Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακή Διατριβή
στην Ασφάλεια Υπολογιστών και Δικτύων

Ανάλυση κρυπτογραφικών ακολουθιών de Bruijn ως προς το προφίλ γραμμικής πολυπλοκότητας

Κωνσταντίνος Ρόζης

Επιβλέπων Καθηγητής
Κωνσταντίνος Λιμνιώτης

Μάιος 2019

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Ανάλυση κρυπτογραφικών ακολουθιών de Bruijn ως προς το προφίλ γραμμικής πολυπλοκότητας

Κωνσταντίνος Ρόζης

Επιβλέπων Καθηγητής
Κωνσταντίνος Λιμνιώτης

Η παρούσα μεταπτυχιακή διατριβή υποβλήθηκε
προς μερική εκπλήρωση των απαιτήσεων για απόκτηση

μεταπτυχιακού τίτλου σπουδών
στην Ασφάλεια Υπολογιστών και Δικτύων

από τη Σχολή Θετικών και Εφαρμοσμένων Επιστημών
του Ανοικτού Πανεπιστημίου Κύπρου

Μάιος 2019

Περίληψη

Σε μια ανταγωνιστική αγορά, η εξασφάλιση του απόρρητου δεδομένων και πληροφοριών και η προστασία της ιδιωτικότητας, εμφανίζονται ως καθοριστικοί παράγοντες σταθερότητας και βιωσιμότητας επιχειρήσεων και Οργανισμών που καλούνται -προκειμένου να προστατευθούν αλλά και να αποδώσουν τα μέγιστα- να εξασφαλίσουν την ασφαλή διακίνηση και επεξεργασία των πληροφοριών και δεδομένων τους.

Οι δυαδικές ακολουθίες de Bruijn χρησιμοποιούνται ιδίως σε κρυπτογραφικούς αλγόριθμους ροής, συγκεκριμένα ως παραγόμενες από μη γραμμικούς καταχωρητές με ανάδραση (NLFSR). Χρησιμοποιούνται ως γεννήτριες κλειδοροής με ισχυρά κρυπτογραφικά χαρακτηριστικά. Αυτές οι ισχυρές ιδιότητες των ακολουθιών de Bruijn όμως, δύνανται να μειωθούν ραγδαία, κατόπιν αλλαγής λίγων μόνο ψηφίων τους.

Παρά τη βιβλιογραφία που έχει αναπτυχθεί παγκοσμίως για τις ακολουθίες de Bruijn, οι κρυπτογραφικές ιδιότητες των ακολουθιών, δεν έχουν εξετασθεί εκτενώς. Η σημασία τους για την περιοχή της κρυπτογραφίας εμφανίζεται τεράστια, καθώς η διαφοροποίησή τους επηρεάζει άμεσα την ισχύ της.

Η παρούσα ερευνητική προσπάθεια σκοπεύει στη διερεύνηση της συμπεριφοράς της γραμμικής πολυπλοκότητας ακολουθίας de Bruijn, μεταβάλλοντας εντός αυτής κάποια ψηφία της. Στο πλαίσιο αυτό, διερευνήθηκε σε συγκεκριμένες ακολουθίες de Bruijn με χρήση του αλγορίθμου Lauder-Paterson, η διαμόρφωση της γραμμικής πολυπλοκότητας k σφαλμάτων. Αυτό το κρυπτογραφικό κριτήριο, παρά τη σημασία του στην κρυπτογραφία, δεν έχει μελετηθεί στο παρελθόν για ακολουθίες de Bruijn. Παράλληλα εξετάσθηκαν, μέσω πειραματικής μεθόδου, τα κρυπτογραφικά κριτήρια των «τροποποιημένων» ακολουθιών de Bruijn.

Τα αποτελέσματα της έρευνας οδήγησαν στο συμπέρασμα ότι μία de Bruijn ακολουθία ενδέχεται να μην εμφανίζει καλή συμπεριφορά ως προς αυτό το κρυπτογραφικό κριτήριο της γραμμικής πολυπλοκότητας k σφαλμάτων, ενώ η συμπεριφορά αυτή φαίνεται να είναι ανεξάρτητη από την τεχνικής παραγωγής της ακολουθίας.

Η παρούσα έρευνα, εστιάζοντας στη μελέτη της διατήρησης ή όχι των κρυπτογραφικών ιδιοτήτων μιας de Bruijn ακολουθίας σε περίπτωση διαφοροποίησης στοιχείων της, αναδεικνύει τη σημασία μελέτης της γραμμικής πολυπλοκότητας k σφαλμάτων και αποτελεί ένα πρώτο βήμα για μελλοντική έρευνα στο επιστημονικό αυτό πεδίο.

Summary

In a competitive market, ensuring confidentiality of data and information and as well as privacy protection are seen as the decisive factors for the stability and sustainability of businesses and organizations, which are invited – in order to get protected but also achieve maximum performance - to ensure safe sharing and processing of their information and data.

The binary de Bruijn sequences are mainly used in stream ciphers, specifically as the output of non-linear feedback shift registers (NLFSR). They are used as keystream generators, with strong cryptographic characteristics. Those strong properties of de Bruijn sequences though may be rapidly reduced after a change in only a few digits.

In spite of the extensive research literature on the subject, there are still open question on the cryptographic properties of de Bruijn sequences. Their particular importance for the cryptography area seems to be enormous as its validity is directly affected by their diversification.

The present study aims to investigate the behavior of a de Bruijn sequence's linear complexity, in case that a few bits are being modified. To achieve this, the Lauder-Paterson algorithm was used to find the so-called k-error linear complexity spectrum of specific de Bruijn sequences. This cryptographic criterion, despite its importance, has not been studied yet with regard to de Bruijn sequence. Moreover the cryptographic criteria of "modified" de Bruijn sequences were examined by the use of experimental methodology.

Findings allow us to conclude that it is possible that a de Bruijn sequence may not behave well in terms of this cryptographic criterion, whereas such a behavior seems to be independent from the method that is being used to construct such a sequence.

Focusing on the preservation of a de Bruijn sequence cryptographic characteristics in case of some bits are being modified, the present study further illustrates the importance of the k-error linear complexity spectrum as a cryptographic criterion for sequences. Apparently, the present results may constitute the first step towards further research on this field.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον κο Κωνσταντίνο Λιμνιώτη που επέβλεψε τη διπλωματική μου εργασία, αφιέρωσε τον πολύτιμο χρόνο του και ήταν υποστηρικτικός καθ' όλη τη διάρκεια της συγγραφής της εργασίας αυτής. Επιθυμώ να εκφράσω την ευγνωμοσύνη και την εκτίμησή μου για την καθοδήγησή του στα ανακύπτοντα ζητήματα του θέματος της εργασίας μου καθώς και την συνολική συμβολή του στην συγγραφή της.

Επίσης ευχαριστώ όλους τους ερευνητές και συγγραφείς που το έργο τους αποτέλεσε τις πηγές αυτής της προσπάθειας και βοήθησε ώστε να γίνει πληρέστερη και επιστημονικά δόκιμη η παρούσα μεταπτυχιακή διατριβή.

Τέλος θα ήθελα να ευχαριστήσω τη σύζυγό μου Άννα για την υποστήριξη και βοήθεια που μου προσέφερε.

Περιεχόμενα

Κεφάλαιο 1
Εισαγωγή

"I am fairly familiar with all the forms of secret writings, and am myself the author of a trifling monograph upon the subject, in which I analyze one hundred and sixty separate ciphers," said Holmes..

—*The Adventure of the Dancing Men*, Sir Arthur Conan Doyle.

"Είμαι αρκετά εξοικειωμένος με όλες τις μορφές μυστικών γραπτών και είμαι ο ίδιος ο συγγραφέας μιας ασήμαντης μονογραφίας πάνω στο θέμα, στο οποίο αναλύω εκατόν εξήντα χωριστούς κώδικες", δήλωσε ο Χολμς.

--Η Περιπέτεια των Χορευτών, Sir Arthur Conan Doyle.

«Ο ανθρώπινος νους μπορεί να σπάσει οποιοδήποτε κρυπτογραφημένο κείμενο που η ανθρώπινη ευρηματικότητα μπορεί να επινοήσει»

--"Το χρυσό έντομο" ("The Gold Bug"), Edgar Allan Poe, 1843.

Σε μια ανταγωνιστική αγορά, η εξασφάλιση του απορρήτου δεδομένων και πληροφοριών και η προστασία της ιδιωτικότητας, εμφανίζονται ως καθοριστικοί παράγοντες σταθερότητας και βιωσιμότητας επιχειρήσεων και Οργανισμών που καλούνται -προκειμένου να προστατευθούν αλλά και να αποδώσουν τα μέγιστα- να εξασφαλίσουν την ασφαλή διακίνηση και επεξεργασία των πληροφοριών και δεδομένων τους.

Η συνεχής προσπάθεια να διατηρηθούν ως απόρρητα για μη εξουσιοδοτημένα άτομα τα - κατά πλειοψηφία - ηλεκτρονικά δεδομένα μέσα στο περιβάλλον ανταγωνισμού που συχνά γίνεται αθέμιτος, βρίσκει εφαρμογή στην κρυπτογραφική προστασία. Επομένως, η κρυπτογράφηση δεδομένων, ως ζήτημα εξέχουσας σπουδαιότητας που συμβάλλει στο απαιτούμενο επίπεδο ασφάλειας προς εξασφάλιση ομαλής λειτουργίας και παραγωγικότητας, δεν μπορεί παρά να στηρίζεται στη χρήση αλγορίθμων, οι οποίοι με χρήση ή όχι παραμέτρων που ονομάζονται κλειδιά, εμφανίζονται με ποικίλους τρόπους (κρυπτογράφηση κειμένου, συναρτήσεις κατακερματισμού, ψηφιακές υπογραφές κλπ) ώστε να επιτελέσουν τον συγκεκριμένο σκοπό.

Στο πλαίσιο ανάπτυξης ισχυρών κρυπτογραφικών αλγορίθμων, ιδιαίτερη βαρύτητα λαμβάνει η παραγωγή ακολουθιών με καλά κρυπτογραφικά χαρακτηριστικά, δηλαδή με ιδιότητες επιθυμητές προκειμένου να μην υπάρχουν ευπάθειες στους αντίστοιχους κρυπτογραφικούς αλγορίθμους οι οποίοι βασίζονται σε αυτές. Επί πολλές δεκαετίες, το ζήτημα της μελέτης κρυπτογραφικών ακολουθιών αποτέλεσε αντικείμενο ερευνητικών προσπαθειών, ιδίως στο πλαίσιο της εφαρμογής τους σε μία σημαντική κατηγορία κατηγορία κρυπτογραφικών αλγορίθμων, αυτής των κρυπταλγορίθμων ροής. Ιδιαίτερη έμφαση έχει δοθεί στη λεγόμενη γραμμική πολυπλοκότητα των ακολουθιών, η οποία εκφράζει το ελάχιστο πλήθος των ψηφίων μίας ακολουθίας τα οποία, εφόσον είναι γνωστά, επαρκούν για την πρόβλεψη ολόκληρου του υπόλοιπου τμήματος της ακολουθίας – ακριβώς και για αυτό το λόγο οι κρυπτογραφικές ακολουθίες πρέπει να έχουν εγγυημένα υψηλή γραμμική πολυπλοκότητα.

Μία σημαντική οικογένεια ακολουθιών με εφαρμογές και στην κρυπτογραφία είναι οι λεγόμενες ακολουθίες de Bruijn. Συγκεκριμένα, οι δυαδικές ακολουθίες de Bruijn παράγονται από μη γραμμικούς καταχωρητές με ανάδραση (NLFSR) , οι οποίοι παράγουν ακολουθίες με τη μέγιστη δυνατή περίοδο – μία ιδιότητα η οποία είναι επιθυμητή σε κρυπτογραφικές εφαρμογές. Οι καταχωρητές αυτοί χρησιμοποιούνται σε γεννήτριες κλειδοροής προκειμένου να παράγονται ακολουθίες με ισχυρά κρυπτογραφικά χαρακτηριστικά, όπως η υψηλή γραμμική πολυπλοκότητα, η ανθεκτικότητα σε αλγεβρικές επιθέσεις, ο υψηλός αλγεβρικός βαθμός και η μη γραμμικότητα. Γενικότερα ωστόσο, παρά το γεγονός ότι οι ακολουθίες de Bruijn μελετώνται

επί δεκαετίες, πολλά συναφή με αυτές ερωτήματα παραμένουν ανοιχτά. Για παράδειγμα, δεν έχει μελετηθεί κατά πόσον είναι μειούμενες οι ισχυρές κρυπτογραφικές ιδιότητες των ακολουθιών de Bruijn με τη διαφοροποίηση ακόμα και λίγων ψηφίων τους.

Το πεδίο έρευνας της παρούσας προσπάθειας, αφορά στην αναγνώριση συγκεκριμένων κριτηρίων που πρέπει να διαθέτει η αρχική de Bruijn ακολουθία, ώστε τα κρυπτογραφικά της κριτήρια να παραμένουν ακόμα και αν μεταβληθούν λίγα ψηφία της. Στο πλαίσιο αυτό, πραγματοποιήθηκε αρχικά βιβλιογραφική επισκόπηση και ακολουθήθηκε ένα ποσοτικό ερευνητικό σχέδιο κατά το οποίο, με χρήση του αλγόριθμου των Lauder-Paterson για την εύρεση του λεγόμενου προφίλ γραμμικής πολυπλοκότητας k σφαλμάτων (το οποίο αποτελεί ένα κρυπτογραφικό κριτήριο που ορίζεται στη συνέχεια), ελέγχθηκαν τα κριτήρια που πρέπει να παρουσιάζει μια ακολουθία de Bruijn ώστε αυτή να διατηρεί τις ισχυρές κρυπτογραφικές ιδιότητές της ακόμα και στην περίπτωση μεταβολής ψηφίων της.

Η έρευνα ολοκληρώνεται με τις επιπτώσεις, τους περιορισμούς της μελέτης και τα μελλοντικά ερευνητικά πεδία.

1.1 Σκοπός της Έρευνας

Παρά τη βιβλιογραφία που έχει αναπτυχθεί παγκοσμίως για τις ακολουθίες de Bruijn, οι κρυπτογραφικές ιδιότητες των ακολουθιών, δεν έχουν εξετασθεί πλήρως. Η σημασία τους για την περιοχή της κρυπτογραφίας εμφανίζεται τεράστια, καθώς η διαφοροποίησή τους επηρεάζει άμεσα την ισχύ της.

Στόχος αυτής της έρευνας είναι, κατόπιν μελέτης βιβλιογραφίας, η συγκέντρωση εμπειρικών δεδομένων μέσω υλοποίησης κατάλληλων αλγορίθμων, ώστε να διερευνηθεί η συμπεριφορά της γραμμικής πολυπλοκότητας ακολουθίας de Bruijn, μεταβάλλοντας εντός αυτής κάποια ψηφία. Μέσω αυτού, θα διερευνήσουμε κατά πόσον συγκεκριμένες τεχνικές παραγωγής ακολουθιών de Bruijn παράγουν ακολουθίες με κοινή συμπεριφορά ως προς αυτό το κρυπτογραφικό κριτήριο.

1.2 Βασικά Ερευνητικά Ερωτήματα

Προς εξυπηρέτηση του σκοπού της έρευνας, θα τεθούν τα ακόλουθα ερευνητικά ερωτήματα:

- Ο τρόπος συμπεριφοράς του προφίλ γραμμικής πολυπλοκότητας μιας ακολουθίας de Bruijn. Προς επίτευξη τούτου, θα αξιοποιηθεί ο αλγόριθμος των Lauder-Paterson.
- Η επιρροή των k-σφαλμάτων (k-error) στη γραμμική πολυπλοκότητα της ακολουθίας de Bruijn.
- Ο βαθμός στον οποίο μία συγκεκριμένη τεχνική παραγωγής ακολουθιών de Bruijn μπορεί να καθορίσει τη συμπεριφορά των παραγόμενων ακολουθιών ως προς αυτό το κρυπτογραφικό κριτήριο.

1.3 Αναγκαιότητα και Σπουδαιότητα της Έρευνας

Όπως προαναφέρθηκε, οι de Bruijn ακολουθίες αποτελούν διαρκές αντικείμενο έρευνας σε πολλούς επιστημονικούς τομείς. Για το χώρο της κρυπτογραφίας, στον οποίο εστιάζει η μελέτη, εξακολουθούν να υπάρχουν πολλά ανοιχτά ερωτήματα.

Η παραγωγή των ακολουθιών από τους μη γραμμικούς καταχωρητές (NLFSR), αποτελεί πεδίο εξαιρετικού ενδιαφέροντος, διότι εμφανίζει σημαντικά πλεονεκτήματα στην ανάπτυξη της κρυπτογραφίας και του ασφαλούς περιβάλλοντος αποθήκευσης και διακίνησης δεδομένων. Παρόλα αυτά, τα ζητήματα που ανακύπτουν δεν είναι εξαντλημένα, αντίθετα, υπάρχει ευρύ φάσμα μελλοντικής έρευνας.

Η έρευνα, εστιάζοντας στη διατήρηση των κρυπτογραφικών ιδιοτήτων μιας de Bruijn ακολουθίας σε περίπτωση διαφοροποίησης στοιχείων της, οδηγεί στην συνέχιση της ισχύος της κρυπτογραφικής ασφάλειας δεδομένων και ανοίγει το δρόμο για μελλοντική έρευνα και εφαρμογές σε διαφορετικά επιστημονικά πεδία (βιολογία, ιατρική).

1.4 Σύντομη Ανασκόπηση Βιβλιογραφίας

Ο μετασχηματισμός ενός μηνύματος σε μη αναγνώσιμη μορφή με σκοπό την προστασία του κατά τη μετάδοση ή την αποθήκευσή του, ορίζεται ως κρυπτογράφηση. Το αντίστροφο, ορίζεται ως αποκρυπτογράφηση, ενώ και οι δύο αποτελούν μέρος της κρυπτογραφίας και επιτυγχάνονται μέσω κρυπτογραφικών αλγορίθμων.

Οι ακολουθίες de Bruijn, αποτελούν ακολουθίες μεγίστης περιόδου και παράγονται από κρυπταλγόριθμους ροής. Η γραμμική πολυπλοκότητά τους χαρακτηρίζει την ισχύ των κρυπτογραφικών ιδιοτήτων των ακολουθιών.

Κατά τη βιβλιογραφική ανασκόπηση (Literature review), πραγματοποιείται αρχικά μια προσπάθεια αναζήτησης της θεωρητικής βάσης των εννοιών, των διαστάσεων και των συσχετισμών τους, βάσει του τρόπου και των ιδεών ερευνητών, που έχουν αντιληφθεί και θέσει διάφορες πλευρές του θέματος, σε θεωρητικό αλλά και εμπειρικό επίπεδο.

Πολλές έννοιες και ορισμοί, όπως αυτοί της ακολουθίας, των γραμμικών καταχωρητών, των κρυπτογραφικών ιδιοτήτων και πολλών άλλων, απαντώνται στη βιβλιογραφία ανάλογα με τους χρήστες και τις αντίστοιχες ερευνητικές προτεραιότητες που έχει ο κάθε ένας τους.

Έτσι, στηριζόμενοι κατά βάση σε άρθρα που δημοσιεύθηκαν σε επιστημονικά περιοδικά από ακαδημαϊκούς αλλά και πεπειραμένους επαγγελματίες, επιχειρείται να συνδυαστεί το επιστημονικό υπόβαθρο με τον πραγματικό κόσμο της ανάπτυξης συστημάτων κρυπτογραφίας υψηλής ισχύος.

Στην συνέχεια, αναλύονται μέσω αλγορίθμων ζητήματα πάνω στις ακολουθίες de Bruijn, ειδικά σε αυτές που εμφανίζουν υψηλή τιμή γραμμικής πολυπλοκότητας, με στόχο την ανάδειξη συγκεκριμένων κριτηρίων πληρότητάς τους έτσι ώστε να συνεχίζουν να λειτουργούν με ισχυρές κρυπτογραφικές ιδιότητες, ακόμα και αν μεταβληθούν ψηφία τους.

1.5 Προτεινόμενη Μεθοδολογία

Η έρευνα στήριξης των υποθέσεων της εργασίας αυτής, θα διεξαχθεί για την διερεύνηση της συμπεριφοράς της γραμμικής πολυπλοκότητας ακολουθίας de Bruijn, μεταβάλλοντας εντός αυτής κάποια ψηφία της, υιοθετώντας πειραματική προσέγγιση. Η μεθοδολογία θα στραφεί στη μετρήσιμη -με συστηματικό τρόπο- διερεύνηση των σχέσεων εντός (μετρήσιμων) μεταβλητών, με στόχο να προβλεφθεί και να εξηγηθεί η μεταβολή της γραμμικής πολυπλοκότητας ακολουθιών de Bruijn, διαφοροποιώντας εντός της ψηφία, με τρόπο που να παραμένουν σε ισχύ οι κρυπτογραφικές τους ιδιότητες (Leedy, 1993). Βάσει της υπόθεσης, θα συλλεγούν πειραματικά δεδομένα που θα χρησιμοποιηθούν με στατιστική συσχέτιση.

1.6 Δομή της Μεταπτυχιακής Διατριβής

Η παρούσα μεταπτυχιακή διατριβή θα αναπτυχθεί και θα παρουσιαστεί δομημένη στα παρακάτω κεφάλαια :

Κεφάλαιο 2 : Κρυπτογραφία

Παρουσίαση του θεωρητικού υπόβαθρου, βασισμένο σε βιβλιογραφική ανασκόπηση της κρυπτογραφίας ως έννοια εξελισσόμενη της οποίας η συμβολή στην ασφάλεια των δεδομένων περιγράφεται συνοπτικά. Γίνεται αναφορά στις κρυπτογραφικές επιθέσεις και αναλύεται ο διαχωρισμός των κρυπτογραφικών αλγορίθμων ανάλογα με τη χρήση μυστικών κλειδιών.

Κεφάλαιο 3: Κρυπτογραφικοί Αλγόριθμοι Ροής

Αναπτύσσεται η αναγκαιότητα ύπαρξης των κρυπτογραφικών αλγορίθμων ροής με αναφορά στα κριτήρια τυχαιότητας που οφείλουν να πληρούν οι παραγόμενες από αυτούς ακολουθίες και περιγράφεται ο τρόπος λειτουργίας τους και οι διατάξεις που λαμβάνουν. Αναλύεται η έννοια της γραμμικής πολυπλοκότητας ως μέγεθος κρυπτογραφικής ισχύος και γίνεται αναφορά στα υπόλοιπα μεγέθη αποτίμησης τυχαιότητας ακολουθίας δημιουργημένης από κρυπτογραφικούς αλγόριθμους ροής. Τέλος, περιγράφονται και αναλύονται οι αλγόριθμοι που υπολογίζουν τα παραπάνω μεγέθη.

Κεφάλαιο 4: Δυαδικές Ακολουθίες de Bruijn

Σύντομη περιγραφή των ακολουθιών de Bruijn με αναφορά στη χρήση τους. Παρουσιάζονται μεθοδολογίες για την επίτευξη αποτελεσματικού τρόπου κατασκευής τους και αναλύονται τα κρυπτογραφικά κριτήρια που πρέπει να διέπουν τις ακολουθίες de Bruijn, ενώ εξηγείται ο λόγος επιδίωξης διατήρησης της υψηλής τιμής της γραμμικής πολυπλοκότητάς τους. Τέλος, αναπτύσσεται μοντέλο επίθεσης προς κατανόηση της σημασίας διατήρησης υψηλής τιμής γραμμικής πολυπλοκότητας.

Κεφάλαιο 5: Πολυπλοκότητα k σφαλμάτων σε Ακολουθίες de Bruijn

Παρουσίαση της πειραματικής προσέγγισης για την παραγωγή των αποτελεσμάτων από αλγόριθμους ανάλυσης κρυπτογραφικών χαρακτηριστικών των ακολουθιών de Bruijn, με χρήση προγραμμάτων των οποίων ο πηγαίος κώδικας είναι γραμμένος σε γλώσσα προγραμματισμού C++.

Κεφάλαιο 6: Επίλογος

Εξαγωγή και ανάλυση των συμπερασμάτων που προέκυψαν από την πειραματική προσέγγιση της έρευνας. Περιορισμοί και συστάσεις για μελλοντική έρευνα.

Βιβλιογραφία. Βιβλιογραφικές και Διαδικτυακές Πηγές.

Παράρτημα Α. Αποτελέσματα μετρήσεων.

Παράρτημα Β. Αποτύπωση πηγαίων κωδίκων των προγραμμάτων που χρησιμοποιήθηκαν κατά τη διάρκεια της έρευνας.

Κεφάλαιο 2
Κρυπτογραφία

Απ τη στιγμή που ο άνθρωπος ένοιωσε και διαπίστωσε την ανάγκη να διαφυλάξει το περιεχόμενο των διακινούμενων μηνυμάτων του, επινόησε μεθόδους κρυπτογράφησής τους. Προσπάθειες ανά την ιστορία καταδεικνύουν την ανθρώπινη αγωνία αλλά και την επιβεβλημένη αναγκαιότητα διασφάλισης της εμπιστευτικότητας της πληροφορίας. Παράλληλα, είναι εμφανής η ευρηματικότητα των δημιουργών συστημάτων της «πρώιμης» κρυπτογράφησης, που, αρχικά, με απλοϊκά εργαλεία και μεθόδους, σταδιακά με πιο σύνθετα, ανέπτυξαν κρυπτογραφικά συστήματα.

Η κρυπτογραφία έχει φύγει οριστικά από τα στεγανά των μυστικών υπηρεσιών και τη στρατιωτική χρήση και είναι έτοιμη να προσφέρει ακόμη περισσότερο τις υπηρεσίες της στο σύνολο της ανθρωπότητας πλέον, προάγοντας τη δημοκρατία, τον σεβασμό της ιδιωτικής ζωής, και τελικά την ενεργό και ισότιμη συμμετοχή όλων στο οικονομικό, πολιτικό, και κοινωνικό γίγνεσθαι.

Η εποχή μας, με την αλματώδη ανάπτυξη της τεχνολογίας και των τηλεπικοινωνιών, την εξάπλωση του διαδικτύου και των εφαρμογών που καλύπτει αυτό, καθιστά την κρυπτογραφία απαραίτητη. Με σκοπό την προστασία των δεδομένων και πληροφοριών στο μέγιστο βαθμό ασφάλειας, με την χρήση των υπολογιστών, αναπτύσσονται μέσω μαθηματικών τεχνικών σύνθετοι αλγόριθμοι, που προάγουν τις κρυπτογραφικές μεθόδους.

Η κρυπτογραφία έχει καταστεί αναπόσπαστο κομμάτι των τεχνολογικών εξελίξεων, είναι δε αντικείμενο ιδιαίτερης ερευνητικής δραστηριότητας που συνέβαλε στη μετεξέλιξή της από τέχνη σε επιστήμη, με αυστηρά κριτήρια και αποδείξεις [54]. Ως πρακτική και μελέτη της απόκρυψης πληροφοριών που διασφαλίζει τη μυστικότητα και ακεραιότητά τους, η σύγχρονη κρυπτογραφία, πιστοποιώντας την ταυτότητα των εμπλεκομένων χρηστών, κάνει χρήση των επιστημονικών πεδίων των μαθηματικών, της επιστήμης των υπολογιστών και των ψηφιακών ηλεκτρονικών [33]. Πρόκειται για τν επιστήμη που ασχολείται με τη μελέτη, την ανάπτυξη και τη χρήση τεχνικών κρυπτογράφησης και αποκρυπτογράφησης, με απώτερο σκοπό την ασφαλή πρόσβαση σε συστήματα και υπηρεσίες, την ανάκτηση και διαχείριση ευαίσθητων δεδομένων, ηλεκτρονικών συναλλαγών και εφαρμογών.

2.1 Κρυπτογραφία, μια Ιστορία Χιλιάδων Ετών. Σύντομη Ιστορική Αναδρομή.

Η κρυπτογραφία, έννοια και εργαλείο με μακρά και εξελικτική πορεία, διασχίζοντας τις ιστορικές περιόδους, χρησιμοποιήθηκε σε δύο πολέμους και συνδέθηκε με τον κρίσιμο ρόλο της προστασίας εθνικών μυστικών και στρατηγικών (Menezes, Katz, VanOorschot, Vanstone, 1996), για να φτάσει τα τελευταία είκοσι χρόνια να αποτελεί αντικείμενο τεράστιας έρευνας. Σε γενικές γραμμές, η ιστορία της κρυπτογραφίας θα ήταν δόκιμο να διαιρεθεί σε τρία στάδια [01]. Στο πρώτο, η κρυπτογράφηση αφορούσε στον τρόπο απεικόνισης των γραμμάτων εντύπως (μελάνι και χαρτί), με διαδικασίες που στηρίχθηκαν σε ανακατατάξεις και αναδιατάξεις των γραμμάτων του αλφαβήτου. Στο δεύτερο, εισάγονται οι κρυπτογραφικές μηχανές, ιδίως κατά την περίοδο του Β΄ Παγκοσμίου Πολέμου, ενώ στο τρίτο και τελευταίο στάδιο, η αλληλεπίδραση των μαθηματικών και υπολογιστών, οδήγησε στα σύγχρονα κρυπτογραφικά συστήματα, τα οποία βεβαίως διαρκώς εξελίσσονται. Η κρυπτογραφία από τέχνη οδηγήθηκε στο να είναι επιστήμη, χωρίς όμως να διαφοροποιηθεί ο στόχος της: η ασφάλεια της επικοινωνίας, μέσω κατασκευής και ανάλυσης πρωτοκόλλων ασφαλείας [26].

2.1.1 Πρώτη Περίοδος Κρυπτογραφίας (1900 π.Χ. – 1900 μ.Χ).

Προσπάθειες κρυπτογράφησης συναντώνται από το 1900 π.Χ. όπου αναπτύχθηκε μεγάλο πλήθος μεθόδων και αλγορίθμων κρυπτογράφησης που στηρίζονταν κυρίως σε απλές αντικαταστάσεις γραμμάτων. Σε αυτή την περίοδο, οι χρησιμοποιούμενες μέθοδοι και αλγόριθμοι δεν απαιτούσαν εξειδικευμένες γνώσεις και πολύπλοκες συσκευές, χρησιμοποιούσαν ως μέσο έντυπη απεικόνιση και στηρίζονταν στην ευφυΐα και την ευρηματικότητα των δημιουργών τους [66].

Ο Ηρόδοτος κάνει αναφορές για κρυπτογραφημένα μηνύματα που μετέφεραν οι αγγελιοφόροι. Μια μικρή σφηνοειδής επιγραφή που ανακαλύφθηκε στις όχθες του ποταμού Τίγρη , οδήγησε στο συμπέρασμα ότι πολιτισμοί που αναπτύχτηκαν στην περιοχή της Μεσοποταμίας ασχολήθηκαν με την κρυπτογραφία από το 1500 π.Χ. . Αργότερα, τον 5ο αιώνα π.Χ. , οι Σπαρτιάτες, για στρατιωτική χρήση, χρησιμοποιούσαν τη «σκυτάλη», μια ξύλινη ράβδο ορισμένης διαμέτρου, στην οποία ήταν τυλιγμένη ελικοειδώς μια λωρίδα περγαμηνής [66].

Ο Ιούλιος Καίσαρας επικοινωνούσε με τους συνεργάτες του χρησιμοποιώντας ένα αλγόριθμο κατά τον οποίο αντικαθιστούσε τα γράμματα του κειμένου με αυτά που βρίσκονται τρείς θέσεις πιο πίσω στο λατινικό αλφάβητο (ο κρυπτογραφικός αλγόριθμος του Καίσαρα), μέθοδος που χρησιμοποιήθηκε ευρύτατα και στους επόμενους αιώνες [66]. Ο Αύγουστος Καίσαρας που διαδέχθηκε τον Ιούλιο Καίσαρα, άλλαξε το αυτοκρατορικό σύστημα ώστε το κάθε γράμμα να αντικαθίσταται από το 2ο επόμενο [71].

Κατά την περίοδο του Μεσαίωνα, στο δυτικό χριστιανικό κόσμο, οποιαδήποτε εμπλοκή με κρυπτολογία ήταν παρεξηγήσιμη ως μορφή αποκρυφισμού και μαύρης μαγείας, με αποτέλεσμα να μην υπάρχει ανάπτυξή της, Αντιθέτως, στον Αραβικό κόσμο, τόσο η κρυπτολογία όσο και τα μαθηματικά, αναπτύσσονται και εξελίσσονται σε τέτοιο βαθμό ώστε οι Άραβες να είναι οι πρώτοι που επινόησαν και χρησιμοποίησαν μεθόδους κρυπτανάλυσης [66].

Την περίοδο της Αναγέννησης, συστηματική χρήση κρυπτογραφικών συστημάτων πραγματοποιήθηκε από τους Βενετούς το 13ο αιώνα, με σκοπό τη διακίνηση διπλωματικής αλληλογραφίας [54].

Δημοσιεύσεις περί κρυπτογραφίας εμφανίστηκαν για πρώτη φορά το 1518 με το σύγγραμμα «Στεγανογραφία» από τον αββά Ιωάννη Τριθέμιο [66], ενώ αργότερα, το 1563, ο Ιταλός φυσικός

και μαθηματικός Giambattista Della Porta, δημοσίευσε το «De furtivis literarum notis» («Περί κρυπτικών συμβόλων και γραμμάτων») [66]. Έκτοτε η κρυπτογραφία απέκτησε ενδιαφέρον και άρχισε να εφαρμόζεται ευρέως[54].

Πολλοί φιλόσοφοι ενδιαφέρθηκαν για την κρυπτογραφία. Μεταξύ αυτών, ο Sir Francis Bacon (1561-1626) που χρησιμοποιούσε σύστημα αντικατάστασης ενός γράμματος από μια λέξη πέντε γραμμάτων και ο Leonardo Da Vinci (1452-1519) μέθοδο με καθρέπτη.

Από τη μέθοδο κρυπτογράφησης που εισήγαγε ο Γάλλος κρυπτογράφος Blaise de Vigenère [67], σημαντικός εκπρόσωπος της περιόδου, ο πίνακας πολυαλφαβητικής αντικατάστασης χρησιμοποιείται ακόμα στις μέρες μας.

Το 18ο αιώνα εμφανίζονται συσκευές, οι οποίες χρησιμοποιούνται αποκλειστικά για το σκοπό της κρυπτογραφίας. Μια τέτοια είναι ο κύλινδρος Jefferson, ο οποίος αποτελείται από 36 δίσκους και εκτελεί κρυπτογράφηση πολυαλφαβητικής αντικατάστασης [39].

2.1.2 Δεύτερη Περίοδος Κρυπτογραφίας (1900 – 1950).

Το δεύτερο στάδιο, καλύπτει την περίοδο μέχρι το μέσο του 20ου αιώνα, περιλαμβάνοντας τους δύο παγκόσμιους πολέμους, κατά τους οποίους η ανάγκη διακίνησης ζωτικών πληροφοριών με ασφάλεια ανέπτυξαν την κρυπτογραφία σημαντικά, όσο δεν είχε αναπτυχθεί τα προηγούμενα 3000 χρόνια.

Χρησιμοποιούνται για στρατιωτικούς σκοπούς πολύπλοκες μηχανικές και ηλεκτρομηχανικές κατασκευές, οι οποίες ονομάζονται κρυπτομηχανές. Για την κρυπτανάλυση τέτοιων συστημάτων απαιτείται μεγάλος αριθμός προσωπικού και εργατοωρών για μακρύ χρονικό διάστημα, ενώ είναι εμφανής η ανάγκη για μεγάλη υπολογιστική ισχύ [66]. Τα συστήματα αποκτούν μεγάλη πολυπλοκότητα, παρόλα αυτά η κρυπτογράφηση της περιόδου ήταν επιτυχημένη.

