Πανεπιστήμιο Αιγαίου
Σχολή Θετικών Επιστημών
Τμήμα Μαθηματικών, Κατεύθυνση Μαθηματικών

Μια εισαγωγή στη θεωρία κωδίκων Reed-Solomon και εφαρμογή της στον κώδικα QR

Πτυχιακή Εργασία

του

ΧΡΗΣΘΕΝΗ ΘΑΝΗ

Επιβλέπων: Χαράλαμπος Κορνάρος
Επίκουρος Καθηγητής

Καρλόβασι, Απρίλιος 2021

Πανεπιστήμιο Αιγαίου
Σχολή Θετικών Επιστημών
Τμήμα Μαθηματικών, Κατεύθυνση Μαθηματικών

Μια εισαγωγή στη θεωρία κωδίκων Reed-Solomon και εφαρμογή της στον κώδικα QR

Πτυχιακή Εργασία

του

ΧΡΗΣΘΕΝΗ ΘΑΝΗ

Επιβλέπων: Χαράλαμπος Κορνάρος
Επίκουρος Καθηγητής

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 21η Απριλίου 2021.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Χαράλαμπος Κορνάρος	Ανδρέας Παπασαλούρος	Αντώνιος Τσολομύτης
Επίκουρος Καθηγητής	Επίκουρος Καθηγητής	Καθηγητής

Καρλόβασι, Απρίλιος 2021

Πανεπιστήμιο Αιγαίου
Σχολή Θετικών Επιστημών
Τμήμα Μαθηματικών, Κατεύθυνση Μαθηματικών

ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ενυπογράφως ότι είμαι αποκλειστικός συγγραφέας της παρούσας Πτυχιακής Εργασίας, για την ολοκλήρωση της οποίας κάθε βοήθεια είναι πλήρως αναγνωρισμένη και αναφέρεται λεπτομερώς στην εργασία αυτή. Έχω αναφέρει πλήρως και με σαφείς αναφορές, όλες τις πηγές χρήσης δεδομένων, απόψεων, θέσεων και προτάσεων, ιδεών και λεκτικών αναφορών, είτε κατά κυριολεξία είτε βάσει επιστημονικής παράφρασης. Αναλαμβάνω την προσωπική και ατομική ευθύνη ότι σε περίπτωση αποτυχίας στην υλοποίηση των ανωτέρω δηλωθέντων στοιχείων, είμαι υπόλογος έναντι λογοκλοπής, γεγονός που σημαίνει αποτυχία στην Πτυχιακή μου Εργασία και κατά συνέπεια αποτυχία απόκτησης του Τίτλου Σπουδών, πέραν των λοιπών συνεπειών του νόμου περί πνευματικών δικαιωμάτων. Δηλώνω, συνεπώς, ότι αυτή η Πτυχιακή Εργασία προετοιμάστηκε και ολοκληρώθηκε από εμένα προσωπικά και αποκλειστικά και ότι, αναλαμβάνω πλήρως όλες τις συνέπειες του νόμου στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής άλλης πνευματικής ιδιοκτησίας.

(Υπογραφή)

.....
Χ. Θανής

21 Απριλίου 2021

Στους γονείς μου

Ευχαριστίες

Θα ήθελα καταρχήν να ευχαριστήσω τον καθηγητή κ. Κορνάρο για την επίβλεψη αυτής της πτυχιακής εργασίας, την καθοδήγησή του και την εξαιρετική συνεργασία που είχαμε. Τέλος, θα ήθελα να ευχαριστήσω θερμά τους γονείς μου για την ηθική συμπαράσταση που μου προσέφεραν όλα αυτά τα χρόνια.

Περιεχόμενα

Πρόλογος

Στόχος της παρούσας πτυχιακής εργασίας είναι η μελέτη, η ανάλυση της θεωρίας των κωδίκων και η εφαρμογή της στον κώδικα QR, σε τέτοιο βαθμό ώστε να μπορέσει να γίνει όσο το δυνατόν περισσότερο κατανοητή σε κοινό χωρίς πολλές μαθηματικές γνώσεις.

Στο πρώτο κεφάλαιο, αρχικά, αναλύουμε τις θεμελιώδεις έννοιες της θεωρίας των κωδίκων μέσα από αρκετά παραδείγματα, με σκοπό ο αναγνώστης να αντιληφθεί τη σπουδαιότητα και τη χρησιμότητα τους στην καθημερινή μας ζωή. Έπειτα, προς το τέλος του κεφαλαίου, γίνεται μια εισαγωγή στα απαραίτητα μαθηματικά εργαλεία που θα χρειαστούμε, ώστε να εφαρμόσουμε αυτή τη θεωρία στον κώδικα QR.

Στο δεύτερο κεφάλαιο, μελετάμε αναλυτικά τον κώδικα QR. Αρχικά, παρουσιάζουμε κάποιες ιστορικές και γενικές πληροφορίες, ώστε να αντιληφθούμε τον αντίκτυπο που μπορεί να έχει στην καθημερινότητα μας. Αναλύουμε εις βάθος τη δομή και τα χαρακτηριστικά του και τέλος επιχειρούμε να κατασκευάσουμε το δικό μας κώδικα QR ο οποίος θα περιέχει το μήνυμα της ιστοσελίδας του πανεπιστημίου μας, https://www.aegean.gr
.

Κεφάλαιο 1

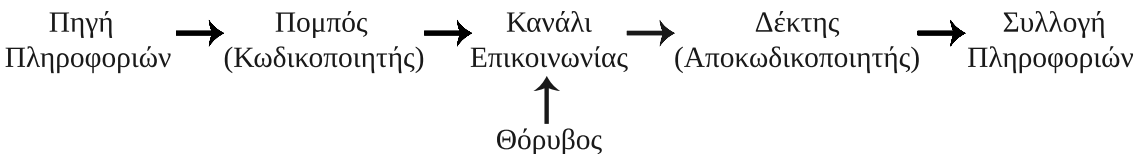
ΕΙΣΑΓΩΓΗ

1.1 Η έννοια της κωδικοποίησης

Η θεωρία κωδικοποίησης (coding theory), ή αλγεβρική θεωρία κωδίκων (algebraic theory of codes) ασχολείται με την εύρεση και τη μελέτη μεθόδων, σύμφωνα με τις οποίες καθίσταται δυνατή η αποτελεσματική και ακριβής μεταφορά πληροφοριών από μια τοποθεσία Α σε μία άλλη τοποθεσία Β. Η έννοια της τοποθεσίας είναι κάπως αφηρημένη, καθώς για παράδειγμα μπορεί να πρόκειται είτε για μεταφορά πληροφοριών μεταξύ δύο πόλεων μέσω τηλεφωνικού καλωδίου, είτε για μεταφορά πληροφοριών από τον προσωπικό μας υπολογιστή σε ένα USB stick, είτε για τη μετάδοση δεδομένων από ένα επίγειο πιάτο σε έναν τηλεπικοινωνιακό δορυφόρο.

Το φυσικό μέσο με το οποίο μεταφέρονται αυτές οι πληροφορίες ονομάζεται κανάλι (channel). Οι τηλεφωνικές γραμμές και η ατμόσφαιρα είναι παραδείγματα τέτοιων καναλιών. Η αιτία της δημιουργίας και της ανάπτυξης αυτής της θεωρίας, είναι ο θόρυβος (noise) και η αντιμετώπισή του. Ως θόρυβο ορίζουμε τις ανεπιθύμητες παρεμβολές οι οποίες εμποδίζουν την ορθή μεταφορά πληροφοριών. Συνέπεια αυτών των παρεμβολών είναι η αλλοίωση των πληροφοριών, καθώς είναι πολύ πιθανό οι πληροφορίες που στάλθηκαν από τη τοποθεσία Α να διαφέρουν με αυτές που έλαβε η τοποθεσία Β. Θόρυβος μπορεί να προκληθεί από ηλεκτρομαγνητικές παρεμβολές, κεραυνούς, βροχή μετεωριτών, φτωχή ακοή, συνωστισμός σε ένα κανάλι και πολλά άλλα.

Ειδικότερα, η θεωρία της κωδικοποίησης έχει ως στόχο να αντιμετωπίσει το πρόβλημα της αλλοίωσης των πληροφοριών, ανιχνεύοντας και διορθώνοντας όποια σφάλματα προκύπτουν κατά τη μετάδοση τους σε ένα κανάλι, εξ'αιτίας της ύπαρξης θορύβου. Το παρακάτω διάγραμμα μας δίνει μια γενική εικόνα ενός συστήματος μετάδοσης πληροφοριών.

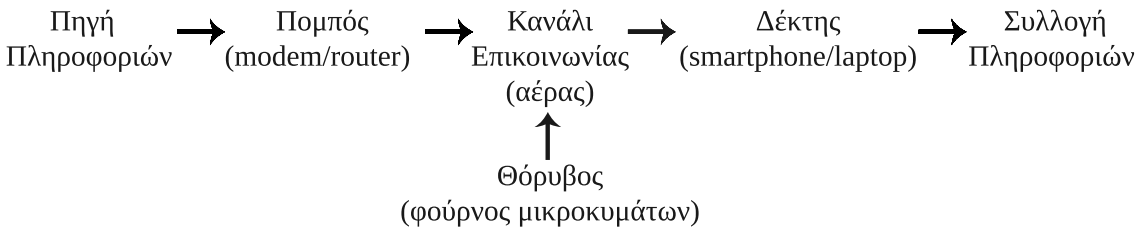


Σχήμα 1.1

Το πιο σημαντικό μέρος του διαγράμματος είναι ο θόρυβος, που εισέρχεται μέσα

στο κανάλι επικοινωνίας. Χωρίς τον θόρυβο, δε θα υπήρχε λόγος για την ανάπτυξη της θεωρίας των κωδίκων. Δυστυχώς τέλειο κανάλι επικοινωνίας δεν υπάρχει. Όσο καλό κανάλι και να επιλέξουμε για τη μετάδοση, όσο και να εφαρμόσουμε κατάλληλα φίλτρα θορύβου, κάποια χρονική στιγμή σίγουρα θα υπάρξουν παρεμβολές που θα οδηγήσουν στην αλλοίωση πληροφοριών.

Ας δούμε, λοιπόν, ένα παράδειγμα τέτοιου θορύβου που είναι πολύ πιθανό να συναντήσουμε στην καθημερινότητά μας. Στην εποχή μας, όπου η τεχνολογία αποτελεί αναπόσπαστο κομμάτι της ζωής μας, σχεδόν κάθε σπίτι πλέον έχει πρόσβαση στο διαδίκτυο (Internet). Συνεπώς, για να επιτύχουμε αυτή την πρόσβαση, απαραίτητος εξοπλισμός αποτελεί μια συσκευή modem/router. Αυτή η συσκευή εκπέμπει συνεχώς ραδιοκύματα σε συχνότητα 2.4Ghz, το οποίο είναι γνωστό σε όλους μας ως Wi-Fi και μπορούμε να συνδεθούμε χρησιμοποιώντας αντίστοιχες συμβατές συσκευές όπως το smartphone και το laptop μας. Όμως, στην ίδια συχνότητα μεταξύ των 2.4Ghz και 2.5Ghz, λειτουργούν και φούρνοι μικροκυμάτων για να ζεστάνουν το φαγητό. Ο λόγος που χρησιμοποιούν το συγκεκριμένο εύρος συχνοτήτων οφείλεται στο γεγονός ότι το νερό, η ζάχαρη και τα λιπαρά απορροφούν τα ραδιοκύματα σε αυτό το μήκος. Αυτό έχει ως αποτέλεσμα, τα απορροφημένα ραδιοκύματα να μετατρέπονται αμέσως σε μόρια άτακτης κίνησης, τα οποία προκαλούν στο φαγητό να θερμανθεί [2]. Το παρακάτω διάγραμμα μας δίνει την εικόνα του παραδείγματός μας, σύμφωνα με το Σχήμα1.1.



Σχήμα 1.2

Στην καθημερινή μας επικοινωνία χρησιμοποιούμε κυρίως λέξεις, είτε γραπτώς είτε προφορικώς, οι οποίες παράγονται από τα γράμματα της αλφαβήτου μας. Τα κωδικοποιούμε με συγκεκριμένο τρόπο ώστε να σχηματίσουμε λέξεις, με τις οποίες έπειτα μπορούμε να μιλήσουμε ή να γράψουμε. Αυτές τις λέξεις τις στέλνουμε μέσω ενός καναλιού, το οποίο είναι ο χώρος μεταξύ του στόματος και του αυτιού του δέκτη (παραλήπτη) ή στην περίπτωση που τις γράφουμε το κανάλι είναι το φως που μεταφέρει την γραπτή πληροφορία στα μάτια του αναγνώστη. Η αποκωδικοποίηση γίνεται διαβάζοντας ή ακούγοντας αυτές τις λέξεις, προσπαθώντας να καταλάβουμε το νόημά τους. Ο θόρυβος μπορεί να προκληθεί από τον οποιοδήποτε λόγο, όπως για παράδειγμα, βιαστική ομιλία, κακή ακουστική του χώρου, ορθογραφικά λάθη, αναγραμματισμός, ηχορύπανση, λανθασμένη πληκτρολόγηση λόγω ελαττωματικού πληκτρολογίου ή και από απροσεξία κατά την ανάγνωση.

Από τη φύση μας όμως, έχουμε μηχανισμούς που προβαίνουν στη διόρθωση αυτών των λαθών. Φανταστείτε ότι εν αγνοία σας, έχετε σταθμεύσει το αυτοκίνητό σας σε μέρος που δεν επιτρέπεται και λαμβάνετε το μήνυμα ”ΠΑΡΑΚΑΛΩ ΤΑΡΤΕ ΤΟ ΟΧΗΜΑ. ΕΝΟΧΗ”. Από τη στιγμή που η γλώσσα μας δεν χρησιμοποιεί όλους τους πιθανούς συν-

δυασμούς γραμμάτων της αλφαβήτου, καταλαβαίνουμε ότι το "ΤΑΡΤΕ" δεν αποτελεί λέξη. Αμέσως, υποσυνείδητα, σκεφτόμαστε ότι το "ΤΑΡΤΕ" θα πρέπει να σχετίζεται με κάποια παραπλήσια λέξη. Οπότε είναι πολύ πιο πιθανό να ήταν "ΠΑΡΤΕ" ή "ΤΑΡΤΑ" από ότι "ΚΑΝΤΕ" ή "ΑΕΡΟΠΛΑΝΟ". Από το περιεχόμενο του μηνύματος, καταλαβαίνουμε ότι είναι πολύ πιο πιθανό η σωστή λέξη να είναι "ΠΑΡΤΕ" αντί "ΤΑΡΤΑ". Επιπλέον, η "ΕΝΟΧΗ" αποτελεί μια σωστή λέξη της γλώσσας μας, όμως πάλι για να συμφωνεί περισσότερο με το περιεχόμενο του μηνύματος και να βγάζει νόημα για εμάς, ωθούμαστε να τη διορθώσουμε σε "ΕΝΟΧΛΕΙ". Όμως, επειδή η λέξη "ΕΝΟΧΛΕΙ" δε στέκεται σαν ορολογία ορθά σύμφωνα με το περιεχόμενο του μηνύματος, θα έπρεπε να τη διορθώσουμε με τη λέξη "ΕΜΠΟΔΙΖΕΙ". Σε αυτή την περίπτωση, να σημειώσουμε ότι το σφάλμα οφείλεται στην πηγή, δηλαδή στο άτομο που το σκέφτηκε και μας το έγραψε και όχι στο θόρυβο που υπάρχει στο κανάλι, όπως για παράδειγμα να το διαβάσαμε εμείς λάθος ή να σβήστηκε μέρος του μηνύματος.

Από τα παραπάνω είδη σφαλμάτων, εμείς θα ασχοληθούμε μόνο με το πρώτο, το οποίο είναι να επιλέγουμε την πιθανότερη λέξη που μπορεί να μεταδόθηκε. Η συνηθισμένη μέθοδος που χρησιμοποιείται για την αντιμετώπιση τέτοιων σφαλμάτων, είναι μέσω του πλεονασμού. Πολλές επιχειρήσεις, όπως για παράδειγμα οι τράπεζες, προθέτουν κάποια επιπλέον ψηφία στους αριθμούς των τραπεζικών λογαριασμών (αυτά ονομάζονται ψηφία ελέγχου), ώστε να είναι εύκολος ο έλεγχος της ορθότητας των αριθμών αυτών στις συναλλαγές. Πιθανόν, αυτή να είναι και η πιο γνωστή και απλή μέθοδος κωδικοποίησης στην πράξη.

1.2 Κάποιες βασικές υποθέσεις

Σε αυτή την ενότητα, θα δούμε κάποιους βασικούς ορισμούς και υποθέσεις, που θα χρησιμοποιήσουμε στις επόμενες ενότητες και κεφάλαια.

Σε πάρα πολλές περιπτώσεις, στον ψηφιακό κόσμο, οι πληροφορίες που πρέπει να σταλθούν, μεταδίδονται από μια αλληλουχία αριθμών μηδέν και ένα. Το μηδέν (0) και το ένα (1) ονομάζονται *ψηφία* (bit). Μία λέξη είναι μια αλληλουχία τέτοιων ψηφίων. Το μήκος μίας λέξης είναι το πλήθος των ψηφίων της. Για παράδειγμα, η αλληλουχία ψηφίων 0110101 είναι μία λέξη με μήκος επτά (7). Μια λέξη μεταδίδεται στέλνοντας τα ψηφία της το ένα μετά το άλλο, μέσω ενός δυαδικού καναλιού. Ο όρος δυαδικός, αναφέρεται στο γεγονός ότι χρησιμοποιούμε μόνο μια δυο ψηφία, το 0 και 1 αποκλειστικά. Προς αποφυγή παρερμηνεύσεων της δυαδικής πληροφορίας από τον λήπτη, στέλνουμε κάθε φορά ένα μοναδικό ψηφίο. Για παράδειγμα, για να μεταδώσουμε τη λέξη 0110101, στέλνουμε κάθε φορά ένα ψηφίο τη φορά, δηλαδή $0 \to 1 \to 1 \to 0 \to 1 \to 0 \to 1$ και όχι $01 \to 10 \to 10 \to 1$ ή χρησιμοποιώντας οποιοδήποτε άλλο συνδυασμό. Κάθε ψηφίο μεταδίδεται μηχανικά, ηλεκτρικά ή μαγνητικά.

Ένας δυαδικός κώδικας είναι ένα σύνολο C από λέξεις. Ας κατασκευάσουμε ένα τέτοιο σύνολο, που το μήκος κάθε δυαδικής λέξης του είναι 2. Αφού θέλουμε να είναι μήκους δύο, θα επιλέξουμε ακριβώς δύο ψηφία, έστω 01. Οπότε το 01, είναι μία λέξη. Με παρόμοιο τρόπο, προκύπτουν οι λέξεις 00,10,11 που είναι επίσης μήκους δύο. Ο

κώδικας που αποτελείται από όλες τις λέξεις μήκους δύο, δηλαδή όλους τους δυνατούς συνδυασμούς, είναι

$$C = \{00, 10, 01, 11\}.$$

Ένας μπλοκ κώδικας, είναι ένας κώδικας που κάθε μία του λέξη έχει το ίδιο μήκος. Το σύνολο C που μόλις είδαμε είναι ένα μπλοκ κώδικα, ενώ το σύνολο $D = \{000, 10, 01, 11\}$ δεν είναι, καθώς η πρώτη του λέξη έχει μήκος 3 και οι υπόλοιπες μήκος 2. Εμείς θα ασχοληθούμε μόνο με μπλοκ κώδικες. Συνεπώς, τον όρο κώδικα από εδώ και στο εξής θα τον ταυτίζουμε πάντα με ένα δυαδικό μπλοκ κώδικα. Τις λέξεις που ανήκουν σε έναν κώδικα C, θα τις λέμε κωδικολέξεις. Θα συμβολίζουμε το πλήθος των κωδικολέξεων ενός κώδικα C με $|C|$. Στο παράδειγμά μας, με τον κώδικα C, τα 00, 10, 01, 11 είναι οι κωδικολέξεις του και το πλήθος αυτών είναι $|C| = 4$.

Θα χρειαστεί επίσης να κάνουμε κάποιες βασικές υποθέσεις σχετικά με το κανάλι, οι οποίες θα μας βοηθήσουν να κατανοήσουμε με απλούστερο τρόπο τη θεωρία των κωδίκων.

Η πρώτη υπόθεση είναι ότι εάν ο πομπός μεταδώσει μια κωδικολέξη μήκους n που αποτελείται από 0 και 1, ο δέκτης θα λάβει επίσης μια κωδικολέξη ίδιου μήκους n που να αποτελείται από 0 και 1, όχι όμως απαραίτητα την ίδια με αυτή που μεταδόθηκε. Για παράδειγμα, έστω ότι ο πομπός χρησιμοποιεί κωδικολέξεις μήκους 2 και μεταδίδει το μήνυμα 0010. Ο δέκτης θα πρέπει να λάβει πάλι κωδικολέξεις μήκους 2, όπως 1010 ή 0011 και όχι 010 ή 00101.

Η δεύτερη υπόθεση είναι ότι δε θα υπάρχει δυσκολία να αναγνωρίσουμε την αρχή της πρώτης παραληφθείσας λέξης. Έτσι, για παράδειγμα, εάν χρησιμοποιούμε κωδικολέξεις μήκους 3 και λάβουμε το μήνυμα 011011001, γνωρίζουμε ότι οι λέξεις που λάβαμε είναι, με τη συγκεκριμένη σειρά, 011, 011, 001. Αυτή η υπόθεση σημαίνει, στο συγκεκριμένο παράδειγμα, ότι το κανάλι δε μπορεί να παραδώσει το μήνυμα 01101 στον παραλήπτη, καθώς ένα ψηφίο έχει χαθεί, αφού έχουμε δύο κωδικολέξεις μήκους 3 και 2 αντίστοιχα.

Η τρίτη και τελευταία υπόθεση είναι ότι ο θόρυβος διασκορπίζεται και κατανέμεται τυχαία σε όλες τις λέξεις που μεταδίδονται μέσα από το κανάλι αντί να συσσωρεύεται σε συγκεκριμένες λέξεις. Αυτή η μορφή θορύβου ονομάζεται *ριπή* (burst). Αυτό σημαίνει ότι η πιθανότητα οποιουδήποτε ψηφίου να επηρεαστεί από τον θόρυβο κατά την μετάδοση του, είναι η ίδια με την πιθανότητα οποιουδήποτε άλλου ψηφίου. Συνεπώς αν ένα ψηφίο παραληφθεί λανθασμένα αυτό δεν συνεπάγεται ότι και τα γειτονικά του ψηφία θα είναι και αυτά λανθασμένα. Αν και αυτή η υπόθεση δεν είναι αρκετά ρεαλιστική σε κάποιους τύπους θορύβων, όπως οι γρατζουνιές στα CD, είναι σημαντική για να κατανοήσουμε σε πρώτο στάδιο τη θεωρία των κωδίκων.

Ένα δυαδικό κανάλι καλείται *συμμετρικό* (binary symmetric channel ή BSC) εάν τα ψηφία 0 και 1 παραλαμβάνονται ισότιμα. Αυτό σημαίνει ότι η πιθανότητα να λάβουμε ένα ψηφίο ορθά δεν εξαρτάται από το ποιο ψηφίο (0 ή 1) ήταν κατά τη στιγμή της αποστολής του. Η αξιοπιστία ενός δυαδικού συμμετρικού καναλιού αντιπροσωπεύεται από έναν πραγματικό αριθμό p, $0 \leq p \leq 1$, όπου p είναι η πιθανότητα ένα τυχαίο ψηφίο που μεταδίδεται μέσα από το κανάλι να μην έχει επηρεαστεί από το θόρυβο, δηλαδή,

το ψηφίο που στάλθηκε να είναι το ίδιο ακριβώς με αυτό που παραλήφθηκε.

