \min_{a,\beta}\left\{H_2(W) - H_2\left(\frac{a}{\beta}\right)\cdot\beta - H_2\left(\frac{W-a}{1-\beta}\right)\cdot(1-\beta) + H_2\left(\frac{a}{\beta}\right)\cdot 2\cdot\frac{\beta}{2} - (\beta - R)\right\} \\ &= \min_{a,\beta}\left\{1 - R - (1-\beta)\cdot H_2\left(\frac{D_{GV}(R)-a}{1-\beta}\right) - \beta + R\right\} \\ &= \min_{a,\beta}\left\{(1-\beta)\cdot\left(1 - H_2\left(\frac{D_{GV}(R)-a}{1-\beta}\right)\right)\right\}. \end{aligned} \tag{8.99}$$

Εν συνεχεία, εξετάζουμε την περίπτωση κατά την οποία το μέγιστο είναι S^*_{PSSD}

$$\begin{aligned} T^*_{PSSD}(R) &= \min_{a,\beta}\left\{H_2(W) - H_2\left(\frac{a}{\beta}\right)\cdot\beta - H_2\left(\frac{W-a}{1-\beta}\right)\cdot(1-\beta) + H_2\left(\frac{a}{\beta}\right)\cdot\frac{\beta}{2}\right\} \\ &= \min_{a,\beta}\left\{H_2(W) - H_2\left(\frac{W-a}{1-\beta}\right)\cdot(1-\beta) - H_2\left(\frac{a}{\beta}\right)\cdot\frac{\beta}{2}\right\} \\ &= \min_{a,\beta}\left\{1 - R - H_2\left(\frac{a}{\beta}\right)\cdot\frac{\beta}{2} - H_2\left(\frac{D_{GV}(R)-a}{1-\beta}\right)\cdot(1-\beta)\right\}. \end{aligned} \tag{8.100}$$

Συνδυάζοντας τις (8.99) και (8.100) τελικά προκύπτει

$$\begin{aligned} T^*_{PSSD}(R) = \min_{a,\beta}\Bigg\{&(1-\beta)\left[1 - H_2\left(\frac{D_{GV}(R)-a}{1-\beta}\right)\right], \\ &1 - R - \frac{\beta}{2}\cdot H_2\left(\frac{a}{\beta}\right) - (1-\beta)\cdot H_2\left(\frac{D_{GV}(R)-a}{1-\beta}\right)\Bigg\}. \end{aligned} \tag{8.101}$$

Ασυμπτωτική συσχέτιση μεταξύ των αλγορίθμων PSSD και FS-ISD

Εν συνεχεία, με βάση το Θεώρημα 8.3 που ακολουθεί, θα αποδείξουμε ότι ο αλγόριθμος FS-ISD είναι ασυμπτωτικά ισοδύναμος με τον αλγόριθμο PSSD.

Θεώρημα 8.3. *Έστω (P,L) ένα σύνολο παραμέτρων για τον αλγόριθμο FS-ISD και $(P,R+L)$ ένα σύνολο παραμέτρων για τον αλγόριθμο PSSD, από όπου προκύπτει $T^*_{FS-ISD}(P,L) = T^*_{PSSD}(P,R+L)$.*

Απόδειξη. Έστω (P,L) ένα έγκυρο σύνολο παραμέτρων για τον αλγόριθμο FS-ISD, δηλαδή $0 \le L \le 1 - R$ και $\max\{0, R + L + W - 1\} \le P \le \min\{R + L, W\}$. Τότε $(a,\beta) = (P, R+L)$ είναι ένα έγκυρο σύνολο παραμέτρων για τον αλγόριθμο PSSD, δηλαδή $\beta = R + L \ge R$ και $\beta = R + L \le 1$ και $\max\{0, \beta + W - 1\} \le a \le \min\{\beta, W\}$.

Διακρίνουμε αρχικά, την περίπτωση κατά την οποία το μέγιστο στην (8.97) είναι $2S^*_{PSSD} - (\beta - R)$, δηλαδή $(\beta - R) \leq S^*_{PSSD}$ και επιπλέον $L \leq S^*_{FS-ISD}$ δηλαδή το μέγιστο στην (6.37) είναι $2S^*_{FS-ISD} - L$. Συνεπώς

$$\begin{aligned} T^*_{FS-ISD}(P,L) &= N^*_{FS-ISD}(P,L) + 2S^*_{FS-ISD}(P,L) - L \\ &= H_2(W) - (R+L)\cdot H_2\left(\frac{P}{R+L}\right) - (1-R-L)\cdot H_2\left(\frac{W-P}{1-R-L}\right) \\ &\quad + (R+L)\cdot H_2\left(\frac{P}{R+L}\right) - L \\ &= H_2(W) - (1-(R+L))\cdot H_2\left(\frac{W-P}{1-(R+L)}\right) - L. \end{aligned} \tag{8.102}$$