Η πιο κυρίαρχη τεχνική του Β' Παγκοσμίου Πολέμου είναι οι μηχανές κινούμενου ρότορα. Το κείμενο κρυπτογραφείται μέσω των διαδοχικών γραναζιών, όπου το καθένα εκτελεί μια μονο-αλφαβητική αντικατάσταση [39]. Γνωστότερη τέτοια κατασκευή είναι η μηχανή Enigma που χρησιμοποιήθηκε με διάφορες παραλλαγές από τους Γερμανούς για κρυπτογράφηση

ραδιοτηλεπικοινωνιών και ήταν ίσως το πιο εξελιγμένο κρυπτοσύστημα της εποχής, εξ ου και πυροδότησε μια από τις εντονότερες προσπάθειες αποκρυπτογράφησης στην ιστορία.

2.1.3 Σύγχρονη εποχή (1950 – σήμερα).

Αρχικά ο Auguste Kerchoff το 1883 και στη συνέχεια ο Claude Shannon το 1948 και το 1949, παρουσίασαν καινοτόμες επιστημονικές εργασίες για τις επικοινωνίες και την ασφάλεια της πληροφορίας εισάγοντας, ο μεν Kercohoff, τη φιλοσοφία της ύπαρξης κλειδιού ως μυστική ποσότητα διαφύλαξης της ασφάλειας συστήματος[63], ο δε Shannon, θεμελιωτής της Θεωρίας Πληροφορίας, την έννοια του κρυπτοσυστήματος και της απόλυτης ασφάλειας. Η ιδέα της ύπαρξης κλειδιού επεκτάθηκε από τους Diffie-Hellman το 1976 με την πρόταση χρήσης δημόσιου κλειδιού, ενώ όλοι οι σύγχρονοι αλγόριθμοι σχεδιάζονται υπό το πρίσμα των εννοιών που εισήγαγε ο Shannon.

Οι εργασίες των Rivest, Shamir και Adleman [03] που ακολούθησαν (1977, MIT), πρότειναν το RSA, ένα κρυπτοσύστημα δημόσιου κλειδιού. Στη δεκαετία του 1980 η εξελικτική πορεία της κρυπτογραφίας συνεχίστηκε με τις έρευνες των Goldwasser, Micali [22] κ.ά., οδηγώντας στην Σύγχρονη Κρυπτογραφία στην οποία κεντρικό ρόλο διαδραματίζει η έννοια της αποδείξιμης ασφάλειας και ο κλάδος της υπολογιστικής πολυπλοκότητας.

Οι κρυπτογραφικές μέθοδοι επιτρέπουν τον έλεγχο της ακεραιότητας των μηνυμάτων, τη χρονική τους σήμανση, την αυθεντικοποίηση, την εξουσιοδότηση. Παράλληλα, είναι σε θέση να παρέχουν ανωνυμία και δυνατότητα άρνησης [54].

2.2 Σύγχρονες Εφαρμογές της Κρυπτογραφίας

Η σύγχρονη κρυπτογραφία έχει επεκτείνει σημαντικά το πεδίο εφαρμογής της πέρα από την ιδιωτική επικοινωνία. Στις αρχικές μορφές της, εφαρμόστηκε κυρίως για στρατιωτικούς ή (κυβερνητικούς) διπλωματικούς σκοπούς, οι σημερινές της όμως εφαρμογές καλύπτουν πολύ περισσοτέρους τομείς, όπως είναι η οικονομία, η δημοκρατία, το εμπόριο, η διασφάλιση της ιδιωτικότητας. Εφαρμογές με χρήση κρυπτογραφικών αλγορίθμων, καθιστούν την κρυπτογραφία απαραίτητη καθώς καλύπτουν τους τομείς :

- ασφαλούς επικοινωνίας,
- ασφαλούς πρόσβασης σε συστήματα και υπηρεσίες,
- πιστοποίησης ταυτότητας,
- πρόσβασης και διαχείρισης βάσεων δεδομένων,
- ηλεκτρονικών συναλλαγών,
- ηλεκτρονικών ψηφοφοριών,
- ψηφιακού νομίσματος (κρυπτονομίσματος),
- μη εξουσιοδοτημένη λήψη εμπιστευτικών πληροφοριών προσωπικού χαρακτήρα
- αποθήκευσης και διαχείρισης προσωπικών δεδομένων και στρατιωτικών εφαρμογών.

Κοινό χαρακτηριστικό των παραπάνω, είναι ότι η ασφάλειά τους εδράζεται, όλο και περισσότερο σε αυστηρές μαθηματικές αποδείξεις. Διενεργούνται πλέον ηλεκτρονικές ψηφοφορίες και δημοπρασίες που παρέχουν αποδείξεις ορθότητας των φάσεων λειτουργίας τους ή έγκυρες συναλλαγές χωρίς κεντρική αρχή, μέσω του Bitcoin ή άλλων κρυπτονομισμάτων με απόδειξη εγκυρότητας και συλλογική επικύρωση, εκδίδονται έξυπνες κάρτες, υπάρχουν ιδιωτικά δίκτυα (VPN) και δορυφορικές εφαρμογές, συστήματα βιομετρικής αναγνώρισης, ασύρματα δίκτυα, τηλεφωνία μέσω διαδικτύου(VOIP), καθώς και πλείστες άλλες εφαρμογές που η ασφάλειά τους στηρίζεται αποκλειστικά σε αριθμητικούς αλγόριθμους που επιτρέπουν την αποδοτική εκτέλεση πράξεων με αριθμούς χιλιάδων ψηφίων, ώστε η υπολογιστική δυσκολία, η πολυπλοκότητα των αντίστροφων πράξεων, να είναι τεράστια.

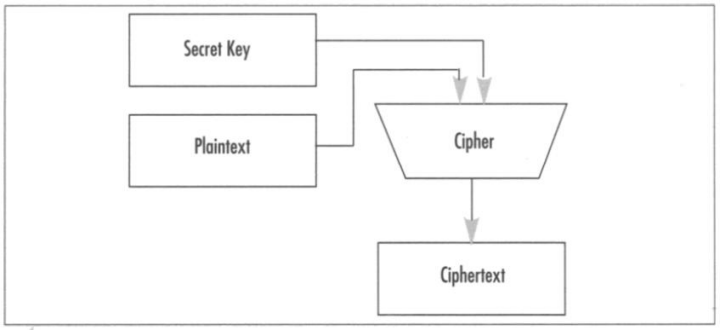
2.3 Η Χρήση Της Κρυπτογραφίας Στη Διασφάλιση Ασφάλειας Των Πληροφοριών.

Επιδίωξη της κρυπτογραφίας είναι η προστασία της πληροφορίας, δεδομένου του ότι αφενός οι δίαυλοι επικοινωνίας είναι ελεύθερα προσπελάσιμοι και αφετέρου ότι υπάρχουν μη πιστοποιημένοι χρήστες που επιθυμούν πρόσβαση στη μεταδιδόμενη ή αποθηκευμένη πληροφορία. Η κρυπτογραφία παρέχει μέσω αλγορίθμων που έχουν αναπτυχθεί, τρόπους αντιμετώπισης για κάθε είδος επιθέσεων καθώς και υπηρεσίες προς εξυπηρέτηση των σκοπών της.

Οι σκοποί που καλύπτονται μέσω της κρυπτογραφίας είναι οι ακόλουθοι :

2.3.1 Εμπιστευτικότητα

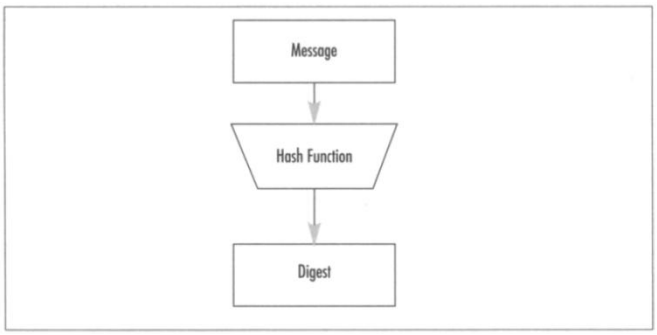
Εμπιστευτικότητα είναι η προστασία αλλά και η απόκρυψη του περιεχομένου της πληροφορίας σε μη εξουσιοδοτημένα άτομα [65]. Επιτυγχάνεται από κρυπτογραφικούς αλγόριθμους συμμετρικού κλειδιού, κατά τους οποίους το μήνυμα, με τη βοήθεια μυστικού κλειδιού, μετατρέπεται σε μη αναγνώσιμη μορφή, το κρυπτοκείμενο. Μόνο η γνώση του κλειδιού επιτρέπει την αναμόρφωση του κρυπτοκειμένου στην αρχική αναγνώσιμη μορφή του (Εικόνα 2.1).



Εικόνα 2.1 : Χρήση Κρυπτογραφίας για Κάλυψη Εμπιστευτικότητας [10].

2.3.2 Ακεραιότητα

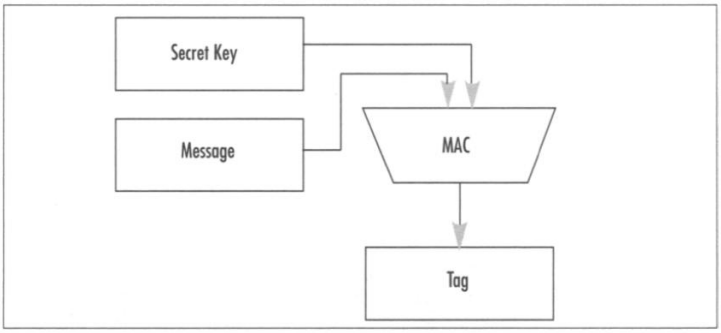
Ακεραιότητα είναι η μη παραποίηση, δηλαδή τροποποίηση, εισαγωγή ή διαγραφή δεδομένων του αρχικού μηνύματος [65]. Αναγνώριση τέτοιου είδους επιθέσεων επιτυγχάνεται μέσω των λεγόμενων συναρτήσεων κατακερματισμού (hash functions), μέσω των οποίων παράγεται ένα ψηφιακό αποτύπωμα σταθερού μήκους που εξαρτάται από το αρχικό μήνυμα και επισυνάπτεται σε αυτό. Οποιαδήποτε αλλαγή στο μήνυμα θα δημιουργήσει διαφορετικό αποτύπωμα. Σύγκριση του παραγόμενου αποτυπώματος με το αρχικό, οδηγεί σε συμπέρασμα εάν έχει τροποποιηθεί ή όχι το μήνυμα (Εικόνα 2.2).



Εικόνα 2.2 : Χρήση Κρυπτογραφίας για Κάλυψη Ακεραιότητας [10].

2.3.3 Πιστοποίηση

Πιστοποίηση ή αυθεντικοποίηση αποστολέα είναι η απόδειξη γνησιότητας του αποστολέα μηνύματος. Κρυπτογραφικά επιτυγχάνεται μέσω κώδικα αυθεντικοποίησης μηνύματος (MAC), ο οποίος αντιστοιχεί σε συνάρτηση κατακερματισμού όπου υπεισέρχεται μυστικό κλειδί στη λειτουργία της. Το αποτέλεσμα προσαρτάται στο μήνυμα. Ο παραλήπτης, γνωρίζοντας το κλειδί, είναι σε θέση να ελέγξει την αυθεντική ή όχι προέλευση του μηνύματος.



Εικόνα 2.3 : Χρήση Κρυπτογραφίας για Κάλυψη Πιστοποίησης [10].

Άλλος τρόπος πιστοποίησης ταυτότητας είναι με χρήση ψηφιακής υπογραφής, η οποία βασίζεται σε αλγόριθμους ασύμμετρης κρυπτογράφησης.

2.3.4 Κατηγοριοποίηση των Επιθέσεων

Μια κατηγοριοποίηση των επιθέσεων ασφάλειας με σκοπό το πλήγμα κάποιων από τους παραπάνω σκοπούς, είναι :

1. Οι παθητικές επιθέσεις, κατά τις οποίες απλά παρακολουθείται το κανάλι επικοινωνίας, υποκλέπτεται η πληροφορία χωρίς να παραποιείται. Τέτοιου είδους επιθέσεις είναι δύσκολο να ανιχνευθούν, είναι εύκολο όμως να αποτραπούν. Σκοπός τους είναι το πλήγμα της εμπιστευτικότητας ή η ανάλυση κίνησης (traffic analysis) [37].

2. Οι ενεργητικές επιθέσεις, κατά τις οποίες υπάρχει ενεργή συμμετοχή και παρέμβαση του επιτιθέμενου. Σε τέτοιου είδους επιθέσεων αφού υποκλαπεί η πληροφορία παραποιείται, ή γίνεται προσπάθεια διακοπής της μετάδοσής της. Ενεργητικές επιθέσεις είναι δύσκολο να αποτραπούν, όμως είναι εύκολο να ανιχνευτούν. Σκοπός τους είναι το πλήγμα :
   - της διαθεσιμότητας, με διακοπή επικοινωνίας,
   - της ακεραιότητας, με αλλοίωση δεδομένων,
   - της πιστοποίησης, με πλαστογραφία [37].

2.4 Κρυπτογραφικές Επιθέσεις

Τα διάφορα κρυπτοσυστήματα έχουν δεχθεί πολλές και διαφορετικές επιθέσεις, που κατηγοριοποιούνται ανάλογα με την αυξητική τάση του βαθμού ισχύος τους. Κοινός στόχος των επιθέσεων είναι η μη εξουσιοδοτημένη απόκτηση του μηνύματος, σπάζοντας το κρυπτοκείμενο [54], για το λόγο αυτό οι επιδόσεις ασφαλείας ενός κρυπτοσυστήματος, αποτελούν κύριο μέλημα. Το κρυπτοσύστημα είναι σε θέση να ισχυριστεί ότι είναι αρκετά ασφαλές, μόνο όταν αντέξει την αξιολόγηση της ασφάλειας με κρυπτανάλυση [34]. Παρακάτω, παρουσιάζονται σε αύξουσα σειρά ισχύος του αντιπάλου, οι κατηγορίες των επιθέσεων.

2.4.1 Επίθεση Μόνο Κρυπτοκειμένου – Ciphertext Only Attack

Πρόκειται για ένα μοντέλο επίθεσης όπου ο εισβολέας υποτίθεται ότι έχει πρόσβαση μόνο σε ένα σύνολο ciphertexts και ενώ δεν έχει κανένα κανάλι που να παρέχει πρόσβαση στο απλό κείμενο πριν από την κρυπτογράφηση, σε όλες τις συγκεκριμένες επιθέσεις (με κρυπτοκείμενο), εξακολουθεί να έχει κάποιες γνώσεις για το απλό κείμενο. Π.χ. ο εισβολέας ενδέχεται να γνωρίζει τη γλώσσα στην οποία γράφεται το απλό κείμενο ή την αναμενόμενη στατιστική κατανομή των χαρακτήρων στο απλό κείμενο. Τα τυπικά δεδομένα πρωτοκόλλου και τα μηνύματα είναι συνήθως μέρος του απλού κειμένου σε πολλά αναπτυσσόμενα συστήματα και συχνά μπορούν να αναγνωριστούν αποτελεσματικά ως μέρος μιας επίθεσης μόνο σε κρυπτοκείμενο σε αυτά τα συστήματα. Εδώ, ο αντίπαλος παρακολουθεί παθητικά το κανάλι επικοινωνίας και συλλέγει κρυπτοκείμενα. Η συγκεκριμένη επίθεση, ισχύει για οποιοδήποτε δημόσιο κανάλι επικοινωνίας.

2.4.2 Επίθεση Γνωστού Μηνύματος – Known Plaintext Attack

Πρόκειται για ιστορικά γνωστή επίθεση, που βασίζεται στη λήψη δειγμάτων τόσο απλού κειμένου όσο και του αντίστοιχου κρυπτογραφημένου ή κρυπτογραφικού κειμένου, για τις διαθέσιμες πληροφορίες που χρησιμοποιούνται για την ανάλυση των δεδομένων, προκειμένου να προσδιοριστεί το μυστικό κλειδί που χρησιμοποιείται για την κρυπτογράφηση των πληροφοριών. Με τις παραδοχές ότι ακόμα και τα απόρρητα πρωτόκολλα περιέχουν μη απόρρητα μηνύματα (π.χ. μηνύματα χειραψίας στην έναρξη επικοινωνίας, τα οποία προέρχονται από ένα συγκεκριμένο σύνολο λέξεων) καθώς και το ότι κρυπτογραφημένα μηνύματα γίνονται

κάποια στιγμή διαθέσιμα [53], ο αντίπαλος, ομοίως με την επίθεση μόνο κρυπτοκειμένου, παραμένει στη θέση του ωτακουστή, γνωρίζοντας όμως κάποια ζεύγη μηνυμάτων και αντιστοίχων κρυπτοκειμένων. Ειδικά στην περίπτωση της μηχανής Enigma που χρησιμοποιήθηκε κατά τη διάρκεια του Β΄ Παγκοσμίου Πολέμου από το γερμανικό στρατό προς κρυπτογράφηση στρατιωτικών και άλλων συναφών μηνυμάτων, το συγκεκριμένο σύνολο λέξεων ήταν γνωστά κρυπτοκείμενα που αντιστοιχούσαν σε μετεωρολογικές προγνώσεις ή - στην Αφρική – τυποποιημένοι χαιρετισμοί, αναφορές που επέτρεψαν στους κρυπτογράφους που εργάζονταν στο Bletchley Park στο Ηνωμένο Βασίλειο, να προχωρήσουν στο σπάσιμο των μηνυμάτων του Enigma. Οι σύγχρονοι ψηφιακοί κώδικες, είναι λιγότερο επιρρεπείς στις «βλάβες» από χρήση της μεθόδου αυτής [64].

2.4.3 Επίθεση Επιλεγμένου Μηνύματος – Chosen Plaintext Attack

Με την εκτέλεση αυτής της επίθεσης, ο αντίπαλος παύει να είναι παθητικός χρήστης και είναι σε θέση να ενεργήσει επιθετικά έχοντας τη δυνατότητα να ζητήσει την κρυπτογράφηση μηνυμάτων επιλογής του. Αν και αρχικά αυτό εμφανίζεται σαν απίθανο σενάριο, εν τούτοις, ένας παράνομος χρήστης μπορεί να ανακτήσει ένα οποιοδήποτε απλό κείμενο από το αντίστοιχο κρυπτογραφικό, με ένα ζευγάρι κρυμμένα κλειδιά αποκρυπτογράφησης [40]. Το ιστορικό παράδειγμα της ναυμαχίας του Midway (1942), όπου το αμερικανικό ναυτικό επιβεβαίωσε τις υποψίες του για επικείμενη επίθεση των Ιαπώνων στην ατόλη Midway στέλνοντας παραπλανητικά ακρυπτογράφητα μηνύματα που περιείχαν την λέξη Midway και παρατηρώντας τις ιαπωνικές επικοινωνίες για πιθανή κρυπτογραφημένη αναμετάδοση (που πράγματι έγινε, συλλέγοντας επικοινωνίες με κρυπτοκείμενα “AF” και συσχετίζοντάς τες με παλαιότερες επικοινωνίες, αφού κωδικοποίησαν το ‘Midway’ σε ‘AF’ – το οποίο ‘AF’ είχε ήδη εντοπιστεί από προηγούμενες αποκρυπτογραφήσεις ως σημείο επίθεσης των Ιαπώνων), είναι χαρακτηριστικό.

2.4.4 Επίθεση Επιλεγμένου Κρυπτοκειμένου – Chosen Ciphertext Attack

Πλέον έχουμε έναν εισβολέα, έναν ισχυρό αντίπαλο που είναι σε θέση να αποκρυπτογραφήσει επιλεγμένα κρυπτοκείμενα. Και αυτή η επίθεση έχει παρατηρηθεί πρακτικά σε πολλές περιπτώσεις. Εκτός από την απόκτηση της κρυπτογράφησης των επιλεγμένων κειμένων, ο

αντίπαλος μπορεί αν πείσει το μυστικό κάτοχο κλειδιού, να αποκρυπτογραφήσει επιλεγμένα κρυπτογραφικά κείμενα της επιλογής του [61]. Εύκολα μπορεί να είναι σε θέση να βγάλει έμμεσα συμπεράσματα από αντιδράσεις σε κρυπτογραφημένα κείμενα όπως την απρόσεκτη απόρριψη κρυπτογραφημένων 'σκουπιδιών' από το πρωτόκολλο, κατόπιν παρεμβολής τροποποιημένων κρυπτοκειμένων στην επικοινωνία τρίτων, τα οποία θα αποκρυπτογραφηθούν σε μη έγκυρα μηνύματα και θα «πεταχθούν» και με τον τρόπο αυτό, να αποκτήσει πολύτιμες αποκρυπτογραφήσεις των τροποποιημένων κρυπτοκειμένων που εισήγαγε στην επικοινωνία. Ας σκεφθούμε μια παρόμοια ενέργεια στον πραγματικό οικονομικό κόσμο, όπως η αγορά ή πώληση μετοχών.

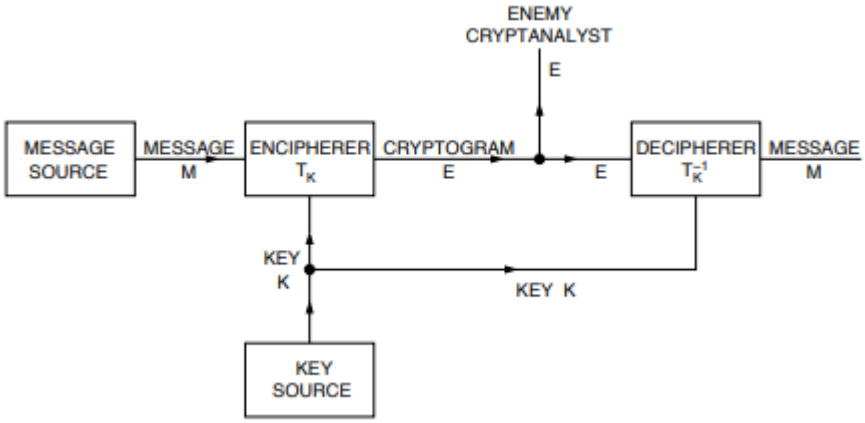
2.5 Η Χρήση Κλειδιών στην Εφαρμογή Κρυπτογραφικών Αλγόριθμων.

Σύμφωνα με τον Auguste Kerchoff, ένα κρυπτογραφικό σύστημα πρέπει να είναι ασφαλές, αν όλα τα στοιχεία για το σύστημα, πλην του κλειδιού, είναι δημοσίως γνωστά [29][63][70]. Η συγκεκριμένη θεωρία, η οποία ήταν μια πρακτική προσέγγιση πάνω σε εφαρμογές στρατιωτικής κρυπτογραφίας, περιλαμβάνει έξι αρχές, με τη δεύτερη να είναι η σπουδαιότερη και ως εκ τούτου καταγεγραμμένη ως «η αρχή του Kerchoff» :

1. Το σύστημα θα πρέπει να είναι πρακτικά, εάν όχι μαθηματικά, ευέλικτο.
2. Δεν απαιτείται να είναι μυστικό και δύναται να αποκτηθεί από τον εχθρό με ευκολία.
3. Το κλειδί πρέπει να είναι μεταδιδόμενο, να διατηρείται χωρίς τη βοήθεια γραπτών σημειώσεων, να μεταβάλλεται και να τροποποιείται με τη θέληση των ανταποκριτών.
4. Πρέπει να εφαρμόζεται στη τηλεγραφική αλληλογραφία.
5. Το σύστημα να έχει τη δυνατότητα υλοποίησης σε οποιοδήποτε περιβάλλον, η χρήση και λειτουργία του να μην απαιτεί εμπλοκή πολλών ατόμων.
6. Είναι απαραίτητο, λαμβάνοντας τις συνθήκες λειτουργίας της εφαρμογή του, να είναι εύκολο στη χρήση, να μην απαιτείται ιδιαίτερη πνευματική ικανότητα, ούτε γνώση εφαρμοσμένων κανόνων.

Η παραπάνω θεμελιώδης αρχή του Kerchoff κατά την οποία το κρυπτοσύστημα υπόκειται σε εκτεταμένες δοκιμές και υφίσταται μεγάλο αριθμό δοκιμαστικών επιθέσεων ώστε να διαπιστωθεί ο πραγματικός βαθμός ασφαλείας που παρέχει, διαμορφώθηκε πληρέστερα από το μαθηματικό Claude Shannon στο άρθρο "Communication theory of secrecy systems" [47].

Αναπτύσσεται η θεωρία ότι «Κάποιος θα έπρεπε να σχεδιάζει συστήματα υπό την προϋπόθεση ότι ο εχθρός θα αποκτήσει αμέσως πλήρη εξοικείωση με αυτά» (μέγιστο Shannon) [70]. Το σύστημα, το οποίο αποτελείται από τα δύο σημεία διακίνησης της πληροφορίας και από μια μονάδα παραγωγής κλειδιού (εικόνα2.5), που παράγει ένα αποκλειστικό για τη μετάδοση κλειδί, το οποίο διαμοιράζεται στους ευρισκόμενους εντός του συστήματος. Η μεταφορά του πραγματοποιείται με τρόπο ασφαλή και αξιόπιστο, για παράδειγμα με αγγελιοφόρο. Το καθαρό μήνυμα που επιθυμείται να αποσταλεί από το σημείο μετάδοσης, κρυπτογραφείται με βάση το κλειδί, και αυτή η κρυπτογραφημένη ποσότητα μπορεί να μεταδοθεί με ένα πιθανώς ανιχνεύσιμο μέσο, π.χ. ραδιόφωνο. Στο τέλος της λήψης του κρυπτογραφήματος, στο σημείο παραλαβής του μηνύματος, το αποκωδικοποιούν συνδυάζοντας με το κλειδί [47].



Εικόνα 2.5 : Γενικό Διάγραμμα συστήματος ασφάλειας κατά Shannon [47].

Η εργασία των Diffie-Hellman το 1976 [11] πρότεινε μια καινοτόμο τεχνική ανταλλαγής κλειδιού από απόσταση, θέτοντας έτσι τη βάση για τη κρυπτογραφία με χρήση δημοσίου κλειδιού [54].

Με τις παραπάνω επιστημονικές εργασίες θεμελιώθηκε η σύγχρονη κρυπτογραφία, όπου εφαρμόζεται από αλγόριθμούς, με χρήση κλειδιών. Ανάλογα με το πλήθος των κλειδιών τα κρυπτογραφικά συστήματα χωρίζονται :

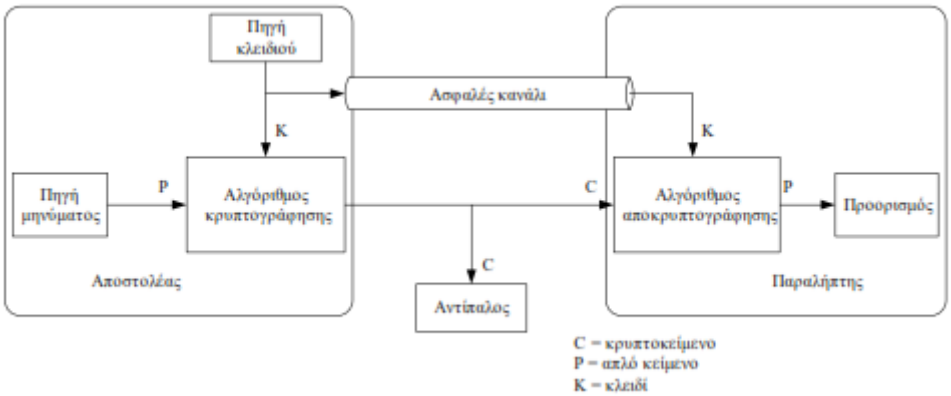
2.5.1 Συμμετρικά Κρυπτογραφικά Συστήματα

Οι αλγόριθμοι που τα υποστηρίζουν ονομάζονται συμμετρικού κλειδιού. Χρησιμοποιείται ένα κλειδί για τη λειτουργία τους, το οποίο καλείται ιδιωτικό κλειδί. Ο αλγόριθμος αποκρυπτογράφησης του κρυπτοκειμένου χρησιμοποιεί το ίδιο κλειδί που χρησιμοποιήθηκε

στον αλγόριθμο κρυπτογράφησής του. Για τη μεταφορά και διανομή του κλειδιού απαιτείται ένα ασφαλές κανάλι [57].

Στα συμμετρικά κρυπτογραφικά συστήματα ο αλγόριθμος αποκρυπτογράφησης εκτελεί τους αντίστροφους μετασχηματισμούς από τον αντίστοιχο αλγόριθμο κρυπτογράφησης, ώστε να δημιουργηθεί το αρχικό «καθαρό» κείμενο από αυτό του κρυπτογραφημένου και το κλειδί [57].

Σχηματικά η λειτουργία ενός τέτοιου συμμετρικού κρυπτοσυστήματος αποτυπώνεται στην εικόνα 2.6.



Εικόνα 2.6 : Συμμετρικό Κρυπτογραφικό σύστημα [57].

Αλγόριθμοι κρυπτογράφησης συμμετρικού κλειδιού χωρίζονται σε δύο κατηγορίες ανάλογα με τη λειτουργία τους :

- στους κρυπταλγόριθμους Ροής (stream ciphers), όπως οι: RC4, ChaCha20 στο πρωτόκολλο SSL/TLS, GSM, Bluetooth και
- στους κρυπταλγόριθμους Τμήματος (block ciphers), όπως οι: DES (το παλαιό πρότυπο κρυπτογράφησης επί δύο δεκαετίες), 3DES, AES (το νυν πρότυπο κρυπτογράφησης).

2.5.2 Ασύμμετρα Κρυπτογραφικά Συστήματα

Οι αλγόριθμοι που τα υποστηρίζουν ονομάζονται επίσης δημοσίου κλειδιού. Για τη λειτουργία τους απαιτείται χρήση δύο κλειδιών. Το κλειδί που χρησιμοποιείται κατά τη διαδικασία της κρυπτογράφησης ονομάζεται δημόσιο, το οποίο διατίθεται ελεύθερα, χωρίς απαίτηση ασφαλούς καναλιού επικοινωνίας, ενώ το κλειδί που χρησιμοποιείται κατά τη διαδικασία

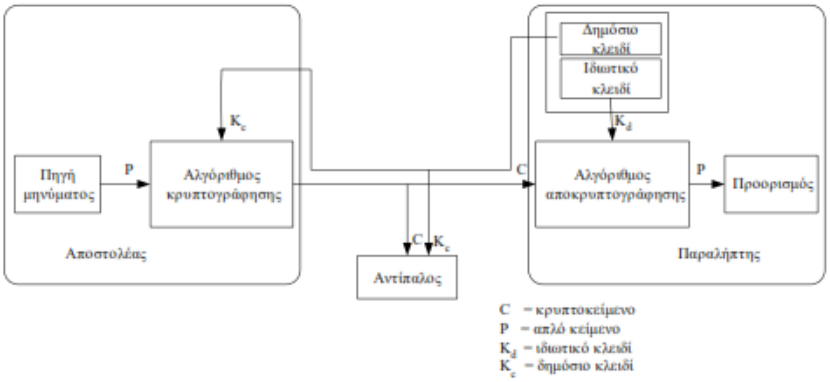
αποκρυπτογράφησης ονομάζεται ιδιωτικό, το οποίο βρίσκεται στην κατοχή του παραλήπτη. Τα κλειδιά αυτά εμφανίζονται πάντοτε ανά ζεύγος και είναι μαθηματικά σχετιζόμενα μεταξύ τους (Εικόνα 2.7) [57].

Εικόνα 2.7 : Ασύμμετρο Κρυπτογραφικό σύστημα [57].

Η λειτουργία αυτών των κρυπτογραφικών συστημάτων βασίζεται στην ύπαρξη κατάλληλων κρυπτογραφικών συναρτήσεων μιας κατεύθυνσης (συναρτήσεις με δυσκολία υπολογισμού των αντίστροφών τους) [39].

Για επικοινωνία με χρήση ασύμμετρου κρυπτογραφικού συστήματος (Εικόνα 2.8):

1. Ο αποστολέας ζητά από τον παραλήπτη, για τις ανάγκες της επικοινωνίας, το δημόσιο κλειδί.
2. Ο παραλήπτης, αφού δημιουργήσει ένα ζεύγος κλειδιών(K_e καιK_d), αποστέλλει το δημόσιο κλειδί K_e στον παραλήπτη διαμέσου μη ασφαλούς καναλιού επικοινωνίας.
3. Ο αποστολέας με χρήση του λαμβανόμενου δημοσίου κλειδιού K_e κρυπτογραφεί το αρχικό κείμενοP και στη συνέχει αποστέλλει στον παραλήπτη το κρυπτογραφημένο κείμενο C.
4. Ο παραλήπτης χρησιμοποιώντας το ιδιωτικό κλειδί K_d αποκρυπτογραφεί το παραγόμενο από το βήμα 3, κρυπτοκείμενο C σε αναγνώσιμη μορφή P [57].



Εικόνα 2.8 : Επικοινωνία με Ασύμμετρο Κρυπτογραφικό σύστημα [57].

Ασύμμετρη κρυπτογραφία εκτός από εφαρμογές αποστολής μηνυμάτων χρησιμοποιείται και σε εφαρμογές όπου απαιτείται ψηφιακή υπογραφή. Το ζεύγος δημιουργημένων κλειδιών χρησιμοποιείται αντίστροφα, δηλαδή ο κάτοχος του ιδιωτικού κλειδιού κρυπτογραφεί το αρχικό κατακερματισμένο μήνυμα, δημιουργώντας τη ψηφιακή υπογραφή του, ενώ με το δημόσιο κλειδί επιβεβαιώνεται η ταυτότητα του αποστολέα [57].

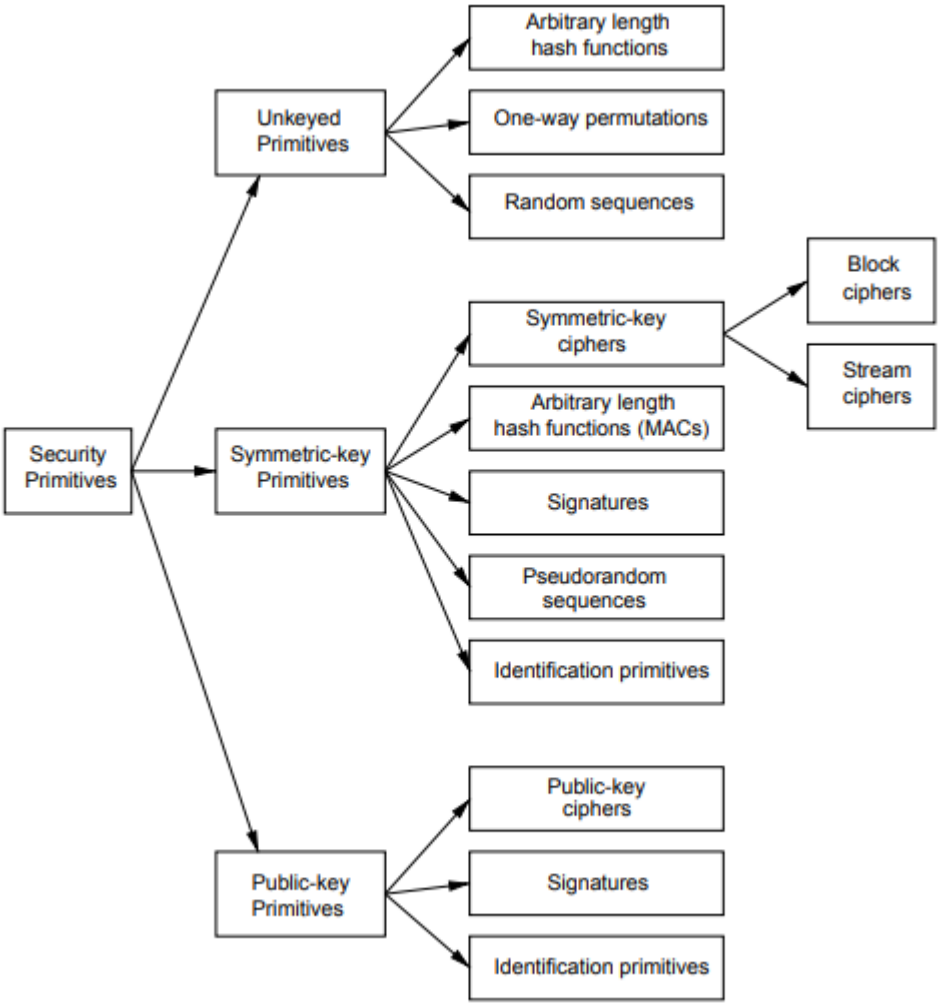
Συγκρίνοντας την συμμετρική κρυπτογραφία με την ασύμμετρη, η δεύτερη -σε ό,τι αφορά στην επιδίωξη του ίδιου επιπέδου ασφαλείας - είναι πιο πολύπλοκη και αργή, ενώ αποδεικνύεται και λιγότερο ασφαλής εάν επιδιώκουμε ίδιο επίπεδο στην ταχύτητα εκτελέσεως. Για τον λόγο αυτό, η ασύμμετρη κρυπτογραφία χρησιμοποιείται, κυρίως, στην ανταλλαγή κλειδιών συμμετρικής κρυπτογραφίας. Το πλέον γνωστό κρυπτοσύστημα ασύμμετρης κρυπτογραφίας είναι το RSA (1978), το οποίο, παρότι αποτελεί εξαιρετικά ασφαλές σύστημα, είναι πολύ αργό, λόγω χρήσεως κλειδιών μήκους τουλάχιστον 1024 bits [55].