Εάν p είναι η πιθανότητα ότι το ψηφίο που στάλθηκε είναι το ίδιο με αυτό που παραλήφθηκε, τότε ως $1-p$ ορίζουμε την πιθανότητα ότι το ψηφίο που παραλήφθηκε να μην είναι το ίδιο με αυτό που στάλθηκε. Να επισημάνουμε όμως, ότι τις περισσότερες φορές θα είναι αρκετά δύσκολο να εκτιμήσουμε την πραγματική τιμή του p για ένα δεδομένο κανάλι. Ωστόσο, η πραγματική τιμή του p δεν μας επηρεάζει σημαντικά στην ανάπτυξη της θεωρίας.

Θα καλούμε ένα κανάλι πιο αξιόπιστο από ένα άλλο εάν η αξιοπιστία του, δηλαδή ο πραγματικός αριθμός p, είναι μεγαλύτερη από την αξιοπιστία του άλλου καναλιού. Στην περίπτωση που έχουμε $p = 1$, τότε σημαίνει ότι δεν υπάρχει περίπτωση κάποιο ψηφίο που στάλθηκε να αλλοιώθηκε κατά τη μετάδοση. Συνεπώς, ο δέκτης έλαβε το ίδιο ψηφίο με αυτό που μετέδωσε ο πομπός, οπότε πρόκειται για ένα τέλειο κανάλι. Στην περίπτωση που έχουμε $p = 0$, τότε σημαίνει ότι κάθε ένα ψηφίο αλλοιώθηκε κατά τη μετάδοσή του και ο δέκτης έλαβε ακριβώς το αντίθετο ψηφίο από αυτού που έστειλε ο πομπός. Οπότε, ο δέκτης μπορεί να αλλάξει τα 0 και 1 που έλαβε σε 1 και 0 αντίστοιχα, ώστε να πάρει εν τέλει το σωστό μήνυμα. Ένα κανάλι με πιθανότητα $0 < p \leq \frac{1}{2}$ μπορεί να μετατραπεί εύκολα σε ένα κανάλι με πιθανότητα $\frac{1}{2} \leq p < 1$, λόγω της δυαδικής συμμετρίας. Για παράδειγμα, εάν έχουμε ένα κανάλι με $p = \frac{1}{4}$, δηλαδή η πιθανότητα να λάβει ο δέκτης το σωστό ψηφίο που έστειλε ο πομπός είναι 25%, μπορεί ο δέκτης κάθε φορά που λαμβάνει ένα ψηφίο να το ερμηνεύει με το αντίθετό του. Έτσι, πλέον έχουμε ένα κανάλι με $p = \frac{3}{4}$, δηλαδή η πιθανότητα να λάβει ο δέκτης το ψηφίο ορθά, όπως του το έστειλε ο πομπός είναι 75%. Συνεπώς, εμείς θα υποθέτουμε πάντοτε ότι χρησιμοποιούμε ένα δυαδικό συμμετρικό κανάλι με πιθανότητα $\frac{1}{2} < p < 1$. Στην περίπτωση όπου έχουμε $p = \frac{1}{2}$, το κανάλι έχει πολύ θόρυβο και η πληροφορία χάνεται.

1.3 Ανίχνευση και διόρθωση σφαλμάτων

Σε αυτή την ενότητα θα πάρουμε μια πρώτη γεύση για το πως μπορούμε να ανιχνεύσουμε και να διορθώσουμε σφάλματα μέσα από απλά παραδείγματα, ενώ μια πιο επίσημη προσέγγιση θα γίνει στις επόμενες ενότητες.

Έστω ότι λάβαμε μία λέξη η οποία δεν είναι κωδικολέξη. Προφανώς κάποιο είδος αλλοίωσης έχει υποστεί κατά τη διαδικασία της μετάδοσης, οπότε εμείς μπορούμε να *ανιχνεύσουμε* ότι (τουλάχιστον) ένα σφάλμα προέκυψε. Εάν ωστόσο λάβαμε μια κωδικολέξη, τότε ενδεχομένως δεν θα υπέστη κάποιο είδος αλλοίωσης κατά τη μετάδοση της κι έτσι δεν μπορούμε να ανιχνεύσουμε κάποιο σφάλμα.

Η ιδέα της διόρθωσης σφαλμάτων είναι κάπως πιο περίπλοκη. Ας θυμηθούμε στην προηγούμενη ενότητα που διορθώσαμε τη λέξη ”ΤΑΡΤΕ” με ”ΠΑΡΤΕ” και όχι με ”ΤΑΡΤΑ”. Υποσυνείδητα ψάξαμε να βρούμε και να διορθώσουμε τη λέξη σε κάποια καλύτερη, κάνοντας σ’ αυτήν όσο το δυνατόν λιγότερες αλλαγές. Επίσης, να επισημάνουμε ότι η υπόθεση που έχουμε κάνει παλαιότερα ότι κανένα ψηφίο δε χάνεται ή δημιουργείται κατά τη διαδικασία της μετάδοσης, δηλαδή ο πομπός εάν στείλει κωδικολέξεις μήκους

n, θα πρέπει ο δέκτης να λάβει επίσης κωδικολέξεις μήκους n, υποδηλώνει ότι δε μπορούμε να διορθώσουμε τη λέξη "ΤΑΡΤΕ", π.χ., σε "ΑΕΡΟΠΛΑΝΟ".

Παράδειγμα 1.3.1. *Έστω* $C_1 = \{00, 01, 10, 11\}$. *Όπως έχουμε δει και παλιότερα, το σύνολο* C_1 *αποτελείται από όλες τις κωδικολέξεις μήκους* 2. *Οπότε εάν οποιαδήποτε κωδικολέξη μήκους 2 ληφθεί ως μήνυμα, τότε ο* C_1 *δε θα μπορέσει να ανιχνεύσει κάποιο σφάλμα. Επίσης, δε θα χρειαστεί να προβεί σε κάποια διόρθωση, καθώς καμία αλλαγή δεν απαιτείται ώστε να σχηματιστεί κωδικολέξη.*

Παράδειγμα 1.3.2. *Έστω ότι τροποποιούμε τον* C_1 *επαναλαμβάνοντας κάθε κωδικολέξη του τρεις φορές. Τότε προκύπτει ο ακόλουθος νέος κώδικας*

$$C_2 = \{000000, 010101, 101010, 111111\}.$$

Έστω ότι λαμβάνουμε το μήνυμα 110101. *Αφού δεν αποτελεί κωδικολέξη, διότι δεν ταυτίζεται με κάποια από τις κωδικολέξεις του* C_2, *μπορούμε να συμπεράνουμε ότι τουλάχιστον ένα σφάλμα έχει προκύψει. Παρατηρούμε ότι εάν από το μήνυμα που λάβαμε αλλάξουμε το πρώτο ψηφίο, προκύπτει η κωδικολέξη* 010101 *η οποία ανήκει στον* C_2. *Επιπλέον, παρατηρούμε ότι για να σχηματίσουμε οποιαδήποτε άλλη κωδικολέξη που να ανήκει στον* C_2, *θα χρειαστεί να αλλάξουμε παραπάνω από δύο ψηφία. Ως εκ τούτου, υποθέτουμε ότι η κωδικολέξη* 010101 *είναι το πιθανότερο να μεταδόθηκε εξ' αρχής από τον πομπό και για αυτό διορθώνουμε το αλλοιωμένο μήνυμα που λάβαμε* 110101 *σε* 010101. *Να σημειώσουμε εδώ, ότι όταν μια κωδικολέξη προκύπτει με τις λιγότερες αλλαγές ψηφίων, τότε αυτή καλείται κοντινότερη κωδικολέξη. Στο συγκεκριμένο παράδειγμα, η* 010101 *είναι η κοντινότερη κωδικολέξη.*

Παράδειγμα 1.3.3. *Έστω ότι τροποποιούμε ξανά τον* C_1 *προσθέτοντας ένα τρίτο ψηφίο σε κάθε κωδικολέξη, έτσι ώστε το πλήθος των* 1 *σε κάθε κωδικολέξη να είναι ζυγός αριθμός. Τότε προκύπτει ο ακόλουθος κώδικας*

$$C_3 = \{000, 011, 101, 110\}.$$

Το ψηφίο που προσθέσαμε ονομάζεται ψηφίο ελέγχου ισοτιμίας (parity-check bit). Έστω ότι λαμβάνουμε το μήνυμα 010. *Αφού το* 010 *δεν αποτελεί κωδικολέξη, διότι δεν ταυτίζεται με κάποια από τις κωδικολέξεις του* C_3, *μπορούμε να συμπεράνουμε ότι τουλάχιστον ένα σφάλμα έχει προκύψει. Παρατηρούμε ότι εάν αλλάξουμε ένα οποιοδήποτε ψηφίο του μηνύματος* 010, *σχηματίζονται οι ακόλουθες τρεις κωδικολέξεις που ανήκουν στον* C_3, 110, 000 *και* 011. *Αυτές οι τρεις κωδικολέξεις είναι επίσης οι κοντινότερες κωδικολέξεις. Θα δούμε όμως στις επόμενες ενότητες το τρόπο με τον οποίο θα μεταχειριζόμαστε τέτοιου είδους περιπτώσεις. Διαφορετικά χειριζόμαστε τις περιπτώσεις που έχουμε να επιλέξουμε την πιθανότερο μεταδιδοθείσα κωδικολέξη όταν υπάρχει μια μόνο κοντινότερη κωδικολέξη στην παραληφθείσα λέξη, όπως στο 1.3.2, από τις περιπτώσεις που υπάρχουν περισσότερες πάνω από μία κοντινότερες κωδικολέξεις και χρειάζεται επιλέξουμε την πιθανότερη απ' αυτές, όπως στο τρέχων παράδειγμά μας.*

1.4 Δείκτης πληροφορίας

Στο Παράδειγμα 1.3.3, είδαμε ότι η προσθήκη ενός επιπλέον ψηφίου ελέγχου ισοτιμίας στις κωδικολέξεις, ίσως συμβάλλει στην αποτελεσματικότερη ανίχνευση και διόρθωση σφαλμάτων. Η κωδικολέξη όμως τότε, εκτός από το μήνυμα, θα συμπεριλαμβάνει και επιπλέον βοηθητικά ψηφία για καλύτερη αντιμετώπιση των σφαλμάτων. Ωστόσο είναι προφανές ότι όσο μεγαλύτερες είναι οι κωδικολέξεις, τόσο περισσότερος χρόνος απαιτείται για τη μετάδοση του κάθε μηνύματος. Ο *δείκτης πληροφορίας* (information rate) ενός κώδικα είναι ένας αριθμός που μετράει την ποσοστό (το κλάσμα) κάθε μίας κωδικολέξης ως προς το μήνυμα που μεταφέρεται. Ορίζουμε ως δείκτη πληροφορίας ενός (δυαδικού) κώδικα C μήκους n την ποσότητα

$$\frac{1}{n}\log_2|C|. \tag{1.1}$$

Μπορούμε να θεωρήσουμε ότι $1 \leq |C| \leq 2^n$. Αυτό διότι το ελάχιστο πλήθος των κωδικολέξεων ενός κώδικα C είναι μία κωδικολέξη, όπως για παράδειγμα ο $C = \{01\}$ έχει $|C| = 1$. Το μέγιστο πλήθος των κωδικολέξεων εξαρτάται από το μήκος n κάθε κωδικολέξης και είναι 2^n διότι μας δίνει όλους τους δυνατούς συνδυασμούς των ψηφίων 0 και 1. Για παράδειγμα, ο ακόλουθος κώδικας

$$C = \{000, 001, 011, 111, 110, 100, 101, 010\}$$

αποτελείται από $|C| = 8 = 2^3$ κωδικολέξεις μήκους 3 η κάθε μία. Συνεπώς, η σχέση (1.1) παίρνει τη τιμή 0 όταν $|C| = 1$, δηλαδή

$$\frac{1}{1}log_2|1| = log_2|1| = 0$$

και τη τιμή 1 όταν $|C| = 2^n$, (αυτό συμβαίνει όταν κάθε λέξη μήκους n αποτελεί μία κωδικολέξη), δηλαδή

$$\frac{1}{n}log_2|2^n| = \frac{n}{n}log_2 2 = log_2 2 = 1.$$

Άρα, το εύρος τιμών του δείκτη πληροφορίας ενός κώδικα C μήκους n είναι

$$0 \leq \frac{1}{n}\log_2|C| \leq 1.$$

Στο Παράδειγμα 1.3.1 ο δείκτης πληροφορίας του κώδικα C_1 ισούται με 1, καθώς έχουμε $n = 2$ και $|C| = 4$, δηλαδή

$$\frac{1}{2}log_2 4 = \frac{1}{2}log_2 2^2 = \frac{2}{2}log_2 2 = 1.$$

Στο Παράδειγμα 1.3.2 ο δείκτης πληροφορίας του κώδικα C_2 ισούται με $\frac{1}{3}$, καθώς έχουμε $n = 6$ και $|C| = 4$, δηλαδή

$$\frac{1}{6}log_2 4 = \frac{1}{6}log_2 2^2 = \frac{2}{6}log_2 2 = \frac{1}{3}.$$

Στο Παράδειγμα 1.3.3 ο δείκτης πληροφορίας του κώδικα C_3 ισούται με $\frac{2}{3}$, καθώς έχουμε $n = 3$ και $|C| = 4$, δηλαδή

$$\frac{1}{3}log_2 4 = \frac{1}{3}log_2 2^2 = \frac{2}{3}log_2 2 = \frac{2}{3}.$$

Κάθε ένας από τους παραπάνω δείκτες πληροφορίας φαίνεται να σχετίζεται με τους αντίστοιχους κώδικες. Τα 2 πρώτα ψηφία από τα 6 της κάθε κωδικολέξης του C_2 μπορούν να θεωρηθούν ότι αποτελούν το μεταδιδόμενο μήνυμα, διότι είναι το ελάχιστο μήκος της κάθε κωδικολέξης έτσι ώστε να είναι διαφορετικές μεταξύ τους. Παρομοίως, ισχύει και για τα 2 πρώτα ψηφία από τα 3 της κάθε κωδικολέξης του C_3. Ας θυμηθούμε ότι ο C_3 προέκυψε από τον C_2 προσθέτοντας το ψηφίο ελέγχου ισοτιμίας σε κάθε κωδικολέξη του. Άρα τα 2 πρώτα ψηφία της κάθε κωδικολέξης του C_3 στην ουσία είναι οι κωδικολέξεις του C_2. Οπότε στην συγκεκριμένη περίπτωση είμαστε σίγουροι ότι αυτά τα ψηφία αποτελούν το μεταδιδόμενο μήνυμα (χωρίς τα επιπλέον πλεονάζοντα ψηφία).

1.5 Ανίχνευση και διόρθωση σφαλμάτων

Σε αυτή την ενότητα θα δείξουμε πόσο σημαντική είναι η χρήση του ψηφίου ελέγχου ισοτιμίας, το οποίο μας βοηθάει στην ανίχνευση σφαλμάτων κατά τη μετάδοση ενός μηνύματος.

Έστω ότι όλες οι $2^{11} = 2048$ λέξεις (μήκους 11) αποτελούν κωδικολέξεις στον κώδικά μας. Αυτό σημαίνει ότι κάθε συνδυασμός των ψηφίων 0 και 1 με μήκος 11 μας δίνει μία κωδικολέξη, οπότε δεν πρόκειται να ανιχνευτεί κάποια αλλοίωση (σφάλμα) λόγω θορύβου κατά τη παραλαβή της οποιασδήποτε λέξης. Έστω ότι η αξιοπιστία του καναλιού είναι $p = 1 - 10^{-8} = 0,99999999$ και έστω ότι τα ψηφία μεταδίδονται με ρυθμό $10^7 = 10.000.000$ ψηφία το δευτερόλεπτο, δηλαδή μεταδίδονται $\frac{10^7}{11} \approx 909.091$ λέξεις το δευτερόπλεπτο. Εάν υποθέσουμε ότι κάθε φορά που μεταδίδουμε 11 ψηφία τα 10 από αυτά μεταδίδονται σωστά, τότε επειδή έχουμε επίσης ότι $p \in [0, 1]$, με τη βοήθεια του διωνυμικού αναπτύγματος υπολογίζουμε ότι περίπου

$$11p^{10}(1 - p) \approx \frac{11}{10^8}$$

είναι η πιθανότητα μία λέξη να έχει αλλοιωθεί κατά την μετάδοση. Λαμβάνοντας υπ'όψιν και τον ρυθμό μετάδοσης συμπεραίνουμε ότι περίπου

$$\frac{11}{10^8} \cdot \frac{10^7}{11} = 0,1 \text{ λέξεις το δευτερόλεπτο}$$

μεταδίδονται αλλοιωμένες χωρίς να ανιχνευτούν. Ισοδύναμα, είναι μια λέξη κάθε 10 δευτερόλεπτα ή 360 την ώρα, το οποίο είναι αρκετά απογοητευτικό!

Έστω τώρα ότι προστίθεται ένα ψηφίου ελέγχου ισοτιμίας σε κάθε μία κωδικολέξη έτσι ώστε το πλήθος των ψηφίων 1 να είναι ζυγό σε κάθε μία από τις 2048 κωδικολέξεις. Οπότε το μήκος κάθε κωδικολέξης από 11 γίνεται 12. Τότε κάθε λέξη με ένα μόνο εσφαλμένο ψηφίο μπορεί πάντα να ανιχνευτεί. Αν όμως έχουν αλλοιωθεί δύο ή

παραπάνω ψηφία της τότε δεν μπορούμε να ισχυριστούμε ότι μπορούμε να την ανιχνεύσουμε! Η πιθανότητα να αλλοιωθούν δύο ψηφία με την αξιοπιστία του καναλιού p να είναι η ίδια, προσεγγίζεται με τη βοήθεια του διωνυμικού αναπτύγματος

$$\binom{12}{2} p^{10}(1-p)^2 \approx \frac{66}{10^{16}}.$$

Οπότε λαμβάνοντας υπ'όψιν τον ίδιο ρυθμό μετάδοσης από πριν, καταλήγουμε ότι περίπου

$$\frac{66}{10^{16}} \cdot \frac{10^7}{12} = \frac{55}{10^8} = 0,00000055$$

λέξεις το δευτερόλεπτο μεταδίδονται αλλοιωμένες χωρίς να ανιχνευτούν. Ισοδύναμα, κάθε 2.000 μέρες αλλοιώνεται μία λέξη!

Συνεπώς, εάν είμαστε πρόθυμοι να μειώσουμε το ρυθμό μετάδοσης των ψηφίων από $\frac{10^7}{11} \approx 909.091$ σε $\frac{10^7}{12} \approx 833.833$ λέξεις το δευτερόλεπτο, αυξάνοντας το μήκος της κάθε κωδικολέξης από 11 σε 12, είναι πολύ πιθανότερο να είμαστε σε θέση να ανιχνεύσουμε τις εσφαλμένες λέξεις. Για να αποφασίσουμε σε ποια ψηφία έχει υπάρξει αλλοίωση, θα χρειαστεί να ζητήσουμε να μεταδοθεί ξανά το μήνυμα. Τεχνικά αυτό σημαίνει ότι είτε η μετάδοση των υπόλοιπων λέξεων από τον πομπό θα πρέπει να μπει σε αναμονή έως ότου ο δέκτης στείλει επιβεβαιωτικό μήνυμα (ότι έλαβε την λέξη ορθά) είτε ότι όλα τα μηνύματα πρέπει να αποθηκεύονται σε κάποια μνήμη πριν αποσταλούν, έστω και προσωρινά, σε περίπτωση που ζητηθεί σε δεύτερο χρόνο η αναμετάδοση κάποιων απ' αυτών. Και οι δύο περιπτώσεις ίσως να μας κοστίσουν ακριβά σε χρόνο ή σε αποθηκευτικό χώρο. Ίσως επίσης το να μεταδοθεί ξανά το μήνυμα να μην είναι τόσο πρακτικό, όπως στην περίπτωση της μετάδοσης δεδομένων μεταξύ ενός επίγειου πιάτου και ενός τηλεπικοινωνιακού δορυφόρου. Αναλυτικότερα, ένας τηλεπικοινωνιακός δορυφόρος βρίσκεται στην γεωστατική τροχιά, έτσι ώστε ο χρόνος περιστροφής του γύρω από τη Γη να ισούται με 24 ώρες. Αυτό σημαίνει ότι ένας παρατηρητής από τη Γη θα βλέπει το δορυφόρο σε σταθερό σημείο στον ουρανό. Το ελάχιστο ύψος που είναι από τον ισημερινό έως τον δορυφόρο υπολογίζεται σε περίπου 36.000 χλμ.[3] Η ταχύτητα του φωτός είναι περίπου 300.000 χλμ το δευτερόλεπτο. Οπότε υπό τέλειες συνθήκες ο ελάχιστος χρόνος για να ταξιδέψει το μήνυμα από το επίγειο πιάτο προς το δορυφόρο και να επιστρέψει ξανά στο επίγειο πιάτο (χρόνος μετάδοσης μετ' επιστροφής ή RTT) ισούται με

$$2 \cdot \frac{36.000}{300.000} = 0,24 \text{ δευτερόλεπτα ή } 240\text{ms}.$$

Ζητώντας την επαναμετάδοση του μηνύματος, θα χρειαστούμε το λιγότερο επιπλέον 240ms, δηλαδή σύνολο 480ms ή ισοδύναμα περίπου μισό δευτερόλεπτο, γεγονός που καθιστά αδύνατη την εύρυθμη λειτουργία εφαρμογών πραγματικού χρόνου, όπως το online gaming και τη ζωντανή μετάδοση (live streaming). Εν κατακλείδι, αυξάνοντας το μήκος της κωδικολέξης εντός του κώδικα, αυξάνουμε τις δυνατότητες ανίχνευσης και διόρθωσης σφαλμάτων από τον κώδικα! Εισάγοντας τέτοιες δυνατότητες ίσως να δυσκολέψουν την κωδικοποίηση και την αποκωδικοποίηση, όμως θα μας βοηθήσουν

στην εξοικονόμηση χρόνου και αποθηκευτικού χώρου που αναφέραμε προηγουμένως.

Ένας απλός τρόπος να αξιοποιήσουμε τη δυνατότητα διόρθωσης σφαλμάτων εντός του κώδικα είναι να σχηματίσουμε έναν κώδικα επανάληψης, όπου κάθε κωδικολέξη, έστω μήκους 11, θα μεταδίδεται τρεις φορές διαδοχικά. Τότε εάν το πολύ μία αλλοίωση (σφάλμα) γίνεται ανά 33 ψηφία μετάδοσης, τουλάχιστον σε δύο από τις τρεις μεταδόσεις, θα πάρουμε την ίδια κωδικολέξη. Για παράδειγμα, έστω ότι ο πομπός μεταδίδει την κωδικολέξη 10101010101 τρεις φορές διαδοχικά και προκύπτει αλλοίωση στο δωδέκατο ψηφίο. Ο δέκτης θα λάβει το ακόλουθο μήνυμα

$$10101010101\mathbf{0}01010101010110101010101.$$

(Στο παραπάνω παράδειγμα έχουμε επισημάνει το εσφαλμένο ψηφίο 0). Το δωδέκατο ψηφίο αποτελεί το πρώτο ψηφίο της δεύτερης κωδικολέξης, άρα η πρώτη και η τρίτη κωδικολέξη μεταδόθηκαν σωστά. Επομένως, αφού η σύγκριση των τριών κωδικολέξεων είναι σχετικά εύκολη, η μόνη θυσία που κάνουμε ώστε να έχουμε τη δυνατότητα να διορθώσουμε ένα σφάλμα είναι ότι ο δείκτης πληροφορίας μειώνεται από 1 σε $\frac{1}{3}$.