Αντικαθιστώντας στην (8.102) όπου $R+L=\beta$ και $P=a$, ισοδύναμα έχουμε

$$\begin{aligned} T^*_{FS-ISD}(P,L) &= N^*_{FS-ISD}(P,L) + 2S^*_{FS-ISD}(P,L) - L \\ &= H_2(W) - (1-\beta)\cdot H_2\left(\frac{W-a}{(1-\beta)}\right) - (\beta - R) \\ &\overset{(8.90),(8.93)}{=} N^*_{PSSD}(P,R+L) + 2S^*_{PSSD}(P,R+L) - (\beta - R) \\ &= T^*_{PSSD}(P,R+L) \end{aligned} \tag{8.103}$$

καθώς $N^*_{FS-ISD}(P,L) = N^*_{PSSD}(P,R+L)$ και $S^*_{FS-ISD}(P,L) = S^*_{PSSD}(P,R+L)$. Εν συνεχεία, εξετάζουμε την περίπτωση κατά την οποία $(\beta - R) \geq S^*_{PSSD}$, υποθέτοντας ότι $L \geq S^*_{FS-ISD}$, δηλαδή το μέγιστο με βάση την (6.37) είναι S^*_{FS-ISD}.

$$\begin{aligned} T^*_{FS-ISD}(P,L) &= N^*_{FS-ISD}(P,L) + S^*_{FS-ISD}(P,L) \\ &= H_2(W) - (1-(R+L))\cdot H_2\left(\frac{W-P}{1-(R+L)}\right) - \frac{R+L}{2}\cdot H_2\left(\frac{P}{R+L}\right) \end{aligned} \tag{8.104}$$

ομοίως, αντικαθιστώντας στην (8.104) όπου $R+L=\beta$ και $P=a$, παίρνουμε

$$\begin{aligned} T^*_{FS-ISD}(P,L) &= N^*_{FS-ISD}(P,L) + S^*_{FS-ISD}(P,L) \\ &= H_2(W) - (1-\beta)\cdot H_2\left(\frac{W-a}{1-\beta}\right) - \frac{\beta}{2}\cdot H_2\left(\frac{P}{R+L}\right) \\ &\overset{(8.90),(8.93)}{=} N^*_{PSSD}(P,R+L) + S^*_{PSSD}(P,R+L) \\ &= T^*_{PSSD}(P,R+L). \end{aligned} \tag{8.105}$$

Τέλος, θεωρούμε την περίπτωση κατά την οποία $(\beta - R) \geq S^*_{PSSD}$, υποθέτοντας ωστόσο ότι $L \leq S^*_{FS-ISD}$, δηλαδή το μέγιστο με βάση την (6.37) είναι $2S^*_{FS-ISD} - L$. Συνεπώς, αντικαθιστώντας στην (8.105) όπου $R+L=\beta$ και $P=a$, ισοδύναμα

έχουμε

$$\begin{aligned} T^*_{FS-ISD}(P,L) &= N^*_{FS-ISD}(P,L) + 2S^*_{FS-ISD}(P,L) - L \\ &= H_2(W) - (1-\beta)\cdot H_2\left(\frac{W-a}{(1-\beta)}\right) - (\beta - R) \\ &< H_2(W) - (1-\beta)\cdot H_2\left(\frac{W-a}{(1-\beta)}\right) - S^*_{PSSD} \\ &\overset{(8.93)}{=} H_2(W) - (1-\beta)\cdot H_2\left(\frac{W-a}{(1-\beta)}\right) - \frac{\beta}{2}\cdot H_2\left(\frac{a}{\beta}\right) \\ &\overset{(8.90)}{=} N^*_{PSSD}(P,R+L) + S^*_{PSSD}(P,R+L) \\ &= T^*_{PSSD}(P,R+L). \end{aligned} \tag{8.106}$$

Τελικά, σε όλες τις περιπτώσεις $T^*_{FS-ISD}(P,L) = T^*_{PSSD}(P,R+L)$. ■

Παρατήρηση 8.4. Συνδυάζοντας το θεώρημα 8.7 με το θεώρημα 8.3, προκύπτει ότι ο αλγόριθμος PSSD είναι ασυμπτωτικά τουλάχιστον, τόσο αποδοτικός όσο ο αλγόριθμος BCD.

8.2 Συγκριτικά αποτελέσματα

8.2.1 Συντελεστές πολυπλοκότητας

Ακολουθεί συγκριτική ανάλυση των ISD αλγορίθμων για $0 < R < 1$ και $W = D_{GV}$, επιλέγοντας τις βέλτιστες παραμέτρους ανά αλγόριθμο. Η επιλογή γίνεται κατόπιν εξαντλητικής αναζήτησης, λαμβάνοντας υπόψη τους φυσικούς περιορισμούς που ικανοποιούν οι εμπλεκόμενες, ανά εξεταζόμενη μέθοδο, παράμετροι. Ειδικότερα, για τους αλγορίθμους των Prange και Lee-Brickell, θα πρέπει να ικανοποιείται ο περιορισμός $0 \leq P \leq W$. Ο αλγόριθμος του Leon θα πρέπει να ικανοποιεί τους ακόλουθους περιορισμούς

- $0 \leq P \leq W$
- $0 \leq L \leq 1 - R$

ενώ οι αλγόριθμοι Stern και FS-ISD, θα πρέπει να ικανοποιούν τους φυσικούς περιορισμούς

- $0 \leq L \leq 1 - R$
- $\max\{0, R + L + W - 1\} \leq P \leq \min\{W, R\}$.