Η μεγάλη πρόκληση για τους κρυπτογραφικούς αλγορίθμους δημοσίου κλειδιού είναι το γεγονός ότι, οι πιο γνωστοί εξ αυτών, δεν θα είναι πλέον ασφαλείς στην εποχή των κβαντικών υπολογιστών. Για αυτό και είναι σε εξέλιξη η ανάπτυξη ασφαλών κρυπτοσυστημάτων δημοσίου κλειδιού τα οποία θα ανήκουν στο λεγόμενο χώρο της μετα-κβαντικής κρυπτογραφίας. Οι αλγόριθμοι συμμετρικού κλειδιού, από την άλλη πλευρά, φαίνεται ότι – με κατάλληλ αύξηση του μεγέθους κλειδιού – θα παραμένουν ασφαλείς και στη μετακβαντική εποχή.

2.5.3 Συνοπτική Αναφορά των Κρυπτογραφικών Αλγορίθμων

Συνοπτική αναφορά των κρυπτογραφικών αλγορίθμων και της χρήσης τους αποτυπώνεται στην εικόνα 2.4 , όπου η επιλογή εφαρμογής τους εξαρτάται από τα ακόλουθα κριτήρια :

- επιπέδου ασφάλειας
- λειτουργικότητας
- μεθόδου λειτουργίας, εξαρτάται από το είδος της εφαρμογής
- αποδοτικότητας εκτέλεσης, εξαρτάται από την απόδοση για τον εκάστοτε τρόπο λειτουργίας
- ευκολίας εφαρμογής, εξαρτάται από το επίπεδο πολυπλοκότητας λογισμικού ή υλικού [37].



Εικόνα 2.4 : Κρυπτογραφικοί αλγόριθμοι [37].

Κεφάλαιο 3
Κρυπτογραφικοί Αλγόριθμοι Ροής

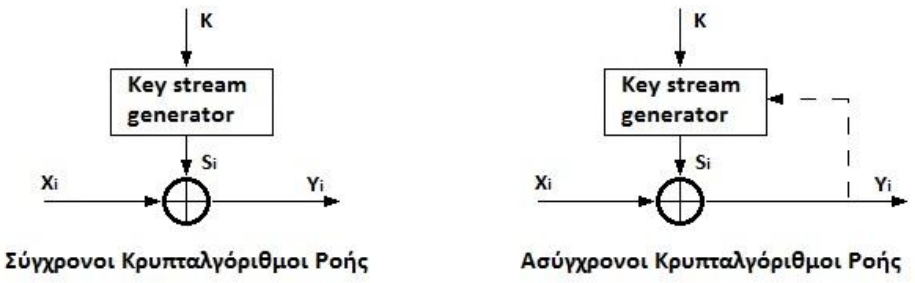
Ο Gilbert S. Vernam, το 1917 ανακάλυψε και υποστήριξε μια εφεύρεση η οποία εκτελούσε τη λειτουργία XOR σε μεμονωμένους παλμούς εισόδου για κωδικοποίηση χαρακτήρων στον κώδικα Baudot [69]. Η NSA αποκάλεσε τη συγκεκριμένη εφεύρεση ως μία από τις σημαντικότερες στην ιστορία της κρυπτογραφίας [58]. Ο Joseph Mauborgne πρότεινε ότι το κλειδί πρέπει να περιέχει τυχαίες πληροφορίες. Με τον συνδυασμό τους, οι δύο παραπάνω ιδέες, οδήγησαν το 1920 στη δημιουργία του σημειωματάριου μιας χρήσης (One Time Pad: OTP) [69]. Στο πεδίο της κρυπτογραφίας έγινε γνωστό ως αλγόριθμος Vernam όπου και εφαρμόστηκε αρχικά σε τηλεγραφικά συστήματα [51] όπου το αρχικό κείμενο αναπαρίσταται ως δυαδική ακολουθία, όπως επίσης και το κλειδί και το κρυπτοκείμενο προκύπτει από την αποκλειστική διάζευξη ανά bit (XOR) του αρχικού κειμένου με το κλειδί.

Ο Claude Shannon στο άρθρο “Communication theory of secrecy systems", αναλύοντας την απεριόριστη ασφάλεια, αναφέρει ότι άνευ όρων ασφάλεια υπάρχει μόνο όταν το κλειδί κρυπτογράφησης έχει μέγεθος ίδιο με το μέγεθος του κειμένου που πρόκειται να κρυπτογραφηθεί, αποδεικνύοντας ότι η τέλεια μυστικότητα επιτυγχάνεται μόνο με το σημειωματάριο μιας χρήσης [47].

Η εφαρμογή των απαιτήσεων του OTP, ενώ αποδεικνύεται ασφαλής, δημιουργεί πρακτικές δυσκολίες τόσο στη δημιουργία όσο και στη διαχείριση και διακίνηση κλειδιών τέτοιου είδους [54]. Κατασκευή κλειδιού με τυχαία τιμή είναι αδύνατο να επιτευχτεί από γεννήτριες ψευδοτυχαίων αριθμών (Pseudo-random generators) με χρήση ντετερμινιστικών μηχανών όπως είναι οι Η/Υ. Πρακτικά είναι ανέφικτη η διανομή τέτοιου μεγέθους κλειδιών σε επικοινωνίες μεγάλης κλίμακας με πολλούς αποδέκτες. Τέλος, είναι άτοπο να διακινηθεί από ασφαλές κανάλι κλειδί μεγέθους ίσο με το κείμενο (που επιθυμείται να μεταφερθεί) και όχι το ίδιο το κείμενο.

Τα παραπάνω προβλήματα προσπάθησαν να καλύψουν οι κρυπτογραφικοί αλγόριθμοι ροής. Από κλειδί μικρού μεγέθους παράγεται δυαδική ακολουθία ψηφίων μεγάλης περιόδου με υψηλό βαθμό τυχαιότητας, η οποία καλείται κλειδοροή, προσομοιάζοντας έτσι το σημειωματάριο μιας χρήσης (OTP) που εισήγαγε ο Vernam.

Οι κρυπτογραφικοί αλγόριθμοι ροής κρυπτογραφούν μεμονωμένα ψηφία (bits) του κειμένου. Αυτό επιτυγχάνεται με την πρόσθεση (XOR) κάθε ψηφίου του κειμένου με το αντίστοιχο της κλειδοροής. Υπάρχουν σύγχρονοι κρυπταλγόριθμοι ροής, στους οποίους η παραγόμενη κλειδοροή εξαρτάται αποκλειστικά από το κλειδί, και ασύγχρονοι, στους οποίους η παραγόμενη κλειδοροή εκτός του κλειδιού εξαρτάται και από το κρυπτοκείμενο. (Εικόνα 3.1) [38].



Εικόνα 3.1 : Σύγχρονοι και Ασύγχρονοι Κρυπταλγόριθμοι Ροής [25].

3.1 Κριτήρια Τυχαιότητας Ακολουθίας Ψηφίων

Ακολουθία δυαδικών ψηφίων ορισμένης περιόδου, θεωρείται τυχαία, όταν δεν υπάρχει μαθηματικός τύπος που μπορεί να την προσδιορίσει. Επειδή καμία ακολουθία η οποία παράγεται από υπολογιστική συσκευή δεν μπορεί να είναι γνήσια τυχαία, εξετάζονται εάν πληρούνται ορισμένες ιδιότητες για την κάθε ακολουθία, οι οποίες συνδέονται με την τυχαιότητα, προκειμένου να χαρακτηριστεί ως ψευδοτυχαία [13]. Οι ιδιότητες τέτοιων ακολουθιών αποτελούν αντικείμενο έρευνας, έχοντας εμφανιστεί στη βιβλιογραφία, από τη δεκαετία του 1950 [05].

Πρώτος ο Golomb στο βιβλίο του «Shift Register Sequences» [24], στα μέσα της δεκαετίας του ’50, εισάγει τρία κριτήρια που πρέπει να πληροί η δυαδική περιοδική ακολουθία. Αργότερα, στο τέλος της δεκαετίας του ’60, ο Elwyn Berlekamp στο βιβλίο του «Algebraic coding theory» [02] περιγράφει ένα αλγόριθμο, ο οποίος μπορεί να κατασκευάσει μια ακολουθία από ορισμένα μόνο διαδοχικά ψηφία της. Εκμεταλλευόμενος αυτό τον αλγόριθμο ο James Massey [36] τον χρησιμοποίησε για να συνθέσει τον LFSR που παράγει την ακολουθία. Το μέγεθος του μικρότερου LFSR που μπορεί να δημιουργήσει την ακολουθία ονομάζεται γραμμική πολυπλοκότητα. Η τιμή γραμμικής πολυπλοκότητας δυαδικής ακολουθίας αποτέλεσε κριτήριο τυχαιότητας [43]. Τέλος δύο ακόμη κριτήρια προστέθηκαν κατά τα μέσα της δεκαετίας του ’80 [43] και τις αρχές της δεκαετίας το ’90, [12] τα οποία ονομάστηκαν αντίστοιχα προφίλ γραμμικής πολυπλοκότητας και σφαιρική γραμμική πολυπλοκότητα ή k-σφαλμάτων γραμμική πολυπλοκότητα [25].

Στη σημερινή εποχή ο NIST (National Institute of Standards and Technology) παραθέτει ένα σετ δοκιμών για γεννήτριες τυχαίων και ψευδοτυχαίων αριθμών (NIST SP 800-22, Απρίλιος 2010) [59], ενώ τα επικαιροποιημένα κριτήρια που πρέπει να πληρούνται περιγράφονται στο NIST SP 800-90A (Δεκέμβριος 2014) [60].

3.1.1 Κριτήρια Τυχαιότητας του Golomb

Στο βιβλίο του «Shift Register Sequences», [24] ο Golomb ορίζει τρία κριτήρια που πρέπει να πληροί η δυαδική περιοδική ακολουθία:

R1. «Balance Property» ή ισο-κατανεμημένο πλήθος 0 και 1.
Σε κάθε περίοδο δυαδικής ακολουθίας το πλήθος των ψηφίων «0» και των ψηφίων «1» θα πρέπει να είναι σχεδόν ίδια. Στην περίπτωση που η ακολουθία έχει περιττό μέγεθος, θα πρέπει να διαφέρουν μόνο κατά ένα [23].

R2. «Run Property» ή τμήμα διαδρομής.
Σε κάθε περίοδο δυαδικής ακολουθίας μεγέθους 2^n-1 θα πρέπει να υπάρχουν 2^{n-2} τμήματα διαδρομής των οποίων οι τιμές εναλλάσσονται. Από αυτά, το ½ θα έχει μέγεθος 1, το ¼ θα έχει μέγεθος 2 και γενικότερα το $\mathbf{1} / \mathbf{2^k}$ θα έχει μέγεθος **k** (1≤k≤n-2) [23].

R3. «Two-level Correlation Property» ή συνάρτηση αυτοσυσχέτισης.
Περίοδος δυαδικής ακολουθίας μεγέθους 2^n-1 συγκρινόμενη με κάθε διαφορετική περίοδο της ίδια ακολουθίας (η οποία προκύπτει από κυκλική μετατόπιση **τ** ψηφίων) θα έχει 2^{n-1}-1 κοινά ψηφία και 2^{n-1} μη κοινά ψηφία [23]. Η συγκεκριμένη ιδιότητα για ακολουθία a0, a1, a2, a3, a4, … περιόδου N=2^n-1 εκφράζεται από τη συνάρτηση :

$$C(\tau) = \sum_{i=0}^{N-1} (-1)^{ai \oplus ai+\tau}$$

όπου **τ** ο αριθμός των ψηφίων μετατόπισης της συγκρινόμενης περιόδου

3.1.2 Ισχυρότερα Κριτήρια Τυχαιότητας

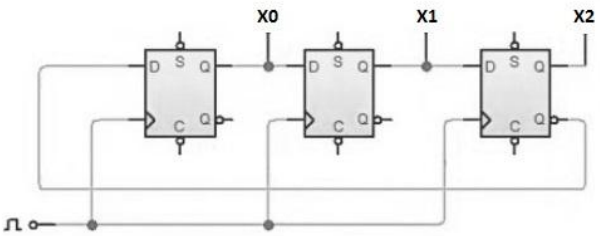
Η ικανοποίηση των κριτήριων τυχαιότητας που έθεσε ο Golomb δεν αρκούν ώστε μια περιοδική δυαδική ακολουθία να χαρακτηριστεί ως ψευδοτυχαία.

Η τιμή, το προφίλ της γραμμικής πολυπλοκότητας και η k-σφαλμάτων γραμμική πολυπλοκότητα ως κριτήριο χαρακτηρισμού ψευδοτυχαίας, καθώς και η επίδρασή τους στη

διαμόρφωση της κρυπτογραφικής ισχύος δυαδικής ακολουθίας, θα αναλυθούν και θα παρουσιαστούν εκτενέστερα στη συνέχεια της μεταπτυχιακής διατριβής.

3.2 Καταχωρητές Ολίσθησης με Ανάδραση (FSR)

Δομικά στοιχεία δημιουργίας ψευδοτυχαίων ακολουθιών είναι οι καταχωρητές ολίσθησης με ανάδραση (Feedback Shift Register - FSR). Έχουν την ιδιότητα ότι δύνανται να παράγουν ακολουθίες μεγάλης περιόδου [49]. Προτιμάται η χρήση τους έναντι άλλων γεννητριών ακολουθιών για συγκεκριμένες εφαρμογές επειδή είναι γρήγοροι, έχουν καλή μαθηματική περιγραφή με εύκολα αναλύσιμες ιδιότητες, ενώ η κατασκευή τους σε επίπεδο Hardware υλοποιείται εύκολα. Μια τυπική διάταξη FSR εμφανίζεται στην εικόνα 3.2. Ακολουθίες κλειδοροής παραγόμενες από FSR χρησιμοποιούνται σε εφαρμογές τηλεπικοινωνιών, πλοήγησης, ασυρμάτων δικτύων, RFID.



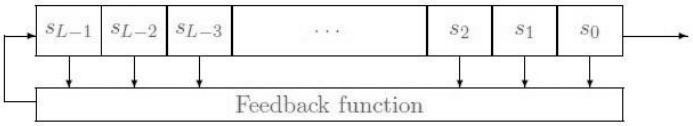
Εικόνα 3.2 : Καταχωρητές σε Διάταξη Ολίσθησης με Ανάδραση [62].

Κάθε καταχωρητής ονομάζεται θέση μνήμης ή βαθμίδα. Το πλήθος των καταχωρητών ορίζει και το μέγεθος του FSR. Ανάλογα με τον βαθμό της συνάρτησης ανάδρασης χωρίζονται σε γραμμικούς LFSR (Linear Feedback Shift Register) και μη γραμμικούς NLFSR. Στην εικόνα 3.3 εμφανίζεται η γενική διάταξη FSR μεγέθους L.

Κάθε βαθμίδα ή μνήμη (S_i) λαμβάνει τιμές από το σώμα GF(2), δηλαδή στο πεπερασμένο σώμα δύο στοιχείων, το 0 και το 1 (εφοδιασμένο με τις πράξεις της πρόσθεσης και του πολλαπλασιασμού mod 2). Το σύνολο των τιμών απ' όλες τις θέσεις μνήμης (S_{L-1}, S_{L-2}, …, S_1, S_0) ονομάζεται κατάσταση του FSR. Οι FSR είναι διατάξεις οι οποίες συγχρονίζονται από εξωτερικό κύκλωμα χρονισμού που ονομάζεται ρολόι (Clock). Με κάθε παλμό χρονισμού αλλάζει η κατάσταση στον FSR. Η πρώτη βαθμίδα (S_{L-1}) λαμβάνει ως τιμή το αποτέλεσμα της συνάρτησης ανάδρασης (Feedback function), ενώ οι υπόλοιπες βαθμίδες (S_{L-2}, S_{L-3}, …, S_1, S_0) λαμβάνουν ως τιμή το περιεχόμενο μνήμης της αμέσως μεγαλύτερης βαθμίδας κατά την προηγούμενη

κατάσταση του FSR (γίνεται ολίσθηση κατά μία θέση). Η τιμή της τελευταίας βαθμίδας (S_0) εξάγεται στην παραγόμενη κλειδοροή. Αρχική κατάσταση στον FSR είναι το κλειδί.

Έτσι, δεχόμενοι ότι κατά τη χρονική στιγμή T_i η κατάσταση στον FSR είναι $X_i = (S_{L-1}, S_{L-2}, \ldots, S_1, S_0)$, τη χρονική στιγμή T_{i+1}, δηλαδή μετά από ένα παλμό του κυκλώματος χρονισμού, η κατάσταση στον FSR θα είναι $X_{i+1} = [f(S_{L-1}, S_{L-2}, \ldots, S_1, S_0), S_{L-1}, S_{L-2}, \ldots, S_2, S_1]$, κ.ο.κ. [41], ενώ η παραγόμενη κλειδοροή θα είναι : $\ldots, S_0, S_1, S_2, \ldots, S_{L-2}, S_{L-1}, f(S_{L-1}, S_{L-2}, \ldots, S_1, S_0), \ldots.$
όπου $f(S_{L-1}, S_{L-2}, \ldots, S_1, S_0)$ είναι το αποτέλεσμα της συνάρτησης ανάδρασης (Feedback function)



Εικόνα 3.3 : Γενική Διάταξη FSR [42]

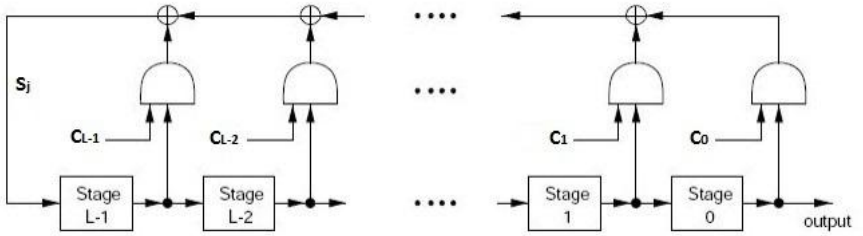
Το πλήθος του συνδυασμού καταστάσεων που είναι σε θέση να δεχτούν οι βαθμίδες ενός FSR είναι πεπερασμένο και η τιμή του ορίζει την περίοδο N της ακολουθίας που θα δημιουργήσει. Δηλαδή ισχύει : $X_{N+i} = X_i$ [48].

3.3 Γραμμικοί Καταχωρητές Ολίσθησης με Ανάδραση (LFSR)

Εύκολα υλοποιήσιμοι οι γραμμικοί καταχωρητές ολίσθησης με ανάδραση (LFSR), χρησιμοποιούνται αρκετά χρόνια ως γεννήτριες παραγωγής κλειδοροών με σχετικά καλά κρυπτογραφικά χαρακτηριστικά τυχαιότητας. Περιγράφονται σαφώς με μαθηματικούς όρους και οι πρωταρχικοί LFSR παράγουν εγγυημένα ακολουθίες με μεγάλη περίοδο, οι οποίες πληρούν και τα τρία κριτήρια τυχαιότητας που όρισε ο Golomb.

Η συνάρτηση ανάδρασης είναι πρώτου βαθμού (ήτοι γραμμική), δηλαδή δεν περιέχει γινόμενα από τις τιμές των καταχωρητών, για το λόγο αυτό ονομάζονται γραμμικοί. Στην εικόνα 3.3. που ακολουθεί, αποτυπώνεται διάταξη γραμμικού καταχωρητή ολίσθησης με ανάδραση, στην οποία διακρίνονται οι L βαθμίδες που ορίζουν και το μέγεθός του, η συνάρτηση ανάδρασης η οποία περιγράφεται ως το άθροισμα των τιμών από τις βαθμίδες που συμμετέχουν, καθώς επίσης και

ο αριθμός των βαθμίδων που μετέχουν στη συνάρτηση ανάδρασης, ο οποίος εξαρτάται από τις τιμές των C_0, C_1, …, C_{L-2}, C_{L-1}.



Εικόνα 3.4 : LFSR Μεγέθους L [37].

Η τιμή S_j, που θα δώσει τιμή στη βαθμίδα L-1, είναι το αποτέλεσμα της συνάρτησης ανάδρασης :
$S_j = f(S_{L-1}, S_{L-2}, \ldots, S_1, S_0) = (C_{L-1} \cdot S_{L-1} \oplus C_{L-2} \cdot S_{L-2} \oplus \ldots \oplus C_1 \cdot S_1 \oplus C_0 \cdot S_0)$,
όπου S_{L-1}, S_{L-2} , …, S_1 , S_0 είναι οι τιμές των βαθμίδων L-1, L-2, …, 1, 0 αντίστοιχα, κατά την προηγούμενη χρονική κατάσταση (προτού εφαρμοστεί παλμός από το κύκλωμα χρονισμού).

Ανάλογα με το ποιές βαθμίδες μετέχουν στη συνάρτηση ανάδρασης, τα C_0, C_1, …, C_{L-2}, C_{L-1} λαμβάνουν τιμές 0 ή 1. Η τιμή του C_0 πρέπει να είναι οπωσδήποτε διαφορετική του 0 ώστε να θεωρηθεί ότι μετέχουν N βαθμίδες στον LFSR [49].

Το πολυώνυμο που περιγράφει την παραπάνω διάταξη είναι :
$P(x) = 1 + C_{L-1} \cdot x + C_{L-2} \cdot x^2 + \ldots + C_1 \cdot x^{L-1} + C_0 \cdot x^L$.

Η παραγόμενη ακολουθία δημιουργείται από τις τιμές που περιέχει η βαθμίδα S_0 σε κάθε παλμό του κυκλώματος χρονισμού. Η τιμή της εξαρτάται από τη συνάρτηση ανάδρασης και από την αρχική τιμή των καταχωρητών, που είναι και το κλειδί. Οι καταστάσεις που λαμβάνει ο LFSR είναι όλοι οι δυνατοί συνδυασμοί πλην της μηδενικής, οι οποίοι εμφανίζονται μόνο μία φορά κατά τη διάρκεια της περιόδου. Η περίοδος των παραγόμενων ακολουθιών πρωταρχικών γραμμικών καταχωρητών ολίσθησης με ανάδραση μεγέθους L, είναι $2^L - 1$. Μειονέκτημα των LFSR αποτελεί το ότι οι ακολουθίες που παράγουν έχουν χαμηλή τιμή γραμμικής πολυπλοκότητας, ίση με τον αριθμό των βαθμίδων τους (μέγεθός τους).

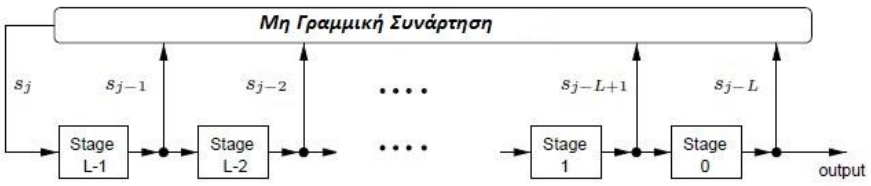
3.4 Μη Γραμμικοί Καταχωρητές Ολίσθησης με Ανάδραση (NLFSR)

Το πρόβλημα της χαμηλής τιμής γραμμικής πολυπλοκότητας σε ακολουθία δημιουργημένη από LFSR, αντιμετωπίστηκε με τη χρήση των μη γραμμικών καταχωρητών ολίσθησης με ανάδραση.

Βασικές διαφορές με τους LFSR είναι :

- Ψηφίο της ακολουθίας εξόδου παράγεται απ' όλες τις δυνατές καταστάσεις του FSR που συμμετέχει.
- Η συνάρτηση ανάδρασης μπορεί να είναι συνδυασμός περισσοτέρων από μία συναρτήσεις.
- Στις συναρτήσεις που λαμβάνουν μέρος, εκτός των πράξεων πρόσθεσης (XOR), μετέχουν και πράξεις γινομένου (AND).
- Η γραμμική πολυπλοκότητα των ακολουθιών που παράγονται από FSR μη γραμμικών συναρτήσεων είναι μεγαλύτερη από τις αντίστοιχες των γραμμικών FSR [55].

Μορφή μη γραμμικών καταχωρητών ολίσθησης με ανάδραση αποτελεί η χρήση μη γραμμικής συνάρτησης ανάδρασης (εικόνα 3.4). Η περίοδος της παραγόμενης ακολουθίας, από μεγέθους N NLFSR, μπορεί να λάβει τη μέγιστη τιμή 2^N, αφού θα περιλαμβάνει όλες τις δυνατές καταστάσεις του FSR.



Εικόνα 3.4 : NLFSR Μεγέθους L με μη γραμμική συνάρτηση ανάδρασης [37].

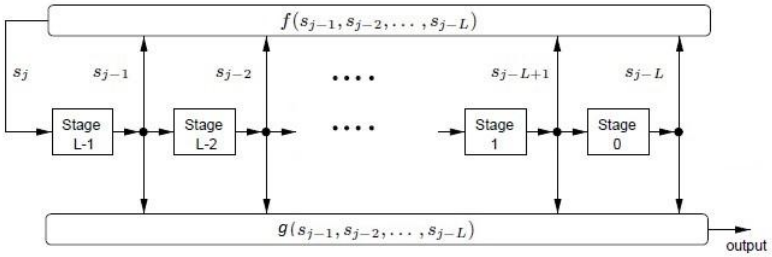
3.5 Γεννήτριες Ακολουθιών Υψηλής Γραμμικής Πολυπλοκότητας

Όπως ήδη αναφέρθηκε, οι LFSR είναι εύκολοι στην υλοποίηση, διαθέτουν σχετικά καλά κρυπτογραφικά χαρακτηριστικά τυχαιότητας και οι πρωταρχικοί LFSR παράγουν εγγυημένα ακολουθίες με μεγάλη περίοδο, οι οποίες πληρούν και τα τρία κριτήρια τυχαιότητας που όρισε ο

Golomb. Τέτοιες διατάξεις όμως παράγουν ακολουθίες χαμηλής γραμμικής πολυπλοκότητας – ίση με το μέγεθος του LFSR. Ακολουθίες χαμηλής γραμμικής πολυπλοκότητας είναι ευάλωτες σε επιθέσεις γνώσης μέρους του αρχικού κειμένου, λόγω του ότι με χρήση του αλγόριθμου Berlekamp-Massey αποκαλύπτεται η δομή του ελάχιστου LFSR που παράγει την ακολουθία – και άρα μπορεί να παραχθεί ολόκληρη η ακολουθία, ξέροντας αρχικά ένα τμήμα αυτής. Προς επίλυση του συγκεκριμένου προβλήματος, οι LFSR δύναται να συνδυαστούν με διατάξεις που θα παρέχουν ακολουθίες με μεγαλύτερη γραμμική πολυπλοκότητα [37].

3.5.1 LFSR με μη Γραμμικά Φίλτρα

Οι LFSR μπορούν να συνδυαστούν με μη γραμμικά φίλτρα για την παραγωγή ακολουθιών υψηλής γραμμικής πολυπλοκότητας (εικόνα 3.5). Η λογική συνάρτηση ανάδρασης f είναι γραμμική. Οι τιμές από τις βαθμίδες του LFSR τροφοδοτούν μια μη γραμμική λογική συνάρτηση g, της οποίας η έξοδος παράγει την ακολουθία κλειδοροής [56].



Εικόνα 3.5 : LFSR Μεγέθους L με μη γραμμικό φίλτρο [37].

Για ισχυρές κρυπτογραφικές ιδιότητες θα πρέπει :

- Το μη γραμμικό φίλτρο να εξασφαλίζει την ισοκατανομή των Bits “0” και “1” στην έξοδό του , δηλαδή να είναι ισοβαρές.
- Το μη γραμμικό φίλτρο να έχει όσο το δυνατόν υψηλό βαθμό. (Βαθμός είναι το πλήθος των μεταβλητών που μετέχουν στο μεγαλύτερο γινόμενο – με την πράξη AND – της συνάρτησης)
- Ο LFSR που τροφοδοτεί το μη γραμμικό φίλτρο να είναι πρωταρχικός και με μεγάλη περίοδο.

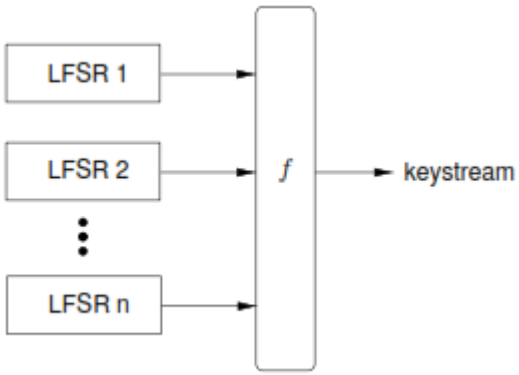
Η περίοδος της παραγόμενης από το μη γραμμικό φίλτρο είναι ίδια με την περίοδο του LFSR. Δηλαδή περίοδος ακολουθίας T(s) που δημιουργείται από μη γραμμικό φίλτρο, το οποίο τροφοδοτείται από LFSR μεγέθους N, θα έχει τιμή $T(s)=2^N - 1$ [13].

Η γραμμική πολυπλοκότητα ακολουθίας που παράγεται από μη γραμμικά φίλτρα είναι :

LS(s) = $\sum_{i=1}^{d}\binom{N}{d}$ όπου N το μέγεθος του LFSR και d ο βαθμός της μη γραμμικής συνάρτησης g του φίλτρου [30].

3.5.2 Μη Γραμμικοί Συνδυαστές

Οι έξοδοι πολλών LFSR χρησιμοποιούνται ως είσοδοι μη γραμμικής λογικής συνάρτησης, η έξοδος της οποίας παράγει την περιοδική ακολουθία κλειδοροής (εικόνα 3.6).



Εικόνα 3.6 : Μη Γραμμικός Συνδυαστής Τροφοδοτούμενος από n LFSR [37].

Επιλογή πρωταρχικών με μεγάλη περίοδο LFSR θα δημιουργήσουν παραγόμενη περιοδική ακολουθία με καλά στατιστικά κρυπτογραφικά χαρακτηριστικά. Η περίοδος της παραγόμενης από τη μη γραμμική λογική συνάρτηση T(s) εξαρτάται από τις επιμέρους περιόδους των παραγομένων από τους LFSR ακολουθιών. Συγκεκριμένα είναι το ελάχιστο κοινό πολλαπλάσιο των τιμών των περιόδων αυτών.

T(s) =lcm $\{\boldsymbol{2^{Ni}-1}\big|\boldsymbol{i=1,2,\dots,n}\}$ [13]

Μέγιστη τιμή που δύναται να λάβει είναι T(s) = $(2^{N1}-1)\cdot(2^{N2}-1)\cdot\dots\cdot(2^{Nn}-1)$, η οποία επιτυγχάνεται με κατάλληλη επιλογή μεγέθους πρωταρχικών LFSR.

Η τιμή της γραμμικής πολυπλοκότητας της παραγόμενης ακολουθίας εξαρτάται από τις επί μέρους τιμές των γραμμικών πολυπλοκοτήτων του κάθε LFSR που λαμβάνει μέρος στη διάταξη, αλλά και της μορφής της μη γραμμικής λογικής συνάρτησης.

Δηλαδή, εάν η μη γραμμική λογική συνάρτηση για εισόδους x1, x2, …, xn είναι g(x)= g(x1, x2, …, xn) , ενώ οι επιμέρους για κάθε LFSR γραμμικές πολυπλοκότητες είναι L1, L2, …, Ln , τότε

γραμμική πολυπλοκότητα της παραγόμενης από τη μη γραμμική συνάρτηση ακολουθίας είναι : LS(s) = g(L1, L2, …, Ln)

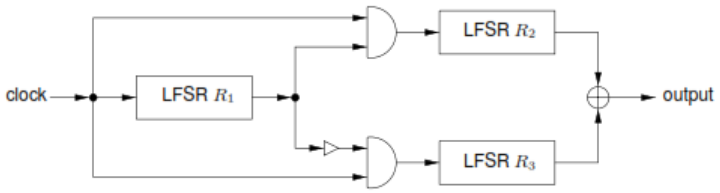
3.5.3 Γεννήτριες Ελεγχόμενες από Ρολόι

Στις ως άνω περιγραφόμενες διατάξεις, ο χρονισμός των FSR πραγματοποιείται από εξωτερικό κύκλωμα, το οποίο ονομάζεται ρολόι (clock) και το σύνολο των FSR δέχεται ταυτόχρονα παλμό χρονισμού από το ίδιο κύκλωμα, ώστε η διάταξη να είναι συγχρονισμένη [37].

Για τη δημιουργία ακολουθιών από γεννήτριες ελεγχόμενες από ρολόι, χρησιμοποιούνται διατάξεις κατά τις οποίες οι LFSR που παράγουν την κλειδοροή χρονίζονται από την έξοδο LFSR προηγούμενης βαθμίδας. Σκοπός τέτοιων διατάξεων είναι η κατασκευή ακολουθιών με μη γραμμικό τρόπο, εφόσον οι LFSR παραγωγής τους χρονίζονται με ακανόνιστο τρόπο [37].

3.5.3.1 Γεννήτρια Εναλλασσόμενου Βήματος

Στη γεννήτρια εναλλασσόμενου βήματος, η έξοδος ενός πρωταρχικού LFSR (R1) ορίζει το χρονισμό των LFSR που συμμετέχουν στην παραγωγή της ακολουθίας (εικόνα 3.7).

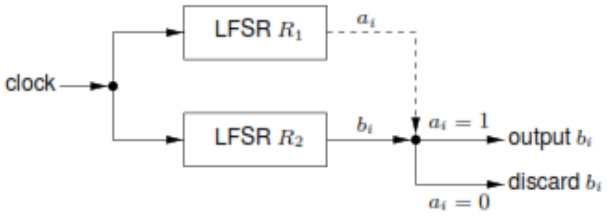


Εικόνα 3.7 : Διάταξη Γεννήτριας Εναλλασσόμενου Βήματος [37].

Η παραγόμενη ακολουθία δημιουργείται από την ένωση των εξόδων των LFSR της τελευταίας βαθμίδας της διάταξης (R2 και R3) με την πράξη XOR.

3.5.3.2 Γεννήτρια Συρρίκνωσης

Στη γεννήτρια συρρίκνωσης η έξοδος ενός πρωταρχικού LFSR (R1) ορίζει εάν θα μετέχει ή όχι το bit εξόδου του LFSR (R2) παραγωγής της ακολουθίας (εικόνα 3.8) [37].



Εικόνα 3.8 : Διάταξη Γεννήτριας Συρρίκνωσης [37].

Όταν η έξοδος του LFSR R1 (a_i) είναι 1, τότε η έξοδος του LFSR R2 (b_i) μετέχει στην παραγόμενη ακολουθία της διάταξης. Αντίθετα, όταν η έξοδος του LFSR R1 (a_i) είναι 0, τότε η έξοδος του LFSR R2 (b_i) απορρίπτεται [37].

3.6 Γραμμική Πολυπλοκότητα και Προφίλ Γραμμικής Πολυπλοκότητας

Προαναφέρθηκε ότι η έννοια της γραμμικής πολυπλοκότητας είναι μέγεθος που χαρακτηρίζει την ισχύ μίας κρυπτογραφικής ακολουθίας που χρησιμοποιείται ως κλειδοροή σε έναν κρυπταλγόριθμο ροής. Συγκεκριμένα, FSR που παράγουν περιοδικές δυαδικές ακολουθίες με χαμηλή γραμμική πολυπλοκότητα θεωρούνται αδύναμοι, διότι κάνοντας χρήση του αλγορίθμου Berlekamp και Massey, αρκούν λίγα μόνο γνωστά ψηφία κειμένου ώστε να αποκαλυφθεί το χαρακτηριστικό πολυώνυμο και εν συνεχεία η διάταξη τους [04].