Αργότερα θα δούμε ότι προσθέτοντας μόνο 4 επιπλέον ψηφία σε κάθε μια κωδικολέξη μήκους 11, θα μπορούμε να διορθώσουμε κάθε κωδικολέξη στην οποία έχει αλλοιωθεί ένα μόνο ψηφίο της. Έτσι θα προκύψει ένας κώδικας με δείκτη πληροφορίας $\frac{11}{15}$, μια σημαντική βελτίωση δεδομένου ότι τα επιπλέον κόστη της κωδικοποίησης και αποκωδικοποίησης δεν μας είναι απαγορευτικά.

Σκοπός μας, λοιπόν, είναι να κατασκευάσουμε κώδικες με όσο το δυνατόν υψηλούς δείκτες πληροφορίας (τιμές κοντά στο 1), χαμηλά κόστη κωδικοποίησης και αποκωδικοποίησης και δυνατότητες ανίχνευσης και διόρθωσης σφαλμάτων εντός της κωδικολέξης, ώστε να αποφύγουμε την περίπτωση επαναμετάδοσης του αλλοιωμένου μηνύματος.

1.6 Εντοπισμός της πιθανότερης μεταδιδόμενης κωδικολέξης

Ας υποθέσουμε ότι έχουμε μια πλήρη εικόνα όλης της διαδικασία μετάδοσης - παραλαβής, γνωρίζοντας κάθε φορά την κωδικολέξη v που μετέδωσε ο πομπός και τη λέξη w είναι που παρέλαβε ο δέκτης. Για οποιαδήποτε v και w όπως τα παραπάνω, ορίζουμε ως $\phi_p(v,w)$ να είναι η πιθανότητα να παραλάβουμε την w εάν μας έχει αποσταλεί η κωδικολέξη v μέσω ενός BSC. Υπενθυμίζουμε ότι έχουμε υποθέσει ότι ο θόρυβος κατανέμεται τυχαία, άρα μπορούμε να αντιμετωπίσουμε τη μετάδοση κάθε ψηφίου ως ένα ανεξάρτητο γεγονός. Οπότε εάν η κωδικολέξη v με τη λέξη w δε ταυτίζονται σε d πλήθος θέσεων, τότε έχουμε d εσφαλμένα ψηφία και $n-d$ ψηφία που μεταδόθηκαν σωστά. Συνεπώς η παραπάνω πιθανότητα με τη βοήθεια της διωνυμικής κατανομής ορίζεται ως

$$\phi_p(v,w) = p^{n-d}(1-p)^d. \tag{1.2}$$

Ο διωνυμικός συντελεστής $\binom{n}{d}$ παραλείπεται καθώς γνωρίζουμε ακριβώς τις θέσεις των σωστών ψηφίων $n-d$.

Παράδειγμα 1.6.1. *Έστω C ένας κώδικας μήκους 5. Τότε για οποιαδήποτε κωδικολέξη v του C, η πιθανότητα η v να ληφθεί σωστά είναι*

$$\phi_p(v,v) = p^{5-0}(1-p)^0 = p^5.$$

Έστω v=10101 να είναι μια κωδικολέξη του C. Έστω ότι ο πομπός την μεταδίδει και ο δέκτης λαμβάνει τη λέξη w=01101. Συγκρίνοντας τις δύο λέξεις παρατηρούμε ότι τα δύο πρώτα τους ψηφία διαφέρουν, άρα έχουμε d=2 εσφαλμένα ψηφία. Τότε

$$\phi_p(10101, 01101) = p^{5-2}(1-p)^2 = p^3(1-p)^2$$

και εάν θέσουμε την αξιοπιστία του καναλιού με p=0.9 τότε

$$\phi_{0.9}(10101, 01101) = (0.9)^3(0.1)^2 = 0.00729.$$

Με άλλα λόγια, εάν στο συγκεκριμένο κανάλι με αξιοπιστία 90% ο πομπός στείλει το μήνυμα 10101, υπάρχει πιθανότητα 0,729% ο δέκτης να λάβει το μήνυμα 01101.

Στην πράξη γνωρίζουμε τη λέξη *w* που λαμβάνουμε, όμως δε γνωρίζουμε την πραγματική κωδικολέξη *v* που μεταδόθηκε. Ωστόσο κάθε μια κωδικολέξη v ορίζει μια συνάρτηση μιας μεταβλητής $\phi_p(v,w)$, όπου η μεταβλητή είναι η *w*. Κάθε τέτοια συνάρτηση είναι ένα μαθηματικό μοντέλο και διαλέγουμε το κατάλληλο μοντέλο (στην περίπτωση μας την κατάλληλη κωδικολέξη v), που συμφωνεί περισσότερο και καλύτερα με τις παρατηρήσεις μας (στην περίπτωσή μας την παραληφθείσα λέξη w). Ο εντοπισμός αυτής της πιθανότερης μεταδιδόμενης λέξης v, βρίσκεται υπολογίζοντας την μέγιστη τιμή όλων $\phi_p(u,w)$:

$$\phi_p(v,w) = max\{\phi_p(u,w) : u \in C\}.$$

Με άλλα λόγια, η παραπάνω σχέση μας λέει ότι δίχως να γνωρίζουμε την μεταδιδόμενη κωδικολέξη *v*, μπορούμε να υπολογίσουμε όλες τις τιμές των πιθανοτήτων $\phi_p(u,w)$, όπου *u* παριστάνει κάθε μία κωδικολέξη του κώδικα *C* και το *w* τη λέξη που έλαβε ο δέκτης και η μέγιστη τιμή αυτών θα ισούται με την πιθανότητα $\phi_p(v,w)$. Το ακόλουθο θεώρημα αποτελεί ένα κριτήριο για να βρούμε μία τέτοια κωδικολέξη *v*.

Θεώρημα 1.6.2. *Έστω ότι έχουμε ένα BSC με $\frac{1}{2} < p < 1$. Έστω v_1 και v_2 κωδικολέξεις και w μια λέξη, κάθε μία μήκους n. Έστω ότι οι v_1 και w διαφέρουν σε d_1 το πλήθος θέσεις και οι v_2 και w διαφέρουν σε d_2 το πλήθος θέσεις. Τότε*

$$\phi_p(v_1,w) \leq \phi_p(v_2,w) \text{ αν και μόνο αν } d_1 \geq d_2.$$

Απόδειξη: Έχουμε ήδη ορίσει ότι $\phi_p(v_1,w) \leq \phi_p(v_2,w)$

$$\iff p^{n-d_1}(1-p)^{d_1} \leq p^{n-d_2}(1-p)^{d_2}$$
$$\iff \left(\frac{p}{1-p}\right)^{d_2-d_1} \leq 1$$
$$\iff d_2 \leq d_1 \quad \text{αφού από υπόθεση} \frac{p}{1-p} > 1. \quad \square$$

Το παραπάνω θεώρημα εισάγει τη μαθηματική μέθοδο που πρέπει να ακολουθήσουμε για να διορθώσουμε τη λέξη που παραλάβαμε. Μέχρι τώρα την διαδικασία αυτή την ακολουθούσαμε διαισθητικά, δηλαδή προτιμούσαμε να διορθώσουμε μια λέξη w σε μία κωδικολέξη v έχοντας ως κριτήριο η w να ταυτίζεται με την v σε όσο το δυνατόν περισσότερες θέσεις (ψηφία ή άλλα σύμβολα).

Παράδειγμα 1.6.3. *Έστω ότι ένας δέκτης μέσω ενός BSC με p=0.98 λαμβάνει τη λέξη w=00110. Εάν* $C = \{01101, 01001, 10100, 10101\}$*, να βρεθεί η κωδικολέξη που είναι το πιθανότερο να μεταδόθηκε. Εάν d είναι το πλήθος των θέσεων όπου η κωδικολέξη v δε ταυτίζεται τη λέξη w, τότε σύμφωνα με το παρακάτω πινακάκι*

v	*d*
01101	*3*
01001	*4*
10100	*2*
10101	*3*

και σύμφωνα με το Θεώρημα 1.6.2 για d=2 η κωδικολέξη 10100 είναι η πιθανότερη να μετέδωσε ο πομπός. Ας σημειώσουμε εδώ ότι δε μας ενδιαφέρει η ακριβής τιμή της αξιοπιστίας του καναλιού, παρά μόνο να γνωρίζουμε ότι $p > \frac{1}{2}$.

1.7 Κάποιες προαπαιτούμενες γνώσεις Άλγεβρας

Ένα βασικό πρόβλημα που καλούμαστε να αντιμετωπίσουμε είναι με ποιο τρόπο μπορούμε πιο αποτελεσματικά να βρούμε την κοντινότερη κωδικολέξη οποιασδήποτε λέξης έχει λάβει ο δέκτης. Εάν ο κώδικας έχει πολλές κωδικολέξεις τότε είναι πρακτικά αναποτελεσματικό να συγκρίνουμε κάθε λέξη *w* με κάθε κωδικολέξη *v* έτσι ώστε να δούμε ποια κωδικολέξη δε ταυτίζεται με την *w* σε όσο το δυνατόν λιγότερες θέσεις. Για παράδειγμα, εάν ο κώδικας περιέχει $2^{12} = 4096$ κωδικολέξεις (όπως χρησιμοποιήθηκε στην αποστολή του Voyager 1 που τον Μάρτιο του 2021 απείχε από τη Γη περίπου 22,7 δισεκατομμύρια χιλιόμετρα, όντας το πιο απομακρυσμένο από τη Γη αντικείμενο ανθρώπινης κατασκευής[4]), τότε η διαδικασία αποκωδικοποίησης δε θα μπορούσε να συνεχιστεί με το ρυθμό μετάδοσης εισερχόμενων πληροφοριών. Για να ξεπεράσουμε αυτό το πρόβλημα, με τη βοήθεια της Άλγεβρας, θα εισάγουμε κάποιες δομές μέσα στον κώδικά μας.

Ορισμός 1.7.1. *Έστω το σύνολο* $K = \{0, 1\}$ *και έστω* K^n *το σύνολο όλων των δυαδικών λέξεων μήκους n. Ορίζουμε την πρόσθεση των στοιχείων του K ως:*

$$0 + 0 = 0, \quad 0 + 1 = 1, \quad 1 + 0 = 1, \quad 1 + 1 = 0$$

και τον βαθμωτό πολλαπλασιασμό των στοιχείων του K ως:

$$0 \cdot 0 = 0, \quad 1 \cdot 0 = 0, \quad 0 \cdot 1 = 0, \quad 1 \cdot 1 = 1.$$

Η πρόσθεση των στοιχείων του K^n ορίζεται κατά συνιστώσες. Για παράδειγμα, για να προσθέσουμε τις λέξεις $v = 01101$ και $w = 11001$, προσθέτουμε κάθε ψηφίο της v με το ψηφίο που βρίσκεται στην αντίστοιχη θέση της w, δηλαδή

$$\begin{array}{ll} 01101 & \\ 11001 & (+) \\ \hline 10100 & \end{array}$$

όπου παίρνουμε $v + w = 10100$. Είναι προφανές ότι η πρόσθεση δύο δυαδικών λέξεων μήκους n η κάθε μία, μας δίνει ως αποτέλεσμα μία δυαδική λέξη μήκους n, άρα το σύνολο K^n είναι κλειστό ως προς την πρόσθεση.

Κάθε στοιχείο του συνόλου K είναι ένας αριθμός, δηλαδή ένα βαθμωτό ή μονόμετρο μέγεθος. Τότε ο βαθμωτός πολλαπλασιασμός του K^n ορίζεται ως εξής: Τα μόνα βαθμωτά πολλαπλάσια μιας λέξης w είναι το $0 \cdot w$, που είναι το στοιχείο του K^n με κάθε συνιστώσα του να είναι 0 (μηδενική λέξη ή μηδενικό στοιχείο), και το $1 \cdot w$ που είναι το w.

Έχοντας ορίσει την πρόσθεση και τον βαθμωτό πολλαπλασιασμό και με τις ιδιότητες που θα αναφέρουμε παρακάτω, μπορούμε να δείξουμε ότι ο K^n είναι ένας *διανυσματικός χώρος*. Για οποιεσδήποτε λέξεις u, v, w μήκους n και για οποιαδήποτε βαθμωτά a και b:

1. Προσεταιριστική: $(u + v) + w = u + (v + w)$
2. Ύπαρξη μηδενικού στοιχείου: $v + 0 = 0 + v = v$, όπου 0 είναι η μηδενική λέξη
3. Ύπαρξη αντίθετου στοιχείου: $v + v' = v' + v = 0$ για κάθε $v' \in K^n$
4. Αντιμεταθετική: $v + w = w + v$
5. Επιμεριστική: $a(v + w) = av + aw$ και $(a + b)v = av + bv$
6. $(ab)v = a(bv)$
7. $1v = v$
8. $v + w \in K^n$
9. $av \in K^n$

Να επισημάνουμε ότι εάν η κωδικολέξη v μεταδοθεί μέσω ενός BSC και η λέξη w ληφθεί χωρίς κάποιο σφάλμα, τότε το ψηφίο 0 προκύπτει στην αντίστοιχη συνιστώσα (θέση) του $v+w$. Ειδάλλως, προκύπτει το ψηφίο 1 εάν παρουσιαστεί κάποιο σφάλμα. Για παράδειγμα, εάν $v = 10101$ είναι η κωδικολέξη που μεταδόθηκε και $w = 01100$ είναι η λέξη που παρελήφθη, τότε τα σφάλματα που προκύπτουν είναι στην πρώτη, δεύτερη και πέμπτη συνιστώσα (θέση).Οπότε το *μοτίβο σφάλματος* (error pattern) ορίζεται να είναι το $v + w = 11001$.

1.8 Βάρος και απόσταση

Σε αυτήν την ενότητα θα δούμε δύο σημαντικούς όρους, το βάρος και την απόσταση.

Έστω v μια λέξη μήκους n. Βάρος Hamming ή απλά *βάρος*, είναι το πλήθος των ψηφίων 1 που εμφανίζονται σε μία λέξη v και το συμβολίζουμε με $wt(v)$. Για παράδειγμα, $wt(110101) = 4$ και $wt(00000) = 0$.

Έστω v και w λέξεις μήκους n. Η απόσταση Hamming ή απλά απόσταση, μεταξύ των v και w είναι το πλήθος των θέσεων όπου οι v και w δε ταυτίζονται και το συμβολίζουμε με $d(v, w)$. Για παράδειγμα, $d(01011, 00111) = 2$ και $d(10110, 10110) = 0$.

Αξίζει να επισημάνουμε σε αυτό το σημείο ότι η απόσταση μεταξύ των v και w είναι η ίδια με τος βάρος του μοτίβου σφάλματος $u = v + w$ δηλαδή

$$d(v, w) = wt(v + w).$$

Για παράδειγμα, εάν $v = 11010$ και $w = 01101$, τότε έχουμε $d(v, w) = d(11010, 01101) = 4$ και $wt(v + w) = wt(11010 + 01101) = wt(10111) = 4$. Συνεπώς, μπορούμε να επαναδιατυπώσουμε τη σχέση (1.2) της Ενότητας 1.6 ως

$$\phi_p(v, w) = p^{n-wt(u)}(1 - p)^{wt(u)}$$

όπου u είναι το μοτίβο σφάλματος $u = v + w$. Οπότε θα συμβολίζουμε πλέον με $\phi_p(v, w)$ την πιθανότητα του μοτίβου σφάλματος $u = v + w$.

Μερικές χρήσιμες ιδιότητες του βάρους και της απόστασης είναι οι ακόλουθες. Έστω u, v και w λέξεις μήκους n και a ένα ψηφίο.

1. $0 \leq wt(v) \leq n$
2. $wt(v) = 0 \iff v = 0$
3. $0 \leq d(v, w) \leq n$
4. $d(v, w) = 0 \iff v = w$
5. $d(v, w) = d(w, v)$
6. $wt(v + w) \leq wt(v) + wt(w)$
7. $d(v, w) \leq d(v, u) + d(u, w)$
8. $wt(av) = a \cdot wt(v)$
9. $d(av, aw) = a \cdot d(v, w)$

1.9 Αποκωδικοποίηση μέγιστης πιθανοφάνειας

Έχοντας σχηματίσει πλέον μια γενικότερη εικόνα της θεωρίας των κωδίκων, θα ασχοληθούμε τώρα με δύο βασικά προβλήματά της, την κωδικοποίηση και την απο-

κωδικοποίηση. Ας υποθέσουμε ότι βρισκόμαστε στο σημείο λήψης ενός BSC και θέλουμε να λάβουμε το μήνυμα που μετέδωσε ο πομπός από το άλλο άκρο. Ο πομπός είναι αυτός που ήδη έχουμε κατασκευάσει και φυσικά ένα βασικό πρόβλημα αποτελεί ο σχεδιασμός ενός κατάλληλου, καθώς και ο έλεγχος της ποιότητάς του και της απόδοσής του.

Υπάρχουν δύο ποσότητες όπου δεν έχουμε καθόλου έλεγχο. Η μία ποσότητα είναι η αξιοπιστία του καναλιού, δηλαδή την πιθανότητα p ότι το BSC θα μεταδώσει κάποιο ψηφίο σωστά. Η δεύτερη είναι το συνολικό πλήθος των πιθανών μηνυμάτων που θα πρέπει μεταδοθούν. Δεν έχει σημασία το περιεχόμενο ενός μηνύματος αλλά το συνολικό πλήθος των μηνυμάτων. Ας δούμε τώρα τα δύο βασικά προβλήματά της θεωρίας μας.

Κωδικοποίηση Πρέπει να κατασκευάσουμε έναν κώδικα που θα χρησιμοποιηθεί για την αποστολή των μηνυμάτων. Πρώτα επιλέγουμε έναν θετικό ακέραιο k, ο οποίος θα αντιπροσωπεύει το μήκος της κάθε λέξης του μηνύματος. Ας συμβολίσουμε με M το σύνολο όλων των δυνατών μηνυμάτων μας. Από τη στιγμή που κάθε μήνυμα από το M αντιστοιχίζεται σε μία μοναδική δυαδική λέξη μήκους k, επιλέγουμε το k να είναι αρκετά μεγάλο έτσι ώστε να εξασφαλίσουμε ότι ο κώδικάς μας έχει περισσότερες κωδικολέξεις απ' το ή με άλλα λόγια θα πρέπει να εξασφαλίσουμε $|M| \leq 2^k$, όπου με $|M|$ παριστάνουμε το πλήθος των στοιχείων του . Έπειτα πρέπει να αποφασίσουμε πόσα ψηφία επιπλέον χρειάζεται να προσθέσουμε σε κάθε μία λέξη μήκους k, έτσι ώστε να διασφαλίσουμε ότι θα ανιχνεύσουμε και θα διορθώσουμε όσο το δυνατόν περισσότερα σφάλματα προκύψουν κατά τη μετάδοση. Συνεπώς, για τη μετάδοση ενός συγκεκριμένου μηνύματος, ο πομπός βρίσκει τη λέξη μήκους k που σχετίζεται με το μήνυμα, προθέτει τα επιπλέον ψηφία ελέγχου σχηματίζοντας την κωδικολέξη μήκους n και εν τέλει την μεταδίδει.

Αποκωδικοποίηση Έστω ότι ο δέκτης λαμβάνει μια λέξη w. Για να αποφασίσουμε ποια λέξη v μεταδόθηκε από τον πομπό, χρησιμοποιούμε μια διαδικασία που λέγεται *αποκωδικοποίηση μέγιστης πιθανοφάνειας* (maximum likelihood decoding ή MLD), η οποία MLD χωρίζεται σε δύο υποκατηγορίες.

1. Πλήρης Αποκωδικοποίηση Μέγιστης Πιθανοφάνειας (Complete Maximum Likelihood Decoding ή CMLD). Στην περίπτωση που υπάρχει μία μοναδική λέξη v στο C που είναι κοντινότερη στη w από οποιαδήποτε άλλη λέξη στο C, αποκωδικοποιούμε τη λέξη w ως v. Αναλυτικότερα, εάν

$$d(v, w) < d(v_1, w) \quad \forall v_1 \in C, v_1 \neq v,$$

   τότε αποκωδικοποιούμε τη w με τη v. Στην περίπτωση που υπάρχουν παραπάνω από μία λέξεις στο C που είναι κοντινότερες με τη w, τότε επιλέγουμε μία τυχαία από αυτές και καταλήγουμε ότι ήταν η κωδικολέξη μου μετέδωσε ο πομπός.

2. Μη-Πλήρης Αποκωδικοποίηση Μέγιστης Πιθανοφάνειας (Incomplete Maximum Likelihood Decoding ή IMLD). Όμοια με την περίπτωση της CMLD, εάν υπάρχει μονα-

δική λέξη v στο C που είναι κοντινότερη στη w, αποκωδικοποιούμε τη w ως v. Όμως, στην περίπτωση που υπάρχουν παραπάνω από μία λέξεις v στο C που είναι κοντινότερες με τη w, δηλαδή έχουν την ίδια απόσταση από τη w, τότε ζητούμε επαναμετάδοση του μηνύματος. Επαναμετάδοση μπορούμε να ζητήσουμε και σε κάποιες περιπτώσεις όπου η λέξη w απέχει πάρα πολύ από κάποια κωδικολέξη v.

Να επισημάνουμε ότι η MLD δε λειτουργεί πάντοτε, όπως για παράδειγμα, εάν εμφανιστούν πάρα πολλά σφάλματα κατά τη μετάδοση των μηνυμάτων μέσω του BSC καναλιού μας.

Όπως αναφέραμε νωρίτερα, η κωδικολέξη $v \in C$ είναι κοντινότερη στη λέξη w όταν η απόσταση τους $d(v,w)$ είναι η ελάχιστη. Συνεπώς από το Θεώρημα 1.6.2, έχει τη μεγαλύτερη πιθανότητα $\phi_p(v,w)$ σε σχέση με όλες τις υπόλοιπες κωδικολέξεις, να είναι η πιο πιθανή κωδικολέξη που στάλθηκε. Στην προηγούμενη ενότητα, αναφέραμε ότι η απόσταση μεταξύ των v και w είναι η ίδια με τος βάρος του μοτίβου σφάλματος $u = v+w$ δηλαδή $d(v,w) = wt(v+w)$. Οπότε μπορούμε να αναδιατυπώσουμε το Θεώρημα 1.6.2 ως εξής:

$$\phi_p(v_1,w) \leq \phi_p(v_2,w) \iff wt(v_1+w) \geq wt(v_2+w),$$

ότι δηλαδή η πιο πιθανή κωδικολέξη που μεταδόθηκε είναι αυτή που το μοτίβο σφάλματος της έχει το μικρότερο βάρος.