Στον αλγόριθμο BCD, καθώς υπεισέρχεται και η παράμετρος Q, οι φυσικοί περιορισμοί που θα πρέπει να ικανοποιούνται είναι

- $0 \leq L \leq 1 - R$

- $0 \leq Q \leq \min\{L, W\}$
- $\max\{0, R + L + W - Q - 1\} \leq P \leq \min\{W - Q, R\}$.

Αντιστοίχως, για τον αλγόριθμο MMT-ISD, οι φυσικοί περιορισμοί που θα πρέπει να ικανοποιούνται είναι

- $0 \leq P \leq W$
- $0 \leq L_1 + L_2 \leq 1 - R - W + P$
- $0 \leq L_2 \leq P$
- $0 \leq L_1$

ενώ, οι παράμετροι που εμπλέκονται στον αλγόριθμο BJMM-ISD, θα πρέπει να ικανοποιούν τους περιορισμούς

- $0 \leq L \leq \min\{1 - R, 1 - R - W - P\}$
- $0 \leq P \leq \min\{W, R + L\}$
- $0 \leq E_1 \leq R + L - P$
- $0 \leq E_2 \leq R + L - P_1$
- $0 \leq R_2 \leq R_1 \leq L$.

Εν συνεχεία, παρατίθενται τα αποτελέσματα των συντελεστών χρονικής και χωρικής πολυπλοκότητας ανά εξεταζόμενη μέθοδο, όπως απεικονίζεται στους αντίστοιχους Πίνακες 8.3-8.10, καθώς επίσης και οι συγκεντρωτικοί Πίνακες 8.11-8.13 των εν λόγω αποτελεσμάτων.

Πίνακας 8.3: Συντελεστές πολυπλοκότητας για τον αλγόριθμο του Prange.

R	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
T^*	.0584	.0913	.1105	.1195	.1199	.1125	.0977	.0753	.0443
S^*	$o(1)$	$o(1)$	$o(1)$	$o(1)$	$o(1)$	$o(1)$	$o(1)$	$o(1)$	$o(1)$
$T^* + S^*$	.0584	.0913	.1105	.1195	.1199	.1125	.0977	.0753	.0443

Πίνακας 8.4: Συντελεστές πολυπλοκότητας για τον αλγόριθμο των Lee-Brickell.

R	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
T^*	.0593	.0925	.1120	.1212	.1217	.1145	.0999	.0777	.0471
S^*	.0081	.0091	.0097	.0101	.0104	.0107	.0109	.0111	.0113
$T^* + S^*$	.0674	.1016	.1217	.1313	.1321	.1252	.1108	.0888	.0584

Πίνακας 8.5: Συντελεστές πολυπλοκότητας για τον αλγόριθμο του Leon.

R	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
T^*	.0599	.0931	.1124	.1216	.1221	.1148	.1002	.0780	.0473
S^*	.0081	.0091	.0097	.0101	.0104	.0107	.0109	.0111	.0113
$T^* + S^*$	.0680	.1022	.1221	.1317	.1325	.1255	.1111	.0891	.0586

Πίνακας 8.6: Συντελεστές πολυπλοκότητας για τον αλγόριθμο του Stern.

R	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
T^*	.0575	.0893	.1075	.1157	.1155	.1077	.0928	.0707	.0406
S^*	.0114	.0194	.0257	.0301	.0332	.0346	.0341	.0301	.0223
$T^* + S^*$	.0689	.1087	.1332	.1458	.1487	.1423	.1269	.1008	.0629

Πίνακας 8.7: Συντελεστές πολυπλοκότητας για τον αλγόριθμο BCD.

R	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
T^*	.0573	.0890	.1072	.1154	.1152	.1075	.0926	.0705	.0405
S^*	.0144	.0232	.0289	.0323	.0377	.0402	.0365	.0318	.0239
$T^* + S^*$	.0717	.1122	.1361	.1477	.1529	.1477	.1291	.1023	.0644

Πίνακας 8.8: Συντελεστές πολυπλοκότητας για τον αλγόριθμο FS-ISD.

R	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
T^*	.0573	.0890	.1072	.1154	.1152	.1074	.0925	.0705	.0405
S^*	.0134	.0214	.0298	.0338	.0376	.0377	.0382	.0312	.0232
$T^* + S^*$	.0707	.1104	.1370	.1492	.1528	.1451	.1307	.1017	.0637

Πίνακας 8.9: Συντελεστές πολυπλοκότητας για τον αλγόριθμο MMT-ISD.

R	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
T^*	.0561	.0865	.1036	.1109	.1102	.1022	.0876	.0662	.0377
S^*	.0220	.0351	.0410	.0479	.0547	.0530	.0549	.0446	.0293
$T^* + S^*$	.0781	.1216	.1446	.1588	.1649	.1552	.1425	.1108	.0670

Πίνακας 8.10: Συντελεστές πολυπλοκότητας για τον αλγόριθμο BJMM-ISD.