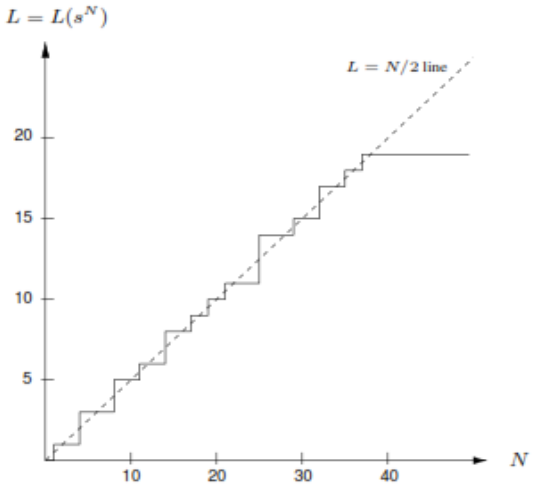
Γραμμική πολυπλοκότητα μιας περιοδικής δυαδικής ακολουθίας ορίζεται ως το μέγεθος του μικρότερου γραμμικού καταχωρητή ολίσθησης με ανάδραση (LFSR) ο οποίος μπορεί να δημιουργήσει την συγκεκριμένη ακολουθία [04].

Για δυαδική, περιοδική ακολουθία S^n = [S0, S1, S2, ….,Sn] η γραμμική πολυπλοκότητά της συμβολίζεται ως $L(S^n)$. Οι ιδιότητες της γραμμικής πολυπλοκότητας είναι :

- Για κάθε $n \geq 1$, ισχύει $0 \leq L(S^n) \leq n$.
- Αν η ακολουθία είναι η μηδενική, S = [0, 0, 0, …], τότε $L(S^n) = 0$, για κάθε $n \geq 1$.
- $L(S^n) = n$ αν και μόνο αν S^n = [0, 0, 0, …, 0, 1].
- Αν δεν υπάρχει γραμμικός καταχωρητής ολίσθησης με ανάδραση ο οποίος να μπορεί να παράγει την ακολουθία S^n, τότε $L(S^n) = \infty$.
- Αν η ακολουθία S^n είναι περιοδική με περίοδο T, τότε $L(S^n) \leq T$.
- Για δύο διαφορετικές ακολουθίες S1 και S2 ισχύει $L(S1 \oplus S2) \leq L(S1)+L(S2)$ [55][37].

Το προφίλ της γραμμικής πολυπλοκότητας για μια παραγόμενη τυχαία ακολουθία, ορίζεται ως η τιμή της γραμμικής πολυπλοκότητας σε σχέση με το μέγεθος της για το κάθε υπο-μέρος της ακολουθίας.

Δηλαδή για μια δυαδική, περιοδική ακολουθία $s_0, s_1, s_2, \ldots$ με γραμμική πολυπλοκότητα L(s), εάν θεωρηθεί ως $L(s^N)$ η γραμμική πολυπλοκότητα της ακολουθίας $s^n = s_0, s_1, s_2, \ldots, s_{n-1}$ - μέρους της αρχικής ακολουθίας - για κάθε N≥1 δημιουργείται το διάγραμμα του προφίλ της γραμμικής πολυπλοκότητας. Για παράδειγμα, για ακολουθία μεγέθους N=20 και s^{20} =1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, το διάγραμμα του προφίλ της γραμμικής πολυπλοκότητας $L(s^N)$ ως προς το μέγεθος N, αποτυπώνεται στην εικόνα 3.9 [37].



Εικόνα 3.9 : Προφίλ της Γραμμικής πολυπλοκότητας Ακολουθίας Μεγέθους L=20 [37].

Οι ιδιότητες που χαρακτηρίζουν τις τιμές για το προφίλ της γραμμικής πολυπλοκότητας της ακολουθίας $s^N = s_0, s_1, s_2, \ldots, s_{N-1}$, μέρους της αρχικής ακολουθίας s = [s0, s1, s2, ….,sn] ορίζονται από τις συνθήκες :

- Εάν J>I τότε $L(s^J) \geq L(s^I)$.
- $L(s^{N+1}) > L(s^N)$ μόνο όταν $L(s^N) \leq N/2$.
- Εάν $L(s^{N+1}) > L(s^N)$ τότε $L(s^{N+1}) + L(s^N) = N+1$ [37].

Μια πραγματικά τυχαία ακολουθία s μήκους N, έχει προφίλ γραμμικής πολυπλοκότητας που ακολουθεί τη συνάρτηση $f[L(s^N)] \approx N/2$ [55].

3.6.1 Αλγόριθμος Berlekamp-Massey

Ένας αποτελεσματικός τρόπος υπολογισμού γραμμικής πολυπλοκότητας δυαδικής ακολουθίας, αλλά και του χαρακτηριστικού πολυωνύμου του LFSR που παράγει την ακολουθία, είναι ο αλγόριθμος Berlekamp-Massey.

Για N ψηφία (bit) ακολουθίας S, η οποία δηλώνεται ως $S^{(N)} = (S_0, S_1, \ldots, S_{N-1})$, χωρίς απαραίτητα να είναι περιοδική, ο αλγόριθμος Berlekamp-Massey υπολογίζει τον LFSR με το μικρότερο αριθμό βαθμίδων ο οποίος μπορεί να παράγει την ακολουθία. Αυτός ο μικρότερος αριθμός βαθμίδων του LFSR που μπορεί να κατασκευάσει την ακολουθία, ονομάζεται γραμμική πολυπλοκότητα της ακολουθίας. Επιτυγχάνεται μέσω μίας αναδρομής N επαναλήψεων, όσα είναι και τα ψηφία της ακολουθίας. Ορίζονται τα χαρακτηριστικά του LFSR ως : LFSR $(f^{(N)}(x), L_N(x))$, όπου $f^{(N)}(x) = 1 + c_1^{(N)} \cdot x + c_2^{(N)} \cdot x^2 + \ldots + c_{LN(S)}^{(N)} \cdot x^{LN(S)}$ το χαρακτηριστικό πολυώνυμο του LFSR και $L_N(x)$ η γραμμική πολυπλοκότητα της ακολουθίας S^N [49].

Ο αλγόριθμος, μέσω της αναδρομής, ενημερώνει το χαρακτηριστικό πολυώνυμο $f^{(n)}(x)$ και τη γραμμική πολυπλοκότητα $L_n(x)$, για κάθε n=1,2, … , N. Δηλαδή $f^{(1)}(x), f^{(2)}(x), \ldots, f^{(n)}(x)$ και $L_1(x)$, $L_2(x)$, … , $L_n(x)$ είναι τα χαρακτηριστικά πολυώνυμα και οι γραμμικές πολυπλοκότητες αντίστοιχα της ακολουθίας $S_0, S_1, \ldots, S_{n-1}$. Ισχύει $f^{(n)}(x) = 1 + \sum_{i=1}^{Ln(S)} c_i^{(n)} \cdot x^i$ [49]

Η διαφορά της ακολουθίας S_n και του (n+1) ψηφίου, το οποίο παράγει ο LFSR $(f^{(n)}(x), L_n(x))$, ορίζεται ως επόμενη απόκλιση $\mathbf{d_n}$. Ισχύει $d_n = S_n + \sum_{i=1}^{Ln(S)} c_i^{(n)} \cdot S_{n-1}$ [49].

Ορίζεται **m** το μήκος της ακολουθίας S_n πριν την τελευταία αλλαγή στην τιμή της γραμμικής πολυπλοκότητας (αριθμός των ελάχιστων βαθμίδων του LFSR παραγωγής). Δηλαδή :
$L_m(S) < L_n(S)$ και $L_{m+1}(S) = L_n(S)$ [49].

LFSR με χαρακτηριστικό πολυώνυμο $f^{(m)}(x)$, με αριθμό βαθμίδων Lm(S), δεν μπορεί να δημιουργήσει ακολουθία $S_0, S_1, \ldots, S_{m-1}, S_m$. Για το λόγο αυτό, $d_m \neq 0$.

Εάν $d_n = 0$ τότε ο LFSR $(f^{(n)}(x), L_n(x))$ παράγει τα n+1 αρχικά ψηφία της ακολουθίας $S_0, S_1, \ldots, S_{n-1}, S_n$. Έτσι $L_{n+1}(S) = L_n(S)$ και $f^{(n)}(x) = f^{(n+1)}(x)$.

Εάν $d_n \neq 0$ τότε πρέπει να υπολογιστούν τα χαρακτηριστικά ενός καινούργιου LFSR που θα μπορεί να παράγει την ακολουθία $S_0, S_1, \ldots, S_{n-1}, S_n$. Γι αυτή την περίπτωση ισχύει :

$f^{(n+1)}(x) = f^{(n)}(x) - d_n d_m^{-1} x^{n-m} f^{(m)}(x)$ και

$Ln+1(S) = MAX [Ln(S), n+1- Ln(S)]$ [49].

1) $1 \rightarrow C(D)$ $\quad 1 \rightarrow B(D)$ $\quad 1 \rightarrow x$
   $0 \rightarrow L$ $\quad 1 \rightarrow b$ $\quad 0 \rightarrow N$
2) If $N = n$, stop. Otherwise compute
$$d = s_N + \sum_{i=1}^{L} c_i s_{N-i}.$$
3) If $d = 0$, then $x + 1 \rightarrow x$, and go to 6).
4) If $d \neq 0$ and $2L > N$, then
   $C(D) - d\, b^{-1} D^x B(D) \rightarrow C(D)$
   $x + 1 \rightarrow x$
   and go to 6).
5) If $d \neq 0$ and $2L \leqq N$, then
   $C(D) \rightarrow T(D)$ [temporary storage of $C(D)$]
   $C(D) - d\, b^{-1} D^x B(D) \rightarrow C(D)$
   $N + 1 - L \rightarrow L$
   $T(D) \rightarrow B(D)$
   $d \rightarrow b$
   $1 \rightarrow x$.
6) $N + 1 \rightarrow N$ and return to 2).

Εικόνα 3.10 : Ο Αλγόριθμος Berlekamp-Massey [36].

Ο ψευδοκώδικας του αλγόριθμου Berlekamp-Massey αποτυπώνεται στην εικόνα 3.10.

Αν μία ακολουθία έχει γραμμική πολυπλοκότητα L και μήκος N, τότε ο ελάχιστου μεγέθους LFSR που την παράγει είναι μοναδικός αν και μόνο αν $L \leq N/2$. Απόρροια αυτού είναι ότι γνώση 2L διαδοχικών bits της ακολουθίας επιτρέπει στον Berlekamp-Massey αλγόριθμο να υπολογίσει τον μοναδικό LFSR μεγέθους L που την παράγει – άρα, τελικά, παράγουμε ολόκληρη την ακολουθία. Για αυτό το λόγο οι κρυπτογραφικές ακολουθίες πρέπει να έχουν υψηλή γραμμική πολυπλοκότητα..

3.6.2 Αλγόριθμος Games-Chan

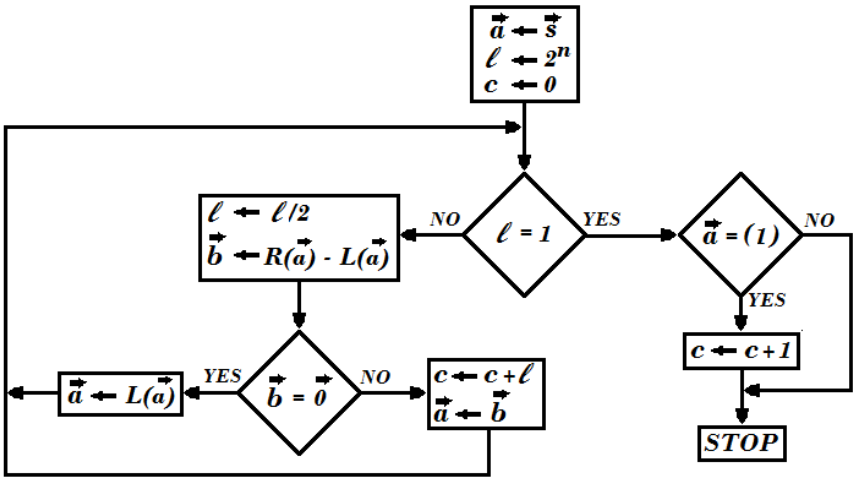
Ο αλγόριθμος Berlekamp-Massey, χρειάζεται δείγμα τουλάχιστον N=2L ψηφίων (bit) ώστε να υπολογιστεί με ακρίβεια η γραμμική πολυπλοκότητα δυαδικής περιοδικής ακολουθίας. Ο αλγόριθμος Games-Chan είναι απλούστερος και έχει τη δυνατότητα να υπολογίσει τη γραμμική πολυπλοκότητα ακολουθίας εισάγοντας αριθμό συνεχόμενων ψηφίων που περιέχονται σε μια περίοδο αυτής.

Για δυαδική ακολουθία μήκους N, όπου $s = (s_0, s_1, \ldots, s_{N-1})$, θεωρούμε (s) το προσαρτημένο αντίγραφο του s, όπου για $i \geq 0$, $(s)_i = s_{i(\mathrm{mod}\, N)}$. Εφόσον η αρχική ακολουθία s είναι περιοδική, τότε

και η ακολουθία (s) θα είναι περιοδική, η οποία μπορεί να έχει μικρότερη περίοδο από αυτή της s. Στην περίπτωση που το μήκος ακολουθίας N είναι δύναμη του 2, μπορεί να υπολογιστεί η γραμμική πολυπλοκότητα σε log N επαναλήψεις, χρησιμοποιώντας έναν απλούστερο αλγόριθμο από αυτόν του Berlekamp-Massey [21]. Το λογικό διάγραμμα του αλγόριθμου Games-Chan αποτυπώνεται στην εικόνα 3.11 και περιγράφεται ακόλουθα.

Αρχικά η μεταβλητή **a** δέχεται την τιμή της ακολουθίας που εξετάζεται ως προς τη γραμμική πολυπλοκότητά της. Η ίδια μεταβλητή **a** σε κάθε επανάληψη του αλγόριθμου θα έχει την τιμή της εξεταζόμενης ακολουθίας. Χρησιμοποιείται μεταβλητή **l** η οποία προσδιορίζει το μήκος της εξεταζόμενης ακολουθίας και αρχικά λαμβάνει τιμή 2^N. Επίσης η μεταβλητή **c** θα δέχεται την τιμή της γραμμικής πολυπλοκότητας με αρχική τιμή 0.

Σε κάθε επανάληψη του αλγορίθμου προστίθεται, εκτελώντας την πράξη XOR, το αριστερό μισό μέρος **L** και το αντίστοιχο δεξιό **R** μισό μέρος της εξεταζόμενης ακολουθίας **a**. Το αποτέλεσμα αποθηκεύεται στη μεταβλητή **b**. Εάν το αποτέλεσμα είναι διαφορετικό της μηδενικής ακολουθίας, τότε αυξάνεται η τιμή **c** της γραμμικής πολυπλοκότητας κατά l/2, όπου **l** είναι το μέγεθος του μήκους της εξεταζόμενης ακολουθίας a. Στην περίπτωση που το αποτέλεσμα της πράξης XOR είναι μηδενικό, δηλαδή τα δύο μισά μέρη της ακολουθίας είναι ίδια, τότε η τιμή **c** της γραμμικής πολυπλοκότητας παραμένει η ίδια, εφόσον η ακολουθία **a** θεωρείται περιοδική και c(s)=c(L). Έτσι, η μεταβλητή **b**, που λαμβάνει την τιμή της προς εξέταση ακολουθίας για την επόμενη επανάληψη του αλγορίθμου, δέχεται την τιμή του μισού αριστερού μέρους της εξεταζόμενης ακολουθίας **a.** Το μήκος της ακολουθίας που βρίσκεται στη μεταβλητή **b** είναι το μισό της αρχικής ακολουθίας **a**. Για το λόγο αυτό, σε κάθε επανάληψη του αλγορίθμου, το μέγεθος της εξεταζόμενης ακολουθίας υποδιπλασιάζεται και ορίζεται ότι l = l /2.



Εικόνα 3.11 : Ο Αλγόριθμος Games-Chan [21].

Επαναλήψεις πραγματοποιούνται έως το μήκος της εξεταζόμενης ακολουθίας **l** λάβει τιμή 1, οπότε δεν μπορεί να υπάρξει περαιτέρω υποδιπλασιασμός της ακολουθίας. Τότε εκτελείται ένας τελευταίος έλεγχος σχετικά με το αν το αποτέλεσμα της προηγηθείσας πράξης είναι ή όχι μηδενικό. Στην περίπτωση μη μηδενικού αποτελέσματος, η μεταβλητή **c** που περιέχει την τιμή της γραμμικής πολυπλοκότητας της αρχικής ακολουθίας, αυξάνεται κατά 1.

3.7 Γραμμική Πολυπλοκότητα k Σφαλμάτων

Ψευδοτυχαίες ακολουθίες που παράγονται από αλγόριθμους ροής και συγκεκριμένα από LFSR, χρησιμοποιούνται ως κλειδοροές προς κρυπτογράφηση και αποκρυπτογράφηση μηνυμάτων που μεταδίδονται από μη ασφαλή κανάλια, με σκοπό την προστασία της πληροφορίας.

Οι περιοδικές αυτές ακολουθίες παρουσιάζουν μετρήσιμα κρυπτογραφικά χαρακτηριστικά, τα οποία καθορίζουν και την κρυπτογραφική ισχύ τους. Πέραν των κριτηρίων τυχαιότητας που περίγραψε ο Golomb [24], σημαντικό μέγεθος κρυπτογραφικής ισχύος είναι η γραμμική πολυπλοκότητα, που περιγράφει το μέγεθος του απλούστερου LFSR που παράγει την συγκεκριμένη ακολουθία. Υψηλή τιμή γραμμικής πολυπλοκότητας διασφαλίζει ισχυρότερα κρυπτογραφικά χαρακτηριστικά της ακολουθίας, εφόσον, όπως προαναφέρθηκε, με γνώση 2L ψηφίων και χρήση του αλγόριθμου Berlekamp-Massey [36] μπορεί να υπολογισθεί το χαρακτηριστικό πολυώνυμο του LFSR που τη δημιουργεί. Στην περίπτωση που η ακολουθία έχει μέγεθος δύναμης του 2 υπάρχει δυνατότητα χρήσης του αλγορίθμου Games-Chan [21]ώστε να μετρηθεί η γραμμική πολυπλοκότητα της ακολουθίας.

Όμως, ως κριτήριο, μόνο η γνώση της τιμής της γραμμικής πολυπλοκότητας, δεν είναι αρκετό. Περιοδική ακολουθία με μορφή (0, 0, 0, …, 0, 1) μεγέθους n ψηφίων, έχει τη μέγιστη δυνατή τιμή γραμμικής πολυπλοκότητας ίση με n, αλλά είναι κρυπτογραφικά αδύναμη, εφόσον η χρήση της ως κλειδοροή, πρακτικά δεν κρυπτογραφεί το κείμενο [50].

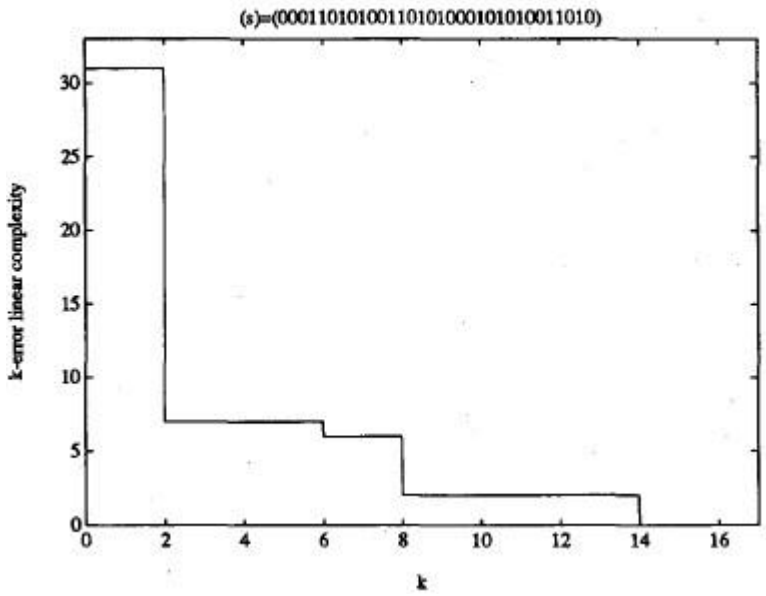
Περιγράφηκε επίσης το μέγεθος του προφίλ της γραμμικής πολυπλοκότητας, όπου αναφέρθηκε ότι σε ακολουθία με καλά χαρακτηριστικά τυχαιότητας πρέπει η σχέση μεταξύ των επιλεγμένων ψηφίων και της γραμμικής πολυπλοκότητάς τους να ακολουθεί τη συνάρτηση $f[L(s^N)] = N/2$.

Οι C. Ding, G. Xiao, και W. Shan, στο «The Stability Theory of Stream Ciphers» (1991) [12], περιγράφουν ένα νέο μετρήσιμο μέγεθος κρυπτογραφικής ισχύς για περιοδική ακολουθία, τη

σφαιρική πολυπλοκότητα (sphere complexity). Ουσιαστικά, εισάγουν την έννοια της Γραμμικής Πολυπλοκότητας k σφαλμάτων (k-error linear complexity) [31].

Οι M. Stamp και C.F. Martin, τον Ιούλιο 1993, αναφέρουν τον όρο k-error linear complexity, όπου ορίζεται ως η μικρότερη τιμή γραμμικής πολυπλοκότητας ακολουθίας της οποίας έχουν τροποποιηθεί k ψηφία και συμβολίζεται $C_k(s)$ [50].

Η σημασία της γραμμικής πολυπλοκότητας k σφαλμάτων είναι η διαμόρφωση της χειρότερης κατάστασης όσον αφορά την τιμή γραμμικής πολυπλοκότητας ακολουθίας, όταν αλλαχθούν k ψηφία της. Στην περίπτωση 0-error, η τιμή της $C_k(s)$ είναι η γραμμική πολυπλοκότητα της μη τροποποιημένης ακολουθίας [50].

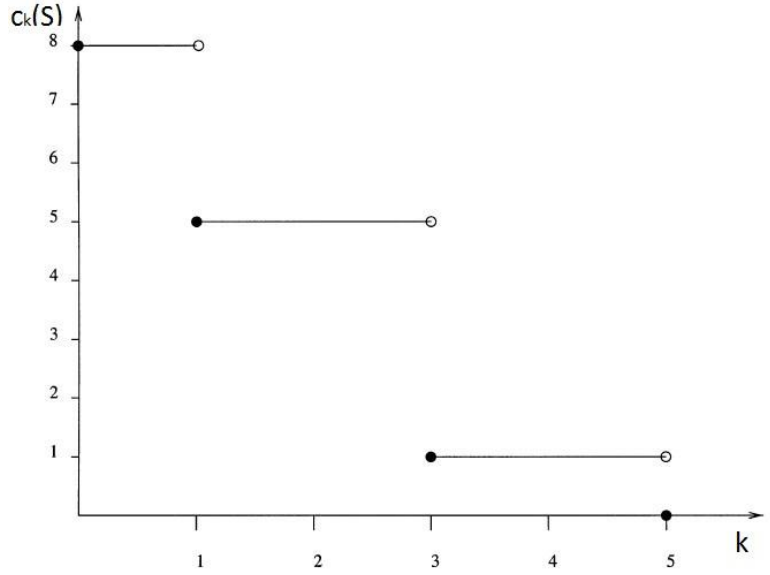


Εικόνα 3.12 : Προφίλ Γραμμικής Πολυπλοκότητας k σφαλμάτων [50].

Για τη δημιουργία του προφίλ γραμμικής πολυπλοκότητας k σφαλμάτων (k-error linear complexity profile) υπολογίζεται η τιμή $C_k(s)$ για k=0, 1, 2, … και αποτυπώνεται σε διάγραμμα $C_k(s) = f(k)$ (εικόνα 3.12).

Τον Ιανουάριο 2003 προστέθηκε, από τους Alan G. B. Lauder και Kenneth G. Paterson, ο όρος **error linear complexity spectrum** (ELCS) ακολουθίας S, ο οποίος περιέχει το πεπερασμένο σύνολο των ζευγών $(k, c_k(S))$ για $0 \leq k \leq \text{cost}(s \to 0)$. Το διάγραμμα ELCS για ακολουθία S αποτυπώνεται στην εικόνα 3.13. Επιπλέον, ορίστηκε η έννοια του **critical point** (CP), ως ζεύγος τιμών $(k, c_k(S))$, τα οποία είναι τα σημεία του διαγράμματος ELCS όπου παρατηρείται μείωση

στην τιμή της γραμμικής πολυπλοκότητας $c_k(S)$. Το φάσμα όλων των κρίσιμων σημείων ακολουθίας S ονομάζεται **critical error linear complexity spectrum** (CELCS) [32].



Εικόνα 3.13 : Διάγραμμα Φάσματος Κρίσιμων Σημείων Γραμμικής Πολυπλοκότητας [32].

3.7.1 Αλγόριθμος Stamp-Martin

Ως αλγόριθμοι υπολογισμού γραμμικής πολυπλοκότητας περιοδικής ακολουθίας περιγράφηκαν ο αλγόριθμος Berlekamp-Massey [33] ή -για ακολουθία με μέγεθος δύναμη του 2- ο αλγόριθμος Games-Chan [21] . Για να υπολογιστεί η γραμμική πολυπλοκότητα k σφαλμάτων ακολουθίας (s) = (s_0, s_1, …, s_{n-1}), ο τετριμμένος τρόπος θα ήταν μέσω υπολογισμού των γραμμικών πολυπλοκοκοτήτων από $\sum_{j=0}^{k} \binom{n}{j}$ ακολουθιών, ο οποίος είναι υπερβολικά δύσκολος ακόμα και για μέτριες τιμές των k και n.

Για περιοδική ακολουθία μεγέθους 2^n, ο αλγόριθμος Games-Chan, χωρίζει σε κάθε επανάληψη του την εξεταζόμενη ακολουθία σε δύο μέρη. Το L περιείχε τα αριστερά ψηφία της και το R τα δεξιά ψηφία. Η ακολουθία b δέχεται το αποτέλεσμα της πράξης L XOR R. Στην περίπτωση όπου L=R, δηλαδή όταν b=0, η τιμή της γραμμικής πολυπλοκότητας δεν αυξάνεται. Αντίθετα, όταν $b \neq 0$ τότε αυξάνεται κατά 2^{n-1}.

Ο προτεινόμενος από τους M. Stamp και C.F. Martin αλγόριθμος υπολογισμού της γραμμικής πολυπλοκότητας k σφαλμάτων, για την περίπτωση που k = 0, λειτουργεί ακριβώς σαν τον αλγόριθμο Games-Chan. Στην περίπτωση που k > 0 επιτρέπεται να αλλαχθεί η ακολουθία σε k ή

λιγότερα ψηφία, ώστε να μειωθεί η τιμή της γραμμικής πολυπλοκότητας όσο το δυνατόν περισσότερο. Με τη λογική του αλγόριθμου Games-Chan, επιδιώκεται η αλλαγή ψηφίων ώστε να επιτευχθεί b=0 [50].

Ο αλγόριθμος Stamp-Martin αποτυπώνεται στην εικόνα 3.14.

```
a = s;  c = 0;  ℓ = 2^n;
cost[i] = 1, for i = 0, 1, ..., ℓ − 1;
while ℓ > 1 do
    ℓ = ℓ/2;  L = a_0 a_1 ··· a_{ℓ−1};  R = a_ℓ a_{ℓ+1} ··· a_{2ℓ−1};
    b = L + R;  T = Σ_{i=0}^{ℓ−1} b_i · min(cost[i], cost[i + ℓ]);
    if T ≤ k then
        k = k − T;
        for i = 0, 1, ..., ℓ − 1 do
            if b_i = 1 then
                if cost[i] ≤ cost[i + ℓ] then
                    L_i = R_i;  cost[i] = cost[i + ℓ] − cost[i];
                else
                    cost[i] = cost[i] − cost[i + ℓ];
                end if
            else
                cost[i] = cost[i] + cost[i + ℓ];
            end if
        a = L;
    else
        c = c + ℓ;
        a = b;
        cost[i] = min(cost[i], cost[i + ℓ]),  for i = 0, 1, ..., ℓ − 1;
    end if
end while
if a_0 = 1 and cost[0] > k then
    c = c + 1;
end if
```

Εικόνα 3.14 : Ο Αλγόριθμος Stamp-Martin [50].

Εισάγεται ως έννοια ο πίνακας **cost[i]** που στοχεύει να μετρήσει το κόστος (το πλήθος των ψηφίων που χρειάζεται να αλλαχθούν από την αρχική ακολουθία) επεμβαίνοντας στο **i** ψηφίο της εξεταζόμενης ακολουθίας **a**, χωρίς να διαταραχθούν τα αποτελέσματα από τα προηγούμενα βήματα επαναλήψεων του αλγόριθμου. Συγκεκριμένα για κάθε επανάληψη του αλγόριθμου, εάν για το **i** ψηφίο της εξεταζόμενης ακολουθίας **a**, το αποτέλεσμα της πράξης $b_i = L_i$ XOR R_i είναι 1, ο αλγόριθμος αλλάζει ένα εκ των L_i ή R_i , ώστε $b_i = 0$. Κόστος αυτής της αλλαγής επιλέγεται να είναι το μικρότερο από τα κόστη αλλαγής L_i και R_i αντίστοιχα, ενώ ενημερώνεται ο πίνακας **cost[i]**. Τέλος, χρησιμοποιείται η μεταβλητή **T**, που για κάθε επανάληψη λαμβάνει την τιμή του συνολικού κόστους αλλαγής ψηφίων, ώστε b=0.

3.7.2 Αλγόριθμος Lauder-Paterson

Το 2003, οι Alan G. B. Lauder και Kenneth G. Paterson, περιγράφουν έναν αλγόριθμο που παράγει φάσμα γραμμικών πολυπλοκοτήτων για ακολουθίες με σφάλματα (error linear complexity spectrum ή ELCS). Πρόκειται για ένα διάγραμμα (εικόνα 3.13) το οποίο παρέχει πληροφορίες για

την συμπεριφορά της γραμμικής πολυπλοκότητας ακολουθίας, καθώς διαφοροποιούνται σε αυτή k ψηφία [32].

Στην ουσία, παράγεται ένα διάγραμμα παρόμοιο με το προφίλ γραμμικής πολυπλοκότητας k σφαλμάτων που δημιουργεί ο αλγόριθμος Stamp-Martin (εικόνα 3.12). Για τη διαμόρφωση όμως του διαγράμματος του προφίλ γραμμικής πολυπλοκότητας k σφαλμάτων, ο αλγόριθμος Stamp-Martin, πρέπει να εκτελεστεί για κάθε k ξεχωριστά μέχρι $c_k(S) = 0$.

Ο αλγόριθμος Lauder-Paterson επιτυγχάνει τη δημιουργία του φάσματος γραμμικών πολυπλοκοτήτων με σφάλματα για μια ακολουθία εκτελούμενος μία φορά, υπολογίζοντας όλα τα κρίσιμα σημεία (critical points), ως ζεύγη τιμών (k, $c_k(S)$), κατά τα οποία παρατηρείται μείωση στην τιμή της γραμμικής πολυπλοκότητας $c_k(S)$.

Για τον σχεδιασμό του αλγορίθμου θεσπίζεται η έννοια της ακολουθίας με κόστος **S**, η οποία έχει τρείς παραμέτρους. Για μήκους l ακολουθία s = s[0], s[1],…, s[l-1] , ορίζεται ως ακολουθία με κόστος S (s, σ, l), όπου **s** η αρχική ακολουθία, **σ** η ακολουθία κόστους για την s και **l** το μήκος της ακολουθίας [32].

```
INPUT: S = (s, σ, ℓ)
OUTPUT: B(S) = (B(s), B(σ), ℓ/2)
  for 0 ≤ i < ℓ/2
    B(s)[i] = s[i] ⊕ s[i + (ℓ/2)]
    B(σ)[i] = min{σ[i], σ[i + (ℓ/2)]}
```

```
INPUT: S = (s, σ, ℓ)
OUTPUT: L(S) = (L(s), L(σ), ℓ/2)
  for 0 ≤ i < ℓ/2
    if s[i] = s[i + (ℓ/2)] then
      L(s)[i] = s[i]
      L(σ)[i] = σ[i] + σ[i + (ℓ/2)]
    if s[i] ≠ s[i + (ℓ/2)] then
       if σ[i] > σ[i + (ℓ/2)] then
         L(s)[i] = s[i]
         L(σ)[i] = σ[i] − σ[i + (ℓ/2)]
       else
         L(s)[i] = s[i + (ℓ/2)]
         L(σ)[i] = σ[i + (ℓ/2)] − σ[i]
```

Εικόνα 3.15 : Ψευδοκώδικας Διαμόρφωσης των B και L [32].

Ορίζονται δύο νέα μεγέθη ακολουθιών με κόστος, το **B** όπου B(S) = (B(s), B(σ), l/2) και το **L** όπου L(S) = (L(s), L(σ), l/2). Η διαμόρφωση αυτών των μεγεθών καθορίζεται από τον ψευδοκώδικα που αποτυπώνεται στην εικόνα 3.15.

Η γραμμική πολυπλοκότητα για δυαδική ακολουθία μήκους 2^n υπολογίζεται :

1. $c(s) = 2^{n-1} + c(B(s))$ εάν $B(s) \neq 0$
2. $c(s) = c(L(s))$ εάν $B(s) = 0$.

Ορίζεται ως **T** το συνολικό κόστος ώστε η ακολουθία B(s) να γίνει μηδενική. Δηλαδή : $T=cost(B(s)\rightarrow 0)$.

Για δεδομένο k και για ακολουθία με κόστος $S = (s, \sigma, 2^n)$ η $c_k(S)$ υπολογίζεται ως :

1. $2^{n-1} + c_k(B(S))$ όταν $0 \leq k < T$ ή
2. $c_{k-T}(L(S))$ όταν $T \leq k$.

Για ακολουθίας με κόστος $S=(s, \sigma, 2^n)$ ορίζεται ως **critical error linear complexity spectrum** (ELCS) το σύνολο των σημείων $\{(k, c_k(S)) : 0 \leq k \leq cost(s\rightarrow 0)\}$. Το ELCS της ακολουθίας S είναι η ένωση των σημείων ELCS για τις ακολουθίες κόστους B(S) και L(S). Δηλαδή :
Για ακολουθία με κόστος $S = (s, \sigma, 2^n)$ υπάρχει ακολουθία $B(S) = (B(s), B(\sigma), 2^{n-1})$ με $T=cost(B(s)\rightarrow 0)$ και ακολουθία $L(S) = (L(s), L(\sigma), 2^{n-1})$ με $U=cost(L(s)\rightarrow 0)$. Εάν υποθέσουμε ότι τα σημεία ELCS για την ακολουθία B(S) είναι $\{(k, c_k(B(S))) : 0 \leq k \leq T\}$ και σημεία ELCS για την ακολουθία L(S) είναι $\{(k, c_k(L(S))) : 0 \leq k \leq U\}$, τότε τα σημεία ELCS της ακολουθίας $S = (s, \sigma, 2^n)$ είναι : $\{(k, 2^{n-1} + c_k(B(S))) : 0 \leq k \leq T\} \cup \{(k+T, c_k(L(S))) : 0 \leq k \leq U\}$ [32].

Τα σημεία στο ELCS των οποίων αυξάνοντας την τιμή του k μειώνεται η τιμή της γραμμικής πολυπλοκότητας k σφαλμάτων ονομάζονται κρίσιμα σημεία **CP** (critical points). Το σύνολο των κρίσιμων σημείων για ακολουθία κόστους S ονομάζεται **critical error linear complexity spectrum** (CELCS) [32].