Εν κατακλείδι, ο σκοπός της MLD είναι να εξετάσει όλα τα μοτίβα σφάλματοςτος $v+w$ για όλες τις κωδικολέξεις v, και να επιλέξει την v της οποίας το μοτίβο σφάλματος έχει το μικρότερο βάρος. Εάν είναι περισσότερα από ένα τα μοτίβα σφάλματοςτος, στην περίπτωση της CMLD επιλέγεται ένα τυχαία, ενώ στην περίπτωση της IMLD ζητείται επαναμετάδοση μέχρι βρεθεί μοναδικό μοτίβο σφάλματος με το μικρότερο βάρος.

Παράδειγμα 1.9.1. *Έστω ότι θέλουμε να μεταδώσουμε δύο μηνύματα* $|M| = 2$ *και διαλέγουμε* $n = 3$ *οπότε και* $C = \{000, 111\}$. *Εάν μεταδώσουμε την* $v = 000$, *θα δούμε σε ποιες περιπτώσεις η IMLD θα μας δώσει την σωστή αποκωδικοποιημένη λέξη* 000 *και σε ποιες περιπτώσεις την εσφαλμένη* 111. *Κατασκευάζουμε τον ακόλουθο πίνακα.*

Ληφθείσα w	*μοτίβο* $000+w$	*Σφάλμα* $111+w$	*Αποκωδικοποιημένη* v
000	***000***	*111*	*000*
100	***100***	*011*	*000*
010	***010***	*101*	*000*
001	***001***	*110*	*000*
110	*110*	***001***	*111*
101	*101*	***010***	*111*
011	*011*	***100***	*111*
111	*111*	***000***	*111*

Η πρώτη στήλη του παραπάνω πίνακα περιέχει όλους τους δυνατούς συνδυασμούς λέξεων μήκους 3, που θα μπορούσε ο δέκτης να λάβει από τη μετάδοση. Η δεύτερη και

τρίτη στήλη περιέχουν το μοτίβο σφάλματος $v + w$ και η έντονη μορφοποίηση δηλώνει ότι η IMLD επέλεξε το συγκεκριμένο, καθώς το βάρος του $v + w$ είναι το μικρότερο. Η τελευταία στήλη περιέχει τις αποκωδικοποιημένες λέξεις $v \in C$, που είναι στην ουσία η κωδικολέξη v από επιλεγμένο μοτίβο σφάλματος $v + w$. Αναλυτικότερα, ας σχολιάσουμε την προτελευταία γραμμή του πίνακα. Ο πομπός μεταδίδει την κωδικολέξη $v = 000$ και ο δέκτης λαμβάνει μέσω ενός BSC τη λέξη $w = 011$ την οποία πρέπει να αποκωδικοποιήσει. Η IMLD σχηματίζει τα μοτίβα σφάλματοςτος για κάθε $v \in C$, δηλαδή τα $000+w = 011$ και $111 + w = 100$. Έπειτα πρέπει να επιλέξει αυτό με το μικρότερο βάρος, δηλαδή το 100 καθώς το πλήθος των ψηφίων 1 είναι λιγότερο έναντι του 011. Συνεπώς, από το επιλεγμένο μοτίβο σφάλματος $111 + w$ η IMLD επιλέγει να αποκωδικοποιήσει τη λέξη $w = 011$ στην κωδικολέξη $v = 111$.

Ας δούμε τώρα τρία σημαντικά κριτήρια που πρέπει να λαμβάνουμε υπ'όψιν μας όσο αφορά την επιλογή κατάλληλου μήκους κωδικολέξεων n και κώδικα C.

1. Μεγαλύτερες λέξεις θα κάνουν περισσότερο χρόνο να μεταδοθούν και να αποκωδικοποιηθούν, άρα το n δε θα πρέπει να είναι πολύ μεγάλο. Πρέπει να επιλέγουμε κατάλληλο ώστε ο δείκτης πληροφορίας να είναι όσο πιο κοντά γίνεται στο 1.

2. Όταν ο δέκτης λαμβάνει πολλά μηνύματα το δευτερόλεπτο, εάν το $|C|$ είναι μεγάλο, έστω μερικές χιλιάδες, για να εφαρμοστεί η διαδικασία της MLD θα είναι χρονοβόρα και θα χρειαστεί πολλούς υπολογιστικούς πόρους. Ευτυχώς υπάρχουν κατάλληλοι έξυπνοι κώδικες με ταχύτατους υπολογισμούς του IMLD.

3. Εάν πολλά σφάλματα γίνονται κατά τη μετάδοση, η MLD δε θα λειτουργήσει σωστά, καθώς θα επιλέγει συνεχώς κωδικολέξεις οι οποίες θα διαφέρουν από τις πραγματικές που έστειλε ο πομπός. Οπότε ο κώδικας C θα πρέπει να επιλέγεται λαμβάνοντας υπ' όψιν ότι η πιθανότητα ότι η MLD θα λειτουργήσει σωστά να είναι αρκετά υψηλή.

Τελικά, καταλήγουμε στο συμπέρασμα ότι ο κύριος σκοπός της θεωρίας των κωδίκων είναι η εύρεση κατάλληλων συνόλων C από λέξεις που θα ικανοποιούν όσο το δυνατόν περισσότερο τα τρία παραπάνω κριτήρια.

1.10 Αξιοπιστία της αποκωδικοποίησης μέγιστης πιθανοφάνειας

Ας υποθέσουμε ότι έχουμε διαλέξει κατάλληλα n και C. Θα προσδιορίσουμε τώρα την πιθανότητα $\theta_p(C, v)$, δηλαδή εάν η κωδικολέξη v αποστέλλεται μέσω ενός BSC πιθανότητας p, τότε η IMLD συμπεραίνει σωστά ότι η v ήταν τελικά αυτή που στάλθηκε.

Βρίσκουμε ένα σύνολο $L(v)$ το οποίο αποτελείται από όλες τις λέξεις του συνόλου K^n που είναι κοντινότερες με την v από οποιαδήποτε άλλη λέξη στο C. Τότε η *αξιοπιστία* $\theta_p(C, v)$ είναι το άθροισμα όλων των πιθανοτήτων $\phi_p(v, w)$ καθώς το w κυμαίνεται στο σύνολο $L(v)$,δηλαδή,

$$\theta_p(C, v) = \sum_{w \in L(v)} \phi_p(v, w).$$

Μπορούμε να βρούμε το σύνολο $L(v)$ από το πινακάκι της IMLD όπως στο Παράδειγμα 1.9.1. Το $L(v)$ είναι ακριβώς το σύνολο όλων των λέξεων στο K^n για τις οποίες η IMLD θα συμπεράνει σωστά ότι η κωδικολέξη v μεταδόθηκε από τον πομπό. Δηλαδή, σε κάθε γραμμή από το πινακάκι όπου προκύπτει η αποκωδικοποιήμενη v στη τελευταία στήλη, η λέξη w στην πρώτη στήλη της ίδιας γραμμής ανήκει στο σύνολο $L(v)$.

Ωστόσο, η $\theta_p(C, v)$ δε λαμβάνει υπόψιν την πιθανότητα της επαναμετάδοσης, η οποία συμβαίνει στην περίπτωση της IMLD, όταν για παράδειγμα η ληφθείσα λέξη w ισαπέχει από δύο κωδικολέξεις v. Τέτοιου είδους περιπτώσεις οδηγούν σε κάποιες ανωμαλίες, όπως $\theta_p(K^n, v) > \theta_p(C, u)$ για κάθε v στο K^n και u στο C, όμως οδηγούν σε μία ασφαλή πρώτη εκτίμηση για τη μέτρηση της αξιοπιστίας. Συγκεκριμένα, η $\theta_p(C, v)$ αποτελεί ένα κάτω φράγμα για την πιθανότητα ότι η v έχει αποκωδικοποιηθεί σωστά.

Παράδειγμα 1.10.1. *Έστω $p = 0,9$, $|M| = 2$, $n = 3$ και $C = \{000, 111\}$, όπως στο Παράδειγμα 1.9.1. Για την αποκωδικοποιημένη λέξη $v = 000$ σύμφωνα με το πινακάκι της IMLD, προκύπτει το σύνολο*

$$L(000) = \{000, 100, 010, 001\},$$

καθώς οι παραπάνω λέξεις w είναι πιο κοντά στην $v = 000$ έναντι της $v = 111$. Τότε έχουμε,

$$\begin{aligned}
\theta_p(C, 000) &= \sum_{w \in L(000)} \phi_p(000, w) \\
&= \phi_p(000, 000) + \phi_p(000, 100) + \phi_p(000, 010) + \phi_p(000, 001) \\
&= p^3 + p^2(1-p) + p^2(1-p) + p^2(1-p) \\
&= p^3 + 3p^2(1-p) \\
&= 0,972 \text{ (για p=0,9).}
\end{aligned}$$

Όμοια, για την αποκωδικοποιημένη λέξη $v = 111$, προκύπτει το σύνολο

$$L(111) = \{110, 101, 011, 111\},$$

οπότε έχουμε

$$\begin{aligned}
\theta_p(C, 111) &= \sum_{w \in L(111)} \phi_p(111, w) \\
&= \phi_p(111, 110) + \phi_p(111, 101) + \phi_p(111, 011) + \phi_p(111, 111) \\
&= p^2(1-p) + p^2(1-p) + p^2(1-p) + p^3 \\
&= 3p^2(1-p) + p^3 \\
&= 0,972 \text{ (για p=0,9).}
\end{aligned}$$

Τελικά, οδηγούμαστε στο συμπέρασμα ότι εάν είτε η κωδικολέξη $v = 000$ είτε η $v = 111$ μεταδοθούν μέσω ενός BSC πιθανότητας $p = 0,9$, τότε η πιθανότητα η IMLD να συμπεράνει σωστά ότι η v ήταν αυτή που τελικά στάλθηκε, είναι 0,972 ή ισοδύναμα

97,2%.

Έστω τώρα αντί του κώδικα $C = \{000, 111\}$ *χρησιμοποιήσουμε τον* $C = \{001, 101\}$. *Δουλεύοντας με παρόμοιο τρόπο βρίσκουμε ότι*

$$\theta_p(C, 001) = \theta_p(C, 101) = 0,900 = 90\%$$

Αυτό σημαίνει ότι με την επιλογή του συγκεκριμένου κώδικα C, θα προκύψουν περισσότερα σφάλματα κατά την αποκωδικοποίηση μέσω της MLD. Συνεπώς, η επιλογή του πρώτου κώδικα C είναι προτιμότερη.

1.11 Κώδικες ανίχνευσης σφαλμάτων

Σε αυτή την ενότητα θα δούμε σε μεγαλύτερο βάθος πότε ένας κώδικας C μπορεί να ανιχνεύσει σφάλματα. Θυμίζουμε ότι εάν μια κωδικολέξη $v \in C$ αποστέλνεται και μία λέξη $w \in K^n$ λαμβάνεται, τότε το $u = v + w$ είναι το μοτίβο σφάλματος. Οποιαδήποτε λέξη $u \in K^n$ μπορεί να προκύψει ως ένα μοτίβο σφάλματος οπότε θέλουμε να γνωρίζουμε ποια μοτίβα σφάλματος ο C θα ανιχνεύσει.

Θα λέμε ότι ο κώδικας C *ανιχνεύει* ένα μοτίβο σφάλματος u αν και μόνο αν το $v + u$ δεν είναι κωδικολέξη για κάθε $v \in C$. Με άλλα λόγια, το u θα ανιχνεύεται εάν για οποιαδήποτε μεταδιδόμενη κωδικολέξη v, ο αποκωδικοποιητής αφού λάβει την $v + u$ θα είναι σε θέση να αναγνωρίσει ότι δεν είναι κωδικολέξη και έτσι θα συμπεραίνει ότι κάποιο σφάλμα συνέβη.

Παράδειγμα 1.11.1. *Έστω* $C = \{001, 101, 110\}$. *Για το μοτίβο σφάλματος* $u = 010$, *υπολογίζουμε το* $v + 010$ *για κάθε* $v \in C$:

$$001 + 010 = 011,$$

$$101 + 010 = 111,$$

$$110 + 010 = 100.$$

Καμία από τις τρεις λέξεις $011, 111, 100$ *δεν ανήκει στον κώδικα C, οπότε ο C ανιχνεύει το μοτίβο σφάλματος 010. Αντιθέτως, για το μοτίβο σφάλματος* $u = 100$ *έχουμε*

$$001 + 100 = 101,$$

$$101 + 100 = 001,$$

$$110 + 100 = 010.$$

Από τη στιγμή που τουλάχιστον μία λέξη από τις παραπάνω είναι κωδικολέξη του C, ο C δεν ανιχνεύει το μοτίβο σφάλματος 100.

Το πινακάκι για την IMLD μπορεί να χρησιμοποιηθεί για να βρούμε ποια μοτίβα σφάλματος ένα κώδικας C θα ανιχνεύσει. Η πρώτη στήλη περιέχει κάθε λέξη στο K^n. Έτσι, η πρώτη στήλη μπορεί να ερμηνευτεί ως όλοι οι πιθανοί συνδυασμοί μοτίβων

σφάλματος όπου στην κάθε περίπτωση οι μοτίβο σφάλματος στήλες, θα περιέχουν τα αθροίσματα $v+u$ για κάθε $v \in C$. Εάν σε κάποια συγκεκριμένη γραμμή κανένα από αυτά τα αθροίσματα δεν είναι κωδικολέξη του C, τότε ο C ανιχνεύει το μοτίβο σφάλματος στην πρώτη στήλη αυτής της γραμμής.

Παράδειγμα 1.11.2. *Έστω ο κώδικας* $C = \{000, 111\}$ *με το ακόλουθο πινακάκι της IMLD*

u	$000+u$	$111+u$
000	000	111
100	100	011
010	010	101
001	001	110
110	110	001
101	101	010
011	011	100
111	111	000

Στην πρώτη στήλη βρίσκονται όλοι οι πιθανοί συνδυασμοί μοτίβο σφάλματος u. Στη δεύτερη και τρίτη στήλη βρίσκονται τα αθροίσματα $v+u$. Εάν σε καμία σειρά από τη δεύτερη και τρίτη στήλη δεν υπάρχει μία κωδικολέξη $v \in C$, δηλαδή ούτε η 000 ούτε η 111, τότε ο C ανιχνεύει το u. Άρα ο C θα ανιχνεύσει όσα μοτίβα σφάλματος u εμφανίζονται στο παραπάνω πινακάκι με έντονη μορφοποίηση.

Μια εναλλακτική και γρηγορότερη μέθοδος είναι να βρεθούν όλα τα μοτίβα σφάλματος που ο C δε μπορεί να ανιχνεύσει. Τότε όλα τα υπολειπόμενα μοτίβα σφάλματος θα μπορούν να ανιχνευτούν από τον C. Για κάθε ζεύγος κωδικολέξεων v και w, εάν το $e = v + w$ τότε το e δε μπορεί να ανιχνευτεί αφού το $v + e = w$, το οποίο είναι μια κωδικολέξη. Οπότε το σύνολο όλων των πρότυπων σφάλμα που δε μπορούν να ανιχνευτούν από τον C είναι το σύνολο όλων των λέξεων που μπορούν να γραφούν σαν άθροισμα δύο κωδικολέξεων.

Παράδειγμα 1.11.3. *Έστω ο κώδικας* $C = \{1000, 0100, 1111\}$. *Τότε προσθέτοντας ανά δύο κωδικολέξεις μεταξύ τους έχουμε*

$$1000 + 1000 = 0000, \quad 1000 + 0100 = 1100$$

$$1000 + 1111 = 0111, \quad 0100 + 1111 = 1011.$$

Να σημειώσουμε εδώ ότι έχουμε παραλείψει σε κάποιες περιπτώσεις την πρόσθεση κωδικολέξεων με τον εαυτό τους, διότι πάντοτε θα παίρνουμε ως αποτέλεσμα την 0000. *Οπότε προκύπτει ότι το σύνολο όλων των μοτίβων σφάλματος που δε μπορούν να ανιχνευτούν από τον C είναι* $\{0000, 1100, 0111, 1011\}$. *Επομένως, όλα τα μοτίβα σφάλματος στο σύνολο* $K^4 \setminus \{0000, 1100, 0111, 1011\}$ *μπορούν να ανιχνευτούν.*

Υπάρχει ακόμη και άλλος τρόπος για την ανίχνευση μοτίβων σφάλματος. Για έναν κώδικα C που περιέχει τουλάχιστον δύο λέξεις, η *απόσταση* του κώδικα C είναι η μικρότερη τιμή της $d(v, w)$ για κάθε διαφορετικό ζεύγος $v, w \in C$. Για παράδειγμα, εάν $C =$

$\{0000, 1010, 0111\}$, τότε $d(0000, 1010) = 2, d(0000, 0111) = 3$ και $d(1010, 0111) = 3$. Άρα η απόσταση του C είναι 2.

Το ακόλουθο θεώρημα συμβάλλει στην ταυτοποίηση πολλών μοτίβων σφάλματος όπου ένας κώδικας C θα ανιχνεύει.

Θεώρημα 1.11.4. *Ένας κώδικας C απόστασης d θα ανιχνεύει τουλάχιστον όλα τα μη-μηδενικά μοτίβα σφάλματος βάρους μικρότερου ή ίσου με $d-1$. Επιπλέον, υπάρχει τουλάχιστον ένα μοτίβο σφάλματος βάρους d το οποίο ο C δε θα μπορέσει να ανιχνεύει.*

Απόδειξη: Έστω u ένα μη-μηδενικό μοτίβο σφάλματος με $wt(u) \leq d-1$, και έστω $v \in C$. Τότε

$$d(v, v+u) = wt(v+v+u) = wt(u) < d.$$

Αφού ο C έχει απόσταση d, τότε το $v+u$ δεν ανήκει στον C. Επομένως ο C ανιχνεύει το u. Από τον ορισμό της απόστασης d, υπάρχουν κωδικολέξεις v και w που ανήκουν στον C με $d(v, w) = d$. Θεωρούμε το μοτίβο σφάλματος $u = v + w$. Τώρα η $w = v + u \in C$, οπότε ο C δε θα ανιχνεύσει το μοτίβο σφάλματος u βάρους d. □

Παράδειγμα 1.11.5. *Έστω ο κώδικας $C = \{000, 111\}$. Τότε έχουμε $d(000, 111) = 3$. Σύμφωνα με το παραπάνω Θεώρημα 1.11.4, ο C θα πρέπει να ανιχνεύει όλα τα μοτίβα σφάλματος βάρους 1 ή 2 και να μην ανιχνεύει το μοναδικό μοτίβο σφάλματος βάρους 3, δηλαδή το 111. Το μοναδικό μοτίβο σφάλματος που δεν καλύπτει το παραπάνω Θεώρημα είναι το μηδενικό, δηλαδή στην περίπτωση μας το 000.*

Το Θεώρημα 1.11.4 δεν εμποδίζει έναν κώδικα C από το να ανιχνεύει μοτίβα σφάλματος βάρους d ή μεγαλύτερα.

Παράδειγμα 1.11.6. *Έστω ο κώδικας $C = \{001, 101, 100\}$. Έχουμε*

$$d(001, 101) = 1, \quad d(001, 100) = 1, \quad d(101, 100) = 1.$$

Άρα ο C έχει απόσταση $d = 1$. Αφού $d - 1 = 0$, το Θεώρημα 1.11.4 δε μπορεί να μας βοηθήσει να προσδιορίσουμε ποια μοτίβα σφάλματος ο C θα ανιχνεύσει. Όμως μας λέει ότι υπάρχει τουλάχιστον ένα μοτίβο σφάλματος βάρους $d = 1$ που ο C δε θα ανιχνεύσει. Ένα τέτοιο είναι το $u = 100$ όπως είδαμε και στο Παράδειγμα 1.11.1. Ωστόσο, ανιχνεύει το $u = 010$ το οποίο είναι επίσης βάρους d=1.

1.12 Κώδικες διόρθωσης σφαλμάτων

Εάν μια λέξη $v \in C$ μεταδίδεται μέσω ενός BSC και αν η λέξη w λαμβάνεται έτσι ώστε να προκύπτει το μοτίβο σφάλματος $u = v + w$, τότε η IMLD συμπεραίνει σωστά ότι η v ήταν που στάλθηκε με την προϋπόθεση ότι η v είναι πιο κοντά στη w σε σχέση με οποιαδήποτε άλλη κωδικολέξη v'. Εάν αυτό συμβαίνει για κάθε τέτοια v και w για τα οποία το άθροισμά τους $v + w$ μας κάνει u, τότε λέμε ότι ο C *διορθώνει* το μοτίβο σφάλματος u. Αναλυτικότερα, ένας κώδικας C διορθώνει το μοτίβο σφάλματος u, εάν για κάθε $v \in C$, το $v + u$ είναι πιο κοντά στο v από οποιαδήποτε άλλη λέξη στο C.

Παράδειγμα 1.12.1. *Έστω* $C = \{000, 111\}$ *και το μοτίβο σφάλματος* $u = 010$. *Για* $v = 000$ *έχουμε,*

$$d(000, v+u) = d(000, 010) = 1$$

$$d(111, v+u) = d(111, 010) = 2.$$

Για $v = 111$,

$$d(000, v+u) = d(000, 101) = 2$$

$$d(111, v+u) = d(111, 101) = 1.$$

Από τα παραπάνω συμπεραίνουμε ότι για την κωδικολέξη $v = 000$ *η* $v+u$ *έχει απόσταση 1 που μάλιστα είναι η μικρότερη. Επίσης, για την κωδικολέξη* $v = 111$ *η* $v+u$ *έχει απόσταση 1 που είναι πάλι η μικρότερη. Οπότε ο C διορθώνει το μοτίβο σφάλματος* $u = 010$.

Αντίθετα, για το μοτίβο σφάλματος $u = 110$ *και για* $v = 000$ *έχουμε,*

$$d(000, v+u) = d(000, 110) = 2$$

$$d(111, v+u) = d(111, 110) = 1.$$

Παρατηρούμε από τις τιμές των αποστάσεων ότι το $v+u$ *δε βρίσκεται πιο κοντά την επιλεγμένη* $v = 000$ *αλλά στην 111. Συνεπώς ο κώδικας C δε θα διορθώσει το μοτίβο σφάλματος* $u = 110$.

Το πινακάκι της IMLD μπορεί να χρησιμοποιηθεί ώστε να προσδιοριστούν ποια μοτίβα σφάλματος ένας κώδικας *C* θα διορθώσει.

Παράδειγμα 1.12.2. *Έστω ο κώδικας* $C = \{000, 111\}$ *και το παρακάτω πινακάκι όπως είδαμε στο Παράδειγμα 1.9.1.*

Ληφθείσα w	*μοτίβο 000+w*	*Σφάλμα 111+w*	*Αποκωδικοποιημένη v*
000	***000***	*111*	*000*
100	***100***	*011*	*000*
010	***010***	*101*	*000*
001	***001***	*110*	*000*
110	*110*	***001***	*111*
101	*101*	***010***	*111*
011	*011*	***100***	*111*
111	*111*	***000***	*111*

Σε κάθε γραμμή του όπου εμφανίζεται το μοτίβο σφάλματος 010 (3η και 6η γραμμή), παρατηρούμε η IMLD διορθώνει και καταλήγει σωστά στο συμπέρασμα για την κωδικολέξη v που στάλθηκε από τον πομπό. Επίσης, σε τουλάχιστον μία γραμμή (4η γραμμή) όπου προκύπτει το μοτίβο σφάλματος 110, εάν η 111 σταλεί και η 001 ληφθεί, η IMLD καταλήγει στο λάθος συμπέρασμα ότι η κωδικολέξη 000 ήταν αυτή που στάλθηκε από τον πομπό.