R	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
T^*	.0532	.0808	.0958	.1017	.1000	.0920	.0779	.0583	.0328
S^*	.0345	.0570	.0690	.0767	.0786	.0767	.0665	.0524	.0306
$T^* + S^*$	.0877	.1378	.1648	.1784	.1786	.1687	.1444	.1107	.0634

Πίνακας 8.11: Συγκεντρωτικός πίνακας χρονικής πολυπλοκότητας T^*.

R	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Prange	.0584	.0913	.1105	.1195	.1199	.1125	.0977	.0753	.0443
Lee-Brickell	.0593	.0925	.1120	.1212	.1217	.1145	.0999	.0777	.0471
Leon	.0599	.0931	.1124	.1216	.1221	.1148	.1002	.0780	.0473
Stern	.0575	.0893	.1075	.1157	.1155	.1077	.0928	.0707	.0406
BCD	.0573	.0890	.1072	.1154	.1152	.1075	.0926	.0705	.0405
FS-ISD	.0573	.0890	.1072	.1154	.1152	.1074	.0925	.0705	.0405
MMT-ISD	.0561	.0865	.1036	.1109	.1102	.1022	.0876	.0662	.0377
BJMM-ISD	.0532	.0808	.0958	.1017	.1000	.0920	.0779	.0583	.0328

Πίνακας 8.12: Συγκεντρωτικός πίνακας χωρικής πολυπλοκότητας S^*.

R	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Prange	$o(1)$	$o(1)$	$o(1)$	$o(1)$	$o(1)$	$o(1)$	$o(1)$	$o(1)$	$o(1)$
Lee-Brickell	.0081	.0091	.0097	.0101	.0104	.0107	.0109	.0111	.0113
Leon	.0081	.0091	.0097	.0101	.0104	.0107	.0109	.0111	.0113
Stern	.0114	.0194	.0257	.0301	.0332	.0346	.0341	.0301	.0223
BCD	.0144	.0232	.0289	.0323	.0377	.0402	.0365	.0318	.0239
FS-ISD	.0134	.0214	.0298	.0338	.0376	.0377	.0382	.0312	.0232
MMT-ISD	.0220	.0351	.0410	.0479	.0547	.0530	.0549	.0446	.0293
BJMM-ISD	.0345	.0570	.0690	.0767	.0786	.0767	.0665	.0524	.0306

Πίνακας 8.13: Συγκεντρωτικός πίνακας συνολικής πολυπλοκότητας $T^* + S^*$.

R	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Prange	.0584	.0913	.1105	.1195	.1199	.1125	.0977	.0753	.0443
Lee-Brickell	.0674	.1016	.1217	.1313	.1321	.1252	.1108	.0888	.0584
Leon	.0680	.1022	.1221	.1317	.1325	.1255	.1111	.0891	.0586
Stern	.0689	.1087	.1332	.1458	.1487	.1423	.1269	.1008	.0629
BCD	.0717	.1122	.1361	.1477	.1529	.1477	.1291	.1023	.0644
FS-ISD	.0707	.1104	.1370	.1492	.1528	.1451	.1307	.1017	.0637
MMT-ISD	.0781	.1216	.1446	.1588	.1649	.1552	.1425	.1108	.0670
BJMM-ISD	.0877	.1378	.1648	.1784	.1786	.1687	.1444	.1107	.0634

8.2.2 Επιτεύξιμη ασφάλεια

Ακολουθούν λεπτομερείς εκτιμήσεις ασφαλείας για τους εξεταζόμενους ISD αλγορίθμους. Στον Πίνακα 8.14 ενδεικτικά, απεικονίζονται έγκυρα σύνολα παραμέτρων δυαδικών ανάγωγων Goppa κωδίκων [54]. Κατόπιν ανά αλγόριθμο, παρατίθενται τα επίπεδα ασφαλείας που μπορούν να επιτευχθούν για μήκος κώδικα $n = 1024$ και $n = 2048$, όπως απεικονίζεται στους Πίνακες 8.15 και 8.16, αντίστοιχα. Τέλος, υπολογίζεται ανά εξεταζόμενη μέθοδο το απαιτούμενο μήκος κώδικα, ώστε να επιτυγχάνονται επίπεδα ασφαλείας τουλάχιστον 128 Bit και 256 Bit (βλ. Πίνακες 8.17 και 8.18, αντίστοιχα).

Πίνακας 8.14: Έγκυρα σύνολα παραμέτρων Goppa κωδίκων.

Μήκος κώδικα n	**Διάσταση κώδικα** k	**Διορθωτική ικανότητα** ω
1024	524	50
1632	1269	34
2048	1608	40
2960	2288	57
4096	3604	41
4624	3389	95
6624	5129	117

Πίνακας 8.15: Επιτεύξιμη ασφάλεια των ISD αλγορίθμων, για $0 \leq R \leq 1$, $W = D_{GV}$ και $n = 1024$.