Το σύνολο των κρίσιμων σημείων (CELCS) της S θα είναι:
Για το B(S) $\{(k_i, ck_i(B(S)) : 0 \leq i \leq t\}$, για κάποια t, όπου $k_0 = 0$ και $k_t = T$.
Για το L(S) $\{(K_i, cK_i(L(S)) : 0 \leq i \leq u\}$, για κάποια u, όπου $K_0 = 0$ και $K_u = U$.
Τότε το CELCS της S θα είναι : $\{(k_0, ck_0(B(S))+2^{n-1}), (k_1, ck_1(B(S))+2^{n-1}), \ldots, (k_{t-1}, ck_{t-1}(B(S))+2^{n-1}), (T+K_0, cK_0(L(S)), (T+K_1, cK_1(L(S)), \ldots, (T+K_u, cK_u(L(S))\}$ [32].

Με τα παραπάνω δεδομένα δημιουργείται ρουτίνα **CELCS** που δέχεται είσοδο την ακολουθία με κόστος προς εξέταση $S = (s, \sigma, l)$ και τρείς τιμές για τις μεταβλητές **tsf**, **lim** και **c**, όπου tsf είναι η τιμή του συνολικού κόστους μέχρι την στιγμή «καλέσματος» της ρουτίνας, lim το όριο του συνολικού κόστους αλλαγών που απαιτείται για την αναζήτηση του κάθε CP και c η τιμή c_k μέχρι την στιγμή «καλέσματος» της ρουτίνας. Ο ψευδοκώδικας της ρουτίνας CELCS αποτυπώνεται στην εικόνα 3.16 [32].

Ο αλγόριθμος Lauder-Paterson βασίζεται στη ρουτίνα CELCS, η οποία καλείται με αναδρομή. Για μήκους 2^n δυαδική ακολουθία s = (s_0, s_1, …, s_{n-1}) δημιουργείται ακολουθία κόστους σ=(σ_0, σ_1, …, σ_{n-1}), η οποία αρχικοποιείται με τιμές 1 (s_i =1 : 0 ≤ i ≤ n-1). Η μεταβλητή **l** λαμβάνει το μήκος της ακολουθίας που εισάγεται στη ρουτίνα CELCS λαμβάνοντας αρχικά τιμή ίση με το μέγεθος της ακολουθίας s (l = 2^n). Η μεταβλητή **tsf** που είναι η τιμή του συνολικού κόστους καθώς και η μεταβλητή **c** που είναι η τιμή της γραμμικής πολυπλοκότητας για k σφάλματα λαμβάνουν μηδενικές αρχικές τιμές (tsf =0, c = 0). Τέλος η μεταβλητή **lim** που ορίζει το όριο του συνολικού κόστους αλλαγών που απαιτείται για την αναζήτηση του κάθε CP αρχικά λαμβάνει τιμή N=2^n (lim = 0), ίση με την υψηλότερη τιμή γραμμικής πολυπλοκότητας που μπορεί να λάβει η ακολουθία s (ουσιαστικά για k = 0). Η πρώτη εκτέλεση της ρουτίνας CELCS γίνεται με παραμέτρους : CELCS ((s, σ, l), 0, N, 0) [32].

```
ALGORITHM CELCS(S = (s, σ, ℓ), tsf, lim, c)

if ℓ > 1
  calculate B(S) and L(S)
  let T = cost (B(s) → 0)
  if T > 0
    CELCS((B(s), B(σ), ℓ/2), tsf,
      min{lim, tsf + T − 1}, c + (ℓ/2))
  if tsf + T ≤ lim
    CELCS((L(s), L(σ), ℓ/2), tsf + T, lim, c)
else \* Case ℓ = 1 *\
  if s[0] = 0
    output (tsf, c)
  if s[0] = 1 and σ[0] > 0
    output (tsf, c + 1)
  if s[0] = 1 and tsf + σ[0] ≤ lim
    output (tsf + σ[0], c)
```

Εικόνα 3.16 : Ψευδοκώδικας της Ρουτίνας CELCS [32].

Σε κάθε εκτέλεση της ρουτίνας CELCS ενημερώνονται άμεσα, από την ακολουθία κόστους B(S) οι ακολουθίες **B(s)** και **B(σ)**, από την ακολουθία κόστους L(S) οι ακολουθίες **L(s)** και **L(σ)**, καθώς και η μεταβλητή **T** η οποία δέχεται την τιμή κόστους ώστε B(s) = 0 (T=cost(B(s)→0)). Έμμεσα, μέσω της διαδικασίας αναδρομικής εκτέλεσής της, ενημερώνονται οι μεταβλητές μήκους της εξεταζόμενης ακολουθίας κόστους **l**, συνολικού κόστους **tsf**, ορίου για το συνολικό κόστος **lim** και τιμής γραμμικής πολυπλοκότητας **c**.

Δημιουργείται ένα δέντρο ώστε να υπολογιστούν τα κρίσιμα σημεία. Όταν ισχύει η συνθήκη ότι T > 0, δηλαδή τα δύο μισά μέρη της εξεταζόμενης ακολουθίας είναι διαφορετικά, καλείται αναδρομικά η CELCS με παράμετρο της εξεταζόμενης ακολουθίας κόστους τη B(S). Σε αυτό το βρόχο κάθε φορά που καλείται η CELCS αυξάνει η τιμή της c κατά l/2 (= 2^{n-1}), η τιμή του tsf δε

μεταβάλλεται, ενώ η τιμή του lim πάντα είναι μικρότερη του T. Αντίστοιχα, με τη συνθήκη ότι ισχύει 0 < k ≤ lim-tsf, τότε tsf + T ≤ lim, καλείται αναδρομικά η CELCS με παράμετρο της εξεταζόμενης ακολουθίας κόστους τη L(S). Η τιμή της μεταβλητής c δεν αυξάνεται, αλλά περιέχει την τιμή που είχε λάβει κατά την τελευταία εκτέλεση της CELCS με παράμετρο τη B(S). Ενημερώνεται η τιμή του συνολικού κόστους tsf (=tsf+T), ενώ η τιμή του lim μένει αμετάβλητη. Η διαδικασία αυτή πραγματοποιείται για κάθε κόμβο του δέντρου.

Η ρουτίνα CELCS καλείται και εκτελείται εφόσον το μήκος της εξεταζόμενης ακολουθίας κόστους είναι μεγαλύτερο της μονάδας (l > 1). Όταν η εξεταζόμενη ακολουθία κόστους αποκτήσει μήκος l=1 τότε εκτελείται το τελευταίο κομμάτι του αλγόριθμου Lauder-Paterson που υπολογίζει τα κρίσιμα σημεία CP στο διάστημα [0, lim-tsf].

3.8 Τιμή Γραμμικής Πολυπλοκότητας Ακολουθίας ως Κριτήριο Κρυπτογραφικής Ισχύος

Ο αλγόριθμος Berlekamp-Massey, λαμβάνοντας ως είσοδο **2l** ψηφία δυαδικής ακολουθίας, όπου l η τιμή της γραμμικής πολυπλοκότητας της ακολουθίας, μπορεί να υπολογίσει το χαρακτηριστικό πολυώνυμο του LFSR ο οποίος την παράγει. Για να εκτιμηθεί η σημαντικότητα διατήρησης της υψηλής τιμής γραμμικής πολυπλοκότητας ακολουθίας, η οποία χρησιμοποιείται ως κλειδοροή κρυπτογράφησης μηνύματος, ακολουθεί περίπτωση και μοντέλο επίθεσης,.

Επιθυμείται να σταλεί κρυπτογραφημένο το κείμενο «**Dear Sir, we would appreciate it if you could send us more detailed information about the project we discussed yesterday**.» χρησιμοποιώντας κλειδοδοροή ακολουθία s. Το κρυπτοκείμενο θα διαμορφωθεί προσθέτοντας κάθε ψηφίο της «κειμενοροής» με το αντίστοιχο της κλειδοροής. Δηλαδή $c_i = m_i \oplus k_i$.

Επιτιθέμενος με πρόσβαση στο κανάλι επικοινωνίας, υποκλέπτει το κρυπτογραφημένο κείμενο. Γνωρίζοντας τα άτομα που συνδιαλέγονται, αλλά και τις και τις συνήθεις προτάσεις που χρησιμοποιούν, κάνει υποθέσεις για τις αρχικές λέξεις του μηνύματος.

- «Good Morning Sir»
- «Dear Sir»
- «Mr. President how are you?»

και άλλες παρόμοιες εκφράσεις. Με τη χρήση υπολογιστών και ενός απλού προγράμματος, γραμμένο σε οποιαδήποτε γλώσσα προγραμματισμού, η διαδικασία που εφαρμόζεται για την

αποκωδικοποίηση κλεμμένου κρυπτοκειμένου είναι πολύ εύκολη και γρήγορη, δίνοντας στον υποκλοπέα τη δυνατότητα πολλών επιλογών στις υποθέσεις αρχικού κειμένου.

Μετατρέποντας κάθε γράμμα του κειμένου, που υπέθεσε ότι υπάρχει στο αρχικό (στο εξής, χάριν συντομίας, θα αποκαλείται «υποθετικό») σε δυαδική μορφή, βάσει της Ascii κωδικοποίησης, δημιουργεί μια «υποθετική» κειμενοροή m'. Έχοντας στην κατοχή του την κλεμμένη κρυπτοροή (το κρυπτοκείμενο σε μορφή ροής δυαδικών ψηφίων), δύναται να κατασκευάσει «υποθετική» κλειδοροή k', εφαρμόζοντας τη σχέση : $k' = m' \oplus c$.

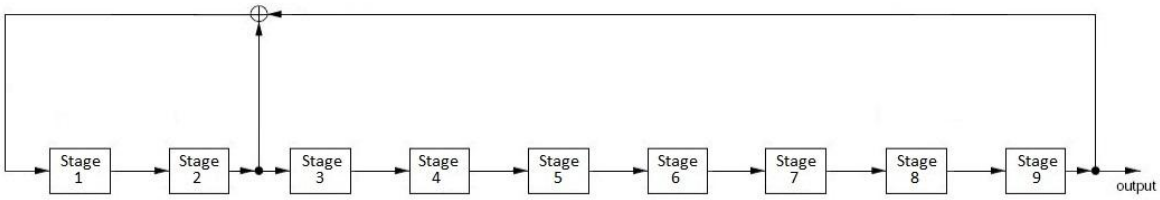
Εισάγοντας στον αλγόριθμο Berlekamp–Massey το πλήθος των ψηφίων που υπολογίστηκαν ως «υποθετική κλειδοροή» k' (από την προηγούμενη σχέση), υπολογίζεται το χαρακτηριστικό πολυώνυμο του LFSR, ο οποίος μπορεί να δημιουργήσει αυτή την «υποτιθέμενη κλειδοροή». Έτσι ο επιτιθέμενος, γνωρίζοντας πλέον τη δομή του LFSR, δύναται να κατασκευάσει όσα ψηφία του χρειάζονται, από την ακολουθία k', ώστε να αποπειραθεί να αποκωδικοποιήσει το κρυπτοκείμενο σε, όσο το δυνατόν περισσότερο αναγνώσιμο κείμενο m' εφαρμόζοντας τη σχέση $m' = k' \oplus c$.

Στην περίπτωση που η ακολουθία η οποία χρησιμοποιήθηκε ως κλειδοροή έχει χαμηλή τιμή γραμμικής πολυπλοκότητας, αρκούν λίγα γνωστά γράμματα του αρχικού μηνύματος, ώστε ο αλγόριθμος Berlekamp–Massey να υπολογίσει χαρακτηριστικό πολυώνυμο LFSR ο οποίος παράγει «υποθετική» ακολουθία παρόμοια με αυτή της κλειδοροής. Αποτέλεσμα θα είναι, αποκρυπτογραφώντας την υποκλεμμένη κρυπτοροή με την ακολουθία που δημιούργησε μέσω του αλγόριθμου Berlekamp–Massey, να παραχθεί κείμενο το οποίο δύναται να αναγνωστεί.

3.8.1 Μοντέλο Επίθεσης 1

Για την παραγωγή της ακολουθίας k περιόδου 511 bits(2^9-1), η οποία θα χρησιμοποιηθεί ως κλειδοροή για την κωδικοποίηση του αρχικού μηνύματος, κατασκευάζεται πρωταρχικός LFSR με χαρακτηριστικό πολυώνυμο $f(x) = x^9 + x^2 + 1$, όπως αυτό απεικονίζεται στην εικόνα 3.17, λαμβάνοντας αρχική τιμή το διάνυσμα [1, 0, 1, 0, 1, 0, 1, 0, 1].

Η παραγόμενη ακολουθία από τον LFSR της εικόνας 3.17, έχει ικανοποιητική περίοδο, όμως η τιμή γραμμικής πολυπλοκότητάς της θεωρείται χαμηλή, εφόσον δε μπορεί να ξεπεράσει τον αριθμό των βαθμίδων του LFSR, δηλαδή 9.



Εικόνα 3.17 : LFSR με χαρακτηριστικό πολυώνυμο $f(x) = x^9 + x^2 + 1$.

Η ακολουθία που παράγεται από την έξοδο του LFSR είναι της μορφής :
k = 1010101011101110110001101101011011011000111001010011 ….

Στο παράρτημα Β.4.1 αποτυπώνεται ο πηγαίος κώδικας προγράμματος σε γλώσσα προγραμματισμού C++, το οποίο δέχεται ως είσοδο αρχείο με το κείμενο προς αναμετάδοση. «**Dear Sir, we would appreciate it if you could send us more detailed information about the project we discussed yesterday.**». Αρχικά, αφού το μετατρέψει σε δυαδική μορφή και αποθηκεύσει αυτή τη ροή δυαδικών ψηφίων του αρχικού μηνύματος στο αρχείο «**OriginalPlainStream.txt**», το κωδικοποιεί με την κλειδοροή, η οποία παράγεται από τον LFSR της εικόνας 3.17. Η κλειδοροή αυτή αποθηκεύεται στο αρχείο «**LFSRKeyStream.txt**». Η συνάρτηση που προσομοιάζει την έξοδο του LFSR είναι η «**LFSROutBit**». Το αποτέλεσμα της κωδικοποίησης είναι το κρυπτοκείμενο, σε μορφή ροής δυαδικών ψηφίων, το οποίο αποθηκεύεται στο αρχείο «**CipherStream.txt**».

Στην συνέχεια το πρόγραμμα εκτελεί την αποκωδικοποίηση του κρυπτοκειμένου με δύο τρόπους: ο πρώτος είναι με τον ίδιο αλγόριθμο που έγινε η κωδικοποίηση (LFSROutBit), ενώ ο δεύτερος χρησιμοποιεί το αρχείο στο οποίο είχε αποθηκευτεί η κλειδοροή (LFSRKeyStream.txt). Και με τους δύο τρόπους εμφανίζεται το αρχικό κείμενο.

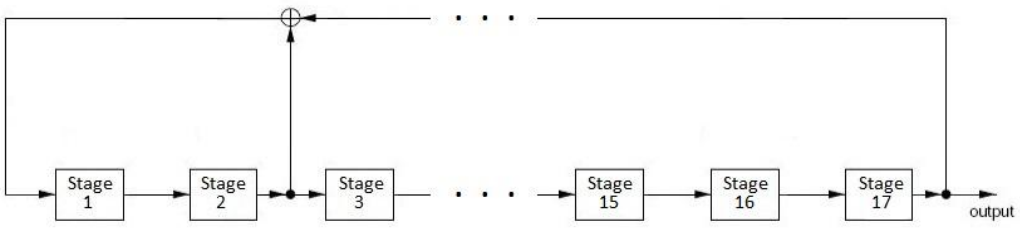
Ο επιτιθέμενος έχει την υποψία ότι το μεταδιδόμενο κείμενο αρχίζει με τη λέξη «Dear» και μετατρέπει τις Ascii τιμές του κάθε χαρακτήρα, από τη λέξη αυτή («Dear»), σε δυαδική μορφή. Με τον τρόπο αυτό δημιουργεί 32 (4*8) ψηφία της κειμενοροής **m'**, που θεωρεί ότι αρχίζει το αρχικό καθαρό κείμενο. Έχοντας πρόσβαση στο ελεύθερο κανάλι επικοινωνίας διαβάζει το κρυπτοκείμενο. Απομονώνει τα πρώτα 32 bits της κρυπτοροής, τα οποία προσθέτει στα αντίστοιχα bits της κειμενοροής m', που είχε υπολογίσει.

Από τον τύπο $k' = m' \oplus c$ δίνεται η δυνατότητα να υπολογιστούν τα 32 πρώτα bits της κλειδοροής, με την εισαγωγή των οποίων στον αλγόριθμο Berlekamp–Massey, εφόσον το πλήθος των ψηφίων είναι μεγαλύτερο από το διπλάσιο της γραμμικής πολυπλοκότητας της

ακολουθίας k που χρησιμοποιήθηκε ως επίσημη κλειδοροή (32 > 2*9), λαμβάνεται το χαρακτηριστικό πολυώνυμο του LFSR που παράγει την ακολουθία κλειδοροής k, καθώς και το μέγεθός του. Έχοντας στην κατοχή του αυτά τα δεδομένα, όπως επίσης και την αρχική κατάσταση του LFSR που είναι τα πρώτα εννέα (9) ψηφία της κλειδοροής που υπολόγισε k', **ο επιτιθέμενος είναι σε θέση να συνθέσει την επίσημη κλειδοροή k και να αποκωδικοποιήσει πλήρως το αρχικό μήνυμα**.

3.8.2 Μοντέλο Επίθεσης 2

Ακολούθως εξετάζεται η περίπτωση να χρησιμοποιείται LFSR μεγέθους 17 βαθμίδων, με χαρακτηριστικό πολυώνυμο $f(x) = x^{17} + x^2 + 1$, ο οποίος παράγει ακολουθία περιόδου 131071 bits ($2^{17} - 1$). Η συγκεκριμένη ακολουθία, με αρχική κατάσταση το διάστημα [1, 0, 1, 0, …, 1, 0, 1], θα χρησιμοποιηθεί ως κλειδοροή k_2 για την κωδικοποίηση και αποκωδικοποίηση του αρχικού μηνύματος (εικόνα 3.18).



Εικόνα 3.18 : LFSR με χαρακτηριστικό πολυώνυμο $f(x) = x^{17} + x^2 + 1$.

Εκτελείται το πρόγραμμα του οποίου ο πηγαίος κώδικας σε γλώσσα προγραμματισμού C++ και αποτυπώνεται στο παράρτημα Β.4.1, με τη διαφορά ότι ορίζεται άλλος LFSR παραγωγής της κλειδοροής k_2. Επιτυγχάνεται μέσω των ορισμών :

```
// LFSR Registers with init values
int Reg[17] = { 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1 };
// FeedBack Function
int FeedBack[17] = { 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0 };
int MaxReg=17; // No of total LFSR Stages
```

Η παραγόμενη ακολουθία κλειδοροής είναι η :

k_2 = 10101010101010101110111011101110111011000110110001101101010110110101…

Κωδικοποιώντας το αρχικό κείμενο με την κλειδοροή k_2 δημιουργείται κρυπτοκείμενο, που σε μορφή ροής δυαδικών ψηφίων είναι :

C_2 = 11101110110011111000111110011100111001101010010101101111…

Επιτιθέμενος, υποκλέπτοντας το κρυπτοκείμενο και υπολογίζοντας τη δυαδική μορφή των Ascii τιμών των τεσσάρων χαρακτήρων της λέξης «Dear», υπολογίζει, από τον τύπο $k_2' = m' \oplus c_2$, τα 32 πρώτα bits της κλειδοροής k_2' τα οποία εισάγει στον αλγόριθμο Berlekamp–Massey. Σημειωτέο ότι, με αυτήν την υπόθεση, ο επιτιθέμενος δεν γνωρίζει 2l bits της ακολουθίας, όπου l η γραμμική πολυπλοκότητα της – άγνωστης σε αυτόν – κλειδοροής, αλλά λιγότερα. Η παραπάνω διαδικασία επιτυγχάνεται με την εκτέλεση του προγράμματος του οποίου ο πηγαίος κώδικας σε γλώσσα προγραμματισμού C++ βρίσκεται στο παράρτημα Β.4.2.

Το πρόγραμμα δέχεται είσοδο το αρχείο κειμένου «**KnownText.txt**» που περιέχει τους χαρακτήρες με τους οποίους ο επιτιθέμενος θεωρεί ότι ξεκινά το αρχικό κείμενο (εδώ, η λέξη «Dear»), και το αρχείο κειμένου «**CipherStream.txt**» που περιέχει το κρυπτοκείμενο σε μορφή ροής δυαδικών ψηφίων, τα περιεχόμενα του οποίου υποκλάπηκαν από το δημόσιο κανάλι επικοινωνίας. Αρχικά μετατρέπονται οι χαρακτήρες του περιεχομένου του αρχείου «KnownText.txt» σε μορφή ροής δυαδικών ψηφίων βάσει της Ascii τιμής του κάθε γράμματος. Το αποτέλεσμα αποθηκεύεται στο αρχείο «**KnownStream.txt**». Στην συνέχεια για κάθε bit του περιεχομένου του αρχείου «KnownStream.txt» εκτελείται η πράξη $m' \oplus c_2$, το αποτέλεσμα της οποίας αποθηκεύεται στο αρχείο «**FakeKeyStream.txt**».
Η υπολογισμένη κλειδοροή k_2' είναι : k_2' = 10101010101010101110111011101110.

Εισάγοντας την τιμή της ακολουθίας k_2' στον αλγόριθμο Berlekamp–Massey, υπολογίζεται το χαρακτηριστικό πολυώνυμο του LFSR παραγωγής της. Επειδή το μέγεθος της ακολουθίας είναι 32 bits, μέγιστος αριθμών βαθμίδων του LFSR μπορεί να είναι μέχρι 16.

Για το συγκεκριμένο μοντέλο, ο αλγόριθμος Berlekamp–Massey υπολόγισε ότι LFSR που δύναται να κατασκευάσει την ακολουθία εισόδου, είναι 16 βαθμίδων με χαρακτηριστικό πολυώνυμο το $f(x) = x^{15} + x^2 + 1$. Αρχική κατάσταση του LFSR θα είναι τα 16 πρώτα ψηφία της κλειδοροής k_2'.

Στην συνέχεια ο επιτιθέμενος παράγει ακολουθία κλειδοροής από LFSR με χαρακτηριστικά που έλαβε από τον αλγόριθμο Berlekamp–Massey και με τα ψηφία της αποπειράται να αποκωδικοποιήσει το υποκλεμμένο κρυπτοκείμενο με την εφαρμογή του τύπου $m' = k_2' \oplus c_2$, ώστε να παραχθεί κείμενο σε αναγνώσιμη μορφή. Το δεύτερο αυτό μέρος της επίθεσης επιτυγχάνεται εκτελώντας το πρόγραμμα, του οποίου ο πηγαίος κώδικας σε γλώσσα προγραμματισμού C++ βρίσκεται στο παράρτημα Β.4.3.

Το πρόγραμμα δέχεται είσοδο το αρχείο κειμένου «**CipherStream.txt**», που περιέχει το κρυπτοκείμενο σε μορφή ροής δυαδικών ψηφίων και τα περιεχόμενά του υποκλάπηκαν από το δημόσιο κανάλι επικοινωνίας. Επίσης χρησιμοποιεί το αρχείο «**FakeKeyStream.txt**» για να δώσει αρχική τιμή στον LFSR. Ακολούθως, αποκωδικοποιεί την ακολουθία του κρυπτοκειμένου με την κλειδοροή, η οποία παράγεται από τον LFSR με τα χαρακτηριστικά που έλαβε από τον αλγόριθμο Berlekamp–Massey. Η διαδικασία εκτελείται για κάθε bit της ακολουθίας του κρυπτοκειμένου με την πράξη $m_i' = k_i' \oplus c_i$. Η προσομοίωση του LFSR γίνεται μέσω της συνάρτησης «**FakeLFSR**» και το αποτέλεσμα αποθηκεύεται σε μορφή καθαρού κειμένου στο αρχείο «**FakePlainText.txt**».

Το συγκεκριμένο κείμενο είναι της μορφής :
Dear¨Ïπι]"Ί'Ιο²JC KφΙ sγKPH'Hm:ωAΘv*]s-
R]Ά7`v†-ν±óU^g†¶qK^'Ω) Ώ0²ρ΄ωDαR#'"›N, ΧΗJΌOM›Δpr\'s TΥ_Ί ;
:-mi](Nu'

το οποίο, πλην των τεσσάρων πρώτων χαρακτήρων, δεν είναι σε αναγνώσιμη μορφή όπως το αρχικό μήνυμα :
«Dear Sir, we would appreciate it if you could send us more detailed information about the project we discussed yesterday. »

3.8.3 Μοντέλο Επίθεσης 3

Για την κρυπτογράφηση του αρχικού κειμένου χρησιμοποιείται κλειδοροή υψηλής τιμής γραμμικής πολυπλοκότητας, η οποία όμως – αλλάζοντας ορισμένα ψηφία της - προσεγγίζει σε μεγάλο βαθμό ακολουθία χαμηλής τιμής γραμμικής πολυπλοκότητας.

Στο παρόν μοντέλο επίθεσης, για την παραγωγή της κλειδοροής χρησιμοποιείται LFSR μεγέθους **717**. Το αρχικό μήνυμα, σε μορφή χαρακτήρων, βρίσκεται στο αρχείο«**Message.txt**», ενώ η κλειδοροή, σε μορφή ροής δυαδικών ψηφίων, στο αρχείο «**LFSRKeyStream3.txt**».

Εκτελώντας – κατάλληλα διαμορφωμένο για τις ανάγκες του μοντέλου επίθεσης 3 - το πρόγραμμα του οποίου ο πηγαίος κώδικας σε γλώσσα προγραμματισμού C++ αποτυπώνεται στο παράρτημα Β.4.1, το αρχικό κείμενο μετατρέπεται σε δυαδική μορφή και στην συνέχεια κωδικοποιείται με τη χρήση της κλειδοροής σε κρυπτοκείμενο το οποίο, σε μορφή ροής δυαδικών ψηφίων, αποθηκεύεται στο αρχείο «**CipherStream3.txt**».

Επιτιθέμενος, υποκλέπτοντας το κρυπτοκείμενο και υποθέτοντας ότι το αρχικό κείμενο αρχίζει από τη λέξη «Dear», μετατρέπει τους 4 χαρακτήρες σε δυαδική μορφή - βάση των Ascii τιμών τους - και στην συνέχεια υπολογίζει, από τον τύπο $k_3' = m' \oplus c_3$, τα 32 πρώτα bits της «υποθετικής» κλειδοροής k_3' τα οποία και εισάγει στον αλγόριθμο Berlekamp–Massey.

Η παραπάνω διαδικασία επιτυγχάνεται με την εκτέλεση του προγράμματος του οποίου ο πηγαίος κώδικας σε γλώσσα προγραμματισμού C++ βρίσκεται στο παράρτημα Β.4.2. Ορίζεται ως είσοδος το αρχείο κειμένου «**KnownText.txt**» που περιέχει τους χαρακτήρες με τους οποίους ο επιτιθέμενος θεωρεί ότι ξεκινά το αρχικό κείμενο (εδώ, η λέξη «Dear»), και το αρχείο κειμένου «**CipherStream3.txt**» που περιέχει το κρυπτοκείμενο σε μορφή ροής δυαδικών ψηφίων, τα περιεχόμενα του οποίου υποκλάπηκαν από το δημόσιο κανάλι επικοινωνίας. Έξοδος του προαναφερθέντος προγράμματος είναι το αρχείο κειμένου «**FakeKeyStream3.txt**», αποτέλεσμα της πράξης $k_3' = m' \oplus c_3$ για κάθε bit της λέξης «Dear», σε δυαδική μορφή βάση των Ascii τιμών των 4 χαρακτήρων της, με τα αντίστοιχα bits του κρυπτοκειμένου που βρίσκονται στο αρχείο «CipherStream3.txt».

Εισάγοντας τα 32 bits του αρχείου «FakeKeyStream3.txt» στον αλγόριθμο Berlekamp–Massey, λαμβάνεται χαρακτηριστικό πολυώνυμο για τον (μεγέθους 9) LFSR που παράγει την ακολουθία, στο παράδειγμα το : $f(x) = x^9 + x^4 + 1$.

Σε αυτή την περίπτωση, ο επιτιθέμενος δεν γνωρίζει 2l bits της ακολουθίας, όπου l η γραμμική πολυπλοκότητα της (άγνωστης σε αυτόν) κλειδοροής, αλλά λιγότερα. Όμως ο υπολογισμένος LFSR, από τον αλγόριθμο Berlekamp–Massey, παράγει ακολουθία η οποία διαφέρει σε λίγα μόνο ψηφία της κλειδοροής.

Ο επιτιθέμενος αναπαράγει ακολουθία, ως κλειδοροή για όλο το μήνυμα, δημιουργώντας τον LFSR που υπολόγισε από τον αλγόριθμο Berlekamp–Massey και ορίζοντας αρχική τιμή τα εννέα (9) πρώτα ψηφία της «υποτιθέμενης κλειδοροής» που είχε υπολογίσει βάσει της λέξης «Dear» και βρίσκεται στο αρχείο κειμένου «FakeKeyStream3.txt».

Στην συνέχεια αποκωδικοποιεί το κλεμμένο κρυπτοκείμενο με την παραγόμενη με τον προαναφερθέντα τρόπο ακολουθία, εφαρμόζοντας τον τύπο $m' = k_3' \oplus c_3$, ώστε να παραχθεί κείμενο σε αναγνώσιμη μορφή. Το δεύτερο αυτό μέρος της επίθεσης επιτυγχάνεται εκτελώντας το πρόγραμμα, του οποίου ο πηγαίος κώδικας σε γλώσσα προγραμματισμού C++ βρίσκεται στο παράρτημα Β.4.3.

Στο πρόγραμμα η μεταβλητή τύπου Array : FeedBack[9] που δέχεται τη συνάρτηση ανάδρασης του LFSR στη συνάρτηση «FakeLFSR» έχει διαμορφωθεί ανάλογα ώστε να προσομοιάζεται ο LFSR που υπολογίστηκε από τον αλγόριθμο Berlekamp-Massey. Είσοδος του προγράμματος είναι η υποκλεμμένη κρυπτοροή, σε μορφή αρχείου κειμένου «CipherStream3.txt», ενώ το αρχείο «FakeKeyStream3.txt» χρησιμοποιείται ώστε να λάβει αρχική κατάσταση ο LFSR παραγωγής της ακολουθίας χρήσης ως κλειδοροής για την αποκωδικοποίηση του κρυπτοκειμένου. Η αποκωδικοποίηση εκτελείται για κάθε bit της ακολουθίας του υποκλεμμένου κρυπτοκειμένου με την πράξη $m_i' = k_i' \oplus c_i$. Η προσομοίωση του LFSR γίνεται μέσω της συνάρτησης «FakeLFSR» και το αποτέλεσμα αποθηκεύεται σε μορφή καθαρού κειμένου στο αρχείο «**FakePlainText3.txt**».

Dear Sir, wε wou|d appreciate it if you could send us more detailed information about the project we discus3ed yesterday.

Το κείμενο που κατάφερε να αποκωδικοποιήσει ο επιτιθέμενος δεν είναι ακριβώς το ίδιο με το αρχικό, αλλά είναι σε σχετικά αναγνώσιμη μορφή που τον οδηγεί στο να λάβει την υποκλεμμένη πληροφορία .

3.8.4 Συμπέρασμα των Παραδειγμάτων Επίθεσης

Από τα περιγραφέντα μοντέλα επίθεσης, παρατηρείται ότι η περίοδος της ακολουθίας που χρησιμοποιείται ως κλειδοροή, δεν έχει ιδιαίτερα μεγάλη σημασία. Αντιθέτως, η τιμή της γραμμικής πολυπλοκότητας είναι πολύ καθοριστική.

Υψηλή τιμή γραμμικής πολυπλοκότητας υποχρεώνει τον επιτιθέμενο στη γνώση ή υπόθεση περισσοτέρων γραμμάτων από το περιεχόμενο του αρχικού κειμένου, ώστε, αποκωδικοποιώντας το κλεμμένο κρυπτοκείμενο,, να υπάρχει δυνατότητα ανάγνωσης του κειμένου.

Η παραπάνω παρατήρηση προκύπτει από το γεγονός ότι ο αλγόριθμος Berlekamp–Massey δεχόμενος ως είσοδο ακολουθία μεγέθους 2N ψηφίων, δύναται να υπολογίσει τον LFSR παράγωγης της ακολουθίας, ο οποίος έχει μέγιστο αριθμό βαθμίδων N. Όσο αυξάνει η τιμή του N, δηλαδή του πλήθους των εισηγμένων στον αλγόριθμο ψηφίων, τόσο αυξάνει η ακρίβεια υπολογισμού του ζητούμενου LFSR.

Με δεδομένο το ότι, βάσει της Ascii κωδικοποίησης, κάθε γράμμα του αλφαβήτου καταλαμβάνει 8 bits σε δυαδική μορφή, τεκμαίρεται ότι η ζητούμενη τιμή γραμμικής πολυπλοκότητας πρέπει να είναι αρκετά υψηλή, ώστε γνώση ορισμένων γραμμάτων του αρχικού μηνύματος να μην οδηγεί σε πλήρη αποκωδικοποίηση του κρυπτοκειμένου και στην συνέχεια στη δυνατότητα ανάγνωσης του αρχικού κειμένου.

Από τα μοντέλα επιθέσεων που εκτέθηκαν υπό των 3.8.1 και 3.8.2, διαπιστώθηκε ότι, εάν το πλήθος bits που γνωρίζει ο επιτιθέμενος είναι έστω και ελάχιστα μικρότερο από την τιμή γραμμικής πολυπλοκότητας της ακολουθίας που χρησιμοποιείται ως κλειδοροή, τότε η κλειδοροή αυτή δεν μπορεί να αναπαραχθεί πιστά αξιοποιώντας τον αλγόριθμο Berlekamp-Massey. Ως αποτέλεσμα είναι η έλλειψη δυνατότητας αποκρυπτογράφησης του κρυπτοκειμένου σε αναγνώσιμη μορφή.

Ωστόσο, αν η κλειδοροή προσεγγίζει σε μεγάλο βαθμό μία άλλη ακολουθία s η οποία έχει χαμηλή γραμμική πολυπλοκότητα l, τότε ακόμα και αν η κλειδοροή έχει μεγάλη γραμμική πολυπλοκότητα μας αρκεί να γνωρίζουμε 2l bits αυτής προκειμένου, μέσω του Berlekamp-Massey, να υπολογίσουμε ολόκληρη την ακολουθία s – άρα, θα έχουμε ουσιαστικά υπολογίσει μία πολύ καλή προσέγγιση της κλειδοροής, το οποίο πρακτικά σημαίνει ότι πάλι μπορούμε να ανακτήσουμε ολόκληρο το μήνυμα με εξαίρεση κάποια λίγα bits (εκείνα στα οποία οι δύο ακολουθίες διαφέρουν). Συνεπώς, και η γραμμική πολυπλοκότητα k σφαλμάτων έχει ιδιάζουσα κρυπτογραφική σημασία.

Κεφάλαιο 4
Δυαδικές Ακολουθίες de Bruijn

Οι ακολουθίες de Bruijn διαδραματίζουν εξέχοντα ρόλο σε πολλά επιστημονικά πεδία και συναντώνται σε πολλές εφαρμογές , όπως είναι το πεδίο της βιολογίας και της βιοπληροφορικής [08], των μαθηματικών και της στατιστικής, των υπολογιστών και της πληροφορικής. Ιδιαίτερο ενδιαφέρον όμως παρουσιάζουν στον τομέα της κρυπτογραφίας, όπου αποτελούν διαρκές πεδίο έρευνας [35] με χρήση ποικίλων μαθηματικών εργαλείων [07].

Προτιμούνται σε εφαρμογές επικοινωνιών, σε συστήματα κρυπτογράφησης και κωδικοποίησης λόγω των πολύ καλών κρυπτογραφικών χαρακτηριστικών που διαθέτουν. Η υψηλή τιμή γραμμικής πολυπλοκότητας, η μέγιστη περίοδος, αλλά και οι υπόλοιπες κρυπτογραφικές ιδιότητες που παρουσιάζουν τις καθιστούν ιδανικές για ορισμένες εφαρμογές κρυπτογραφίας.