1.13 Γραμμικοί και κυκλικοί κώδικες

Ένας κώδικας C θα λέγεται *γραμμικός* κώδικας εάν το $v+w$ είναι μία λέξη που ανήκει στον C για οποιεσδήποτε κωδικολέξεις v,w που ανήκουν στον C. Αυτό σημαίνει ότι ένας γραμμικός κώδικας είναι ένας κώδικας ο οποίος είναι κλειστός ως προς την πρόσθεση λέξεων. Για παράδειγμα, έστω ο κώδικας $C = \{000, 111\}$ ο οποίος είναι γραμμικός καθώς για τις κωδικολέξεις $v = 000$ και $w = 111$ έχουμε ότι τα αθροίσματα τους

$$000 + 000 = 000, \quad 111 + 000 = 111,$$

$$000 + 111 = 111, \quad 111 + 111 = 000,$$

ανήκουν και αυτά στον C. Αντίθετα, ο $C_1 = \{000, 001, 101\}$ δεν είναι γραμμικός καθώς εάν επιλέξουμε $v = 001$ και $w = 101$, παρόλο που ανήκουν στον C, παρατηρούμε ότι το άθροισμά τους $001 + 101 = 100$ δεν ανήκει στον C.

Ένα πλεονέκτημα που έχει ένας γραμμικός κώδικας έναντι ενός μη-γραμμικού κώδικα είναι ότι η απόσταση του είναι ευκολότερο να υπολογισθεί, καθώς η απόσταση ενός γραμμικού κώδικα ισούται με το μικρότερο βάρος οποιασδήποτε μη μηδενικής κωδικολέξης του C. Αυτό οφείλεται στο γεγονός ότι κάθε συνδυασμός αθροισμάτων κωδικολέξεων $v + w$ θα ισούται με κάποια κωδικολέξη που θα ανήκει στον γραμμικό κώδικα. Οπότε είναι προφανές ότι αρκεί να υπολογίσουμε μόνο το βάρος της κάθε κωδικολέξης του γραμμικού κώδικα.

Έστω v μία λέξη μήκους n. Η κυκλική μετάθεση $\pi(v)$ του v είναι μία λέξη πάλι μήκους n η οποία προκύπτει από τη v παίρνοντας το τελευταίο ψηφίο της v και τοποθετώντας το στην αρχή, έτσι ώστε όλα τα υπόλοιπα ψηφία να μετακινηθούν μία θέση προς τα δεξιά. Για παράδειγμα, εάν $v = 10010$ τότε $\pi(v) = 01001$. Οπότε, ένας γραμμικός κώδικας C θα λέγεται κυκλικός κώδικας, εάν η κυκλική μετάθεση σε κάθε μία από τις κωδικολέξεις του μας δίνει ξανά μια κωδικολέξη. Οι κώδικες QR που θα μελετήσουμε είναι κυκλικοί κώδικες.

1.14 Σώμα Galois

Μέχρι στιγμής έχουμε αναφερθεί στο σύνολο $K = \{0, 1\}$ και στο σύνολο K^n που παριστάνει το σύνολο όλων των δυαδικών λέξεων μήκους n. Αυτά τα σύνολα με τις ιδιότητες που έχουμε ορίσει στην Ενότητα 1.7 μπορούμε να τα δούμε ως πεπερασμένα σώματα, γνωστά και ως σώματα Galois. Ένα σώμα Galois (Galois field) είναι ένα πεπερασμένο σώμα του οποίου τα στοιχεία δεν είναι άπειρα. Συμβολίζεται με $\mathrm{GF}(p^k)$, όπου p πρώτος αριθμός και k θετικός ακέραιος. Στην παρούσα εργασία θα ασχοληθούμε μόνο για $p = 2$ και συγκεκριμένα όταν μελετήσουμε τον κώδικα QR για $k = 8$. Καθώς βρισκόμαστε στο δυαδικό σύστημα η πράξη της πρόσθεσης δύο στοιχείων γίνεται με modulo 2.Θα συμβολίζουμε την πρόσθεση modulo 2 δύο στοιχείων με $\oplus$. Για παράδειγμα, $001 \oplus 111 = 110$.

Ένα πολυώνυμο βαθμού n πάνω από το $GF(2)$ είναι ένα πολυώνυμο της μορφής

$\alpha_0 + \alpha_1 x + \cdots + \alpha_n x^n$ όπου οι συντελεστές του $\alpha_0, \ldots, \alpha_n$ είναι δυαδικά στοιχεία του συνόλου $GF(2)$. Για παράδειγμα, στο GF(2^8) το πολυώνυμο $x^4 + x^3 + x^2 + 1$ έχει τους συντελεστές $x^4, x^3, x^2, x^0 = 1$ και $x^7, x^6, x^5, x = 0$. Συνεπώς, η αντίστοιχη μορφή του στο δυαδικό σύστημα θα εκφράζεται ως 00011101, όπου το πρώτο bit από αριστερά θα λέγεται Most Significant Bit.

Ο βαθμός ενός πολυωνύμου $f(x)$ θα συμβολίζεται με $\deg(f(x))$. Ένα πολυώνυμο $f(x)$ βαθμού m λέγεται ανάγωγο εάν δε μπορεί να διαιρεθεί με οποιοδήποτε άλλο πολυώνυμο πάνω από το $GF(2)$ βαθμού μεγαλύτερου του μηδενός, αλλά μικρότερου του m. Ένα ανάγωγο πολυώνυμο πάνω από το $GF(2)$ βαθμού $n > 1$ λέγεται πρωταρχικό (primitive) εάν δεν είναι διαιρέτης του $1 + x^m$ για κάθε $m < 2^n - 1$. Θα δούμε ότι ένα ανάγωγο πολυώνυμο βαθμού n πάντοτε διαιρεί το $1 + x^m$ όταν $m = 2^n - 1$. Καλούμε το $\alpha \in GF(2^r)$ *πρωταρχικό* στοιχείο (primitive element) εάν $\alpha^m \neq 1$ για $1 \leq m < 2^r - 1$. Ισοδύναμα, το α λέγεται πρωταρχικό στοιχείο εάν κάθε μη-μηδενική λέξη στο $GF(2^r)$ μπορεί να εκφραστεί ως μια δύναμη του α.

Η χρήση ενός πρωταρχικού πολυωνύμου για την κατασκευή του $GF(2^r)$ διευκολύνει τους υπολογισμούς στο σώμα σε σύγκριση με ένα ανάγωγο μη-πρωταρχικό πολυώνυμο, καθώς τα στοιχεία του σώματος Galois μπορούν να παραχθούν. Εμείς θα χρειαστούμε το $GF(2^8) = GF(256)$ με πρωταρχικό πολυώνυμο το $x^8 + x^4 + x^3 + x^2 + 1$ για την δημιουργία του κώδικα QR, όμως καθώς η λογική είναι η ίδια, ας δούμε πως κατασκευάζεται το $GF(8)$. Στο $GF(2^3)$, λοιπόν, με πρωταρχικό πολυώνυμο το $h(x) = x^3 + x^2 + 1$ έχουμε:

Κωδικολέξη	Πολυώνυμο $x^i \bmod h(x)$	Δύναμη του α	Ακέραιος
000	0	—	0
001	1	$\alpha^0 = 1$	1
010	x	α	2
100	x^2	α^2	4
101	$1 + x^2$	α^3	5
111	$1 + x + x^2$	α^4	7
011	$1 + x$	α^5	3
110	$x + x^2$	α^6	6

Πίνακας 1.1: *Κατασκευή του $GF(8)$ με το $h(x) = x^3 + x^2 + 1$*

Σύμφωνα, λοιπόν, με τον παραπάνω πίνακα έχουμε καταφέρει να εκφράσουμε όλα τα στοιχεία του $GF(8)$ με μοναδικό τρόπο. Ας δούμε αναλυτικότερα πώς κατασκευάζεται ο παραπάνω πίνακας. Επιθυμούμε, λοιπόν, να εκφράσουμε κάθε κωδικολέξη ως μια δύναμη του πρωταρχικού στοιχείου α.

Η κάθε μία κωδικολέξη είναι της μορφής $x^2 x^1 x^0$ και προκύπτει από το υπόλοιπο της διαίρεσης $x^i \bmod h(x)$. Για παράδειγμα, το πολυώνυμο x^5 όταν διαιρεθεί με το πρωταρχικό πολυώνυμο $h(x) = x^3 + x^2 + 1$, θα δώσει υπόλοιπο $1 + x$. Συνεπώς, η κωδικολέξη έχει τη μορφή $0x^2 1x^1 1x^0$, δηλαδή θα προκύψει η 011. Αντιστοιχίζεται στην ακέραια τιμή 3, διότι $0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$.

Εν κατακλείδι, οι πληροφορίες που συλλέγουμε από το $x^5 \bmod h(x)$ είναι ότι αντιστοιχεί στην κωδικολέξη 011, η οποία αντιστοιχεί στην ακέραια τιμή 3, η οποία αντι-

στοιχεί στο α^5, δηλαδή $\alpha^5 = 3$. Να σημειώσουμε εδώ, ότι ο συμβολισμός $\alpha^i = n$ είναι σημαντικός, καθώς θα τον συναντήσουμε παρακάτω στον κώδικα QR στη διαδικασία παραγωγής των κωδικολέξεων διόρθωσης σφαλμάτων, όταν χρειαστούμε να πολλαπλασιάσουμε δύο στοιχεία του πεπερασμένου σώματος κατά τη διαδικασία διαίρεσης πολυωνύμων.

Στον κώδικα QR, το σώμα Galois που θα χρησιμοποιήσουμε θα είναι το $GF(256)$, το οποίο κατασκευάζεται παρόμοια όπως πράξαμε με το $GF(8)$. Επειδή στην ουσία θα χρειαστούμε μόνο την αντιστοιχία μεταξύ των a^i και των ακέραιων τιμών στο $GF(256)$, δίνουμε έτοιμο τον πίνακα B′.1 που βρίσκεται στο Παράρτημα Β.

1.15 Reed-Solomon

Οι κώδικες Reed-Solomon (ή εν συντομία RS) είναι γραμμικά μπλοκ κωδίκων και αποτελούν ένα υποσύνολο των κωδίκων BCH (κώδικες των Bose-Chaudhuri- Hocquenghem). Είναι οι πιο ισχυροί στην κατηγορία των γραμμικά μπλοκ κωδίκων, καθώς έχουν την ικανότητα να διορθώσουν τυχαία σφάλματα και πολλαπλά σφάλματα ριπών. Οι κώδικες RS είναι εν γένει μη-δυαδικοί κώδικες και κατασκευάζονται από ένα σώμα Galois με q στο πλήθος στοιχεία, GF(q). Όμως στο ψηφιακό κόσμο, ενδιαφέρον αποτελούν μόνο οι δυαδικοί κώδικες. Για το λόγο αυτό το σώμα GF(q) θα περιοριστεί στο GF(2^m), όπου m είναι ένας θετικός ακέραιος. Το πλήθος και το είδος των σφαλμάτων που μπορούν να διορθώσουν εξαρτάται από τα χαρακτηριστικά του κώδικα RS. Συμβολίζεται με $RS(n, k)$ και έχει τις ακόλουθες παραμέτρους:

- $n = 2^m - 1$: Μέγιστο μήκος της κάθε κωδικολέξης του κώδικα
- k : Μήκος του μηνύματος
- $(n - k) = 2r$: Πλήθος των bytes της κωδικολέξης που χρησιμοποιείται για την διόρθωση των σφαλμάτων κατά την αποκωδικοποίηση (λέγονται και parity σύμβολα)
- r : Μέγιστο πλήθος εσφαλμένων bytes που μπορεί να διορθωθούν κατά την αποκωδικοποίηση.

Το GF(2^m) θα παριστάνει το αλφάβητο μας, όπως θα δούμε παρακάτω στον κώδικα QR. Επιπλέον, τα στοιχεία του GF(2^m) θα εκφράζονται πλέον από ένα πολυώνυμο βαθμού $< 2^m$ με συντελεστές, πίνακες διάστασης $1 \times m$, δηλαδή διανύσματα μήκους m. Κάθε τέτοιο διάνυσμα θα αντιπροσωπεύει μια κωδικολέξη. Για παράδειγμα, έστω στο GF(2^8) το τυχαίο πολυώνυμο

$$P(x) = a_{255}x^{255} + a_{254}x^{254} + a_{253}x^{253} + \cdots + a_2x^2 + a_1x + a_0,$$

όπου οι συνελεστές μας είναι διανύσματα των 8-bits:

$$a_{255} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, a_{254} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \cdots \quad , a_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, a_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}.$$

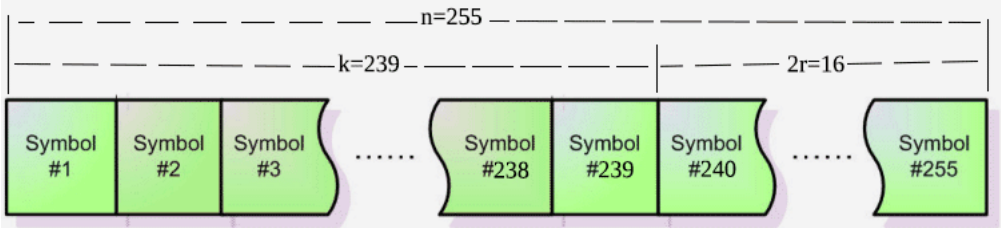
Το παραπάνω πολυώνυμο θα περιέχει ένα μήνυμα σε κωδικοποιημένη μορφή, π.χ. https://www.aegean.gr . Για τη δημιουργία των παραπάνω διανυσμάτων (κωδικολέξεων), πρέπει να βρούμε ένα τρόπο συσχέτισης του συνόλου των συμβόλων (σημεία στίξης) και των αγγλικών χαρακτήρων σε ένα σύνολο αριθμών δυαδικής μορφής. Η χρήση, λοιπόν, ενός πίνακα κωδικοποίησης χαρακτήρων, π.χ. ISO 8859-1, μας λύνει το πρόβλημα. Αυτός ο πίνακας κωδικοποίησης χαρακτήρων A′.1, όπως παρουσιάζεται στο Παράρτημα Α, περιέχει $2^8 = 256$ στοιχεία και συνεπώς θα αποτελεί το αλφάβητο μας. Για δική μας ευκολία, θα συμβολίζουμε τους συντελεστές του πολυωνύμου με τη δεκαδική μορφή αντί με τη μορφή διαδικών διανυσμάτων.

Ο RS είναι επίσης και συστηματικός κώδικας (systematic code) διότι τα δεδομένα μας (δηλ. το κωδικοποιημένο μήνυμά μας) βρίσκεται στα αριστερά της κωδικολέξης και το υπόλοιπό της έχει καλυφθεί με τα parity σύμβολα δηλ. τα ψηφία ελέγχου ισοτιμίας. Το παρακάτω σχήμα 1.3 απεικονίζει μια τέτοια συστηματική κωδικολέξη Reed-Solomon.

Ας δούμε τώρα ένα παράδειγμα ενός δημοφιλούς κώδικα Reed-Solomon. Έστω ο RS(255,239). Αποτελείται από 8-bit κωδικολέξεις, βρισκόμαστε στο GF(2^8) και έχουμε:

- $n = 255$: Μήκος κάθε κωδικολέξης.
- $k = 239$: Μήκος μηνύματος.
- $(n - k) = 2r = 16$.
- $r = 8$: Μέγιστο μήκος από bytes που μπορεί να διορθωθεί

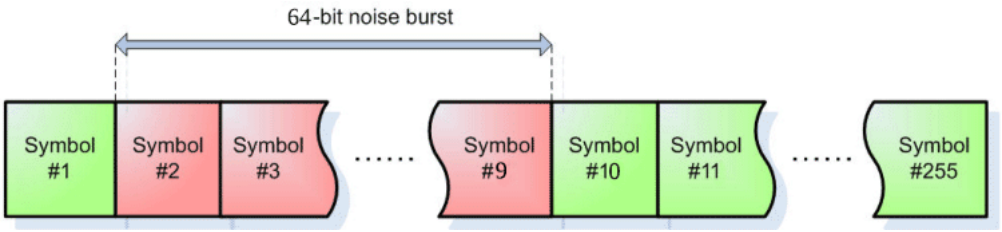
Ακολούθως δίνεται η γραφική αναπαράσταση του. Αναλυτικότερα, έχουμε μία κωδικολέξη μήκους n=255 bytes, η οποία σχηματίζεται από 255 επιμέρους σύμβολα (κωδικολέξεις). Από αυτές, οι 239 περιέχουν το μήνυμάς μας σε κωδικοποιήμενη μορφή και οι υπόλοιπες 16 στα δεξιά είναι οι κωδικολέξεις διόρθωσης σφαλμάτων.



Σχήμα 1.3: *Κωδικολέξη μήκους 255bytes με τη μέθοδο Reed-Solomon*

Επιπλέον, οι κώδικες Reed Solomon έχουν τη δυνατότητα να εκφραστούν σε μία πιο «σύντομη μορφή», δηλαδή να σμικρυνθούν, διατηρώντας όμως το ίδιο μέγεθος αλφαβήτου m. Αυτό επιτυγχάνεται μετατρέποντας ένα πλήθος από δεδομένα σε μηδενικά στον κωδικοποιητή, τα οποία όμως δε μεταδίδονται, και έπειτα ξαναεισάγονται στον αποκωδικοποιητή. Για παράδειγμα, ο κώδικας Reed Solomon RS(255,239) μεγέθους $m = 8$, μπορεί να σμυκρινθεί στον RS(44,28) διατηρώντας το ίδιο $m = 8$. Ο κωδικοποιητής παίρνει ένα μπλοκ από 28 κωδικολέξεις (bytes), προσθέτει 211 μηδενικά ψηφία (bits), δημιουργεί μια κωδικολέξη RS(255,239) και μεταδίδει μόνο τις 28 κωδικολέξεις και τις 16 κωδικολέξεις διόρθωσης σφαλμάτων. Επίσης, για τον κώδικα $RS(44, 28)$, που θα ασχοληθούμε μετέπειτα στον κώδικα QR, ο οποίος έχει $n = 44$, $k = 28$, $(n - k) = 16$ και $r = 8$, το αρχικό μας μήνυμα θα μπορέσει να ανακτηθεί με τη μέθοδο αποκωδικοποίησης RS μόνο εάν το πλήθος των κωδικολέξεων που αλλοιώθηκαν είναι μικρότερο ή ίσο του 8. Η σμύκρινση ενός κώδικα RS προκύπτει ανάλογα το μήκους του μηνύματος που θέλουμε να κωδικοποιήσουμε και είναι χρήσιμη αφού έχουμε τη δυνατότητα να γλυτώσουμε τόσο σε χρόνο όσο επίσης και σε υπολογιστή ισχύ.

Μία ακόμη σπουδαία ιδιότητα των κωδίκων Reed-Solomon που αναφέραμε και τώρα θα την μελετήσουμε αναλυτικότερα, είναι ότι λειτουργούν αρκετά αποτελεσματικά ενάντια στην ύπαρξη θορύβου. Έστω ο δημοφιλής κώδικας Reed-Solomon, RS(255,239), όπου κάθε κωδικολέξη (symbol) προκύπτει από $m = 8$ bits. Αφού όπως έχουμε δει, $n - k = 16 = 2 \cdot 8$, αυτό σημαίνει ότι ο τρέχων κώδικας Reed-Solomon έχει τη δυνατότητα να διορθώσει οποιαδήποτε σφάλματα προκύψουν σε 8 κωδικολέξεις. Ας υποθέσουμε ότι παρατηρείται μία ριπή θορύβου (noise burst), η οποία έχει μέγεθος 64-bit και αλλοιώνει μία κωδικολέξη, όπως φαίνεται στο παρακάτω σχήμα.



Σχήμα 1.4: *Κωδικολέξη μήκους 255bytes όπου έχουν αλλοιωθεί 8 κωδικολέξεις εξ αιτίας της ύπαρξης θορύβου.*

Παρατηρούμε μια ριπή θορύβου η οποία διαρκεί για 64 συνεχόμενα bits, αλλοιώνει ακριβώς 8 κωδικολέξεις. Ο αποκωδικοποιητής RS για τον κώδικα RS(255,239) θα καταφέρει να διορθώσει επιτυχώς και τις 8 αλλοιωμένες κωδικολέξεις, ανεξαρτήτως του μεγέθους αλλοίωσης που έχει υποστεί η καθεμία. Ας δούμε αναλυτικότερα τι εννοούμε με την έκφραση μέγεθος αλλοίωσης. Υπενθυμίζουμε ότι κάθε κωδικολέξη (byte) αποτελείται από $m = 8$ bits. Έστω οι κωδικολέξεις $s_1 = 10101010$ και $s_2 = 01010101$ οι οποίες εξαιτίας του θορύβου αλλοιώνονται και προκύπτουν οι αντίστοιχες $s'_1 = 0100001$ και $s'_2 = 11010101$. Παρατηρούμε ότι η s_1 αλλοιώθηκε σε μεγάλο βαθμό (6 bits), ενώ η s_2 σε πολύ μικρό βαθμό (1bit). Παρόλα αυτά, η μέθοδος Reed-Solomon θα καταφέρει να διορθώσει με απόλυτη επιτυχία τα αλλοιωμένα bits στις κωδικολέξεις s'_1, s'_2 και να επιστρέψει σαν αποτέλεσμα τις ορθές κωδικολέξεις s_1, s_2. Αυτός είναι και

ο λόγος που οι κώδικες Reed-Solomon είναι αρκετά δημοφιλής, χάρη στην εξαιρετική ικανότητα τους να διορθώνουν σφάλματα ριπών.

Κεφάλαιο 2

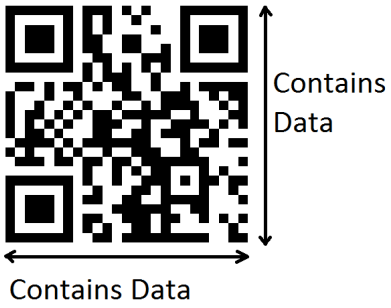
ΚΩΔΙΚΑΣ QR

2.1 Γενικές πληροφορίες και χρησιμότητά του

Ένας κώδικας γρήγορης ανταπόκρισης (Quick Response Code ή QR Code) είναι ένα σύμβολο ή ισοδύναμα ένα barcode δύο διαστάσεων που εφευρέθηκε από την Ιαπωνική εταιρία Denso Wave. Οι πληροφορίες μπορούν να κωδικοποιηθούν και στην οριζόντια και στην κάθετη κατεύθυνση, επιτρέποντας την αποθήκευση πολλών περισσότερων δεδομένων σε αντίθεση με ένα παραδοσιακό barcode.