R	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Prange	59.80	93.49	113.15	122.36	122.77	115.20	100.04	77.10	45.36
Lee-Brickell	60.72	94.72	114.69	124.11	124.62	117.24	102.30	79.56	48.23
Leon	61.33	95.33	115.10	124.52	125.03	117.55	102.60	79.87	48.43
Stern	58.88	91.44	110.08	118.48	118.27	110.28	95.03	72.40	41.57
BCD	58.67	91.14	109.77	118.17	117.96	110.08	94.82	72.19	41.47
FS-ISD	58.67	91.14	109.77	118.17	117.96	109.98	94.72	72.19	41.47
MMT-ISD	57.45	88.57	106.09	113.56	112.85	104.65	89.70	67.79	38.60
BJMM-ISD	54.48	82.74	98.10	104.14	102.40	94.21	79.77	59.70	33.59

Πίνακας 8.16: Επιτεύξιμη ασφάλεια των ISD αλγορίθμων, για $0 \leq R \leq 1$, $W = D_{GV}$ και $n = 2048$. Η γκρίζα σκίαση αντιστοιχεί σε επίπεδα ασφαλείας τουλάχιστον 128 Bit.

R	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Prange	119.60	186.98	226.30	244.74	245.55	230.40	200.08	154.21	90.72
Lee-Brickell	121.45	189.44	229.38	248.22	249.24	234.50	204.59	159.13	96.46
Leon	122.62	190.67	230.19	249.04	250.06	235.11	205.21	159.74	96.87
Stern	117.76	182.88	220.16	236.95	236.54	220.57	190.05	144.79	83.15
BCD	117.35	182.27	219.55	236.34	235.93	220.16	189.64	144.38	82.94
FS-ISD	117.35	182.27	219.55	236.34	235.93	219.95	189.44	144.38	82.94
MMT-ISD	114.89	177.15	212.17	227.12	225.69	209.30	179.40	135.58	77.21
BJMM-ISD	108.95	165.48	196.20	208.28	204.80	188.42	159.54	119.40	67.17

Πίνακας 8.17: Το μήκος του κώδικα που απαιτείται ανά αλγόριθμο για $0 \leq R \leq 1$, προκειμένου να επιτευχθούν επίπεδα ασφαλείας τουλάχιστον 128 Bit.

R	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Prange	2192	1402	1159	1072	1068	1138	1311	1700	2890
Lee-Brickell	2159	1384	1143	1057	1052	1118	1282	1648	2718
Leon	2137	1375	1389	1053	1049	1115	1278	1641	2707
Stern	2227	1434	1191	1107	1109	1189	1380	1811	3153
BCD	2234	1439	1194	1110	1112	1191	1383	1816	3161
FS-ISD	2234	1439	1194	1110	1112	1192	1384	1816	3161
MMT-ISD	2282	1480	1236	1155	1162	1253	1462	1934	3396
BJMM-ISD	2407	1585	1337	1259	1280	1392	1644	2196	3903

Πίνακας 8.18: Το μήκος του κώδικα που απαιτείται ανά αλγόριθμο για $0 \leq R \leq 1$, προκειμένου να επιτευχθούν επίπεδα ασφαλείας τουλάχιστον 256 Bit.

R	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Prange	4384	2804	2317	2143	2136	2276	2621	3400	5779
Lee-Brickell	4318	2768	2286	2113	2104	2236	2563	3295	5436
Leon	4274	2750	2278	2106	2097	2231	2556	3283	5413
Stern	4453	2867	2382	2213	2217	2378	2759	3621	6306
BCD	4468	2877	2389	2219	2223	2382	2765	3632	6322
FS-ISD	4468	2877	2389	2219	2223	2384	2768	3632	6322
MMT-ISD	4564	2960	2472	2309	2324	2505	2923	3868	6791
BJMM-ISD	4813	3169	2673	2518	2560	2783	3287	4392	7805

ΚΕΦΑΛΑΙΟ 9

Συμπεράσματα

Ύστερα από εκτεταμένη ανασκόπηση της βιβλιογραφίας μελετήσαμε πλήθος ISD αλγορίθμων, καθώς και την ασυμπτωτική τους ανάλυση. Διαπιστώσαμε ωστόσο, ότι το ζήτημα της χωρικής πολυπλοκότητας δεν μελετάται εκτενώς, καθώς ο κύριος σκοπός των ερευνητών, είναι ο περιορισμός του χρόνου εκτέλεσης στο μέγιστο δυνατό βαθμό. Άρα, ο σχεδιασμός χωρικά-αποδοτικών αλγορίθμων παραμένει ανοικτό ζήτημα για περαιτέρω έρευνα.

Ερώτημα ωστόσο τίθεται και για την πιθανή βελτιστοποίηση του BJMM-ISD αλγορίθμου, με την περαιτέρω προσθήκη επιπρόσθετων αναδρομικών επιπέδων, που επεκτείνεται μάλιστα σε ευρύτερες εφαρμογές της τεχνικής αναπαράστασης, όπως το *πρόβλημα αθροίσματος υποσυνόλων* (subset-sum problem) [61]. Ερώτημα που απαιτεί πλήρως εναλλακτική μοντελοποίηση και ανάλυση για την επίλυσή του. Σύμφωνα πάντως με τους συγγραφείς [65], και βάσει αριθμητικών πειραμάτων, ο αριθμός των ήδη υπαρχόντων επιπέδων (δηλ. 2-3) προσδίδει τα καλύτερα δυνατά αποτελέσματα.