Οι ακολουθίες de Bruijn δημιουργούνται από μη γραμμικούς καταχωρητές ολίσθησης. Λόγω της μεγαλύτερης ανάπτυξης των μαθηματικών μοντέλων ανάλυσης της γραμμικής συνάρτησης ανάδρασης έναντι αυτής για τη μη γραμμική, δεν υπάρχει αλγόριθμος για την αποτελεσματική

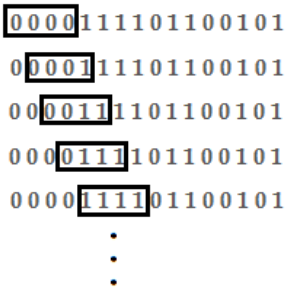
παραγωγή όλων των ακολουθιών de Bruijn μιας δεδομένης ακολουθίας. Πολλοί από τους αλγόριθμους χρησιμοποιούν διάφορα συνδυαστικά αποτελέσματα για τη δημιουργία υποσυνόλων των ακολουθιών de Bruijn [27].

Ονομάστηκαν έτσι προς τιμήν του Ολλανδού μαθηματικού Nicolaas Govert de Bruijn, ο οποίος το 1946 στην εργασία του «A combinatorial problem» [09] περιγράφοντάς τις ως T-nets υπολόγισε τον αριθμό $|N|$ όλων των πιθανών διαφορετικών αλληλουχιών για τις ακολουθίες αυτές.

Ακολουθία de Bruijn, στα συνδυαστικά μαθηματικά, είναι μια κυκλική αλληλουχία όπου για **k** όρους και τάξης **n** έχει περίοδο $\mathbf{k^n}$ και είναι διαμορφωμένη έτσι ώστε κάθε στοιχείο V του σώματος GF(k), μεγέθους n να εμφανίζεται μία φορά. Συμβολίζεται ως **B(k,n)** [18].

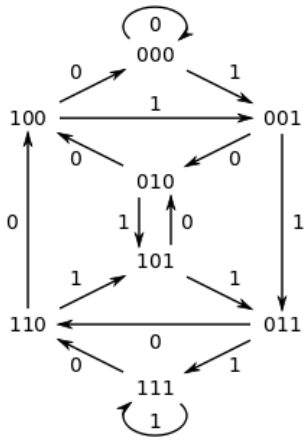
Στις κρυπτογραφικές εφαρμογές όπου χρησιμοποιούνται (και εξετάζονται στην παρούσα μεταπτυχιακή διατριβή), οι ακολουθίες de Bruijn λαμβάνουν τιμές από το σώμα GF(2), δηλαδή είναι δυαδικές. Αυτό σημαίνει ότι, σε μία δυαδική ακολουθία τάξης n, κάθε δυνατή n-άδα από bits εμφανίζεται ακριβώς μία φορά σε μία περίοδό της. Το πλήθος όλων των δυνατών συνδυασμών διαφορετικών αλληλουχιών για ακολουθίες η βαθμού δίδεται από τον τύπο $B_n = 2^{2^{n-1}-n}$ [41].

Παράδειγμα ακολουθίας de Bruijn βαθμού n=4 είναι η : 0 0 0 0 1 1 1 1 0 1 1 0 0 1 0 1. Περιέχονται όλα τα πιθανά στοιχεία μεγέθους 4 ακριβώς μία φορά όπως φαίνεται στην εικόνα 4.1, ενώ στην εικόνα 4.2 αποτυπώνεται ο γράφος που την παράγει [68].



Εικόνα 4.1 : Εμφανιζόμενα Στοιχεία της n=4 Ακολουθίας de Bruijn.

Τα δεκαέξι (16) μοναδικά στοιχεία, μεγέθους 4, που περιέχονται στην ακολουθία είναι :
(0000), (0001), (0011), (0111), (1111), (1110), (1101), (1011),
(0110), (1100), (1001), (0010), (0101), (1010), (0100), (1000).



Εικόνα 4.2 : Γράφος Ακολουθίας de Bruijn [68].

Από τα ανωτέρω προκύπτει ότι κάθε ακολουθία που παράγεται από έναν NLFSR μεγέθους n ο οποίος παράγει ακολουθίες με τη μέγιστη δυνατή περίοδο 2^n – μία ιδιότητα επιθυμητή σε κρυπτογραφικές εφαρμογές – είναι ακολουθία de Bruijn.

4.1 Τρόποι Κατασκευής Ακολουθίας de Bruijn

Εκτός από δημιουργία νέας ακολουθίας, τρόποι παραγωγής ακολουθιών de Bruijn μπορεί να είναι η σύνθεση δύο υπαρχουσών ή η επέκταση άλλης μικρότερης περιόδου. Λίγοι είναι οι γνωστοί αποτελεσματικοί τρόποι παραγωγής τέτοιων ακολουθιών [45]:

- με γεννήτριες κλειδοροής (δηλαδή εύρεση κατάλληλων NLFSR, οι οποίοι διέρχονται από όλες τις πιθανές καταστάσεις),
- με τους τρεις διαφορετικούς αλγόριθμους για τη δημιουργία της λεξικογραφικά μικρότερης ακολουθίας de Bruijn (επίσης γνωστή ως ακολουθία Ford), έναν αλγόριθμο αλληλοσυμπλήρωσης Lyndon λέξεων από τους Fredricksen και Maiorana [19], μια διαδοχική προσέγγιση του κανόνα από τον Fredricksen [16] και έναν αλγόριθμο συγκολλήσεως μπλοκ από την Ralston [42],
- με τον αλγόριθμο αλφαριθμητικής συνενώσεως της λεξικογραφικής σύνθεσης από τους Fredricksen και Kessler [18],
- με τους τρεις διαφορετικούς αλγόριθμους αλληλοσύνδεσης κύκλου από τους Fredricksen [17], Etizon και Lempel [14] και Huang [28] ,
- με αλγόριθμο κατασκευής βασισμένη σε υπάρχουσα λεξικογραφική σύνθεση από Sawada, Stevens και Williams [44] ή Sawada, Williams και Wong [46] και
- με αλγόριθμο κατασκευής βασισμένη σε μετατόπιση υπάρχουσας λεξικογραφικής σύνθεσης από τους Sawada, Williams, και Wong [45].

4.1.1 Παραγωγή με NLFSR

Στην παράγραφο 3.4, της παρούσας, περιγράφηκε η λειτουργία των μη γραμμικών καταχωρητών ολίσθησης με ανάδραση (NLFSR). Με επιλογή κατάλληλης συνάρτησης ανάδρασης, ο NLFSR δύναται να παράγει ακολουθίες μέγιστης περιόδου και γραμμικής πολυπλοκότητας.

Για παράδειγμα, ένας NLFSR μεγέθους 3, ο οποίος έχει μη γραμμική συνάρτηση ανάδρασης :
$f(x1, x2, x3) = 1 \oplus x2 \oplus x3 \oplus x1 \cdot x2$
παρουσιάζει την αλληλουχία καταστάσεων που απεικονίζεται στον πίνακα 4.3.

x1	x2	x3	out
0	0	0	0
1	0	0	0
1	1	0	0
1	1	1	1
0	1	1	1
1	0	1	1
0	1	0	0
0	0	1	1

Πίνακας 4.3 : Καταστάσεις NLFSR με $f(x1, x2, x3) = 1 \oplus x2 \oplus x3 \oplus x1 \cdot x2$.

Ορίζοντας αρχική κατάσταση των καταχωρητών την τιμή [0, 0, 0], η ακολουθία εξόδου του NLFSR θα είναι η de Bruijn : **00011101** [38].

4.1.2 Υπόλοιπες Τεχνικές Κατασκευής Ακολουθίας de Bruijn

Εκτός των μη γραμμικών καταχωρητών ολίσθησης με ανάδραση (NLFSR), παραγωγή ακολουθιών de Bruijn μπορεί να πραγματοποιηθεί με τις παρακάτω μεθόδους.

- Με το γράφο (εικόνα 4.2), όπου ακολουθώντας διαφορετική κάθε φορά «Euler» διαδρομή, δύναται να κατασκευαστεί ξεχωριστή ακολουθία de Bruijn.
- Με την προσέγγιση μέσω «άπλειστου» (greedy) αλγορίθμου, κατά την οποία προστίθεται κάθε φορά το μεγαλύτερο ψηφίο, το οποίο δημιουργεί την υπο-ακολουθία του στοιχείου μεγέθους n και δεν είναι αντίγραφο υπάρχουσας.
- Με διαδοχικό κανόνα, μια συνάρτηση που επιστρέφει την τιμή του επόμενου στοιχείου ακολουθίας de Bruijn (εικόνα 4.4) [52].

$$f(b_1 b_2 \cdots b_n) = \begin{cases} b_2 b_3 \cdots b_n \overline{b_1} & \text{if } b_2 b_3 \cdots b_n 1 \text{ is a necklace;} \\ b_2 b_3 \cdots b_n b_1 & \text{otherwise.} \end{cases}$$

Εικόνα 4.4 : Διαδοχικός Κανόνας Ακολουθίας de Bruijn [45].

4.2 Κριτήρια Κρυπτογραφικής Ισχύος Τροποποιημένης Ακολουθίας de Bruijn

Ένα από τα πιο σημαντικά μεγέθη τυχαιότητας μιας ακολουθίας είναι η γραμμική της πολυπλοκότητα, που είναι ο βαθμός του μικρότερου γραμμικού καταχωρητή ο οποίος παράγει την ακολουθία. Στις ακολουθίες de Bruijn έχει εξεταστεί η γραμμική πολυπλοκότητα και είναι πλέον γνωστό ότι είναι υψηλή. Ωστόσο, όπως ειπώθηκε νωρίτερα, ακόμα και αν μία ακολουθία έχει υψηλή γραμμική πολυπλοκότητα, εάν αυτή μειώνεται δραματικά στην περίπτωση αλλαγής λίγων μόνο ψηφίων της τότε η ακολουθία δεν είναι κρυπτογραφικά ισχυρή. Ενώ η γραμμική πολυπλοκότητα των δυαδικών ακολουθιών de Bruijn έχει εκτεταμένα ερευνηθεί, δεν έχει γίνει ιδιαίτερη αναφορά στις κρυπτογραφικές ιδιότητες των τροποποιημένων αυτών ακολουθιών de Bruijn Αυτό αποτελεί και το κύριο ερευνητικό αντικείμενο της παρούσας διατριβής.

Οι A. H. Chan, R. A. Games και E. L. Key, στο άρθρο τους «On the complexities of de Bruijn sequences» (1982) [06], αποδεικνύουν ότι η γραμμική πολυπλοκότητά ακολουθίας de Bruijn **S** τάξης **n** κυμαίνεται $\mathbf{2^{n-1}+n \le C(S) \le 2^n-1}$. Στο ίδιο άρθρο αναφέρεται ότι η χαμηλότερη τιμή της γραμμικής πολυπλοκότητας, $C(S) = 2^{n-1}+n$, επιτυγχάνεται για $3 \le n \le 6$. Οι T. Etzion και A. Lempel στο «Construction of de Bruijn Sequences of Minimal Complexity» (1984), δείχνουν ότι η χαμηλότερη αυτή τιμή της γραμμικής πολυπλοκότητας, $C(S) = 2^{n-1}+n$, για ακολουθίες de Bruijn μπορεί να επιτευχθεί για οποιοδήποτε βαθμό $\mathbf{n \ge 3}$ [15].

Στην παρούσα διατριβή, κατά την εξέταση του προφίλ της γραμμικής πολυπλοκότητας (3.6, σελ. 36), αναφέρθηκε ότι έχει αποδειχθεί ότι το προφίλ της γραμμικής πολυπλοκότητας μιας πραγματικά τυχαίας δυαδικής ακολουθίας μήκους N bits, είναι η ευθεία N/2. Δηλαδή κρυπτογραφικά ισχυρές είναι οι ακολουθίες οι οποίες πληρούν τη συνθήκη :
$f[C(s^N)] \approx N/2$ [55].

Στον αλγόριθμο Stamp-Martin (3.7.1, σελ. 42 της παρούσης), διατυπώθηκε ότι το διάνυσμα **b**, που περιέχει το αποτέλεσμα της πρόσθεσης του αριστερού μέρους με το δεξί της προς εξέταση

ακολουθίας, σε κάθε επανάληψη του αλγόριθμου, εξαναγκάζεται να πάρει μηδενική τιμή. Αυτό επιτυγχάνεται αλλάζοντας ψηφία στην ακολουθία, της οποίας το συνολικό κόστος της αλλαγής αυτής αποθηκεύεται στη μεταβλητή **T**.

Όταν το διάνυσμα **b** λάβει μηδενική τιμή σημαίνει ότι τα δύο μέρη της εξεταζόμενης ακολουθίας μεγέθους **l**, το αριστερό και το δεξί, είναι όμοια και για το λόγο αυτό η ακολουθία είναι περιοδική μεγέθους **l/2** και επαναλαμβάνεται δύο φορές. Στην περίπτωση αυτή, η τιμή γραμμικής πολυπλοκότητας της ακολουθίας δεν αυξάνεται κατά **l/2**.

Στην περίπτωση που για ακολουθία **S**, βαθμού **n**, μεγέθους $\mathbf{N = 2^n}$, το διάνυσμα **b** λάβει μηδενική τιμή κατά την πρώτη επανάληψη του αλγόριθμου, το μέγεθος της εξεταζόμενης ακολουθίας είναι **N**, η απώλεια στην τιμή της γραμμικής πολυπλοκότητας είναι η μέγιστη δυνατή, ίση με **N/2** ή $\mathbf{2^{n-1}}$, ενώ η μεταβλητή **T**, η οποία περιέχει το συνολικό κόστος κατά την συγκεκριμένη επανάληψη του αλγόριθμου, απεικονίζει τον αριθμό **k** των ψηφίων της αρχικής ακολουθίας τα οποία αλλοιώθηκαν.

Λαμβάνοντας υπόψη τα παραπάνω, εξετάζεται, για κάθε ακολουθία de Bruijn, το πλήθος των αλλοιωμένων ψηφίων **k**, που διαμορφώνει το μέγεθος της γραμμικής πολυπλοκότητας σε τιμές $\mathbf{C_k(S) < 2^{n-1}+n}$. Ιδιαίτερο ενδιαφέρον έχει το πλήθος **k**, το οποίο οδηγεί τη γραμμική πολυπλοκότητα ακολουθίας de Bruijn να λάβει τιμή $\mathbf{C_k(S) \leq 2^{n-1}}$.

Κεφάλαιο 5
Πολυπλοκότητα k σφαλμάτων σε Ακολουθίες de Bruijn

Ανωτέρω, παρουσιάστηκε το θεωρητικό πλαίσιο της κρυπτογραφίας και των εφαρμογών της, των κρυπτογραφικών συστημάτων και της χρήσης τους. Ειδικότερα, έγινε αναφορά στα συμμετρικά κρυπτογραφικά συστήματα με εστίαση στους κρυπτογραφικούς αλγόριθμους ροής.

Μη γραμμικοί καταχωρητές ολίσθησης με ανάδραση οι οποίοι παράγουν δυαδικές ακολουθίες με μέγιστη περίοδο είναι επιθυμητοί ως δομικά συστατικά γεννητριών κλειδοροής σε κρυπταλγορίθμους ροής. Τέτοιες ακολουθίες μεγίστης περιόδου ονομάζονται de Bruijn και εν γένει εμφανίζουν πολύ καλές κρυπτογραφικές ιδιότητες, όπως η γραμμική πολυπλοκότητα. Δεν έχουν όμως μελετηθεί ως προς το πώς συμπεριφέρεται η γραμμική τους πολυπλοκότητα με τη διαφοροποίηση ακόμα και λίγων ψηφίων της. Αν αυτή εμφανίζεται ραγδαία μειούμενη, τότε τίθεται ζήτημα ως προς την καταλληλότητα των αντίστοιχων ακολουθιών de Bruiijn σε κρυπτογραφικές εφαρμογές.

Η παρούσα έρευνα, εστιάζει στην πειραματική μελέτη της τιμής της γραμμικής πολυπλοκότητας ακολουθιών de Bruijn κατόπιν αλλαγής k ψηφίων με τη χρήση αλγορίθμων και προγραμμάτων σε γλώσσα προγραμματισμού C++.

Χρησιμοποιώντας τους αλγόριθμους Games-Chan, Stamp–Martin και Lauder–Paterson, διαπιστώνεται η επίπτωση που έχει στη διαμόρφωση της τιμής της γραμμικής πολυπλοκότητας η αλλαγή ορισμένων ψηφίων μιας αρχικής de Bruijn ακολουθίας.

Επίσης δίνεται ιδιαίτερη σημασία στις τιμές που λαμβάνει η μεταβλητή **k** που δηλώνει τον αριθμό των τροποποιημένων ψηφίων της αρχικής ακολουθίας, όταν η γραμμική πολυπλοκότητά **n** βαθμού ακολουθίας de Bruijn **S** λαμβάνει τιμές $\mathbf{C(S) < 2^{n-1}}$.

5.1 Κριτήρια της Έρευνας

Ο αλγόριθμος Berlekamp–Massey δεχόμενος ως είσοδο ακολουθία μεγέθους 2N ψηφίων, δύναται να υπολογίσει τον LFSR παραγωγής της ακολουθίας, ο οποίος έχει μέγιστο αριθμό βαθμίδων N. Επιδίωξη για δημιουργία κρυπτογραφικά ισχυρών ακολουθιών οι οποίες δύναται να χρησιμοποιηθούν ως κλειδοροή, είναι η υψηλή τιμή γραμμικής πολυπλοκότητας.

Ωστόσο, αν η κλειδοροή έχει υψηλή τιμή γραμμικής πολυπλοκότητας αλλά προσεγγίζεται σε μεγάλο βαθμό από άλλη ακολουθία s χαμηλής γραμμικής πολυπλοκότητας - η οποία διαφέρει της αρχικής σε k ψηφία - τότε αρκεί γνώση 2l bits αυτής της ακολουθίας s ώστε να υπολογιστεί ολόκληρη και συνεπώς να οδηγηθούμε σε ουσιαστική προσέγγιση της κλειδοροής. Σε αυτή την περίπτωση επιδίωξη, αποδεκτής κρυπτογραφικά κλειδοροής, αποτελεί η ύπαρξη υψηλού αριθμού k ώστε να θεωρηθεί ότι η ακολουθία s δεν προσεγγίζει σε μεγάλο βαθμό την κλειδοροή.

Κατά την εξέταση του προφίλ της γραμμικής πολυπλοκότητας αναφέρθηκε ότι το προφίλ της γραμμικής πολυπλοκότητας μιας πραγματικά τυχαίας δυαδικής ακολουθίας μήκους N bits, είναι η ευθεία N/2. Δηλαδή $f[C(s^N)] \approx N/2$ [55].

Με βάση τα ανωτέρω, θεωρείται ότι, εάν σε ακολουθία s, η οποία χρησιμοποιείται ως κλειδοροή, η γραμμική πολυπλοκότητα λάβει τιμή $C_k(s) < 2^{n-1}$, τροποποιώντας k ψηφία της, αυτή θεωρείται κρυπτογραφικά αδύναμη. Ο αριθμός k των τροποποιημένων ψηφίων επιδρά στη διαμόρφωση της τιμής γραμμικής πολυπλοκότητας $C_k(s)$ της ασθενούς κρυπτογραφικά ακολουθίας. Αυξάνοντας το πλήθος των τροποποιημένων ψηφίων μειώνεται η τιμής γραμμικής

πολυπλοκότητας της ακολουθίας. Όταν όμως το πλήθος k είναι αρκετά υψηλό θεωρείται ότι η τροποποιημένη αυτή ακολουθία δεν προσεγγίζει την αρχική.

Ως κριτήριο αποτίμησης κρυπτογραφικής αξίας τροποποιημένης ακολουθίας de Bruijn ορίζεται δείκτης που ισούται με $\mathbf{2 \cdot C_k(s)+k}$. Όταν σε ακολουθία de Bruijn τροποποιούνται **k** ψηφία οδηγώντας την γραμμική πολυπλοκότητα σε τιμή $\mathbf{C_k(s)}$ και ο συγκεκριμένος δείκτης λάβει τιμή μικρότερη από **N** (2^n) τότε αυτή η ακολουθία θεωρείται κρυπτογραφικά αδύναμη. Ο συγκεκριμένος δείκτης ορίστηκε στο πλαίσιο της παρούσας διατριβής ως απόρροια του γεγονότος ότι γνώση $2 \cdot C_k(s)$ bits της τροποποιηθείσας σε k θέσεις ακολουθίας επιτρέπει τον υπολογισμό ολόκληρης της τροποποιηθείσας ακολουθίας (και, εφόσον ο ανωτέρω δείκτης λαμβάνει τιμή μικρότερη του N, καθίσταται πιθανό να αρκεί γνώση των $2 \cdot C_k(s)$ bits της αρχικής ακολουθίας – το οποίο είναι και το ρεαλιστικό σενάριο).

Δείκτης κρυπτογραφικής αξίας υπολογίζεται σε κάθε κρίσιμο σημείο (Critical Point) ακολουθίας de Bruijn. Εάν σε κρίσιμο σημείο, παρουσιάζεται ο συγκεκριμένος δείκτης με τιμή μικρότερη του N, τότε είναι πιθανό γνωρίζοντας τμήμα της τροποποιημένης ακολουθίας, ίσο με $2 \cdot C_k(s)$, να παρέχεται η δυνατότητα να ανακτηθεί ολόκληρη. Αυτό συμβαίνει όταν το γνωστό τμήμα της ακολουθίας δεν περιέχει τροποποιημένα ψηφία. Για τον λόγο αυτό υπολογίζεται η αναλογία τροποποιημένων ψηφίων στην ακολουθία ώστε να χαρακτηριστεί η κρυπτογραφική ισχύς της.

5.2 Δημιουργία Ακολουθιών de Bruijn

Για την πραγματοποίηση της έρευνας χρησιμοποιούνται ακολουθίες de Bruijn μεγέθους (περιόδου) 32, 64, 128, 256, 512. 1024 και 2048 bits.

Για το λόγο αυτό αναπτύχθηκε πρόγραμμα σε γλώσσα προγραμματισμού C++, βασισμένο σε κώδικα των D. Gabric, J. Sawada, A. Williams, και D. Wong, ο οποίος δημοσιεύτηκε στο άρθρο «A framework for constructing de Bruijn sequences via simple successor rules» (2018) [20] και διαμορφώθηκε από τον γράφοντα για τις ανάγκες της συγκεκριμένης έρευνας. Ο πηγαίος κώδικας του προγράμματος αποτυπώνεται στο Παράρτημα Β.1.

Το πρόγραμμα δημιουργεί ακολουθίες de Bruijn περιόδου (μεγέθους) 2^n για $5 \leq n \leq 11$. Για κάθε n δημιουργούνται επτά (7) διαφορετικές ακολουθίες de Bruijn περιόδου (μεγέθους) 2^n που όλες, χωρίς βλάβη της γενικότητας, ξεκινούν από n μηδενικά. Τα αποτελέσματα του προγράμματος,

ανάλογα με το μέγεθος της κάθε παραχθείσας ακολουθίας de Bruijn, τοποθετούνται στα ακόλουθα αρχεία κειμένου :

- Για **n=5**, δηλαδή ακολουθίες μεγέθους 2^5 = 32 ψηφίων στο αρχείο : **De_Bruijn_5.txt**
- Για **n=6**, δηλαδή ακολουθίες μεγέθους 2^6 = 64 ψηφίων στο αρχείο : **De_Bruijn_6.txt**
- Για **n=7**, δηλαδή ακολουθίες μεγέθους 2^7 = 128 ψηφίων στο αρχείο : **De_Bruijn_7.txt**
- Για **n=8**, δηλαδή ακολουθίες μεγέθους 2^8 = 256 ψηφίων στο αρχείο : **De_Bruijn_8.txt**
- Για **n=9**, δηλαδή ακολουθίες μεγέθους 2^9 = 512 ψηφίων στο αρχείο : **De_Bruijn_9.txt**
- Για **n=10**, δηλαδή ακολουθίες μεγέθους 2^{10} = 1025 ψηφίων στο αρχείο : **De_Bruijn_10.txt**
- Για **n=11**, δηλαδή ακολουθίες μεγέθους 2^{11} = 2048 ψηφίων στο αρχείο : **De_Bruijn_11.txt**

Στην συνέχεια τα αρχεία εμπλουτίστηκαν με πέντε (5) ακολουθίες ($7 \leq n \leq 11$),οι οποίες προέκυψαν από εφαρμογή ενός πρόσφατου αλγορίθμου για την εύρεση de Bruijn [35] που παράγει ακολουθίες εισάγοντας στοιχεία τυχαιότητας. Στην ακολουθία προστίθεται ψηφίο «0» ή «1» για όσο πληρείται η de Bruijn ιδιότητα, δηλαδή κάθε *n*-άδα που δημιουργείται να μην έχει εμφανιστεί νωρίτερα. Σκοπός εμπλουτισμού των αρχείων είναι να εξετασθούν ακολουθίες που δεν παράγονται εγγυημένα από δεδομένους αλγόριθμους τεχνικών παραγωγής.

Οι ακολουθίες που περιέχονται στα ανώτερα αρχεία θα αποτελέσουν δοκίμια της έρευνας και εφεξής θα καλούνται **ακολουθίες προς δοκιμή**. Το περιεχόμενο των αρχείων για n=5 έως n=8, αποτυπώνεται στο Παράρτημα Α.1.

5.3 Πρόγραμμα Stamp-Martin

Στο άρθρο «Computing the error linear complexity spectrum of a binary sequence of period 2^n» των A. Lauder και K. Paterson, η μέθοδος υπολογισμού του k- error linear complexity profile με τη χρήση του αλγορίθμου Stamp–Martin, περιγράφεται ως απλή [32].

Στην παρούσα διατριβή, με τη χρήση του αλγορίθμου Stamp–Martin, επιχειρείται για πρώτη φορά μια τέτοια προσέγγιση, ώστε να διαπιστωθεί η σχέση του αριθμού αλλοιωμένων ψηφίων της αρχικής ακολουθίας k, με τη μείωση της τιμής της γραμμικής πολυπλοκότητας που παρουσιάζει αυτή η τροποποιημένη ακολουθία ck(s).

Το πρόγραμμα, του οποίου ο πηγαίος κώδικας σε γλώσσα προγραμματισμού C++ αποτυπώνεται στο Παράρτημα Β.2, εξετάζει χρησιμοποιώντας τον αλγόριθμο Stamp–Martin, για κάθε ακολουθία μεγέθους N=32 bits και N=64 bits που είναι προς δοκιμή, τη διαμόρφωση της γραμμικής πολυπλοκότητας ck(s) στις συνθήκες αύξησης της τιμής των αλλοιωμένων ψηφίων k.

Επιλογή τιμών μεγέθους των ακολουθιών αλλά και των ονομάτων αρχείων εισόδου και εξόδου πραγματοποιείται μέσω των εντολών ορισμού (ακολούθως για ακολουθίες μεγέθους N =32 bits, προερχόμενες από το αρχείο «De_Bruijn_5.txt» και Excel αρχείο προορισμού το «StampMartin_5.csv») :

```
#define N 32
#define FileName "De_Bruijn_5.txt"
#define StampMartinExcelFile "StampMartin_5.csv"
```

Αρχικά εκτελείται η ρουτίνα **GetDBSequence**, η οποία ενημερώνει τον πίνακα **DBSequ[7][N]** τοποθετώντας τις τιμές των επτά (7) ακολουθιών μεγέθους N προς δοκιμή.

Στην συνέχεια, για κάθε μία από τις προαναφερθείσες ακολουθίες, μέσω της επανάληψης (Loop) **for (int j = 0; j < 7; j++)** , υπολογίζεται η τιμή της γραμμικής πολυπλοκότητας ck(s), για αριθμό αλλοιωμένων ψηφίων τη αρχικής k = 0, 1, 2, … , μέχρι αυτή να μηδενιστεί.

Η συγκεκριμένη διαδικασία πραγματοποιείται μέσω των εντολών :

```
k = 0;
do  {
     ck = StampMartin(sDB, k);
     k++; }
while (ck > 0);
```

όπου **StampMartin** είναι συνάρτηση υπολογισμού της γραμμικής πολυπλοκότητας ck(s) βάσει του αλγορίθμου Stamp–Martin, η οποία δέχεται ως παραμέτρους (ορίσματα) την προς εξέταση ακολουθία **sDB** και την τιμή του πλήθους των αλλοιωμένων ψηφίων **k** .

Τέλος, εκτελώντας τη ρουτίνα **SaveResults ,** αποθηκεύονται τα αποτελέσματα σε αρχείο τύπου **csv**. Τα αποτελέσματα του προγράμματος για τις ακολουθίες που εξετάστηκαν μεγέθους N=32 bits αποθηκεύονται στο αρχείο **De_Bruijn_5.csv**. Αντίστοιχα τα αποτελέσματα για τις ακολουθίες που εξετάστηκαν μεγέθους N=64 bits αποθηκεύονται στο αρχείο **De_Bruijn_6.csv**. Τα περιεχόμενα των δύο αυτών αρχείων αποτυπώνονται το Παράρτημα Α.2.

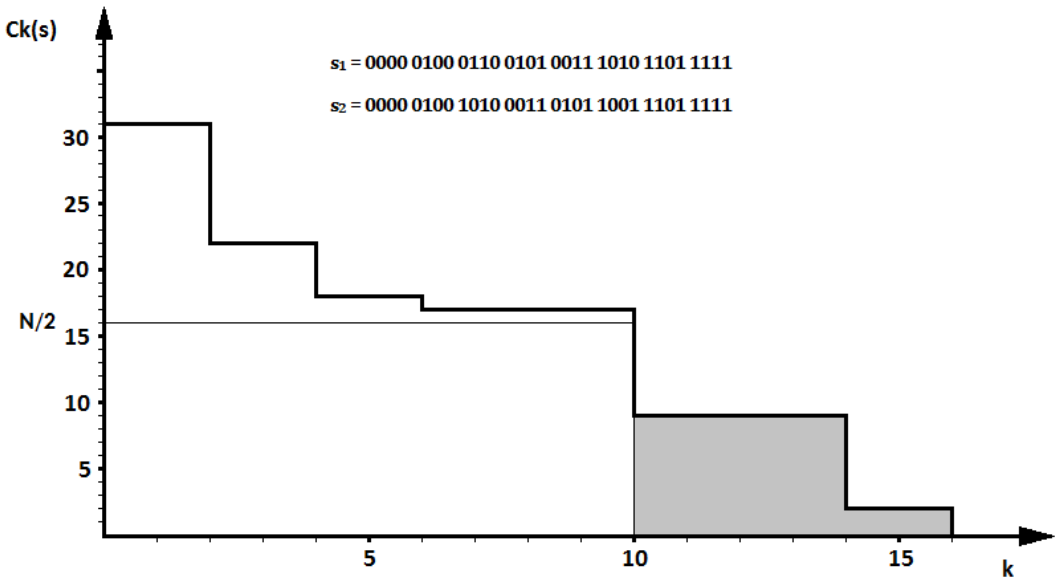
5.4 Αποτελέσματα Προγράμματος Stamp–Martin

Λαμβάνοντας για κάθε ακολουθία από τις εξεταζόμενες τις τιμές της γραμμικής πολυπλοκότητας k σφαλμάτων Ck(s) σε συνάρτηση με τον αριθμό αλλοιωμένων ψηφίων της αρχικής k, δύναται να σχεδιαστεί το διάγραμμα του προφίλ της γραμμικής πολυπλοκότητας k σφαλμάτων. Στα παρακάτω διαγράμματα (ανά ακολουθία), το σκιασμένο μέρος εμφανίζει το πλήθος των αλλοιωμένων ψηφίων k που οδηγούν σε τιμές γραμμικής πολυπλοκότητας μικρότερης του N/2.

5.4.1 Ακολουθίες de Bruijn μεγέθους 32 bits

Αποτυπώνονται σε διαγράμματα τα προφίλ γραμμικής πολυπλοκότητας k σφαλμάτων των ακολουθιών, των οποίων οι τιμές $Ck(s) = f(k)$ αναγράφονται στον πίνακα Α.1 του παραρτήματος Α.2.1. Οι έξι (6) αυτές ακολουθίες βρίσκονται στο αρχείο De_Bruijn_5.txt, και έχουν αριθμηθεί ως № 1 ÷ № 6 .

Η ακολουθία № 1 και η ακολουθία № 2 παρουσιάζουν ίδιες τιμές $Ck(s) = f(k)$:
s_1 = 0000 0100 0110 0101 0011 1010 1101 1111
s_2 = 0000 0100 1010 0011 0101 1001 1101 1111
των οποίων το προφίλ της γραμμικής πολυπλοκότητας k σφαλμάτων αποτυπώνεται στο διάγραμμα 5.1.



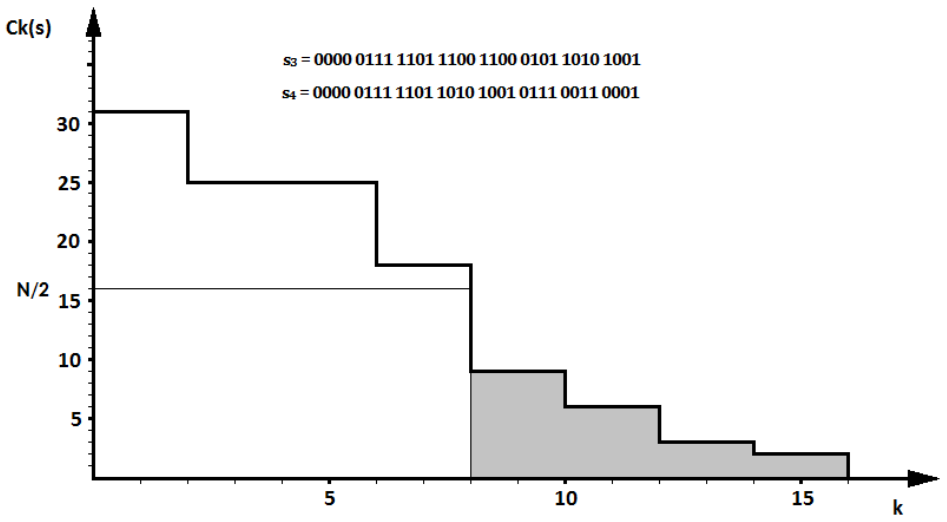
Διάγραμμα 5.1 : Προφίλ Γραμμικής Πολυπλοκότητας k-σφαλμάτων ακολουθιών s_1 και s_2.

Η ακολουθία № 3 και η ακολουθία № 4 παρουσιάζουν ίδιες τιμές Ck(s) = ƒ(k) :

s_3 = 0000 0111 1101 1100 1100 0101 1010 1001

s_4 = 0000 0111 1101 1010 1001 0111 0011 0001

των οποίων το προφίλ της γραμμικής πολυπλοκότητας k σφαλμάτων αποτυπώνεται στο διάγραμμα 5.2.

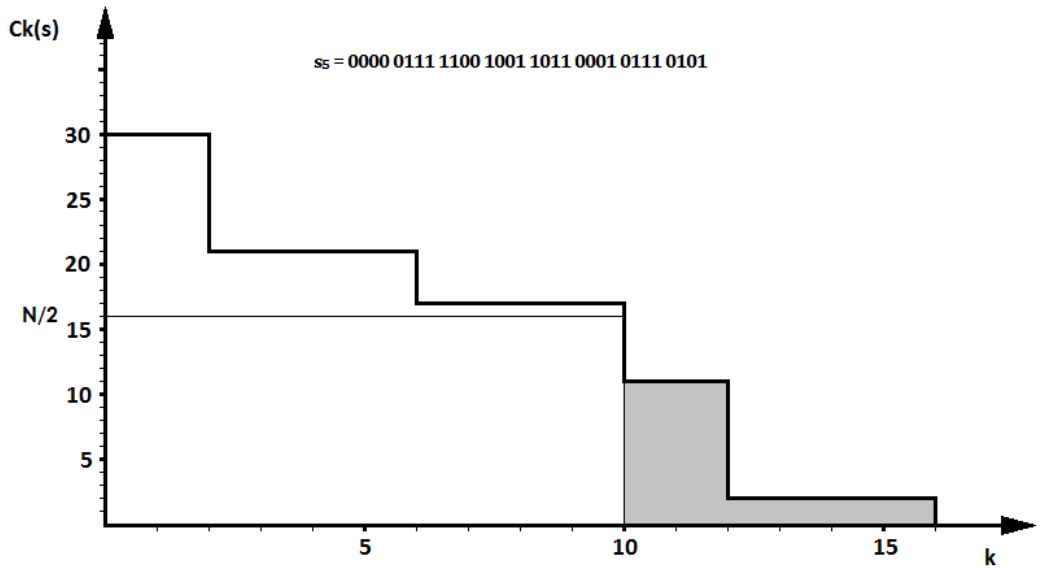


Διάγραμμα 5.2 : Προφίλ Γραμμικής Πολυπλοκότητας k-σφαλμάτων ακολουθιών s_3 και s_4.