Σχήμα 2.1: *Παραδοσιακό Barcode*

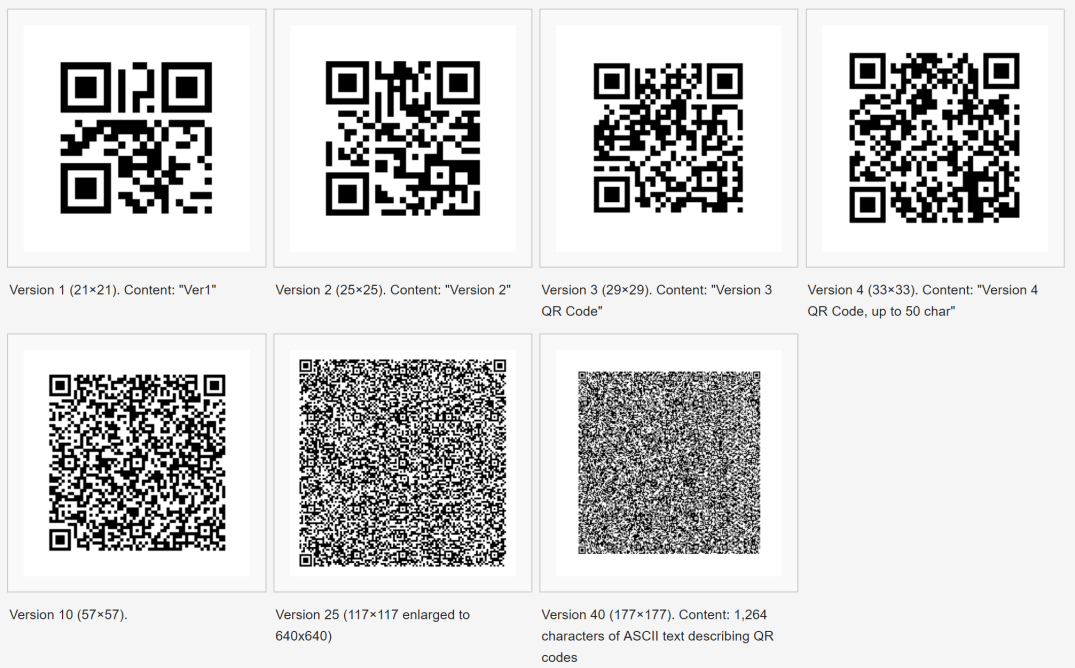
Τα δεδομένα μπορούν να διαβαστούν πολύ απλά από την κάμερα του smartphone μας και να αποκωδικοποιηθούν από κάποια εφαρμογή QR reader. Για να είναι ευανάγνωστος ένας κώδικας QR θα πρέπει να υπάρχει αντίθεση μεταξύ των κελιών που περιέχουν το ψηφίο 1 και αυτών που περιέχουν το ψηφίο 0. Κατά καιρούς έχουν δημιουργηθεί πολλές παραλλαγές τους, όμως εμείς θα μελετήσουμε τον κλασικό κώδικα QR και συγκεκριμένα την Version 2.



Σχήμα 2.2: *Κώδικας QR Version 2*

Οι κώδικες QR είναι χρήσιμοι καθώς μπορούν να μας διευκολύνουν σε πάρα πολλούς τομείς της καθημερινότητάς μας. Για παράδειγμα, μπορούμε να εξοφλήσουμε τους λογαριασμούς μας εύκολα, γρήγορα και με ασφάλεια, σαρώνοντας απλώς τον κώδικα QR που βρίσκεται σε αυτούς, ο οποίος περιέχει τη ταυτότητα οφειλής. Επίσης, ένας κώδικας QR μπορεί να αποτελέσει το εισιτήριο επιβίβασης μας σε κάποιο μέσο μεταφοράς, όπου η εγκυρότητα του επιβεβαιώνεται όταν σαρώνεται κατά την επιβίβασή μας.

Η πιο σημαντική και επίκαιρη χρησιμότητά του, κατά τη γνώμη μου, είναι ότι θα χρησιμοποιηθεί στα ψηφιακά πράσινα πιστοποιητικά για τον covid-19. Το ψηφιακό πράσινο πιστοποιητικό θα περιέχει έναν κωδικό QR με ψηφιακή υπογραφή για την προστασία του από τυχόν πλαστογράφηση. Κατά τον έλεγχο ενός πιστοποιητικού, θα γίνεται σάρωση του κωδικού QR και επαλήθευση της υπογραφής. Όλοι οι φορείς έκδοσης (π.χ. νοσοκομεία, κέντρα εξετάσεων, υγειονομικές αρχές) θα διαθέτουν το δικό τους κλειδί ψηφιακής υπογραφής. Όλα αυτά θα αποθηκεύονται σε ασφαλή βάση δεδομένων σε κάθε χώρα. Η Ευρωπαϊκή Επιτροπή θα δημιουργήσει μια πύλη μέσω της οποίας όλες οι υπογραφές των πιστοποιητικών θα μπορούν να επαληθευτούν παντού στην ΕΕ. Τα κωδικοποιημένα προσωπικά δεδομένα που περιέχει το πιστοποιητικό δεν μεταβιβάζονται μέσω της πύλης, αφού αυτό δεν είναι αναγκαίο για την επαλήθευση της ηλεκτρονικής υπογραφής. Η Επιτροπή θα βοηθήσει επίσης τα κράτη μέλη να αναπτύξουν λογισμικό το οποίο θα μπορούν να χρησιμοποιούν οι αρχές για τον έλεγχο των κωδικών QR.[1]



Σχήμα 2.3: *Κάποιες από τις 40 versions του κώδικα QR.*

Υπάρχουν 40 εκδόσεις (versions) του κώδικα QR. Όσο μεγαλύτερη η έκδοση, τόσο πυκνότερος ο κώδικας QR και η δυνατότητα αποθήκευσης περισσότερων πληροφο-

[1] https://ec.europa.eu/info/live-work-travel-eu/coronavirus-response/safe-covid-19-vaccines-europeans/covid-19-digital-green-certificates_el

ριών. Ένας κώδικας QR version 40 μπορεί να αποθηκεύσει έως 7089 χαρακτήρες (ψηφία, σύμβολα, γράμματα από οποιοδήποτε αλφάβητο), ενώ ένα barcode μπορεί έως και 20 ψηφία. Στατιστικά, οι κώδικες QR έχουν τη δυνατότητα να κωδικοποιήσουν την ίδια ποσότητα δεδομένων χρησιμοποιώντας περίπου σε χώρο το ένα δέκατο σε σχέση με ένα παραδοσιακό barcode. Κάποια ακόμη πλεονεκτήματα ενός κώδικας QR είναι ότι μπορεί να σαρωθεί από οποιαδήποτε γωνία και ακόμη να σαρωθεί επιτυχώς εάν υπάρξει κάποια αλλοίωση του, όπως παρουσιάζονται στο παρακάτω σχήμα. Ο αριστερός κώδικας QR έχει αλλοιωθεί εσκεμμένα, καθώς του προσθέσαμε το σύμβολο του πανεπιστημίου Αιγαίου. Ο μεσαίος κώδικας έχει παραμορφωθεί. Ο δεξιός κώδικας είναι όπως φαίνεται από την πίσω όψη. Παρόλα αυτά, και στις τρεις περιπτώσεις, όταν σαρωθεί θα μας δώσει το μήνυμα *https://www.aegean.gr*.

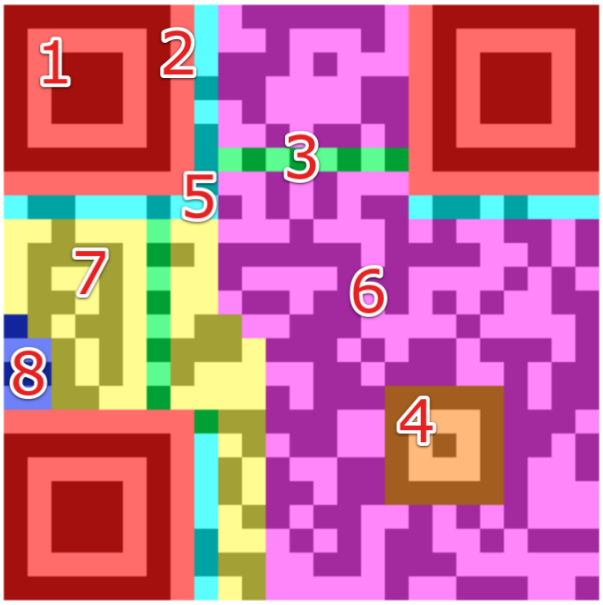
Σχήμα 2.4: *Αλλοιωμένοι κώδικες QR που σαρώνονται επιτυχώς*

Αυτό είναι εφικτό καθώς οι QR readers έχουν τη δυνατότητα να προσδιορίσουν το σωστό τρόπο αποκωδικοποίησης της εικόνας, χάρη στα τρία συγκεκριμένα τετράγωνα που βρίσκονται στις γωνίες του συμβόλου και τα μοτίβα ευθυγράμμισης που θα δούμε παρακάτω. Ένας κώδικας QR ποτέ δεν πρόκειται να δώσει λανθασμένο μήνυμα. Αντιθέτως, εάν κατά τη διάρκεια της σάρωσης του δεν επιστρέφεται κάποιο αποτέλεσμα, αυτό δηλώνει την αποτυχία αποκωδικοποίησης του.

2.2 Δομή ενός κώδικα QR

Οι κώδικες QR αποτελούνται στις περιπτώσεις από μαύρα και άσπρα τετραγωνάκια που ονομάζονται κελιά (modules). Κάθε ένα κελί παριστάνει ένα ψηφίο (bit) στο δυαδικό σύστημα αρίθμησης. Εάν το κελί είναι άσπρο ισούται με το ψηφίο 0, ενώ αν είναι μαύρο ισούται με το ψηφίο 1. Αυτά τα bits ομαδοποιούνται σε bytes, όπου κάθε byte αποτελείται από 8 bits. Κάθε τέτοιο byte θα αποτελεί μία κωδικολέξη. Ο κώδικας QR αποτελείται από διάφορες περιοχές οι οποίες είναι κατοχυρωμένες για συγκεκριμένες σκοπούς. Στο παρακάτω σχήμα αναφερόμαστε στην version 2 ενός κώδικα QR, επειδή η version 1 δεν περιέχει όλα τα μοτίβα.

- **Μοτίβο Εύρεσης (1):** Το μοτίβο εύρεσης (finder pattern) αποτελείται από τρεις ίδιες δομές (τετράγωνα) τα οποία βρίσκονται σε κάθε γωνία του κώδικα QR εκτός από την κάτω δεξιά γωνία. Κάθε μοτίβο εύρεσης κατασκευάζεται από έναν 3×3



Σχήμα 2.5: *Δομή ενός QR Code Version 2*

πίνακα από μαύρα κελιά (black modules), δηλαδή έχουμε το μικρό μαύρο τετραγωνάκι με πλήθος 9 κελιών. Κάθε τέτοιο τετραγωνάκι περιβάλλεται από άσπρα κελιά σχηματίζοντας ένα ακόμη άσπρο τετράγωνο το οποίο περιβάλλεται επίσης από μαύρα κελιά. Οπότε προκύπτει ένα τετράγωνο 7×7. Συνεπώς, το πλήθος των κελιών και των τριών τετραγώνων ισούται με $49 \times 3 = 147$. Τα μοτίβα εύρεσης είναι χρήσιμα καθώς δίνουν τη δυνατότητα στον QR reader να αναγνωρίσει ότι σαρώνει έναν κώδικα QR και να προσδιορίσει το σωστό προσανατολισμό.

- **Διαχωριστές (2):** Οι διαχωριστές (seperators) είναι κελιά με άσπρο χρώμα και βοηθούν στον εντοπισμό των μοτίβων εύρεσης καθώς τα διαχωρίζουν από τα πραγματικά δεδομένα, δηλαδή τις πληροφορίες που περιέχει ο κώδικας QR. Το πλήθος τους ισούται με 45 κελιά.

- **Χρονικό Μοτίβο (3):** Το χρονικό μοτίβο (timing pattern) αποτελείται από εναλλασσόμενα άσπρα και μαύρα κελιά. Ο σκοπός του είναι να βοηθήσει τον QR reader να προσδιορίσει το πλάτος του κελιού. Το πλήθος των κελιών εξαρτάται από το μέγεθος του κώδικα QR και δίνεται από το τύπο $2(N-16)$ όπου N είναι το μέγεθος $N \times N$ του κώδικα QR. Για το συγκεκριμένο σχήμα έχουμε $N = 25$.

- **Μοτίβα Ευθυγράμμισης (4):** Τα μοτίβα ευθυγράμμισης (alignment patterns) βοηθούν τον QR reader να εντοπίσει και να διορθώσει τυχόν στρεβλώσεις έχουν προκύψει. Οι κώδικες QR Version 1 δεν έχουν μοτίβα ευθυγράμμισης. Εμείς θα μελετήσουμε κώδικες QR Version 2 όπου είναι ο μικρότερος κώδικας με μοτίβο ευθυγράμμισης. Μέχρι και την Version 6 χρησιμοπιείται ένα μοτίβο ευθυγράμμισης. Από την Version 7 και έπειτα, χρησιμοποιούνται τουλάχιστον 6 μοτίβα ευθυγράμμισης. Το πλήθος των κελιών ενός μοτίβου ευθυγράμμισης ισούται με $5\times5=25$ κελιά.

- **Πληροφορίες Μορφοποίησης (5):** Οι πληροφορίες μορφοποίησης (format information) αποτελούνται από μία κωδικολέξη μήκους 15 bits η τοποθετείται δίπλα από τους διαχωριστές. Είναι χρήσιμες καθώς περιέχουν πληροφορίες σχετικά με το επίπεδο διόρθωσης σφαλμάτων και του επιλεγμένου μοτίβου μάσκας του κώδικα QR. Το πλήθος των κελιών τους ισούται με 31. Το ένα κελί αντιστοιχεί στο λεγόμενο Dark Module. Στα υπόλοιπα 30 κελιά τοποθετείται δύο φορές η κωδικολέξη μήκους 15bits. Περισσότερες λεπτομέρειες θα δούμε μετέπειτα.
- **Περιοχή Δεδομένων (6):** Στην περιοχή δεδομένων (data area) οι πληροφορίες μετατρέπονται σε μια ακολουθία από bit και αποθηκεύονται σε ομάδες όπου η κάθε μια ομάδα αποτελείται από 8 bit (τις λεγόμενες κωδικολέξεις).
- **Διόρθωση Σφαλμάτων (7):** Όμοια με την περιοχή δεδομένων, η διόρθωση σφαλμάτων (error correction) αποθηκεύεται σε κωδικολέξεις μήκους 8 bit.
- **Υπολειπόμενα Bits (8):** Αυτή η περιοχή αποτελείται από άδεια bits εάν τα bits των δεδομένων και της διόρθωσης σφαλμάτων δε μπορούν να διαιρεθούν σε κωδικολέξεις των 8 bit χωρίς υπόλοιπο.
- **Ήσυχη Ζώνη:** Ολόκληρος ο κώδικας QR θα πρέπει να περιβάλλεται από μία ήσυχη ζώνη (quiet zone), μια περιοχή από τουλάχιστον τέσσερα λευκά κελιά ώστε να διευκολυνθεί ο QR reader στην ανάγνωση του κώδικα.

Σύμφωνα με το παραπάνω Σχήμα 2.5, τα (1), (2), (3), (4), (5) λέγονται *μοτίβα λειτουργιών* (function patterns) καθώς μας δίνουν πληροφορίες για τρόπο κατασκευής και λειτουργίας του συμβόλου QR. Τα (6), (7), (8) αποτελούν *περιοχή κωδικοποίησης* (encoding region) καθώς εμπεριέχουν το περιεχόμενο του μηνύματος και τις απαραίτητες πληροφορίες για τη διόρθωση σφαλμάτων. Γνωρίζοντας τα παραπάνω, μπορούμε να συμπεράνουμε ότι το πλήθος των κελιών των μοτίβων λειτουργιών ενός συμβόλου QR Version 2 ισούται σύμφωνα με το παρακάτω πινακάκι

Μοτίβα λειτουργιών	κελιά
Μοτίβο εύρεσης	147
Διαχωριστές	45
Χρονικό μοτίβο	18
Μοτίβα ευθυγράμμισης	25
Πληροφορίες μορφοποίησης	31
Σύνολο	266

2.3 Χωρητικότητα και διόρθωση σφαλμάτων

Η χωρητικότητα ενός κώδικα QR εξαρτάται από κάποιους παράγοντες, όπως την έκδοση (version) η οποία καθορίζει το μέγεθός του (δηλαδή το πλήθος των κελιών), το επίπεδο διόρθωσης σφαλμάτων και το τύπο των κωδικοποιημένων δεδομένων.

Υπάρχουν 40 διαφορετικές εκδόσεις κωδίκων QR οι οποίες κυρίως διαφέρουν στο πλήθος των κελιών. Το μέγεθος μιας version V είναι ένας $N \times N$ πίνακας, όπου $N =$

$17+4V$. Επομένως, έχουμε ότι η Version 1 αποτελείται από 21×21 κελιά (modules) και έως 133 κελιά (εάν επιλεγεί το χαμηλότερο επίπεδο διόρθωσης σφαλμάτων) τα οποία μπορούν να αποθηκεύσουν τα κωδικοποιημένα δεδομένα. Ο μεγαλύτερος κώδικας QR (Version 40) έχει μέγεθος 177×177 κελιών και μπορεί να αποθηκεύσει κωδικοποιημένα δεδομένα σε έως και 23.648 κελιά.

Τα σύμβολα QR χρησιμοποιούν τους κώδικες Reed-Solomon για την ανίχνευση και τη διόρθωση σφαλμάτων. Μια σειρά από κωδικολέξεις διόρθωσης σφαλμάτων παράγεται, οι οποίες προστίθενται μετά το κωδικοποιημένο μήνυμά μας με σκοπό να συμβάλουν στην ακεραιότητα των δεδομένων όταν το σύμβολο QR υποστεί κάποια ζημιά. Στο παρακάτω πινακάκι βλέπουμε τα τέσσερα επίπεδα διόρθωσης σφαλμάτων από τα οποία διαλέγει ένα ο χρήστης κατά τη δημιουργία του κώδικα QR.

Επίπεδο L (Low)	≈7%
Επίπεδο M (Medium)	≈15%
Επίπεδο Q (Quartile)	≈25%
Επίπεδο H (High)	≈30%

Επιλέγοντας υψηλότερο επίπεδο διόρθωσης σφαλμάτων αυξάνουμε την πιθανότητα να διορθώσουμε τις εσφαλμένες κωδικολέξεις, όμως μειώνουμε το πλήθος των δεδομένων που θα μπορέσουν να αποθηκευτούν μέσα στον κώδικα.

Οι κώδικες QR μπορούν να χρησιμοποιήσουν διαφορετικούς τρόπους κωδικοποίησης. Η πολυπλοκότητα τους επηρεάζει αρνητικά το πλήθος των χαρακτήρων που θα μπορέσουν να αποθηκευτούν μέσα στον κώδικα, δηλαδή στην περιοχή δεδομένων. Για πράδειγμα, ένας κώδικας QR Version 2 με το χαμηλότερο επίπεδο διόρθωσης σφαλμάτων μπορεί να αποθηκεύσει έως 77 αριθμητικούς χαρακτήρες, αλλά μόνο 10 Kanji χαρακτήρες.

Για έναν κώδικα QR Version 2 γνωρίζουμε ότι το μέγεθός του είναι $25\times25=625$ κελιά. Παραπάνω είδαμε ότι το πλήθος των κελιών των μοτίβων λειτουργιών ισούται με 266 κελιά. Συνεπώς, μπορούμε να υπολογίσουμε το πλήθος των κελιών της περιοχής κωδικοποίησης ως:

$$\begin{aligned}(\text{Πλήθος κελιών συμβόλου QR}) =&(\text{Πλήθος κελιών μοτίβων λειτουργιών})\\ &+(\text{Πλήθος κελιών περιοχής κωδικοποίησης}),\end{aligned}$$

οπότε προκύπτει ότι η περιοχή κωδικοποίησης αποτελείται από 359 κελιά. Αυτό σημαίνει ότι έχουμε διαθέσιμα 359bits για να αποθηκεύσουμε το μήνυμα μας και να κατασκευάσουμε τη διόρθωση σφαλμάτων χρησιμοποιώντας τον κώδικα Reed-Solomon. Ας θυμηθούμε όμως ότι μία κωδικολέξη αποτελείται από 8bits=1byte. Άρα το σύνολο των κωδικολέξεων μας θα ισούται με το ακέραιο μέρος του αποτελέσματος της διαίρεσης του πλήθους των κελιών της περιοχής κωδικοποίησης με τον αριθμό 8, δηλαδή έχουμε

$$\frac{359}{8} = 44 \text{ bytes ή ισοδύναμα 44 κωδικολέξεις.}$$

Παρατηρούμε ότι η παραπάνω διαίρεση δεν είναι τέλεια, καθώς παίρνουμε υπόλοιπο ίσο με 7. Αυτά τα 7bits λοιπόν τοποθετούνται στη θέση (8) σύμφωνα με το Σχήμα 2.5.

Ο παρακάτω πίνακας μας δίνει μια συγκεντρωτική εικόνα.

Version	Επίπεδο διόρθωσης σφαλμάτων	Πλήθος κωδικολέξεων	Πλήθος κωδικολέξεων μηνύματος	Πλήθος κωδικολέξεων διόρθωσης σφαλμάτων
2	L	44	34	10
2	M	44	28	16
2	Q	44	22	22
2	H	44	16	28

Να σημειώσουμε εδώ ότι στο Πλήθος κωδικολέξεων μηνύματος έχουμε συμπεριλάβει τις επιπλέον δύο κωδικολέξεις, *Δείκτης λειτουργίας* (Mode Indicator) και *Δείκτης αρίθμησης χαρακτήρων* (Character Count Indicator). Για τον υπολογισμό του μέγιστου επιτρεπτού πλήθους των κωδικολέξεων του μηνύματός μας θα πρέπει να λάβουμε υπόψιν μας τους παραπάνω δείκτες. Για παράδειγμα, το μήνυμα *https://www.aegean.gr* αποτελείται από 21 χαρακτήρες οι οποίοι όπως θα δούμε παρακάτω μπορούν να μετατραπούν σε 21 κωδικολέξεις. Συνεπώς, έχουμε τη δυνατότητα να χρησιμοποιήσουμε μόνο δύο από τα τέσσερα διαθέσιμα επίπεδα διόρθωσης σφαλμάτων, είτε το L είτε το M. Ο Δείκτης λειτουργίας προσδιορίζει ποια λειτουργία για την κωδικοποίηση χαρακτήρων θα χρησιμοποιηθεί. Εμείς θα μελετήσουμε τη λειτουργία Byte (Byte mode) η οποία αντιστοιχεί στη δυαδική τιμή 0100_2. Ο Δείκτης αρίθμησης χαρακτήρων προσδιορίζει το σύνολο των χαρακτήρων του μηνύματός μας και εκφράζεται σε δυαδική μορφή μήκους 8bit[3] , π.χ. για 21 χαρακτήρες έχουμε 00010101.

2.4 Διαδικασία κωδικοποίησης μηνύματος

Σε αυτή την ενότητα θα δούμε τα βήματα που χρειάζεται να ακολουθήσουμε ώστε να μετατρέψουμε το μήνυμα μας https://www.aegean.gr σε κωδικοποιημένη μορφή με σκοπό να το εισάγουμε σε έναν κώδικα QR.

Βήμα 1-Ανάλυση δεδομένων: Μελετούμε τα δεδομένα που θέλουμε να κωδικοποιήσουμε και την κατάλληλη λειτουργία εισαγωγής πληροφοριών σύμφωνα με τις ανάγκες μας. Οι πιο γνωστές είναι:

- Αριθμητική λειτουργία: Ακέραιοι αριθμοί 0 έως και 9. Μπορούν να αποθηκευτούν έως και 7,089 χαρακτήρες.
- Αλφαριθμητική λειτουργία: Ακέραιοι αριθμοί 0 έως και 9, κεφαλαία γράμματα αγγλικού αλφαβήτου A-Z, σύμβολα κενό$%+-./:) Μπορούν να αποθηκευτούν έως και 4,296 χαρακτήρες.
- Byte λειτουργία: Χαρακτήρες αντλούνται από έναν πίνακα χαρακτήρων 8-bit (ISO 8859-1). Μπορούν να αποθηκευτούν έως και 2,953 χαρακτήρες.