Επιπλέον, όπως παρατηρήσαμε οι αλγόριθμοι ISD επιτρέπουν την επίλυση του προβλήματος LPN πιο αποδοτικά, από τον (ασυμπτωτικά) ταχύτερο αλγόριθμο BKW (βλ. Σχ. 9.1), στην περίπτωση που η πιθανότητα σφάλματος είναι σχετικά μικρή, δηλ. μικρότερη από $\frac{1}{8}$. Συνεπώς, κάθε σοβαρή πρόταση παραμέτρων για κρυπτογραφικές κατασκευές που βασίζεται στην δυσκολία του προβλήματος LPN, θα πρέπει να λαμβάνει σοβαρά υπόψη τους ISD αλγορίθμους. Τα παραπάνω, εγείρουν το ερώτημα αν μπορούν να προκύψουν πιο αποδοτικές επιθέσεις, συνδυάζοντας τον αλγόριθμο BKW με ιδέες προερχόμενες από την αποκωδικοποίηση συνόλου πληροφορίας,

Συνοψίζοντας, να επισημάνουμε ότι το κυριότερο τμήμα της μελέτης μας, η επίλυση του CSD προβλήματος, σχετίζεται με έναν από τους ισχυρότερους ερευνητικούς τομείς στην μετα-κβαντική κρυπτογραφία, καθώς σε συνδυασμό με τα συγγενή προβλήματα LPN και LWE, δεν επιδέχεται σημαντικής κβαντικής επιτάχυνσης, ακόμα και υπό την χρήση ισχυρών τεχνικών, όπως η κβαντική δειγματοληψία Fourier [39]. Συνεπώς, είναι παραδεκτό στην κρυπτογραφική κοινότητα ότι απαιτείται η πραγματοποίηση περαιτέρω έρευνας προς την κατασκευή αποδοτικών κρυπτοσυστημάτων και πρωτοκόλλων, βασιζόμενων στα

k \ η	$\frac{1}{100}$	$\frac{1}{20}$	$\frac{1}{8}$	$\frac{1}{4}$
128	BJMM 16	BJMM 26	BKW 36	BKW 42
256	BCD 21	BCD 38	BKW 55	BKW 63
512	BCD 27	BJMM 38	BKW 87	BKW 99
768	BJMM 34	BJMM 72	BKW 124	BKW 142
1024	BJMM 37	BJMM 89	BKW 143	BKW 161

Σχήμα 9.1: Αλγόριθμοι BCD, BJMM-ISD, BKW και η πολυπλοκότητά τους $\log t$ (βλ. Ορισμό 7.15). Για κάθε ζεύγος παραμέτρων (k, η) αναπαρίσταται η πιο αποδοτική επίθεση, με καταγεγραμμένο το πραγματικό επίπεδο ασφαλείας. Η γκρίζα σκίαση αντιστοιχεί σε ανασφαλή ζεύγη παραμέτρων (δηλ. $\log t \leq 80$).

προβλήματα CSD, LPN, LWE και ισοδύναμα αυτών. Με κύριο στόχο, την υλοποίηση νέων κατασκευών, με αποδοτικότερους τρόπους αποκωδικοποίησης για την ταχύτερη μεταγωγή των δεδομένων, μικρότερα μήκη κλειδιών και μεγαλύτερα επίπεδα ασφαλείας.

Βιβλιογραφία

[1] E. Prange, *The use of information sets in decoding cyclic codes*, IRE Trans., vol. IT-8, pp. S5-S9, 1962.

[2] T. Kasami, *A decoding procedure for multiple-error-correcting cyclic codes*, IEEE Trans. Information Theory, vol. IT-IO, pp. 134-138, 1964.

[3] W. Feller, *An Introduction to Probability Theory and Its Applications*, New York: Wiley, vol. 1, pp. 50 -53. 1968.

[4] L. D. Baumert, R. J. McEliece, and G. Solomon, *Decoding with multipliers*, Jet Propulsion Laboratory Deep Space Network Progress Report, 42-34, pp. 42-46, 1976.

[5] F. MacWilliams and N. Sloane, *The Theory of Error Correcting Codes*. North–Holland, Amsterdam, 1977.

[6] McEliece R. J, *The Theory of Information and Coding*, (Vol. 3 of The Encyclopedia of Mathematics and Its Applications.), Reading Mass., Addison - Wesley 1977.

[7] Rivest, R.L., Shamir, A., and Adleman, L.M., *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*, Communications of the ACM, 21, (2), pp.120-126, 1978.

[8] E. R. Berlekamp, McEliece R. J., Henk C. A Van Tilbork *On the Inherent Intractability of Certain Coding Problems*, IEEE Transactions on information theory, vol. IT-24, no.3, May 1978.