Της ακολουθίας № 5 :

s_5 = 0000 0111 1100 1001 1011 0001 0111 0101

το προφίλ της γραμμικής πολυπλοκότητας k σφαλμάτων αποτυπώνεται στο διάγραμμα 5.3.

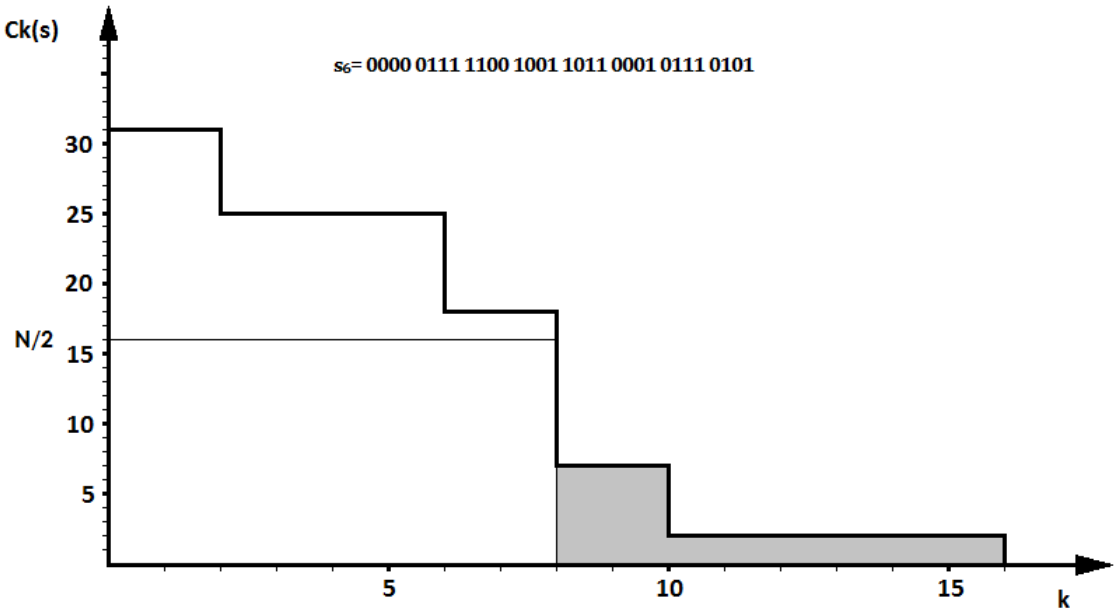


Διάγραμμα 5.3 : Προφίλ Γραμμικής Πολυπλοκότητας k-σφαλμάτων ακολουθίας s_5.

Της ακολουθίας № 6 :

s_6= 0000 0111 1100 1001 1011 0001 0111 0101

το προφίλ της γραμμικής πολυπλοκότητας k σφαλμάτων αποτυπώνεται στο διάγραμμα 5.4.



Διάγραμμα 5.4 : Προφίλ Γραμμικής Πολυπλοκότητας k-σφαλμάτων ακολουθίας s_6.

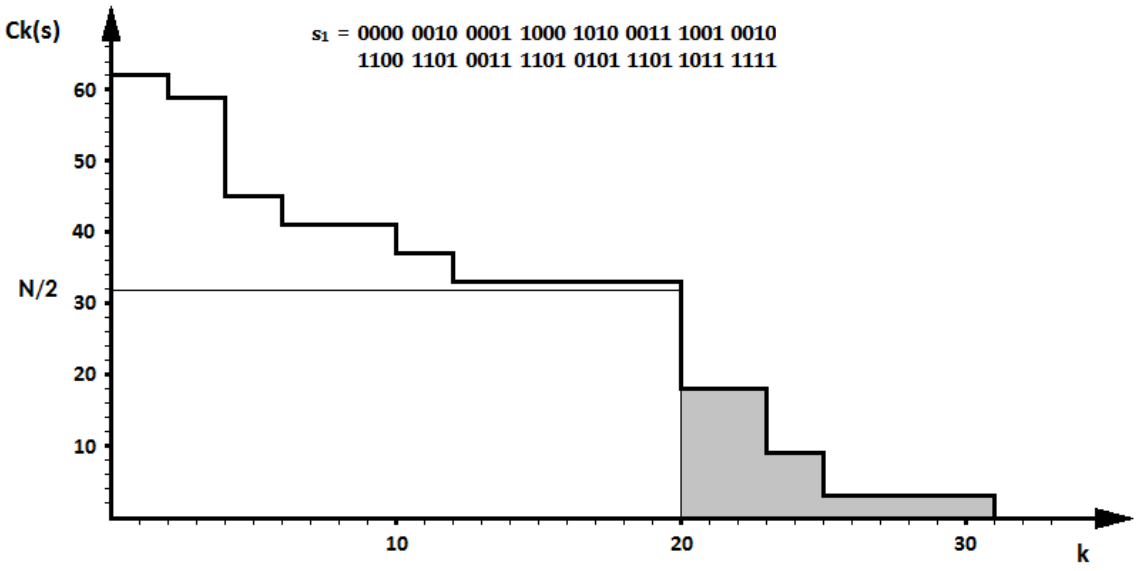
5.4.2 Ακολουθίες de Bruijn μεγέθους 64 bits

Αποτυπώνονται σε διαγράμματα τα προφίλ γραμμικής πολυπλοκότητας k σφαλμάτων των ακολουθιών, των οποίων οι τιμές Ck(s) = f(k) αναγράφονται στον πίνακα Α.2 του παραρτήματος Α.2.2. Οι επτά (7) αυτές ακολουθίες βρίσκονται στο αρχείο De_Bruijn_7.txt, και έχουν αριθμηθεί ως № 1 ÷ № 7.

Της ακολουθίας № 1 :

s_1 = 0000 0010 0001 1000 1010 0011 1001 0010 1100 1101 0011 1101 0101 1101 1011 1111

το προφίλ της γραμμικής πολυπλοκότητας k σφαλμάτων αποτυπώνεται στο διάγραμμα 5.5.

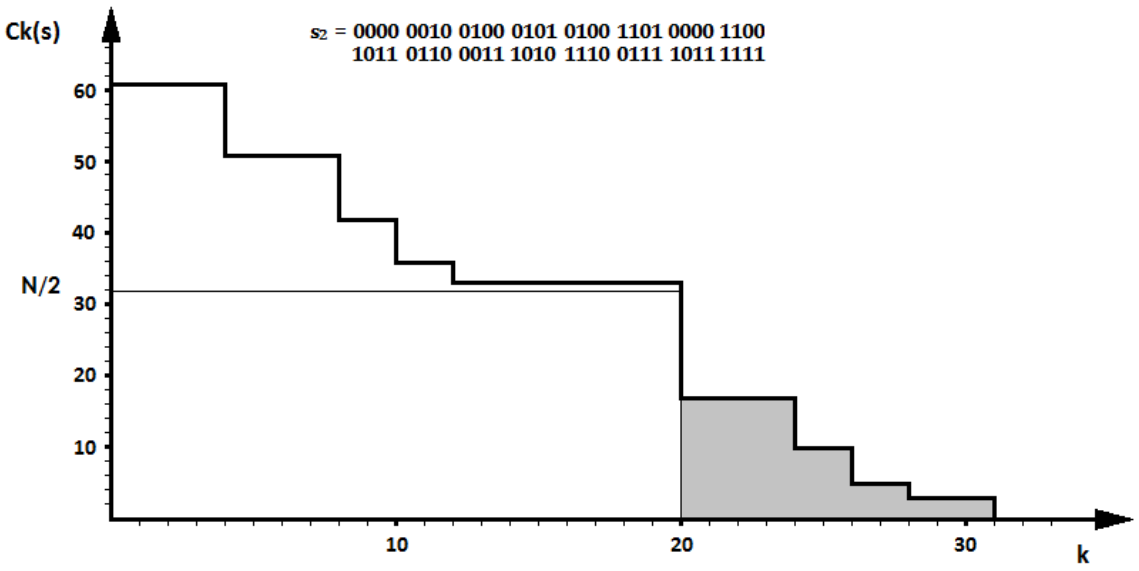


Διάγραμμα 5.5 : Προφίλ Γραμμικής Πολυπλοκότητας k-σφαλμάτων ακολουθίας s_1.

Της ακολουθίας № 2 :

s_2 = 0000 0010 0100 0101 0100 1101 0000 1100 1011 0110 0011 1010 1110 0111 1011 1111

το προφίλ της γραμμικής πολυπλοκότητας k σφαλμάτων αποτυπώνεται στο διάγραμμα 5.6.

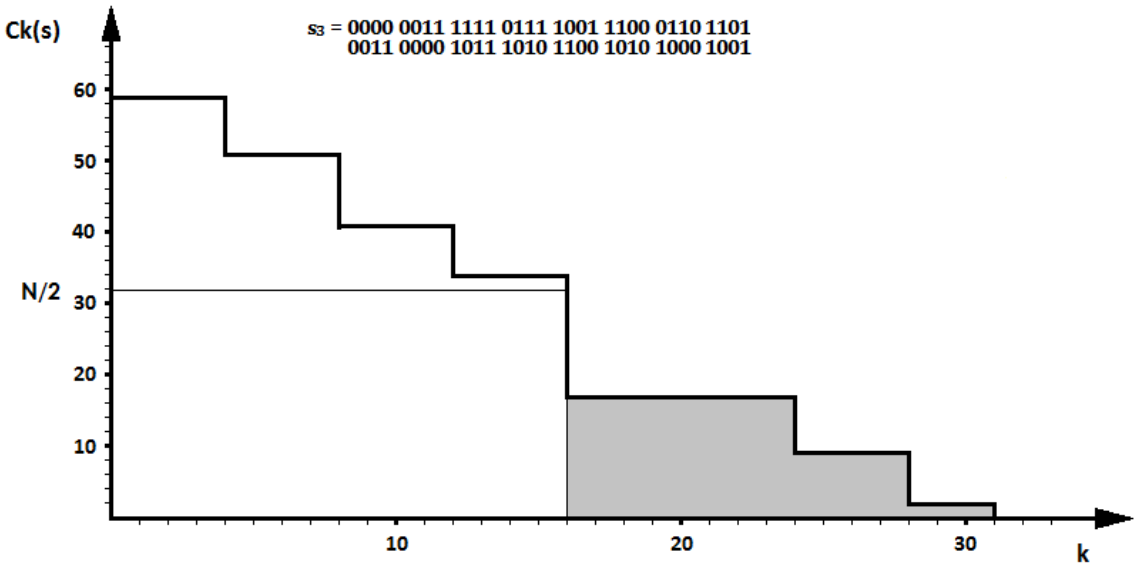


Διάγραμμα 5.6 : Προφίλ Γραμμικής Πολυπλοκότητας k-σφαλμάτων ακολουθίας s_2.

Της ακολουθίας № 3 :

s_3 = 0000 0011 1111 0111 1001 1100 0110 1101 0011 0000 1011 1010 1100 1010 1000 1001

το προφίλ της γραμμικής πολυπλοκότητας k σφαλμάτων αποτυπώνεται στο διάγραμμα 5.7.

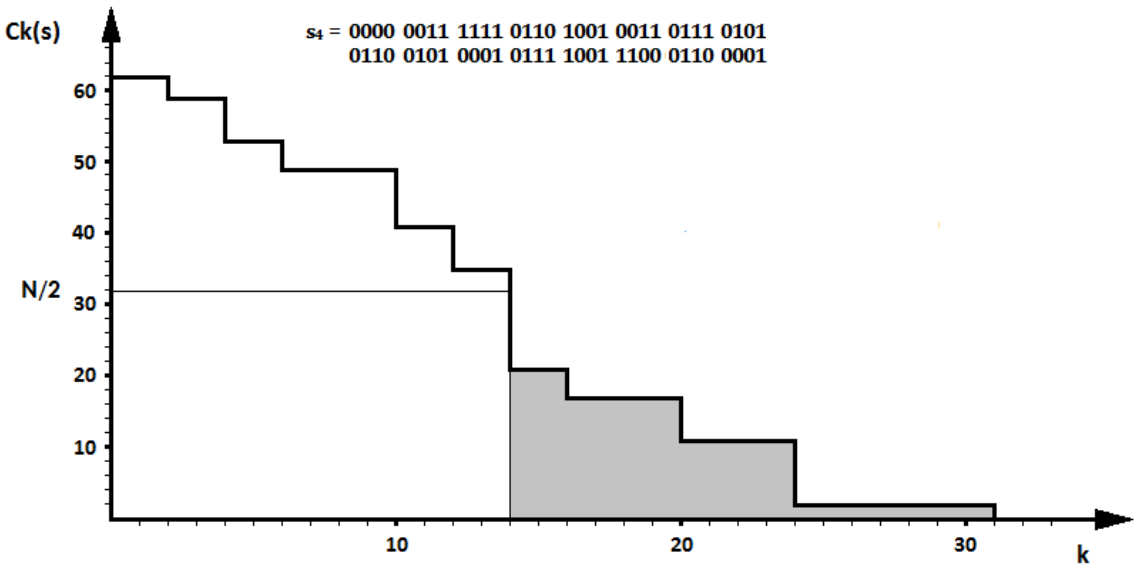


Διάγραμμα 5.7 : Προφίλ Γραμμικής Πολυπλοκότητας k-σφαλμάτων ακολουθίας s_3 .

Της ακολουθίας № 4 :

s_4 = 0000 0011 1111 0110 1001 0011 0111 0101 0110 0101 0001 0111 1001 1100 0110 0001

το προφίλ της γραμμικής πολυπλοκότητας k σφαλμάτων αποτυπώνεται στο διάγραμμα 5.8.

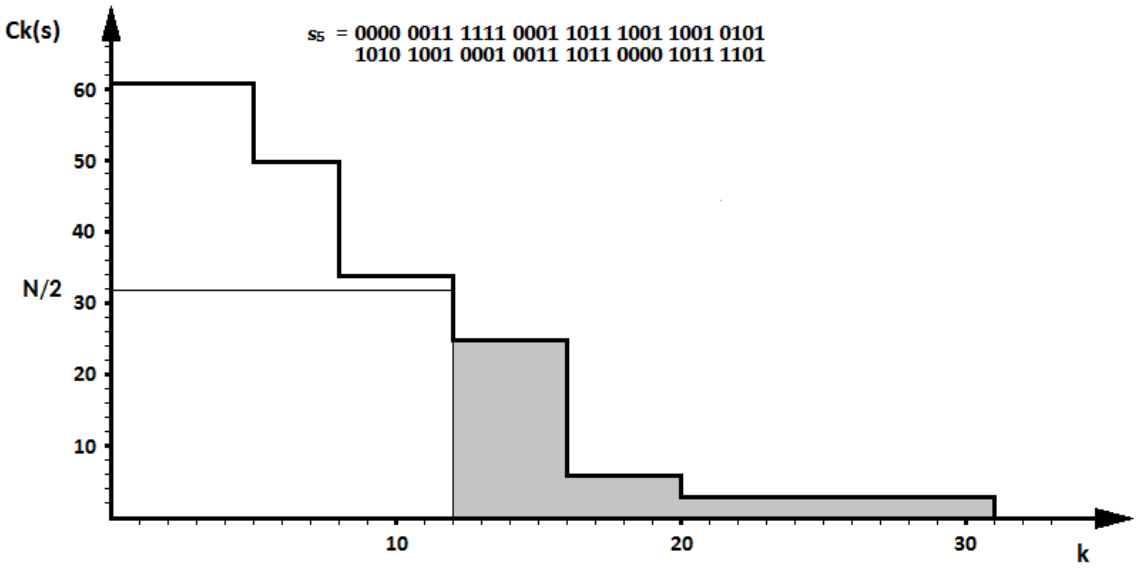


Διάγραμμα 5.8 : Προφίλ Γραμμικής Πολυπλοκότητας k-σφαλμάτων ακολουθίας s_4 .

Της ακολουθίας № 5 :

s_5 = 0000 0011 1111 0001 1011 1001 1001 0101 1010 1001 0001 0011 1011 0000 1011 1101

το προφίλ της γραμμικής πολυπλοκότητας k σφαλμάτων αποτυπώνεται στο διάγραμμα 5.9.



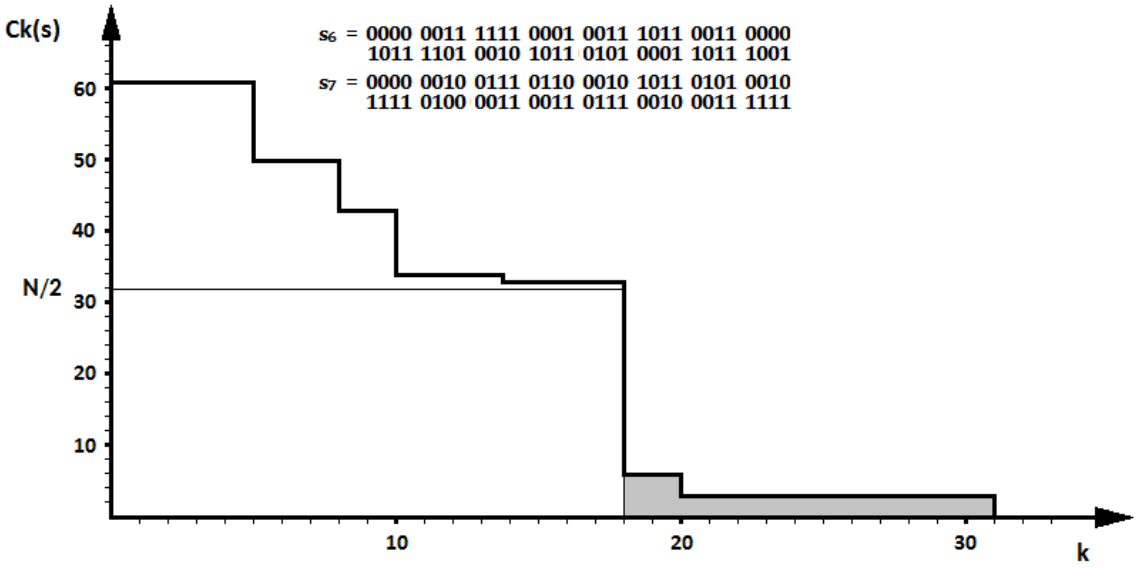
Διάγραμμα 5.9 : Προφίλ Γραμμικής Πολυπλοκότητας k-σφαλμάτων ακολουθίας s_5 .

Η ακολουθία № 6 και η ακολουθία № 7 παρουσιάζουν ίδιες τιμές Ck(s) = f(k) :

s_6 = 0000 0011 1111 0001 0011 1011 0011 0000 1011 1101 0010 1011 0101 0001 1011 1001

s_7 = 0000 0010 0111 0110 0010 1011 0101 0010 1111 0100 0011 0011 0111 0010 0011 1111

των οποίων το προφίλ της γραμμικής πολυπλοκότητας k σφαλμάτων αποτυπώνεται στο διάγραμμα 5.10.



Διάγραμμα 5.10 : Προφίλ Γραμμικής Πολυπλοκότητας k-σφαλμάτων ακολουθιών s_6 και s_7.

5.5 Πρόγραμμα Lauder–Paterson

Οι Alan G. B. Lauder και Kenneth G. Paterson, στο άρθρο «Computing the Error Linear Complexity Spectrum of a Binary Sequence of Period 2^n» (2003), περιγράφουν αλγόριθμο που παράγει φάσμα γραμμικών πολυπλοκοτήτων για ακολουθίες με σφάλματα (error linear complexity spectrum ή ELCS). Αυτό επιτυγχάνεται υπολογίζοντας όλα τα κρίσιμα σημεία (critical points) ως ζεύγη τιμών (k, $C_k(s)$), κατά τα οποία παρατηρείται μείωση στην τιμή της γραμμικής πολυπλοκότητας $C_k(s)$ [32].

Ο κώδικας που χρησιμοποιήθηκε αντλήθηκε από την αναφερόμενη στο άρθρο διαδικτυακή διεύθυνση http://www.isg.rhul.ac.uk/~kp/CELCS.c και τροποποιήθηκε ανάλογα για τις ανάγκες της παρούσης έρευνας, από τον γράφοντα. Ο πηγαίος κώδικας του προγράμματος που χρησιμοποιήθηκε για την έρευνα της μεταπτυχιακής διατριβής, σε γλώσσα προγραμματισμού C++, βρίσκεται στο Παράρτημα Β.3.

Πρόκειται για πρόγραμμα υπολογισμού των κρίσιμων σημείων (critical points) των εξεταζομένων ακολουθιών de Bruijn (εισάγονται από το αρχείο «**De_Bruijn_X.txt**» με X=5,6, …, 11) με χρήση του αλγόριθμου Lauder–Paterson , το οποίο βασίζεται στη συνάρτηση celcs. Στην συνέχεια, για κάθε κρίσιμο σημείο που υπολογίστηκε, ελέγχεται εάν η τιμή της γραμμικής πολυπλοκότητας $C_k(s)$ εμφανίζεται μικρότερη από $\mathbf{2^{n-1}}$ καθώς και το πλήθος των αλλοιωμένων ψηφίων k που την διαμορφώνουν. Τέλος πραγματοποιείται έλεγχος κρυπτογραφικής ισχύος της κάθε ακολουθίας βάσει των ερευνητικών κριτηρίων (παρ. 5.1).

Επιλογή τιμών μεγέθους των ακολουθιών αλλά και των ονομάτων αρχείων εισόδου και εξόδου, πραγματοποιείται μέσω των εντολών ορισμού (παρακάτω για ακολουθίες μεγέθους N =128 bits, προερχόμενες από το αρχείο «De_Bruijn_7.txt» και Excel αρχείο προορισμού το «LauderPaterson_7.csv») :

```
#define N 128/* N is the period of the input sequence */
#define H 12 /* H is the number of sequences */
#define FileName "De_Bruijn_7.txt"  /* Input de Bruijn sequences */
#define LauderPatersonExcelFile "LauderPaterson_7.csv" /* Output Results to Excel */
```

Αρχικά εκτελείται η ρουτίνα **GetDBSequence**, η οποία ενημερώνει τον πίνακα **DBSequ[H][N]** τοποθετώντας τις τιμές των ακολουθιών μεγέθους N προς δοκιμή. Στην συνέχεια, μέσω ενός επαναληπτικού βρόγχου (loop) για το πλήθος των προς εξέταση ακολουθιών (**H**) , ο πίνακας **s**

λαμβάνει τα περιεχόμενα της κάθε φορά εξεταζόμενης ακολουθίας, ενώ αρχικοποιείται ο πίνακας **cost** με 1.

Εντός του προαναφερθέντος βρόγχου καλείται η συνάρτηση **celcs**, εκκινώντας την για κάθε ακολουθία. Αρχικές παράμετροι (ορίσματα) ορίζονται η ακολουθία προς εξέταση **s**, έχοντας αρχική τιμή την κάθε ακολουθία του πίνακα DBSequ , το διάνυσμα κόστους **cost** αρχικοποιημένο με τιμές 1, το αρχικό μήκος της ακολουθίας προς εξέταση **N**, το αρχικό συνολικό κόστος **0**, το αρχικό όριο κοστών **N** και η αρχική τιμή γραμμικής πολυπλοκότητας **0**.

```
GetDBSequence( );  /*  input the initial sequence of N bits */
for (j = 0; j < H; j++)  {
        /*  setting s sequence for examine and all costs to 1 */
        for (i = 0; i< N; i++)  {
                s[i] = DBSequ[j][i];
                cost[i] = 1;  }
        /* now run the celcs algorithm  */
        celcs(s, cost, N, 0, N, 0);
}
```

Η συνάρτηση **celcs** καλείται αναδρομικά, με προσαρμογή των παραμέτρων της κάθε φορά, ώστε να δημιουργηθεί δέντρο υπολογισμών για κρίσιμα σημεία. Όταν το μέγεθος της ακολουθίας που εξετάζεται, σε κάθε κλάδο του δέντρου λάβει τιμή 1, υπολογίζονται και αποθηκεύονται τα κρίσιμα σημεία.

Παράλληλα, όπως προαναφέρθηκε, για κάθε υπολογισθέν κρίσιμο σημείο, εκτελείται έλεγχος μέσω της ρουτίνας **CheckResults** σχετικά με το αν η τιμή της γραμμικής πολυπλοκότητας $C_k(s)$ είναι μικρότερη απο $\mathbf{2^{n-1}}$. Επίσης, γίνεται έλεγχος στην τιμή του δείκτη αποτίμησης ισχύος.

Η αποθήκευση των κρίσιμων σημείων αρχικά γίνεται στον τρισδιάστατο πίνακα **Results**, ενώ, με την ολοκλήρωση της διαδικασίας για όλες τις ακολουθίες προς δοκιμή, τα περιεχόμενά του σώζονται σε αρχείο τύπου Excel μέσω της ρουτίνας **SaveResults**.

5.6 Αποτελέσματα Προγράμματος Lauder–Paterson

Για κάθε μία από τις εξεταζόμενες ακολουθίες de Bruijn προς δοκιμή, το πρόγραμμα υπολογίζει τα κρίσιμα σημεία της, ως ζεύγος τιμών <k, $C_k(s)$>. Ελέγχεται - για την κάθε ακολουθία - το πλήθος των αλλοιωμένων ψηφίων k που οδηγεί τη γραμμική πολυπλοκότητα ακολουθίας de

Bruijn σε τιμή $C_k(s) \leq 2^{n-1}$ και υπολογίζεται ο δείκτης αποτίμησης ισχύος. Στην περίπτωση που αυτός λάβει τιμή μικρότερη του N εμφανίζονται τα αποτελέσματα προς μελέτη.

Στους πίνακες εμφάνισης των αποτελεσμάτων, ανάλογα με το μέγεθος των εξεταζομένων ακολουθιών de Bruijn, ερευνητικό ενδιαφέρον παρουσιάζει το πρώτο ζεύγος τιμών <k, $C_k(s)$> κατά το οποίο ο δείκτης αποτίμησης κρυπτογραφικής ισχύος $P = 2 \cdot C_k(s) + k$ λαμβάνει τιμή μικρότερη από N. Τούτο συμβαίνει διότι το πλήθος των αλλοιωμένων ψηφίων k είναι το ελάχιστο δυνατό διαμόρφωσης της τροποποιημένης ακολουθίας ώστε αυτή να λάβει χαμηλή τιμή γραμμικής πολυπλοκότητας και να θεωρηθεί κρυπτογραφικά αδύναμη. Οι πίνακες εμφάνισης των αναλογιών, που παρουσιάζουν τα κρυπτογραφικά μεγέθη που εξετάζει η μεταπτυχιακή διατριβή, είναι ποσοστά επι τοις εκατό σε σχέση με τη μέγιστη τιμή που μπορεί να λάβει το κάθε μέγεθος.

5.6.1 Ακολουθίες de Bruijn μεγέθους 32 bits

Για τις ακολουθίες de Bruijn μήκους N=32 bits, n=5, οι οποίες βρίσκονται στο αρχείο «De_Bruijn_5.txt», τα ζεύγη τιμών <k, $C_k(s)$> που υπολογίστηκαν ως κρίσιμα σημεία από το πρόγραμμα και αποθηκεύτηκαν στο αρχείο «LauderPaterson_5.csv», απεικονίζονται στον Πίνακα Α.3 του Παραρτήματος Α.3.1. Τα ζεύγη τιμών <k, $C_k(s)$> κατά τα οποία η γραμμική πολυπλοκότητα εμφανίζεται μικρότερη του N/2 (2^{n-1}) - με έντονη γραμματοσειρά όταν ο δείκτης αποτίμησης κρυπτογραφικής αξίας παρουσιάζει τιμές μικρότερες του N (2^n) - αποτυπώνονται στον Πίνακα 5.11.

Sequences Length N=**32** (n=5)			
Sequence	**$C_k(s)$**	**k**	**P**
No 1 & No 2	**9**	**10**	**28**
	2	**14**	**18**
No 3 & No 4	**9**	**8**	**26**
	6	**10**	**22**
	3	**12**	**18**
	2	**14**	**18**
No 5	11	10	32
	2	**12**	**16**
No 6	**7**	**8**	**22**
	2	**10**	**14**

Πίνακας 5.11 : Τιμές Ζευγών <k, $C_k(s)$> με $P = 2 \cdot C_k(s) + k$ (n=5).

Παρατηρείται ότι μόνο οι πρώτες από τις τιμές των ζευγών <k, $C_k(s)$>, για κάθε ακολουθία, εμφανίζει ερευνητικό ενδιαφέρον, εφόσον για τα υπόλοιπα ζεύγη είτε η τιμή γραμμικής πολυπλοκότητας $C_k(s)$ είναι πολύ χαμηλή (οπότε η ακολουθία θεωρείται εκ των προτέρων κρυπτογραφικά ασθενής), είτε το πλήθος των αλλοιωμένων ψηφίων k είναι υψηλό (άρα θεωρείται ότι η τροποποιημένη ακολουθία δεν προσεγγίζει την αρχική).

Στον πίνακα 5.12 αποτυπώνονται συνοπτικά οι αναλογίες των μεγεθών γραμμικής πολυπλοκότητας k σφαλμάτων $C_k(s)$, πλήθους αλλοιωμένων ψηφίων k και δείκτη αποτίμησης κρυπτογραφικής αξίας P, για τα πρώτα ζεύγη τιμών – ανά ακολουθία - που ο δείκτης αποτίμησης κρυπτογραφικής αξίας παρουσιάζει τιμές μικρότερες του N

Seq	$C_0(s)$	$C_k(s)$	%	k	%	P	%
No 1	31	9	29.03	10	31.25	28	87.50
No 2	31	9	29.03	10	31.25	28	87.50
No 3	31	9	29.03	8	25.00	26	81.25
No 4	31	9	29.03	8	25.00	26	81.25
No 5	30	2	6.66	12	37.50	16	50.00
No 6	31	7	22.58	8	25.00	22	68.75
Mean		**7.3**	**24.22**	**9.2**	**29.16**	**24.2**	**76.04**

Πίνακας 5.12 : Αναλογίες Μεγεθών για τις Πρώτες Τιμές Ζευγών <k, $C_k(s)$> όταν P < N (n=5).

Για ακολουθίες μεγέθους 32 bits η 5η ακολουθία εμφανίζει μικρή τιμή του δείκτη κρυπτογραφικής αξίας, μόλις 16 με ποσοστό 50%, αλλά το δείγμα θεωρείται μικρό.

5.6.2 Ακολουθίες de Bruijn μεγέθους 64 bits

Για τις ακολουθίες de Bruijn μήκους N=64 bits, n=6, οι οποίες βρίσκονται στο αρχείο «De_Bruijn_6.txt», τα ζεύγη τιμών <k, $C_k(s)$> που υπολογίστηκαν ως κρίσιμα σημεία από το πρόγραμμα και αποθηκεύτηκαν στο αρχείο «LauderPaterson_6.csv», απεικονίζονται στον Πίνακα Α.4 του Παραρτήματος Α.3.2. Τα ζεύγη τιμών <k, $C_k(s)$> κατά τα οποία η γραμμική πολυπλοκότητα εμφανίζεται μικρότερη του N/2 (2^{n-1}) - με έντονη γραμματοσειρά όταν ο δείκτης αποτίμησης κρυπτογραφικής αξίας παρουσιάζει τιμές μικρότερες του N (2^n) - αποτυπώνονται στον Πίνακα 5.13.

Όλες οι ακολουθίες παρουσιάζουν αναμενόμενα αποτελέσματα, πλην των ακολουθιών 6 και 7 που εμφανίζουν πολύ ασθενή κρυπτογραφικά χαρακτηριστικά, δηλαδή χαμηλή τιμή γραμμικής πολυπλοκότητας σε συνάρτηση με το πλήθος αλλοιωμένων ψηφίων.

Sequences Length N=**64** (n=6)			
Sequence	**$C_k(s)$**	**k**	**P**
No 1	**18**	**20**	**56**
	9	**24**	**42**
	3	**26**	**32**
No 2	**17**	**20**	**54**
	10	**24**	**44**
	5	**26**	**36**
	3	**28**	**34**
No 3	**17**	**16**	**50**
	9	**24**	**42**
	2	**28**	**32**
No 4	**21**	**14**	**56**
	17	**16**	**50**
	11	**20**	**42**
	2	**24**	**28**
No 5	**25**	**12**	**62**
	6	**16**	**28**
	3	**20**	**26**
No 6 & No 7	**6**	**18**	**30**
	3	**20**	**26**

Πίνακας 5.13 : Τιμές Ζευγών <k, $C_k(s)$> με $P = 2 \cdot C_k(s) + k < 2^n$ (n=6).

Στον πίνακα 5.14 αποτυπώνονται συνοπτικά οι αναλογίες των μεγεθών γραμμικής πολυπλοκότητας k σφαλμάτων $C_k(s)$, πλήθους αλλοιωμένων ψηφίων k και δείκτη αποτίμησης κρυπτογραφικής αξίας P, για τα πρώτα ζεύγη τιμών – ανά ακολουθία - που ο δείκτης αποτίμησης κρυπτογραφικής αξίας παρουσιάζει τιμές μικρότερες του N

Seq	**$C_0(s)$**	**$C_k(s)$**	**%**	**K**	**%**	**P**	**%**
No 1	62	18	29.03	20	31.25	56	87.50
No 2	61	17	27.86	20	31.25	54	84.37
No 3	59	17	28.81	16	25.00	50	78.12
No 4	62	21	33.87	14	21.87	56	87.50
No 5	61	25	40.98	12	18.75	62	96.87
No 6	61	6	9.83	18	28.12	30	46.87
No 7	61	6	9.83	18	28.12	30	46.87
Mean		**15.7**	**25.74**	**16.8**	**26.33**	**48.3**	**75.44**

Πίνακας 5.14 : Αναλογίες Μεγεθών για τις Πρώτες Τιμές Ζευγών <k, $C_k(s)$> όταν P < N (n=6).

5.6.3 Ακολουθίες de Bruijn μεγέθους 128 bits

Για τις ακολουθίες de Bruijn μήκους N=128 bits, n=7, οι οποίες βρίσκονται στο αρχείο «De_Bruijn_7.txt», τα ζεύγη τιμών <k, $C_k(s)$> που υπολογίστηκαν ως κρίσιμα σημεία από το πρόγραμμα και αποθηκεύτηκαν στο αρχείο «LauderPaterson_7.csv», απεικονίζονται στους Πίνακες Α.5.1 και Α.5.2 του Παραρτήματος Α.3.3. Τα ζεύγη τιμών <k, $C_k(s)$> κατά τα οποία η γραμμική πολυπλοκότητα εμφανίζεται μικρότερη του N/2 (2^{n-1}) - με έντονη γραμματοσειρά όταν ο δείκτης αποτίμησης κρυπτογραφικής αξίας παρουσιάζει τιμές μικρότερες του N (2^n) - αποτυπώνονται στον Πίνακα 5.15.