[2]ISO/IEC 18004:2015 σελ.23 Table 2
[3]ISO/IEC 18004:2015 σελ.23 Table 3

Εμείς θα ασχοληθούμε μόνο με τον πίνακα χαρακτήρων 8bit και θα επιλέξουμε τον πίνακα χαρακτήρων ISO 8859-1. Επίσης, σε αυτό το βήμα επιλέγουμε το επιθυμητό επίπεδο ανίχνευσης και διόρθωσης σφαλμάτων από τα τέσσερα που είναι διαθέσιμα (L,M,Q,H), ανάλογα τις ανάγκες μας.

Βήμα 2-Κωδικοποίηση δεδομένων: Μετατρέπουμε το μήνυμά μας, δηλαδή τους χαρακτήρες, σε μια ακολουθία από bits. Για να το επιτύχουμε αυτό, πρέπει να συμβουλευτούμε τον πίνακα A′.1. Κάθε χαρακτήρας της στήλης Char (Character) αντιστοιχίζεται σε έναν μοναδικό αριθμό σε δεκαδική μορφή της στήλης Dec (Decimal). Παρατηρούμε ότι σύνολο των χαρακτήρων του ισούται με $256 = 2^8$. Επομένως, κάθε χαρακτήρας μέσω του αντίστοιχου δεκαδικού αριθμού μπορεί να μετατραπεί σε δυαδική μορφή, δηλαδή σε μία κωδικολέξη μήκους 8bit. Ας δοκιμάσουμε λοιπόν να κωδικοποιήσουμε το μήνυμα *https://www.aegean.gr* Ο πρώτος χαρακτήρας μας, *h*, αντιστοιχίζεται στον αριθμό 104. Οπότε πρέπει να τον μετατρέψουμε σε δυαδική μορφή. Έχουμε:

$$104 = (0\cdot 2^7)+(1\cdot 2^6)+(1\cdot 2^5)+(0\cdot 2^4)+(1\cdot 2^3)+(0\cdot 2^2)+(0\cdot 2^1)+(0\cdot 2^0) = 01101000.$$

Άρα η πρώτη κωδικολέξη του μηνύματός μας είναι η 01101000. Όμοια μετατρέπουμε και τους υπόλοιπους χαρακτήρες σε κωδικολέξεις. Έτσι λοιπόν, το μήνυμά μας πλέον μπορεί να εκφραστεί με τις παρακάτω κωδικολέξεις σε μία ακολουθία από ψηφία (bits) μήκους 8 η κάθε μία. Το τέλος του κωδικοποιημένου μηνύματός μας σηματοδοτείται με τη χρήση μιας Ακολουθίας τερματισμού (Terminator Sequence) από τέσσερα στο πλήθος 0 ψηφία. Προσθέτοντας στην αρχή τον Δείκτη λειτουργίας, έπειτα τον Δείκτη αρίθμησης χαρακτήρων, έπειτα το μήνυμάς μας και τέλος την Ακολουθία τερματισμού, προκύπτει η ακόλουθη αλληλουχία bit:

0100	00010101	01101000	01110100	01110100	01110000	01110011	00111010
Δ. λειτ.	Δ. αριθ. χαρ.	h	t	t	p	s	:

00101111	00101111	01110111	01110111	01110111	00101110	01100001
/	/	w	w	w	.	a

01100101	01100111	01100101	01100001	01101110	00101110	01100111
e	g	e	a	n	.	g

01110010	0000
r	Ακ. Τερμ.

Η παραπάνω αλληλουχία bit αποτελείται στο πλήθος από 184 ψηφία ή ισοδύναμα από 23 κωδικολέξεις. Επειδή πρέπει το πλήθος των κωδικολέξεων μηνύματος να ισούται με 28, χρειαζόμαστε ακόμη 5 κωδικολέξεις. Αυτό επιτυγχάνεται με την προσθήκη Pad κωδικολέξεων (Pad Codewords) *11101100* και *00010001* εναλλάξ[4]. Έτσι λοιπόν παίρνουμε την παρακάτω αλληλουχία bit μήκους 224 ψηφίων ή ισοδύναμα 28 κωδικο-

[4]ISO/IEC 18004:2015 Ενότητα 7.4.10

λέξεων.

0100	00010101	01101000	01110100	01110100	01110000	01110011	00111010
Δ. λειτ.	Δ. αριθ. χαρ.	h	t	t	p	s	:
00101111	00101111	01110111	01110111	01110111	00101110	01100001	
/	/	w	w	w	.	a	
01100101	01100111	01100101	01100001	01101110	00101110	01100111	
e	g	e	a	n	.	g	
01110010	0000	11101100	00010001	11101100	00010001	11101100	
r	Ακ. Τερμ.	Pad κωδ.	Pad κωδ.	Pad κωδ.	Pad κωδ.	Pad κωδ.	

Να υπενθυμίσουμε σε αυτό το σημείο ότι κάθε κωδικολέξη μας πρέπει να έχει μήκος 8-bit. Επειδή ο Δείκτης λειτουργίας έχει μήκος 4-bit, για να σχηματιστεί η πρώτη κωδικολέξη θα πρέπει να χρησιμοποιηθούν ακόμη 4-bit από το Δείκτη αρίθμησης χαρακτήρων. Οπότε η πρώτη κωδικολέξη θα είναι η 01000001 η οποία αντιστοιχεί στον δεκαδικό αριθμό 65. Σχηματίζοντας, λοιπόν, και τις υπόλοιπες 27 κωδικολέξεις θα προκύψει η παρακάτω αλληλουχία bit

01000001	01010110	10000111	01000111	01000111	00000111	00110011
65	86	135	71	71	7	51
10100010	11110010	11110111	01110111	01110111	01110010	11100110
162	242	247	119	119	114	230
00010110	01010110	01110110	01010110	00010110	11100010	11100110
22	86	118	86	22	226	230
01110111	00100000	11101100	00010001	11101100	00010001	11101100
119	32	236	17	236	17	236

οι οποίοι θα είναι οι συντελεστές του πολυωνύμου μηνύματος $M(x)$. Το $M(x)$ έχει τη μορφή [9]

$$M(x) = m_{k-1}x^{k-1} + m_{k-2}x^{k-2} + \cdots + m_1 x + m_0 \tag{2.1}$$

όπου k το πλήθος κωδικολέξεων του μηνύματος όπως ορίζεται σύμφωνα με το αντίστοιχο επίπεδο διόρθωσης σφαλμάτων και την έκδοση του QR που αποφασίσαμε στο προηγούμενο βήμα. Οπότε το πολυώνυμο μηνύματος μας θα είναι το

$$\begin{aligned} M(x) =& 65x^{27} + 86x^{26} + 135x^{25} + 71x^{24} + 71x^{23} + 7x^{22} + 51x^{21} \\ &+ 162x^{20} + 242x^{19} + 247x^{18} + 119x^{17} + 119x^{16} + 114x^{15} + 230x^{14} + 22x^{13} \\ &+ 86x^{12} + 118x^{11} + 86x^{10} + 22x^9 + 226x^8 + 230x^7 + 119x^6 + 32x^5 \\ &+ 236x^4 + 17x^3 + 236x^2 + 17x + 236. \end{aligned}$$

Βήμα 3-Παραγωγή κωδικολέξεων διόρθωσης σφαλμάτων:

Όπως έχουμε αναφέρει και προηγουμένως, οι κώδικες QR χρησιμοποιούν τη μέθοδο Reed-Solomon για την ανίχνευση και τη διόρθωση σφαλμάτων.

Οι κωδικολέξεις διόρθωσης σφαλμάτων μπορούν να διορθώσουν δύο είδη εσφαλμένων κωδικολέξεων, τις διαγραφές (erasures) και τα σφάλματα (errors). Ως διαγραφή

θεωρείται ένας χαρακτήρας συμβόλου που δε σαρώθηκε ή δεν αποκωδικοποιήθηκε. Ως σφάλμα θεωρείται ένας χαρακτήρας συμβόλου που αποκωδικοποιήθηκε λανθασμένα. Επειδή οι κώδικες QR αποτελούνται από άσπρα και μαύρα κελιά, η λανθασμένη μετατροπή ενός άσπρου κελιού σε μαύρο και αντίστροφα, θα έχει ως αποτέλεσμα τη λανθασμένη αποκωδικοποίησή του η οποία όμως θα αποτελεί μια έγκυρη κωδικολέξη.

Οι κωδικολέξεις διόρθωσης σφαλμάτων προκύπτουν από τη σχέση

$$E(x) = x^{2r}M(x) \bmod g(x), \tag{2.2}$$

όπου $2r$ είναι το πλήθος κωδικολέξεων διόρθωσης σφαλμάτων και $g(x)$ είναι ένα πολυώνυμο γεννήτορας[9]. Η επιλογή του $g(x)$ εξαρτάται από το πλήθος κωδικολέξεων διόρθωσης σφαλμάτων. Εν γένει, υπάρχουν 36 πολυώνυμα γεννήτορες για τον κώδικα QR[5]. Στη δική μας περίπτωση, QR version 2-M, το πλήθος κωδικολέξεων διόρθωσης σφαλμάτων πρέπει να ισούται με 16. Οπότε έχουμε:

$$\begin{aligned} g(x) =& x^{16} + \alpha^{120}x^{15} + \alpha^{104}x^{14} + \alpha^{107}x^{13} + \alpha^{109}x^{12} + \alpha^{102}x^{11} + \alpha^{161}x^{10} \\ &+ \alpha^{76}x^{9} + \alpha^{3}x^{8} + \alpha^{91}x^{7} + \alpha^{191}x^{6} + \alpha^{147}x^{5} + \alpha^{169}x^{4} \\ &+ \alpha^{182}x^{3} + \alpha^{194}x^{2} + \alpha^{225}x + \alpha^{120} \end{aligned}$$

όπου α είναι πρωταρχικό στοιχείο του GF(256).

Για την εύρεση των κωδικολέξεων διόρθωσης σφαλμάτων απαραίτητη είναι η διαίρεση πολυωνύμων σύμφωνα με τη σχέση (2.2).Το υπόλοιπο που θα προκύψει από την παραπάνω διαίρεση θα είναι ένα πολυώνυμο βαθμού ($2r$-1), έστω το $r(x)$, και θα περιέχει όλες τις κωδικολέξεις διόρθωσης σφαλμάτων. Ο συντελεστής του μεγαλύτερου βαθμού του $r(x)$ θα είναι η πρώτη κωδικολέξη διόρθωσης σφαλμάτων και αντίστοιχα ο συντελεστής του μικρότερου βαθμού (μηδενικού βαθμού) θα είναι η τελευταία κωδικολέξη διόρθωσης σφαλμάτων.

Συνεπώς, από τη σχέση (2.2) προκύπτει το πολυώνυμο

$$\begin{aligned} E(x) =& 144x^{15} + 213x^{14} + 13x^{13} + 21x^{12} + 99x^{11} + 156x^{10} + 151x^{9} \\ &+ 30x^{8} + 83x^{7} + 73x^{6} + 36x^{5} + 204x^{4} + 47x^{3} + 6x^{2} + 35x + 141. \end{aligned}$$

Οπότε οι 16 κωδικολέξεις διόρθωσης σφαλμάτων είναι οι συντελεστές του $E(x)$. Η δυαδική τους μορφή είναι η ακόλουθη:

$$\underbrace{10010000}_{144}\,\underbrace{11010101}_{213}\,\underbrace{00001101}_{13}\,\underbrace{00010101}_{21}\,\underbrace{01100011}_{99}\,\underbrace{10011100}_{156}\,\underbrace{10010111}_{151}$$

$$\underbrace{00011110}_{30}\,\underbrace{01010011}_{83}\,\underbrace{01001001}_{73}\,\underbrace{00100100}_{36}\,\underbrace{11001100}_{204}\,\underbrace{00101111}_{47}\,\underbrace{00000110}_{6}$$

$$\underbrace{00100011}_{35}\,\underbrace{10001101}_{141}$$

Βήμα 4-Κατασκευή τελικού μηνύματος Εφόσον έχουμε βρει και μετατρέψει τις κωδικολέξεις διόρθωσης σφαλμάτων σε ψηφία μήκους 8, μπορούμε να κατασκευάσουμε το

[5]Περισσότερες πληροφορίες: ISO/IEC 18004:2015 Annex A

τελικό κωδικοποιημένο μήνυμα μας για τον κώδικα QR.

Η τελική μας κωδικολέξη $C(x)$ προκύπτει σύμφωνα με την παρακάτω σχέση:

$$C(x) = \left(x^{2r}M(x)\right) + \left(x^{2r}M(x) \bmod g(x)\right) = x^{2r}M(x) + E(x) \tag{2.3}$$

όπου $2r$ είναι το πλήθος κωδικολέξεων διόρθωσης σφαλμάτων και $g(x)$ είναι το πολυώνυμο γεννήτορας[9].

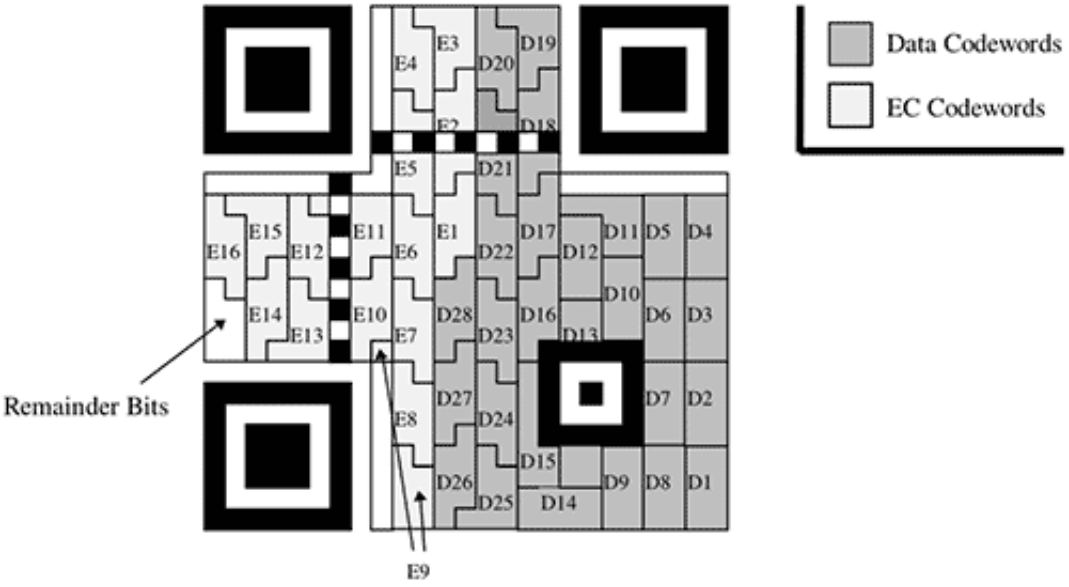
Συνεπώς, σύμφωνα με την παραπάνω σχέση, τοποθετώντας στο τέλος του μηνύματός μας τις 16 κωδικολέξεις διόρθωσης σφαλμάτων παίρνουμε:

$$\underbrace{01000001}_{65}\underbrace{01010110}_{86}\underbrace{10000111}_{135}\underbrace{01000111}_{71}\underbrace{01000111}_{71}\underbrace{00000111}_{7}\underbrace{00110011}_{51}$$
$$\underbrace{10100010}_{162}\underbrace{11110010}_{242}\underbrace{11110111}_{247}\underbrace{01110111}_{119}\underbrace{01110111}_{119}\underbrace{01110010}_{114}\underbrace{11100110}_{230}$$
$$\underbrace{00010110}_{22}\underbrace{01010110}_{86}\underbrace{01110110}_{118}\underbrace{01010110}_{86}\underbrace{00010110}_{22}\underbrace{11100010}_{226}\underbrace{11100110}_{230}$$
$$\underbrace{01110111}_{119}\underbrace{00100000}_{32}\underbrace{11101100}_{236}\underbrace{00010001}_{17}\underbrace{11101100}_{236}\underbrace{00010001}_{17}\underbrace{11101100}_{236}$$
$$\underbrace{10010000}_{144}\underbrace{11010101}_{213}\underbrace{00001101}_{13}\underbrace{00010101}_{21}\underbrace{01100011}_{99}\underbrace{10011100}_{156}\underbrace{10010111}_{151}$$
$$\underbrace{00011110}_{30}\underbrace{01010011}_{83}\underbrace{01001001}_{73}\underbrace{00100100}_{36}\underbrace{11001100}_{204}\underbrace{00101111}_{47}\underbrace{00000110}_{6}$$
$$\underbrace{00100011}_{35}\underbrace{10001101}_{141}$$

Οπότε πλέον είμαστε σε θέση να εισάγουμε το κωδικοποιημένο μήνυμά μας μαζί με τις κωδικολέξεις διόρθωσης σφαλμάτων στον κώδικα QR. Στις θέσεις D1 έως και D28 θα τοποθετήσουμε τις κωδικολέξεις από το κωδικοποιημένο μήνυμα (Data Codewords) και στις θέσεις E1 έως και E16 τις κωδικολέξεις διόρθωσης σφαλμάτων (EC Codewords). Τα υπολειπόμενα bits (Remainder Bits) είναι τα ψηφία που περισσεύουν, καθώς είναι στο πλήθος λιγότερα από 8 και συνεπώς δεν μπορούν να αποτελέσουν κάποια κωδικολέξη. Στην Version 2, το πλήθος τους είναι πάντα 7.

Η τοποθέτηση των bits μέσα στη στήλη θα πρέπει να γίνεται από τα δεξιά προς τα αριστερά είτε κατευθυνόμαστε προς τα πάνω είτε προς τα κάτω εκτός κάποιων εξαιρέσεων που θα δούμε παρακάτω. Το πρώτο ψηφίο που τοποθετούμε στη στήλη D1 είναι το Most Significant Bit (το 7 στο σχήμα 2.7), το πρώτο δηλαδή από τα αριστερά ψηφίο της πρώτης κωδικολέξης μας. Αναγκαστικά κινούμαστε προς τα επάνω για να τοποθετήσουμε τα ψηφία στο κελί D1, όπως απεικονίζεται στο σχήμα 2.8.

Σχήμα 2.7: *Τοποθέτηση των ψηφίων προς τα πάνω και προς τα κάτω*



Σχήμα 2.6: *Τοποθέτηση κωδικολέξεων στον κώδικα QR version 2-M*

Για παράδειγμα, η πρώτη κωδικολέξη μας είναι η 01000001 (65). Άρα το Most Significant Bit είναι το ψηφίο 0 και το τοποθετούμε στο κελί 7. Το δεύτερο ψηφίο μας, δηλαδή το 1, το τοποθετούμε στα αριστερά, δηλαδή στο κελί 6. Το τρίτο ψηφίο πάλι δεξιά, δηλαδή στο κελί 5 κόκ. Οπότε η στήλη D1 θα πάρει τη μορφή

1	0
0	0
0	0
1	0

Σχήμα 2.8: *Τοποθέτηση των ψηφίων της κωδικολέξης στη στήλη D1*

Όταν συναντήσουμε κάποιο οριζόντιο μοτίβο ευθυγράμμισης ή οριζόντιο χρονικό μοτίβο, συνεχίζουμε προς τα πάνω ή προς τα κάτω αγνοώντας το τελείως. Για παράδειγμα, τέτοιες περιπτώσεις εντοπίζονται στις στήλες D18, E5 όπως και στις D9 και D10, όπου για να πάμε από την D9 στην D10 αγνοούμε πλήρως την ύπαρξη του μοτίβου ευθυγράμμισης.

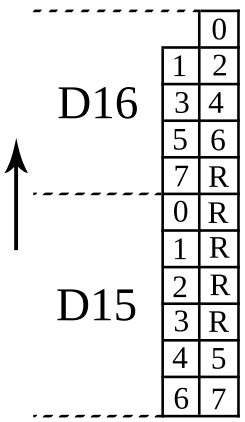
Όταν φτάσουμε σε κάποιο άνω ή κάτω σύνορο της περιοχής κωδικοποίησης, όπως για παράδειγμα στη στήλη D11, όποια ψηφία της συγκεκριμένης κωδικολέξης περισσεύουν θα πρέπει να τοποθετηθούν στην επόμενη στήλη στα αριστερά.

0	1	2	3
		4	5
		6	7

Σχήμα 2.9: *Τοποθέτηση των ψηφίων όταν φτάσουμε σε κάποιο σύνορο*

Όταν συναντήσουμε κάποιο κάθετο μοτίβο ευθυγράμμισης ή κάθετο χρονικό μοτίβο, συνεχίζουμε προς τα πάνω ή προς τα κάτω αγνοώντας κάθε φορά τα δεσμευμένα κελιά

από αυτά. Μια τέτοια περίπτωση παρατηρούμε στις στήλες D15 και D16. Το παρακάτω σχήμα απεικονίζει το τρόπο τοποθέτησης των ψηφίων.



Σχήμα 2.10: *Τοποθέτηση των ψηφίων όταν φτάσουμε σε κάθετο μοτίβο. Το R συμβολίζει τη δεσμευμένη περιοχή*

2.4.1 Μοτίβο μάσκας

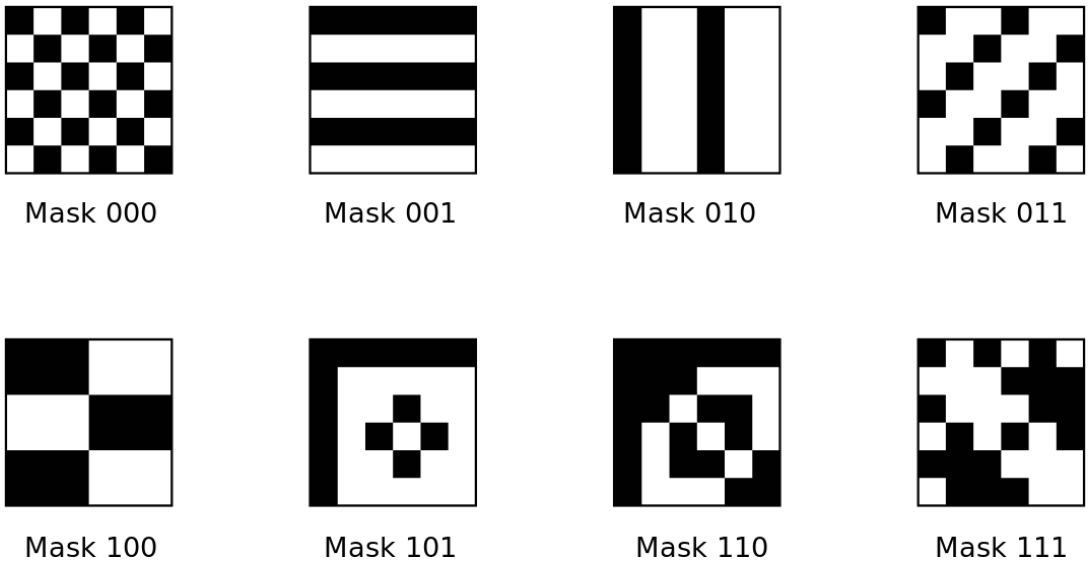
Για την αξιόπιστη ανάγνωση ενός κώδικα QR, πρέπει τα μαύρα και λευκά κελιά να κατανεμηθούν ισόποσα όσο το δυνατόν περισσότερο. Το μοτίβο ψηφίων 1011101 το οποίο συναντάται στο μοτίβο εύρεσης θα πρέπει να αποφεύγεται η εμφάνισή του σε άλλες περιοχές όσο το δυνατόν περισσότερο γίνεται. Για την ικανοποίηση των παραπάνω κριτηρίων γίνεται η χρήση του μοτίβου μάσκας. Ο παρακάτω πίνακας μας απεικονίζει τη δυαδική αλληλουχία bit που θα χρησιμοποιηθεί στις πληροφορίες μορφοποίησης συναρτήσει της αντίστοιχης συνθήκης. Το μοτίβο της μάσκας που θα παραχθεί εφαρμόζεται μόνο στην περιοχή κωδικοποίησης για την οποία η συνθήκη είναι αληθής. Ως i ορίζεται η γραμμή στον κώδικα QR και ως j η στήλη του. Με $(i,j) = (0,0)$ ορίζεται το τέρμα πάνω αριστερό κελί.