[9] McEliece, R.J., *A Public-Key Cryptosystem Based on Algebraic Coding Theory*, DSN Progress Report, 42-44, pp.114-116, 1978.

[10] Rabin, M.O., *Digital Signatures and Public-Key Functions as Intractable as Factorization*, MIT Lab. For Computer Science, Technical Report, MIT/LCS/TR-212, Jan 1979.

[11] Tai-Yang Hwang, *Decoding Linear Block Codes for Minimizing Word Error Rate*, IEEE Transactions on Information Theory, vol. IT-25, no. 6, November 1979.

[12] N. G. de Bruijn, *Asymptotic Methods in Analysis*, New York: Dover, pp. 3-10, 1981.

[13] A. H. Chan and R. A. Games, *(n, k,t)-covering systems and error-trapping decoding*, IEEE Trans. Inform. Theory, vol. IT-27, pp. 643-646, 1981.

[14] Shu Lin and Daniel J. Costello, *Error Control Coding, Fundamentals and Applications*, Prentice-Hall, Inc. Englewood Cliffs, New Jersey, 1983.

[15] Blum, M., and Goldwasser, S., An efficient Probabilistic Public-Key Encryption Scheme Which Hides All Partial Information, Advances in Cryptology-CRYPTO '84, Springer-Verlag, pp. 289-299, 1985.

[16] Benny Chor, Ronald L. Rivest, *A Knapsack type Public Key Cryptosystem Based On Arithmetic in Finite Fields*, CRYPTO '84, LNCS 196, pp. 54-65, 1985.

[17] Lev. B. Levitin, Carlos Hartmann, *A New Approach to the General Minimum Distance Decoding Problem: The Zero-Neighbors Algorithm*, IEEE Transactions on Information Theory, vol. IT-31, no. 3, May 1985.

[18] A. G. Hamilton, *A FIRST COURSE in Linear Algebra*, Cambridge University Press, pp.1-10, 1987.

[19] P. J. Lee, E. F. Brickell, *An Observation on the Security of McEliece's Public-Key Cryptosystem*, In: Gunther, C.G. (ed.) EUROCRYPT 1988. LNCS, VOL.330, PP.275-280. Springer, Heidelberg, 1988.

[20] J. S. Leon, *A Probabilistic Algorithm for Computing Minimum Weights of Large Error-Correcting Codes*, IEEE Trans. Information Theory, vol. 34, no. 5, 1988.

[21] I. Dumer, *Two decoding algorithms for linear codes*, Problems of Information Transmission, 1989.

[22] J. Stern, *A method of finding codewords of small weight*, In Lecture Notes in Computer Science, Coding Theory & Applications, vol. 388, pp. 106-113, Springer, 1989.

[23] Carlisle M. Adams, Henk Melter, *Security-Related Comments Regarding McEliece's Public-Key Cryptosystem*, IEEE Transactions on Information Theory, vol. 35, No. 2, March 1989.

[24] J. T. Coffey, R.M.Goodman, *The complexity of Information Set Decoding*, IEEE Transactions on Information Theory, 1990.

[25] Valery I. Korzhik, Andrey I. Turkin, *Cryptanalysis of McEliece Public-Key Cryptosystem*, USSR, 1991.

[26] I. Dumer, *On minimum distance decoding of linear codes*, Joint Soviet-Swedish International Workshop Information Theory, 1991.

[27] Stephane G. Mallat, Zhifeng Zhang,*Matching Pursuits with time-frequency dictionaries*, IEEE Transactions on signal processing, vol. 41, no. 12, December 1993.

[28] Mitsuru Matsui, *Linear Cryptanalysis Method for DES Cipher*, Advances in Cryptology, Lecture Notes in Computer Science vol. 765, 1994, pp 386-397, EUROCRYPT '93, 1993.

[29] F. Chabaud, *On the Security of Some Cryptosystems Based on Error-correcting Codes* , Advances in Cryptology, Lecture Notes in Computer Science vol. 950, 1995, pp 131-139, EUROCRYPT'94, 1994.

[30] Yuan Xing Li, Robert H. Deng, Xin Mei Wang, *On the Equivalence of McEliece's and Niederreiter's Public-Key Cryptosystems*, IEEE Transactions on Information Theory, vol. 40, pp. 271-273, 1994.

[31] Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.

[32] Thomas A. Berson, *Failure of the McEliece Public-Key Cryptosystem Under Message-Resend and Related-Message Attack*, USA. at Crypt 1997.

[33] Shor P., *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*, SIAM Journal of Computing, vol. 26(5), 1997.

[34] Hung-Min Sun, *Improving the Security of the McEliece Public-Key Cryptosystem*, Taiwan, ASIACRYPT '98, LNCS 1514, pp 200-213, 1998.

[35] D. Knuth, *Art of Computer Programming: Sorting and Searching*, 2nd edition, vol.3, Addison-Wesley Professional, 1998.

[36] A. Barg, *Complexity Issues in Coding Theory*, Chapter 7 in [PHB98]

[37] V. Pless, W.C, Huffman and R.Brualdi, *Handbook of Coding Theory*, Elsevier Science, 1998.