Δυαδική αναφορά	Συνθήκη
000	$(i+j) \bmod 2 = 0$
001	$i \bmod 2 = 0$
010	$j \bmod 3 = 0$
011	$(i+j) \; mod 3 = 0$
100	$((i \text{ div } 2) + (j \text{ div } 3)) \bmod 2 = 0$
101	$(ij) \bmod 2 + (ij) \bmod 3 = 0$
110	$((ij) \bmod 2 + (ij) \bmod 3) \bmod 2 = 0$
111	$((ij) \bmod 3 + (i+j) \bmod 2) \bmod 2 = 0$

Πίνακας 2.1: *Παραγωγή μοτίβων μάσκας ανάλογα τη συνθήκη*

Το παρακάτω Σχήμα 2.11 δείχνει το οπτικά το μοτίβο που σχηματίζεται ανάλογα με τη συνθήκη σε πλήθος λευκών κελιών 6×6=36. Όταν η συνθήκη είναι αληθής το αντίστοιχο κελί μετατρέπεται σε μαύρο. Εν γένει, όταν εφαρμόζεται το μοτίβο της μάσκας

σε κελιά της περιοχής κωδικοποίησης στον κώδικα QR, το χρώμα τους αντιστρέφεται.



Σχήμα 2.11: *Απεικόνιση των μοτίβων μάσκας ανά συνθήκη*

Αφού εφαρμοστεί το κάθε μοτίβο μάσκας στον κώδικα QR μας, δηλαδή θα προκύψουν 8 κώδικες QR, ο καθένας με το αντίστοιχο μοτίβο μάσκας, είναι προτιμότερο (αλλά όχι αναγκαίο) να επιλεχθεί αυτός που θα συγκεντρώσει τους λιγότερους πόντους ποινής[6]. Εμείς για την παραγωγή του δικού μας κώδικα QR θα εφαρμόσουμε το μοτίβο μάσκας 010.

2.4.2 Μοτίβο πληροφοριών μορφοποίησης

Το μοτίβο των πληροφοριών μορφοποίησης είναι μια 15-bit ακολουθία η οποία αποτελείται από δεδομένα των 5-bit και 10-bit διόρθωσης σφαλμάτων. Ο υπολογισμός γίνεται των bit διόρθωσης σφαλμάτων γίνεται ξανά με τη χρήση της μεθόδου των Reed-Solomon. Όμως, στην συγκεκριμένη περίπτωση τα πράγματα είναι ευκολότερα καθώς τα πολυώνυμα έχουν το πολύ 15 όρους και οι συντελεστές τους είναι είτε 0 είτε 1.

Από τα 5-bit δεδομένων, τα δύο πρώτα bit περιέχουν την πληροφορία για το επίπεδο διόρθωσης σφαλμάτων του κώδικα QR, και η εκπροσώπηση τους στο δυαδικό γίνεται σύμφωνα με τον παρακάτω πίνακα

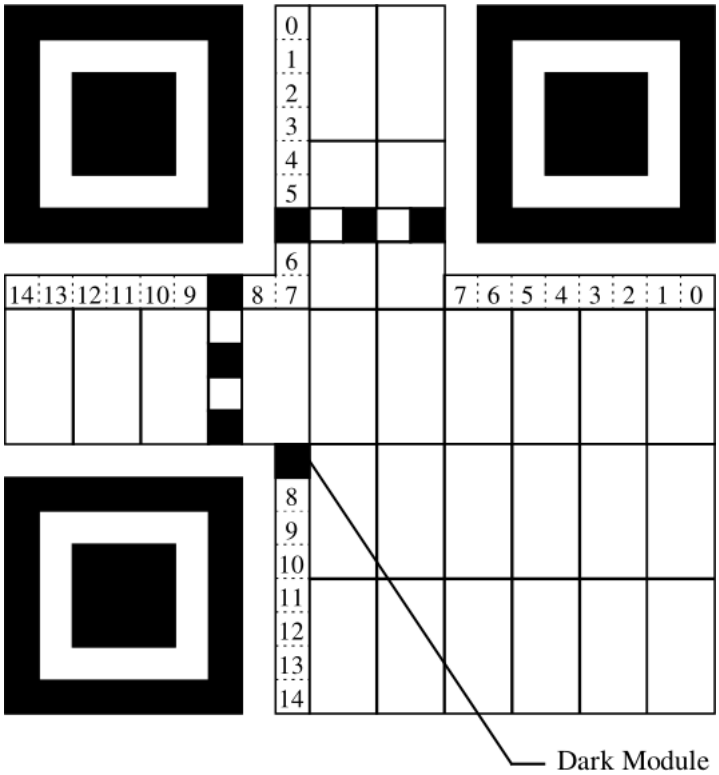
Δυαδική αναφορά	Επίπεδο διόρθωσης σφαλμάτων
01	L
00	M
11	Q
10	H

Πίνακας 2.2: *Δυαδική αναφορά σχετικά με το επίπεδο διόρθωσης σφαλμάτων στο μοτίβο πληροφοριών μορφοποίησης*

[6] Περισσότερες πληροφορίες: ISO/IEC 18004:2015 Ενότητα 7.8.3

Τα υπόλοιπα 3-bit περιέχουν την πληροφορία σχετικά με το μοτίβο μάσκας που έχει χρησιμοποιηθεί όπως αναγράφονται στον Πίνακα 2.1. Για παράδειγμα, εμείς στο δικό μας κώδικα QR που θα χρησιμοποιήσουμε το μοτίβο μάσκας 010 και το επίπεδο διόρθωσης σφαλμάτων M, η ακολουθία δεδομένων 5-bit θα είναι η 00010. Για την παραγωγή των 10 ψηφίων διόρθωσης σφαλμάτων, επειδή το σύνολο όλων των δυνατών συνδυασμών της δυαδικής αναφοράς του μοτίβου μάσκας και του επιπέδου διόρθωσης σφαλμάτων ισούται με 32, δεν είναι απαραίτητο να γίνεται κάθε φορά ο υπολογισμός τους. Επομένως, η 15-bit ακολουθία που θα προκύψει σύμφωνα με το παράδειγμά μας θα είναι η 101111001111100.[7]

Το μοτίβο πληροφοριών εφαρμόζεται στην περιοχή 5 σύμφωνα με το Σχήμα 2.5. Η μέθοδος τοποθέτησης των 15 ψηφίων γίνεται σύμφωνα με το παρακάτω σχήμα. Το μαύρο κελί (Dark Module) τοποθετείται πάντοτε στη θέση (4*V*+9,8) όπου *V* ο αριθμός της έκδοσης (Version) του κώδικα QR. Το μαύρο κελί πρέπει να είναι πάντα μαύρο και δεν αποτελεί μέρος του μοτίβου πληροφοριών μορφοποίησης.



Σχήμα 2.12: *Τοποθέτηση των ψηφίων στον κώδικα QR για το σχηματισμό του μοτίβου πληροφοριών μορφοποίησης.*

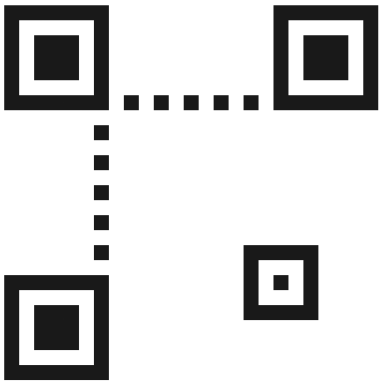
2.5 Κατασκευή QR σύμφωνα με το μήνυμα μας

Οπότε πλέον είμαστε σε θέση να ξεκινήσουμε τη δημιουργία του κώδικα QR μας. Θέλουμε να κατασκευάσουμε ένα τετράγωνο 25x25=625 κελιών οπότε πρέπει να τοποθετήσουμε κατάλληλα το μοτίβο εύρεσης, τους διαχωριστές και τα χρονικά μοτίβα,

[7] Περισσότερες πληροφορίες: https://www.thonky.com/qr-code-tutorial/format-version-tables

όπως έχουμε περιγράψει στο Σχήμα 2.5.

Ανάλογα με την Version του κώδικα QR, προκύπτει και το πλήθος των μοτίβων ευθυγράμμισης που θα χρησιμοποιηθούν. Η Version 2,όπως έχουμε αναφέρει, χρησιμοποιεί ένα μοτίβο ευθυγράμμισης, σύμφωνα με το Σχήμα 2.5. Τοποθετείται στη θέση με κέντρο του τις συντεταγμένες $(i,j) = (6, 18)$ στον κώδικα QR[8]. Συνεπώς, ο κώδικας QR μας θα έχει την παρακάτω πρώτη μορφή



Σχήμα 2.13: *Τοποθέτηση μοτίβου εύρεσης, διαχωριστών, χρονικών μοτίβων και μοτίβου ευθυγράμμισης στον κώδικα QR.*

Έπειτα τοποθετούμε τα ψηφία από τις κωδικολέξεις δεδομένων και τα ψηφία από τις κωδικολέξεις διόρθωσης σφαλμάτων όπως έχουμε περιγράψει σύμφωνα με το Σχήμα 2.6. Έτσι ο κώδικας QR μας θα πάρει την παρακάτω μορφή

Σχήμα 2.14: *Ο κώδικας QR περιλαμβάνει μοτίβο εύρεσης, διαχωριστές, χρονικά μοτίβα, μοτίβο ευθυγράμμισης και τις κωδικολέξεις δεδομένων και διόρθωσης σφαλμάτων*

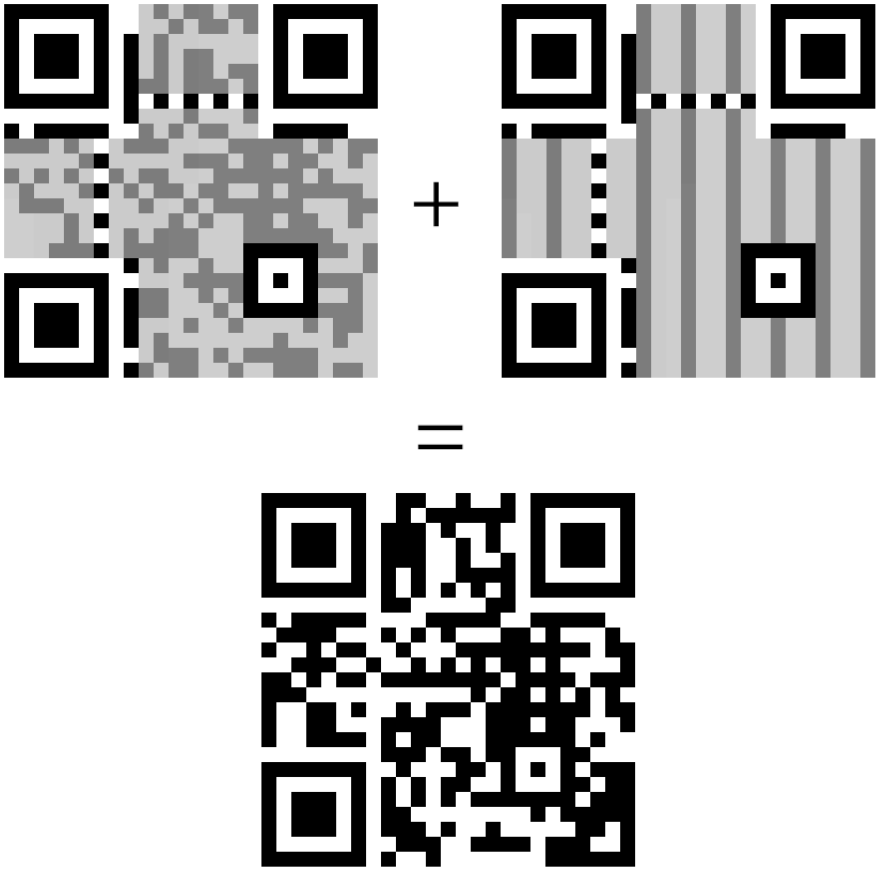
Έτσι λοιπόν, έχουμε συμπληρώσει (χρωματίσει) τα κελιά ανάλογα με τα ψηφία από όλες τις περιοχές σύμφωνα με το Σχήμα 2.5, εκτός όμως από τα κελιά που βρίσκονται

[8] Περισσότερες πληροφορίες: ISO/IEC 18004:2015 Annex E

στην περιοχή 5 τα οποία περιέχουν τις πληροφορίες μορφοποίησης. Όπως έχουμε αναφέρει, οι πληροφορίες μορφοποίησης περιέχουν το επίπεδο διόρθωσης σφαλμάτων (L,M,Q,H) και το μοτίβο μάσκας, το οποίο ακόμη δεν έχουμε εφαρμόσει. Οπότε προσθέτοντας τις πληροφορίες μορφοποίησης, το παρακάτω σχήμα πλέον περιέχει όλες τις πληροφορίες σχετικά με τα μοτίβα λειτουργιών και την περιοχή κωδικοποίησης.

Σχήμα 2.15: *Ο κώδικας QR περιλαμβάνει όλες τις πληροφορίες σχετικά με τα μοτίβα λειτουργιών και την περιοχή κωδικοποίησης, εκτός του μοτίβου μάσκας.*

Όμως, ακόμη δεν είναι ολοκληρωμένος για να διαβαστεί από έναν QR reader, καθώς πρέπει να εφαρμοστεί το μοτίβο μάσκας. Το μοτίβο μάσκας όπως έχουμε αναφέρει αντιστρέφει το χρώμα των κελιών ανάλογα με την αντίστοιχη συνθήκη, τα οποία κελιά βρίσκονται αυστηρώς στην περιοχή κωδικοποίησης. Στο παρακάτω σχήμα, για δική μας ευκολία και καλύτερη κατανόηση, η γκρίζα περιοχή αντιπροσωπεύει την περιοχή κωδικοποίησης όπου επιτρέπεται να εφαρμοστεί το μοτίβο μάσκας. Η εφαρμογή του μοτίβου μάσκας, συγκεκριμένα του 010, μας δίνει τη τελική μορφή του κώδικα QR μας που περιέχει το μήνυμα https://www.aegean.gr



Σχήμα 2.16: *Ο τελικός κώδικας QR που προκύπτει εφαρμόζοντας το μοτίβο μάσκας 010.*

Παραρτήματα

Παράρτημα Α΄

Πίνακας κωδικοποίησης για το ISO 8859-1

Dec	Char	Dec	Char	Dec	Char	Dec	Char	Dec	Char	Dec	Char	Dec	Char	Dec	Char
0	NUL	32	SPC	64	@	96	‘	128	PAD	160	NBS	192	À	224	à
1	SOH	33	!	65	A	97	a	129	HOP	161	¡	193	Á	225	á
2	STX	34	”	66	B	98	b	130	BPH	162	¢	194	Â	226	â
3	ETX	35	#	67	C	99	c	131	NBH	163	£	195	Ã	227	ã
4	EOT	36	$	68	D	100	d	132	IND	164	¤	196	Ä	228	ä
5	ENQ	37	%	69	E	101	e	133	NEL	165	¥	197	Å	229	å
6	ACK	38	&	70	F	102	f	134	SSA	166	¦	198	Æ	230	æ
7	BEL	39	’	71	G	103	g	135	ESA	167	§	199	Ç	231	ç
8	BS	40	(	72	H	104	h	136	HTS	168	¨	200	È	232	è
9	TAB	41	)	73	I	105	i	137	HTJ	169	©	201	É	233	é
10	LF	42	*	74	J	106	j	138	VTS	170	ª	202	Ê	234	ê
11	VT	43	+	75	K	107	k	139	PLD	171	«	203	Ë	235	ë
12	FF	44	,	76	L	108	l	140	PLU	172	¬	204	Ì	236	ì
13	CR	45	-	77	M	109	m	141	RI	173	SHY	205	Í	237	í
14	SO	46	.	78	N	110	n	142	SS2	174	®	206	Î	238	î
15	SI	47	/	79	O	111	o	143	SS3	175	¯	207	Ï	239	ï
16	DLE	48	0	80	P	112	p	144	DCS	176	°	208	Đ	240	ð
17	DC1	49	1	81	Q	113	q	145	PU1	177	±	209	Ñ	241	ñ
18	DC2	50	2	82	R	114	r	146	PU2	178	²	210	Ò	242	ò
19	DC3	51	3	83	S	115	s	147	STS	179	³	211	Ó	243	ó
20	DC4	52	4	84	T	116	t	148	CCH	180	´	212	Ô	244	ô
21	NAK	53	5	85	U	117	u	149	MW	181	µ	213	{	245	õ
22	SYN	54	6	86	V	118	v	150	SPA	182	¶	214	Ö	246	ö
23	ETB	55	7	87	W	119	w	151	EPA	183	·	215	×	247	÷
24	CAN	56	8	88	X	120	x	152	SOS	184	¸	216	Ø	248	ø
25	EM	57	9	89	Y	121	y	153	SGCI	185	¹	217	Ù	249	ù
26	SUB	58	:	90	Z	122	z	154	SCI	186	º	218	Ú	250	ú
27	ESC	59	;	91	[	123	{	155	CSI	187	»	219	Û	251	û
28	FS	60	<	92	\	124	|	156	ST	188	¼	220	Ü	252	ü
29	GS	61	=	93	]	125	}	157	OSC	189	½	221	Ý	253	ý
30	RS	62	>	94	^	126	~	158	PM	190	¾	222	Þ	254	þ
31	US	63	?	95	_	127	DEL	159	APC	191	¿	223	ß	255	ÿ

Πίνακας Α΄.1: *Πίνακας κωδικοποίησης-αποκωδικοποίησης χαρακτήρων για το ISO 8859-1. Πηγή: ISO/IEC 18004*

Παράρτημα Β′

Πίνακας αντιστοιχίας πρωταρχικού στοιχείου σε ακέραια τιμή στο σώμα Galois $GF(256)$

Δύναμη του α	Ακέραιος	Δύναμη του α	Ακέραιος	Δύναμη του α	Ακέραιος	Δύναμη του α	Ακέραιος	Δύναμη του α	Ακέραιος
0	1	51	10	102	68	153	146	204	221
1	2	52	20	103	136	154	57	205	167
2	4	53	40	104	13	155	114	206	83
3	8	54	80	105	26	156	228	207	166
4	16	55	160	106	52	157	213	208	81
5	32	56	93	107	104	158	183	209	162
6	64	57	186	108	208	159	115	210	89
7	128	58	105	109	189	160	230	211	178
8	29	59	210	110	103	161	209	212	121
9	58	60	185	111	206	162	191	213	242
10	116	61	111	112	129	163	99	214	249
11	232	62	222	113	31	164	198	215	239
12	205	63	161	114	62	165	145	216	195
13	135	64	95	115	124	166	63	217	155
14	19	65	190	116	248	167	126	218	43
15	38	66	97	117	237	168	252	219	86
16	76	67	194	118	199	169	229	220	172
17	152	68	153	119	147	170	215	221	69
18	45	69	47	120	59	171	179	222	138
19	90	70	94	121	118	172	123	223	9
20	180	71	188	122	236	173	246	224	18
21	117	72	101	123	197	174	241	225	36
22	234	73	202	124	151	175	255	226	72
23	201	74	137	125	51	176	227	227	144
24	143	75	15	126	102	177	219	228	61
25	3	76	30	127	204	178	171	229	122
26	6	77	60	128	133	179	75	230	244
27	12	78	120	129	23	180	150	231	245
28	24	79	240	130	46	181	49	232	247
29	48	80	253	131	92	182	98	233	243
30	96	81	231	132	184	183	196	234	251
31	192	82	211	133	109	184	149	235	235
32	157	83	187	134	218	185	55	236	203
33	39	84	107	135	169	186	110	237	139
34	78	85	214	136	79	187	220	238	11
35	156	86	177	137	158	188	165	239	22
36	37	87	127	138	33	189	87	240	44
37	74	88	254	139	66	190	174	241	88
38	148	89	225	140	132	191	65	242	176
39	53	90	223	141	21	192	130	243	125
40	106	91	163	142	42	193	25	244	250
41	212	92	91	143	84	194	50	245	233
42	181	93	182	144	168	195	100	246	207
43	119	94	113	145	77	196	200	247	131
44	238	95	226	146	154	197	141	248	27
45	193	96	217	147	41	198	7	249	54
46	159	97	175	148	82	199	14	250	108
47	35	98	67	149	164	200	28	251	216
48	70	99	134	150	85	201	56	252	173
49	140	100	17	151	170	202	112	253	71
50	5	101	34	152	73	203	224	254	142

Πίνακας Β′.1: *Η πρώτη στήλη απεικονίζει τη δύναμη στην οποία είναι υψωμένο το πρωταρχικό στοιχείο $\alpha = 2$ στο GF(256). Η άλλη στήλη μας δίνει την αντίστοιχη ακέραια τιμή του α.*

Βιβλιογραφία

[1] D.R. Hankerson, D.G. Hoffman, D.A. Leonard, C.C. Lindner, K.T. Phelps, C.A. Rodger, J.R. Wall *Coding Theory And Cryptography - The Essentials*. 2nd Edition. , Taylor and Francis Group. Boca Raton, FL. 2000.

[2] N. Azmi, L.M. Kamarudin, M. Mahmuddin, A. Zakaria, A.Y.M Shakaff, S. Khatun, M.N. Morshed, *Interference Issues and Mitigation Method in WSN 2.4Ghz ISM Band: A Survey*. 2nd International Conference on Electronic Design, Penang, Malaysia, August 19-21, 2014.

[3] Gérard Maral, Jean-Jacques de Ridder, Barry G. Evans, Madhavendra Richharia *Low earth orbit satellite systems for communications*. International Journal of Satellite Communications, 1991.

[4] https://voyager.jpl.nasa.gov/

[5] Θ. Θεοχάρη-Αποστολίδη, Χ. Χαραλάμπους, Χ.Βαβατσούλας *Εισαγωγή στη Γραμμική Άλγεβρα*, Θεσσαλονίκη, 2006.

[6] ISO/IEC 18004:2015

[7] Peter Kieseberg, Manuel Leithner, Martin Mulazzani, Lindsay Munroe, Sebastian Schrittwieser, Mayank Sinha, Edgar Weippl *QR Code Security*, SBA Research, Vienna, Austria.

[8] https://www.thonky.com/qr-code-tutorial/

[9] Anindya Sundar Das, Satyajit Das, Jaydeb Bhaumik *Design of RS (255, 251) Encoder and Decoder in FPGA.* International Journal of Soft Computing and Engineering, 2013.

[10] Raymond S. Lim *A Deconding Procedure for the Reed-Solomon Codes*, NASA Technical Paper 1286, 1978.

[11] William A. Geisel *Tutorial on Reed-Solomon Error Correction Coding*, NASA Technical Memorandum 102162, 1990.