[38] M. Mosca, A. Ekert, *The Hidden Subgroup Problem and Eigenvalue Estimation on a Quantum Computer*, Quantum Computing and Quantum Communications, Lecture Notes in Computer Science, vol. 1509, pp 174-188, 1999.

[39] L. Hales, S. Hallgren, *Quantum Fourier Sampling Simplified*, Proceeding STOC '99 Proceedings of the thirty-first annual ACM symposium on Theory of Computing, pp 330-338, 1999.

[40] A. Barg, E. Krouk, Henk C. A. van Tilborg, *On the complexity of Minimum Distance Decoding of Long Linear Codes*, IEEE Transactions on Information Theory, vol. 45, No. 5, July 1999.

[41] C. Meyer, *Matrix Analysis and Applied Linear Algebra*, SIAM Pubs, 2000.

[42] D. P. Bertsekas, J. N. Tsitsiklis, *Introduction to Probability*, Athena Scientific, 2000.

[43] A. Kh. Al Jabri, *A Statistical Decoding Algorithm for General Linear Block Codes*, Cryptography and Coding, 2001.

[44] A. Blum, A. Kalai and H. Wasserman, *Noise-tolerant Learning, the Parity Problem, and the Statistical Query Model*, in Journal of the ACM, vol. 50, 2003.

[45] R. Overbeck, *A New Structural Attack for GPT and variants*, In Poc. of MyCrypt 2005, vol. 3715 of LNCS, pages 50-63. Springer Verlag, 2005.

[46] Jorg Rothe, *Complexity Theory and Cryptology*, Texts in Theoretical Computer Science. An EATCS Series, Springer, 2005.

[47] D. Engelbert, R. Overbeck and A. Schmidt, *A Summary of McEliece-Type Cryptosystems and their Security*, pages 8-14, May 2006.

[48] R. Overbeck, *Statistical Decoding Revisited*, Information Security and Privacy, ACISP 2006, 2006.

[49] T. M. Cover, J. A. Thomas *Elements of Information Theory*, 2nd Edition Wiley Series in Telecommunications and Signal Processing, 2006.

[50] D. J. Bernstein, J. Buchmann, E. Dahmen *Post-Quantum Cryptography*, Springer, 2008.

[51] D. Micciancio and O. Regev, *Lattice-based Cryptography*, book chapter in *Post-quantum cryptography*, Springer, 2008.

[52] O. Goldreich, *Computational Complexity: A Conceptual Perspective*, Cambridge University Press, US 2008.

[53] N. Alon, J. Spencer, *The Probabilistic Method*, A John Wiley and Sons, INC. Publication, 3rd edition, 2008.

[54] D. J. Bernstein, T. Lange and C. Peters, *Attacking and Defending the McEliece Cryptosystem*, Post-Quantum Cryptography - PQCrypto 2008, 2008.

[55] M. Finiasz, N. Sendrier, *Security Bounds for the Design of Code-Based Cryptosystems*, In: Matsui, Lncs, vol.5912, pp.88-105, Springer, Heidelberg, ASIACRYPT 2009, 2009.

[56] N. Howgrave-Graham, A.Joux, *New Generic Algorithms for Hard Knapsacks*, Advances in Cryptology - EUROCRYPT 2010, 2010.

[57] Oded Regev, *The Learning with Errors Problem*, Invited survey in CCC 2010, 2010.

[58] H. Dinh, C. Moore and Russell, *McEliece and Niederreiter Cryptosystems That Resist Quantum Fourier Sampling Attacks*, Advances in Cryptography, CRYPTO 2011, 2011.

[59] D. J. Bernstein, T. Lange, and C. Peters, *Ball-collision Decoding*, International Association for Cryptologic Research, LNCS 6841, pp. 743–760, CRYPTO 2011, 2011.

[60] D. J. Bernstein, T. Lange, and C. Peters, *Smaller Decoding Exponents: Ball-collision Decoding*, International Advanced in Cryptology, CRYPTO 2011, 2011.

[61] A. Becker, J.S. Coron and A. Joux, *Improved Generic Algorithms for Hard Knapsacks*, Advances in Cryptology, EUROCRYPT 2011, 2011.

[62] N. Kalouptsidis, N. Kolokotronis, *Fast Decoding of Regular LDPC Codes Using Greedy Approximation Algorithms*, IEEE International Symposium on Information Theory Proceedings, 2011.

[63] A. May, A. Meurer and E. Thomae, *Decoding Random Linear Codes in* $\widetilde{O}(2^{0.054n})$, Advances in Cryptology - ASIACRYPT 2011, 2011.

[64] T. Johansson, C. Löndahl, *An improvement to Stern's algorithm*, Lund University Libriaries, Sweden 2011.

[65] A. Becker, A. Joux, A. May, A. Meurer, *Decoding Random Binary Linear Codes in* $2^{n/20}$*: How* $1 + 1 = 0$ *Improves Information Set Decoding*, Advances in Cryptology - EUROCRYPT 2012, 2012.

[66] Alexander Meurer, *A Coding-Theoretic Approach to Crypanalysis*, PhD Thesis, RUB University, November 2012.