Sequences Length N=**128** (n=7)							
Sequence	**$C_k(s)$**	**k**	**P**	**Sequence**	**$C_k(s)$**	**k**	**P**
No 1	**37**	**38**	**112**	No 2	**37**	**38**	**112**
	34	**40**	**108**		**33**	**40**	**106**
	33	**42**	**108**		**21**	**46**	**88**
	18	**44**	**80**				
No 3	**38**	**34**	**110**	No 4	49	32	130
	34	**36**	**104**		**35**	**36**	**106**
	33	**38**	**104**		**33**	**38**	**104**
	22	**42**	**86**		**20**	**40**	**80**
No 5	**37**	**36**	**110**	No 6	**21**	**36**	**78**
	33	**38**	**104**		**19**	**40**	**78**
	25	**40**	**90**		**13**	**42**	**68**
No 7	**41**	**30**	**112**	No 8	49	38	136
	35	**36**	**106**		**33**	**40**	**106**
	21	**38**	**80**		**11**	**46**	**68**
No 9	**34**	**34**	**102**	No 10	49	30	128
	33	**40**	**106**		**41**	**32**	**114**
	25	**42**	**92**		**36**	**34**	**106**
					26	**36**	**88**
No 11	**34**	**36**	**104**	No 12	**43**	**28**	**114**
	21	**40**	**82**		**34**	**30**	**98**
					33	**34**	**100**
					18	**44**	**80**

Πίνακας 5.15 : Τιμές Ζευγών <k, $C_k(s)$> με $P = 2 \cdot C_k(s) + k < 2^n$ (n=7).

Από τις ακολουθίες de Bruijn με μεγέθους N=128 bits που εξετάστηκαν ιδιαίτερο ενδιαφέρον παρουσιάζει η ακολουθία № 6, της οποίας η γραμμική πολυπλοκότητα - τροποποιώντας 36 ψηφία της αρχικής - λαμβάνει τιμή εξαιρετικά χαμηλή, ίση με 21. Αντιθέτως η ακολουθία № 7

(όπου στον πίνακα 5.13 –ακολουθίας μεγέθους 64 bits κατασκευασμένη με την ίδια μέθοδο - εμφάνιζε ασθενή κρυπτογραφικά χαρακτηριστικά) εμφανίζει κρυπτογραφική συμπεριφορά παρόμοια με τις υπόλοιπες.

Στον πίνακα 5.16 αποτυπώνονται συνοπτικά οι αναλογίες των μεγεθών γραμμικής πολυπλοκότητας k σφαλμάτων $C_k(s)$, πλήθους αλλοιωμένων ψηφίων k και δείκτη αποτίμησης κρυπτογραφικής αξίας P, για τα πρώτα ζεύγη τιμών – ανά ακολουθία - που ο δείκτης αποτίμησης κρυπτογραφικής αξίας παρουσιάζει τιμές μικρότερες του N

Seq	$C_0(s)$	$C_k(s)$	%	k	%	P	%
No 1	127	37	29.13	38	29.68	112	87.50
No 2	127	37	29.13	38	29.68	112	87.50
No 3	127	38	29.92	34	26.56	110	85.93
No 4	126	35	27.77	36	28.12	106	82.81
No 5	125	37	29.60	36	28.12	110	85.93
No 6	125	21	16.80	36	28.12	78	60.93
No 7	126	41	32.53	30	23.43	112	87.50
No 8	126	33	26.19	40	31.25	106	82.81
No 9	118	34	28.81	34	26.56	102	79.68
No 10	126	41	32.53	32	25.00	114	89.06
No 11	127	34	26.77	36	28.12	104	81.25
No 12	124	43	34.67	28	21.87	114	89.06
Mean		**35.9**	**28.65**	**34.8**	**27.2**	**106.7**	**83.33**

Πίνακας 5.16 : Αναλογίες Μεγεθών για τις Πρώτες Τιμές Ζευγών <k, $C_k(s)$> όταν P < N (n=7).

5.6.4 Ακολουθίες de Bruijn μεγέθους 256 bits

Για τις ακολουθίες de Bruijn μήκους N=256 bits, n=8, οι οποίες βρίσκονται στο αρχείο «De_Bruijn_8.txt», τα ζεύγη τιμών <k, $C_k(s)$> που υπολογίστηκαν ως κρίσιμα σημεία από το πρόγραμμα και αποθηκεύτηκαν στο αρχείο «LauderPaterson_8.csv», απεικονίζονται στους Πίνακες Α.6.1 και Α.6.2 του Παραρτήματος Α.3.4. Τα ζεύγη τιμών <k, $C_k(s)$> κατά τα οποία η γραμμική πολυπλοκότητα εμφανίζεται μικρότερη του N/2 (2^{n-1}) - με έντονη γραμματοσειρά όταν ο δείκτης αποτίμησης κρυπτογραφικής αξίας παρουσιάζει τιμές μικρότερες του N (2^n) - αποτυπώνονται στον Πίνακα 5.17.

Sequences Length N=**256** (n=8)							
Sequence	**$C_k(s)$**	**K**	**P**	**Sequence**	**$C_k(s)$**	**k**	**P**
No 1	98	60	256	No 2	99	58	256
	89	**62**	**240**		**97**	**60**	**254**
	81	**64**	**226**		**75**	**62**	**212**
	69	**66**	**204**		**69**	**64**	**202**
No 3	113	48	274	No 4	101	60	262
	105	50	260		**77**	**62**	**216**
	97	**52**	**246**		**69**	**64**	**202**
	81	**56**	**218**		**43**	**72**	**158**
No 5	**55**	**52**	**162**	No 6	**71**	**44**	**186**
	9	**54**	**72**		**69**	**48**	**186**
					37	**52**	**126**
No 7	**69**	**42**	**180**	No 8	105	58	268
	23	**54**	**100**		101	60	262
	21	**58**	**100**		**75**	**62**	**212**
					73	**64**	**210**
					52	**72**	**176**
No 9	**81**	**60**	**222**	No 10	97	64	258
	72	**64**	**208**		**85**	**68**	**238**
	66	**66**	**198**		**82**	**70**	**234**
	53	**70**	**176**		**65**	**78**	**208**
					41	**88**	**170**
No 11	97	70	264	No 12	97	64	258
	77	**72**	**226**		**81**	**66**	**228**
	65	**76**	**206**		**77**	**68**	**222**
	41	**92**	**174**		**51**	**82**	**184**

Πίνακας 5.17 : Τιμές Ζευγών <k, $C_k(s)$> με $P = 2 \cdot C_k(s) + k < 2^n$ (n=8).

Η ακολουθία No 5 με μέγεθος 256 bits εμφανίζει ενδιαφέρον, διότι παρουσιάζεται με ασθενέστερα κρυπτογραφικά χαρακτηριστικά σε σχέση με την αντίστοιχη μεγέθους 128 bits.

Στον πίνακα 5.18 αποτυπώνονται συνοπτικά οι αναλογίες των μεγεθών γραμμικής πολυπλοκότητας k σφαλμάτων $C_k(s)$, πλήθους αλλοιωμένων ψηφίων k και δείκτη αποτίμησης κρυπτογραφικής αξίας P, για τα πρώτα ζεύγη τιμών – ανά ακολουθία - που ο δείκτης αποτίμησης κρυπτογραφικής αξίας παρουσιάζει τιμές μικρότερες του N.

Seq	$C_0(s)$	$C_k(s)$	%	K	%	P	%
No 1	254	89	35.03	62	24.21	240	93.75
No 2	251	97	38.64	60	23.43	254	99.21
No 3	255	97	38.03	52	20.31	246	96.09
No 4	255	77	30.19	62	24.21	216	84.37
No 5	248	55	22.17	52	20.31	162	63.28
No 6	247	71	28.74	44	17.18	186	72.65
No 7	248	69	27.82	42	16.40	180	70.31
No 8	254	75	29.52	62	24.21	212	82.81
No 9	253	81	32.01	60	23.43	222	86.71
No 10	255	85	33.33	68	26.56	238	92.96
No 11	254	77	30.31	72	28.12	226	88.28
No 12	255	81	31.76	66	25.78	228	89.06
Mean		**79.5**	**31.46**	**58.5**	**22.84**	**217.5**	**84.95**

Πίνακας 5.18 : Αναλογίες Μεγεθών για τις Πρώτες Τιμές Ζευγών <k, $C_k(s)$> όταν P < N (n=8).

Η ακολουθία № 5 παρουσιάζει ασθενή κρυπτογραφικά χαρακτηριστικά, οι δε ακολουθίες № 6 και № 7 εμφανίζουν χαμηλές τιμές ποσοστών σε όλα τα μετρήσιμα μεγέθη, ιδιαίτερα στις τιμές του k με ποσοστά μικρότερα του 20%. Ειδικά στην ακολουθία № 7 :

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

τροποποιώντας 42 ψηφία από τα συνολικά 256 (ποσοστό 16,4%), η γραμμική πολυπλοκότητα λαμβάνει τιμή 69 (ποσοστό 27,82%).

Οι υπόλοιπες ακολουθίες εμφανίζουν αναμενόμενα αποτελέσματα.

5.6.5 Ακολουθίες de Bruijn μεγέθους 512 bits

Για τις ακολουθίες de Bruijn μήκους N=512 bits, n=9, οι οποίες βρίσκονται στο αρχείο «De_Bruijn_9.txt», τα ζεύγη τιμών <k, $C_k(s)$> που υπολογίστηκαν ως κρίσιμα σημεία από το πρόγραμμα και αποθηκεύτηκαν στο αρχείο «LauderPaterson_9.csv», απεικονίζονται στους Πίνακες Α.7.1 και Α.7.2 του Παραρτήματος Α.3.5. Τα ζεύγη τιμών <k, $C_k(s)$> κατά τα οποία η γραμμική πολυπλοκότητα εμφανίζεται μικρότερη του N/2 (2^{n-1}) - με έντονη γραμματοσειρά όταν ο δείκτης αποτίμησης κρυπτογραφικής αξίας παρουσιάζει τιμές μικρότερες του N (2^n) - αποτυπώνονται στον Πίνακα 5.19.

Sequences Length N=**512** (n=9)							
Sequence	**$C_k(s)$**	**k**	**P**	**Sequence**	**$C_k(s)$**	**k**	**P**
No 1	201	134	536	No 2	195	142	532
	162	**136**	**460**		**149**	**144**	**442**
	149	**138**	**436**		**145**	**148**	**436**
	97	**160**	**354**		**101**	**156**	**358**
No 3	195	132	522	No 4	193	138	524
	193	134	520		**177**	**142**	**496**
	162	**136**	**460**		**150**	**144**	**444**
	100	**154**	**354**		**78**	**178**	**334**
No 5	201	120	522	No 6	195	122	512
	195	122	512		193	126	512
	166	**124**	**456**		**148**	**130**	**426**
	101	**150**	**352**		**106**	**146**	**358**
No 7	209	138	556	No 8	201	128	530
	197	142	536		193	130	516
	195	144	534		**163**	**136**	**462**
	163	**146**	**472**		**147**	**138**	**432**
	100	**156**	**354**		**103**	**160**	**366**
No 9	225	116	566	No 10	201	110	512
	193	**118**	**504**		**196**	**112**	**504**
	178	**122**	**478**		**193**	**114**	**500**
	101	**154**	**356**		**105**	**150**	**360**
No 11	197	140	534	No 12	193	130	516
	164	**142**	**470**		**162**	**134**	**458**
	155	**144**	**454**		**147**	**136**	**430**
	97	**168**	**362**		**105**	**148**	**358**

Πίνακας 5.19 : Τιμές Ζευγών <k, $C_k(s)$> με $P = 2 \cdot C_k(s) + k < 2^n$ (n=9).

Ενώ ήταν αναμενόμενο οι ακολουθίες 5, 6 και 7 να παρουσιάζουν ασθενέστερα χαρακτηριστικά από τις υπόλοιπες, εντούτοις όλες οι ακολουθίες de Bruijn αυτού του μεγέθους που δοκιμάστηκαν, παρουσιάζουν παρόμοια κρυπτογραφικά χαρακτηριστικά, τα οποία είναι σχετικά καλά.

Στον πίνακα 5.20 αποτυπώνονται συνοπτικά οι αναλογίες των μεγεθών γραμμικής πολυπλοκότητας k σφαλμάτων $C_k(s)$, πλήθους αλλοιωμένων ψηφίων k και δείκτη αποτίμησης κρυπτογραφικής αξίας P, για τα πρώτα ζεύγη τιμών – ανά ακολουθία - που ο δείκτης αποτίμησης κρυπτογραφικής αξίας παρουσιάζει τιμές μικρότερες του N.

Seq	$C_0(s)$	$C_k(s)$	%	k	%	P	%
No 1	507	162	31.95	136	26.56	460	89.84
No 2	509	149	29.27	144	28.12	442	86.32
No 3	511	162	31.70	136	26.56	460	89.84
No 4	510	177	34.70	142	27.73	496	96.87
No 5	510	166	32.54	124	24.21	456	89.06
No 6	510	148	29.01	130	25.39	426	83.20
No 7	511	163	31.89	146	28.51	472	92.18
No 8	511	163	31.89	136	26.56	462	90.23
No 9	509	193	37.91	118	23.04	504	98.43
No 10	511	196	38.35	112	21.87	504	98.43
No 11	511	164	32.09	142	27.73	470	91.79
No 12	511	162	31.70	134	26.17	458	89.45
Mean		**167.1**	**32.75**	**133.3**	**26.03**	**467.5**	**91.30**

Πίνακας 5.20 : Αναλογίες Μεγεθών για τις Πρώτες Τιμές Ζευγών <k, $C_k(s)$> όταν P < N (n=9).

Στον πίνακα 5.20 παρατηρείται ότι η αναλογία των τροποποιημένων ψηφίων k παρουσιάζει τιμές μεγαλύτερες του 20%, ενώ η αναλογία του δείκτη κρυπτογραφικής αξίας P κυμαίνεται σε τιμές 83 % ÷ 98 %. Οι τιμές αυτές στα συγκεκριμένα μεγέθη καθιστούν τις ακολουθίες de Bruijn μεγέθους 512 bits που εξετάστηκαν, κρυπτογραφικά ισχυρές.

Παρατηρείται στα έως εδώ αποτελέσματα ότι, αυξανομένου του μεγέθους των προς δοκιμή ακολουθιών de Bruijn, αυξάνεται το ποσοστό του δείκτη κρυπτογραφικής αξίας για τα πρώτα ζεύγη τιμών στα οποία ο δείκτης αποτίμησης κρυπτογραφικής αξίας παρουσιάζει τιμές μικρότερες του N.

5.6.6 Ακολουθίες de Bruijn μεγέθους 1024 bits

Για τις ακολουθίες de Bruijn μήκους N=1024 bits, n=10, οι οποίες βρίσκονται στο αρχείο «De_Bruijn_10.txt», τα ζεύγη τιμών <k, $C_k(s)$> που υπολογίστηκαν ως κρίσιμα σημεία από το πρόγραμμα και αποθηκεύτηκαν στο αρχείο «LauderPaterson_10.csv», απεικονίζονται στους Πίνακες Α.8.1 και Α.8.2 του Παραρτήματος Α.3.6. Τα ζεύγη τιμών <k, $C_k(s)$> κατά τα οποία η γραμμική πολυπλοκότητα εμφανίζεται μικρότερη του N/2 (2^{n-1}) - με έντονη γραμματοσειρά όταν ο δείκτης αποτίμησης κρυπτογραφικής αξίας παρουσιάζει τιμές μικρότερες του N (2^n) - αποτυπώνονται στον Πίνακα 5.21.

Sequences Length N=**1024** (n=10)							
Sequence	**$C_k(s)$**	**k**	**P**	**Sequence**	**$C_k(s)$**	**k**	**P**
No 1	385	284	1054	No 2	401	274	1076
	322	**292**	**936**		389	278	1056
	321	**296**	**938**		**323**	**282**	**928**
	198	**338**	**734**		**165**	**354**	**684**
No 3	393	266	1052	No 4	**403**	**208**	**1014**
	353	**268**	**974**		**395**	**210**	**1000**
	329	**270**	**928**		**391**	**212**	**994**
	163	**362**	**688**		**193**	**306**	**692**
No 5	421	244	1086	No 6	417	298	1132
	389	246	1024		385	300	1070
	338	**250**	**926**		**321**	**304**	**946**
	169	**332**	**670**		**164**	**346**	**674**
No 7	386	270	1042	No 8	394	260	1048
	385	272	1042		391	262	1044
	333	**276**	**942**		**355**	**264**	**974**
	169	**318**	**656**		**177**	**330**	**684**
No 9	386	260	1032	No 10	449	260	1158
	337	**264**	**938**		401	262	1064
	331	**266**	**928**		387	264	1038
	161	**342**	**664**		**345**	**266**	**956**
					170	**338**	**678**
No 11	389	250	1028	No 12	390	268	1048
	387	252	1026		386	270	1042
	341	**258**	**940**		**341**	**274**	**956**
	169	**330**	**668**		**169**	**342**	**680**

Πίνακας 5.21 : Τιμές Ζευγών <k, $C_k(s)$> με $P = 2 \cdot C_k(s) + k < 2^n$ (n=10).

Παρατηρείται ότι όλες οι δοκιμασμένες ακολουθίες de Bruijn παρουσιάζουν παρόμοια καλά κρυπτογραφικά χαρακτηριστικά.

Στον πίνακα 5.22 αποτυπώνονται συνοπτικά οι αναλογίες των μεγεθών γραμμικής πολυπλοκότητας k σφαλμάτων $C_k(s)$, πλήθους αλλοιωμένων ψηφίων k και δείκτη αποτίμησης κρυπτογραφικής αξίας P, για τα πρώτα ζεύγη τιμών – ανά ακολουθία - που ο δείκτης αποτίμησης κρυπτογραφικής αξίας παρουσιάζει τιμές μικρότερες του N.

Seq	$C_0(s)$	$C_k(s)$	%	k	%	P	%
No 1	1023	322	31.47	292	28.51	936	91.40
No 2	1023	323	31.57	282	27.53	928	90.62
No 3	1022	353	34.54	268	26.17	974	95.11
No 4	1023	403	39.39	208	20.31	1014	99.02
No 5	1022	338	33.07	250	24.41	926	90.42
No 6	1021	321	31.43	304	29.68	946	92.38
No 7	1016	333	32.77	276	26.95	942	91.99
No 8	1020	355	34.80	264	25.78	974	95.11
No 9	1021	337	33.00	264	25.78	938	91.60
No 10	1023	345	33.72	266	25.97	956	93.35
No 11	1022	341	33.36	258	25.19	940	91.79
No 12	1020	341	33.43	274	26.75	956	93.35
Mean		**342.6**	**33.54**	**267.2**	**26.08**	**952.5**	**93.01**

Πίνακας 5.22 : Αναλογίες Μεγεθών για τις Πρώτες Τιμές Ζευγών <k, $C_k(s)$> όταν P < N (n=10).

Επιβεβαιώνεται ο συλλογισμός ότι αυξάνοντας το μέγεθος των ακολουθιών de Bruijn που δοκιμάζονται αυξάνει το ποσοστό του δείκτη κρυπτογραφικής αξίας, και ως εκ τούτου τα κρυπτογραφικά χαρακτηριστικά των ακολουθιών.

5.6.7 Ακολουθίες de Bruijn μεγέθους 2048 bits

Για τις ακολουθίες de Bruijn μήκους N=2048 bits, n=11, οι οποίες βρίσκονται στο αρχείο «De_Bruijn_11.txt», τα ζεύγη τιμών <k, $C_k(s)$> που υπολογίστηκαν ως κρίσιμα σημεία από το πρόγραμμα και αποθηκεύτηκαν στο αρχείο «LauderPaterson_11.csv», απεικονίζονται στους Πίνακες Α.9.1 και Α.9.2 του Παραρτήματος Α.3.7. Τα ζεύγη τιμών <k, $C_k(s)$> κατά τα οποία η γραμμική πολυπλοκότητα εμφανίζεται μικρότερη του N/2 (2^{n-1}) - με έντονη γραμματοσειρά όταν ο δείκτης αποτίμησης κρυπτογραφικής αξίας παρουσιάζει τιμές μικρότερες του N (2^n) - αποτυπώνονται στον Πίνακα 5.23.

Sequences Length N=**2048** (n=11)							
Sequence	**$C_k(s)$**	**k**	**P**	**Sequence**	**$C_k(s)$**	**k**	**P**
No 1	773	552	2098	No 2	897	528	2322
	673	**558**	**1904**		803	530	2136
	338	**712**	**1388**		788	532	2108
No 3	785	534	2104		785	534	2104
	770	538	2078		781	538	2100
	769	542	2080		773	540	2086
	659	**546**	**1864**		771	546	2088
	329	**714**	**1372**		770	548	2088
No 4	769	514	2052		**705**	**550**	**1960**
	677	**544**	**1898**		**369**	**710**	**1448**
	337	**684**	**1358**	No 5	801	526	2128
No 6	781	538	2100		777	532	2086
	775	540	2090		**661**	**538**	**1860**
	774	542	2090		**330**	**656**	**1316**
	769	544	2082	No 8	785	508	2078
	675	**558**	**1908**		778	512	2068
	337	**666**	**1340**		773	514	2060
No 7	789	510	2088		772	516	2060
	785	512	2082		769	518	2056
	769	514	2052		**668**	**524**	**1860**
	705	**524**	**1934**		**333**	**670**	**1336**
	354	**642**	**1350**	No 9	897	500	2294
No 10	805	506	2116		833	502	2168
	802	508	2112		801	506	2108
	785	510	2080		785	508	2078
	779	514	2072		777	512	2066
	775	518	2068		773	516	2062
	773	520	2066		771	518	2060
	705	**524**	**1934**		**705**	**520**	**1930**
	361	**668**	**1390**		**353**	**668**	**1374**
No 11	785	510	2080	No 12	801	488	2090
	779	516	2074		787	490	2064
	777	518	2072		785	492	2062
	771	520	2062		778	494	2050
	769	524	2062		777	496	2050
	675	**530**	**1880**		**771**	**498**	**2040**
	339	**650**	**1328**		**386**	**628**	**1400**

Πίνακας 5.23 : Τιμές Ζευγών <k, $C_k(s)$> με $P = 2 \cdot C_k(s) + k < 2^n$ (n=11).

Στον πίνακα 5.24 αποτυπώνονται συνοπτικά οι αναλογίες των μεγεθών γραμμικής πολυπλοκότητας k σφαλμάτων $C_k(s)$, πλήθους αλλοιωμένων ψηφίων k και δείκτη αποτίμησης κρυπτογραφικής αξίας P, για τα πρώτα ζεύγη τιμών – ανά ακολουθία - που ο δείκτης αποτίμησης κρυπτογραφικής αξίας παρουσιάζει τιμές μικρότερες του N.

Seq	$C_0(s)$	$C_k(s)$	%	k	%	P	%
No 1	2047	673	32.87	558	27.24	1904	92.96
No 2	2046	705	34.45	550	26.85	1960	95.70
No 3	2047	659	32.19	546	26.66	1864	91.01
No 4	2046	677	33.08	544	26.56	1898	92.67
No 5	2047	661	32.29	538	26.26	1860	90.82
No 6	2047	675	32.97	558	27.24	1908	93.16
No 7	2047	705	34.44	524	25.58	1934	94.43
No 8	2046	668	32.64	524	25.58	1860	90.82
No 9	2047	705	34.44	520	25.39	1930	94.23
No 10	2047	705	34.44	524	25.58	1934	94.43
No 11	2047	675	32.97	530	25.87	1880	91.79
No 12	2047	771	37.66	498	24.31	2040	99.60
Mean		**689.9**	**33.70**	**534.5**	**26.09**	**1914.3**	**93.47**

Πίνακας 5.24 : Αναλογίες Μεγεθών για τις Πρώτες Τιμές Ζευγών <k, $C_k(s)$> όταν P < N (n=11).

Όλες οι ακολουθίες παρουσιάζουν βελτιωμένα χαρακτηριστικά σε σχέση με τις τιμές των μεγεθών από τους αντίστοιχους προηγούμενους πίνακες ακολουθιών μικρότερου μεγέθους.

5.7 Παρατηρήσεις επί των Αποτελεσμάτων

Ανωτέρω περιγράφηκαν και αναλύθηκαν τα αποτελέσματα του προγράμματος Lauder–Paterson κατά περίπτωση, δηλαδή ανά μέγεθος ακολουθίας de Bruijn. Από αυτά δύναται να εξαχθούν τα παρακάτω γενικά συμπεράσματα.

Μικρή αύξηση στο πλήθος των αλλοιωμένων ψηφίων επιφέρει μεγάλη μείωση στην τιμή της γραμμική πολυπλοκότητας των εξεταζόμενων ακολουθιών. Όσο μεγαλώνει το μέγεθος της ακολουθίας, τόσο μεγαλύτερη είναι η μείωση της γραμμικής πολυπλοκότητας, αλλοιώνοντας επιπλέον ψηφία της.

Μείωση στην τιμή γραμμικής πολυπλοκότητας των ακολουθιών de Bruijn πραγματοποιείται με αύξηση άρτιου πλήθους αλλοιωμένων ψηφίων. Οι αρχικές ακολουθίες είναι ισοβαρείς, με ισοκατανεμημένο πλήθος «0» και «1». Για τον λόγο του ότι το μέγεθός τους είναι δύναμη του 2,

το πλήθος των «1» ή των «0» στην αρχική ακολουθία είναι πάντα άρτιο. Από τον αλγόριθμο Games-Chan, έγινε εμφανές ότι για να μην υπάρξει αύξηση στην τιμή της γραμμικής πολυπλοκότητας ακολουθίας, πρέπει το αριστερό μέρος της να είναι ίδιο με το δεξί, γεγονός που σημαίνει ότι οι αλλαγές πραγματοποιούνται πάντα σε άρτιο πλήθος ψηφίων.

Για ακολουθίες de Bruijn μεγέθους μέχρι 256 bits η συμπεριφορά κρυπτογραφικής αξίας είναι ανεξάρτητη από τη μέθοδο κατασκευής της ακολουθίας. Δηλαδή, ενώ είναι αναμενόμενο ακολουθίες de Bruijn (ανεξαρτήτως μεγέθους) που κατασκευάζονται με την εκτέλεση του ίδιου αλγορίθμου (με την ίδια μέθοδο) να παρουσιάζουν παρόμοιες κρυπτογραφικές ιδιότητες, αυτό δεν έχει ισχύ εδώ.

Αντιθέτως, ακολουθίες de Bruijn μεγάλου μεγέθους συμπεριφέρονται με καλά κρυπτογραφικά χαρακτηριστικά, όσο δε αυξάνεται το μέγεθος της ακολουθίας, τόσο βελτιώνεται η κρυπτογραφική συμπεριφορά των ακολουθιών.

Γενικά οι ακολουθίες de Bruijn, αλλοιώνοντας κάποια ψηφία τους, παρουσιάζουν όχι βέλτιστα, αλλά καλά κρυπτογραφικά χαρακτηριστικά, με λίγες εξαιρέσεις που εμφανίζονται με μη ισχυρή κρυπτογραφική συμπεριφορά.

Συγκεντρωτική εικόνα ανά ακολουθία για τις αναλογίες των μεγεθών της γραμμικής πολυπλοκότητας k σφαλμάτων Ck(s), του πλήθους αλλοιωμένων ψηφίων k και του δείκτη αποτίμησης κρυπτογραφικής αξίας P, για τα πρώτα ζεύγη τιμών που ο δείκτης αποτίμησης κρυπτογραφικής αξίας παρουσιάζει τιμές μικρότερες του N αποτυπώνεται στον πίνακα 5.25. Για τις πρώτες επτά (7) ακολουθίες εφαρμόστηκε η ίδια τεχνική κατασκευής - ανά ακολουθία – ανεξαρτήτως μεγέθους. Για τις συγκεκριμένες επτά ακολουθίες ο πίνακας 5.25 παρέχει δυνατότητα σύγκρισης των ανωτέρω κρυπτογραφικών μεγεθών.

Οι ακολουθίες de Bruijn έχουν πολύ καλά κρυπτογραφικά κριτήρια, όπως η μέγιστη περίοδος ή η υψηλή τιμή γραμμικής πολυπλοκότητας, δείχνουν όμως να μην παρουσιάζουν τη βέλτιστη συμπεριφορά εάν μεταβληθούν κάποια ψηφία τους. Μέσω των αποτελεσμάτων εμφανίζεται η σπουδαιότητα του μεγέθους της γραμμικής πολυπλοκότητας k σφαλμάτων (k-error linear complexity) καθώς και του προφίλ γραμμικής πολυπλοκότητας k σφαλμάτων.

	№ 1			№ 2			№ 3		
N [bits]	**$C_k(s)$** [%]	**k** [%]	**P** [%]	**$C_k(s)$** [%]	**k** [%]	**P** [%]	**$C_k(s)$** [%]	**k** [%]	**P** [%]
32	29.03	31.25	87.50	29.03	31.25	87.50	29.03	25.00	81.25
64	29.03	31.25	87.50	27.86	31.25	84.37	28.81	25.00	78.12
128	29.13	29.68	87.50	29.13	29.68	87.50	29.92	26.56	85.93
256	35.03	24.21	93.75	38.64	23.43	99.21	38.03	20.31	96.09
512	31.95	26.56	89.84	29.27	28.12	86.32	31.70	26.56	89.84
1024	31.47	28.51	91.40	31.57	27.53	90.62	34.54	26.17	95.11
2048	32.87	27.24	92.96	34.45	26.85	95.70	32.19	26.66	91.01
	№ 4			**№ 5**			**№ 6**		
N [bits]	**$C_k(s)$** [%]	**k** [%]	**P** [%]	**$C_k(s)$** [%]	**k** [%]	**P** [%]	**$C_k(s)$** [%]	**k** [%]	**P** [%]
32	29.03	25.00	81.25	6.66	37.50	50.00			
64	33.87	21.87	87.50	40.98	18.75	96.87	9.83	28.12	46.87
128	27.77	28.12	82.81	29.60	28.12	85.93	16.80	28.12	60.93
256	30.19	24.21	84.37	22.17	20.31	63.28	28.74	17.18	72.65
512	34.70	27.73	96.87	32.54	24.21	89.06	29.01	25.39	83.20
1024	39.39	20.31	99.02	33.07	24.41	90.42	31.43	29.68	92.38
2048	33.08	26.56	92.67	32.29	26.26	90.82	32.97	27.24	93.16
	№ 7			**№ 8**			**№ 9**		
N [bits]	**$C_k(s)$** [%]	**k** [%]	**P** [%]	**$C_k(s)$** [%]	**k** [%]	**P** [%]	**$C_k(s)$** [%]	**k** [%]	**P** [%]
32	22.58	25.00	68.75						
64	9.83	28.12	46.87						
128	32.53	23.43	87.50	26.19	31.25	82.81	28.81	26.56	79.68
256	27.82	16.40	70.31	29.52	24.21	82.81	32.01	23.43	86.71
512	31.89	28.51	92.18	31.89	26.56	90.23	37.91	23.04	98.43
1024	32.77	26.95	91.99	34.80	25.78	95.11	33.00	25.78	91.60
2048	34.44	25.58	94.43	32.64	25.58	90.82	34.44	25.39	94.23
	№ 10			**№ 11**			**№ 12**		
N [bits]	**$C_k(s)$** [%]	**k** [%]	**P** [%]	**$C_k(s)$** [%]	**k** [%]	**P** [%]	**$C_k(s)$** [%]	**k** [%]	**P** [%]
32									
64									
128	32.53	25.00	89.06	26.77	28.12	81.25	34.67	21.87	89.06
256	33.33	26.56	92.96	30.31	28.12	88.28	31.76	25.78	89.06
512	38.35	21.87	98.43	32.09	27.73	91.79	31.70	26.17	89.45
1024	33.72	25.97	93.35	33.36	25.19	91.79	33.43	26.75	93.35
2048	34.44	25.58	94.43	32.97	25.87	91.79	37.66	24.31	99.60

Πίνακας 5.25 : Αναλογίες Μεγεθών ανά Ακολουθία για τα Πρώτα Κρίσιμα Σημεία όταν $P < N$.

Κεφάλαιο 6
Επίλογος

Η μεγάλη πρόκληση για τους κρυπτογραφικούς αλγορίθμους δημοσίου κλειδιού είναι το ότι οι πιο γνωστοί εξ αυτών, δεν θα είναι πλέον ασφαλείς στην εποχή των κβαντικών υπολογιστών. Από την άλλη πλευρά, οι αλγόριθμοι συμμετρικού κλειδιού, φαίνεται ότι – με κατάλληλη αύξηση του μεγέθους κλειδιού – θα παραμένουν ασφαλείς και στη μετακβαντική εποχή.

Οι δυαδικές ακολουθίες de Bruijn, παράγονται από κρυπτογραφικούς αλγόριθμους ροής και συγκεκριμένα από μη γραμμικούς καταχωρητές με ανάδραση (NLFSR). Χρησιμοποιούνται ως γεννήτριες κλειδοροής με ισχυρά κρυπτογραφικά χαρακτηριστικά. Οι ισχυρές ιδιότητες των ακολουθιών de Bruijn όμως, δύνανται να μειωθούν ραγδαία, κατόπιν αλλαγής λίγων μόνο ψηφίων τους.

Η παραγωγή των ακολουθιών - ιδιαιτέρως των ακολουθιών de Bruijn- από τους μη γραμμικούς καταχωρητές (NLFSR), αποτελεί πεδίο εξαιρετικού ενδιαφέροντος, διότι εμφανίζει σημαντικά πλεονεκτήματα στην ανάπτυξη της κρυπτογραφίας και του ασφαλούς περιβάλλοντος αποθήκευσης και διακίνησης δεδομένων. Όπως προαναφέρθηκε, οι de Bruijn ακολουθίες

αποτελούν διαρκές αντικείμενο έρευνας σε πολλούς επιστημονικούς τομείς. Παρά την σχετική βιβλιογραφία που έχει αναπτυχθεί παγκοσμίως, οι κρυπτογραφικές ιδιότητες των ακολουθιών de Bruijn δεν έχουν εξετασθεί εκτενώς. Η σημασία τους για την περιοχή της κρυπτογραφίας εμφανίζεται τεράστια, καθώς η διαφοροποίησή τους επηρεάζει άμεσα την ισχύ της.

Η έρευνα, εστιάζοντας στη διατήρηση των κρυπτογραφικών ιδιοτήτων μιας de Bruijn ακολουθίας σε περίπτωση διαφοροποίησης στοιχείων της, οδηγεί στην συνέχιση της ισχύος της κρυπτογραφικής ασφάλειας δεδομένων και ανοίγει το δρόμο για μελλοντική έρευνα και εφαρμογές σε διαφορετικά επιστημονικά πεδία.

6.1 Σύνοψη

Η παρούσα ερευνητική προσπάθεια σκοπεύει στη διερεύνηση της διαμόρφωσης της γραμμικής πολυπλοκότητας μιας ακολουθίας de Bruijn, μεταβάλλοντας εντός αυτής κάποια από τα ψηφία της. Στο πλαίσιο αυτό, έγινε χρήση του αλγορίθμου Lauder-Paterson με στόχο την εύρεση - σε συγκεκριμένες ακολουθίες de Bruijn – του προφίλ της γραμμικής πολυπλοκότητας k σφαλμάτων (k-error linear complexity spectrum) και, παράλληλα, εξετάσθηκαν τα κρυπτογραφικά κριτήρια των «τροποποιημένων» ακολουθιών de Bruijn.

6.1.1 Θεωρητική Παρουσίαση

Αρχικά, βάσει βιβλιογραφικών αναφορών, έγινε κατανοητό το θεωρητικό υπόβαθρο και το πλαίσιο ανάπτυξης της έρευνας. Αναλύθηκαν έννοιες όπως η κρυπτογραφία, οι κρυπτογραφικοί αλγόριθμοι ροής, τα κρυπτογραφικά κριτήρια που πρέπει να διέπουν τις παραγόμενες από FSR ακολουθίες. Έγινε ιδιαίτερη αναφορά σε μεγέθη κρυπτογραφικής ισχύος όπως η γραμμική πολυπλοκότητα, το προφίλ γραμμικής πολυπλοκότητας καθώς και το μέγεθος της γραμμικής πολυπλοκότητας k σφαλμάτων.

Στην συνέχεια παρουσιάστηκαν και επεξηγήθηκαν αλγόριθμοι που χρησιμοποιούνται στον υπολογισμό των προαναφερθέντων κρυπτογραφικών μεγεθών. Έμφαση δόθηκε στον αλγόριθμο Lauder-Paterson, βασικό εργαλείο στην παρούσα έρευνα, με τον οποίο εισήχθησαν οι έννοιες των κρίσιμων σημείων και του φάσματος γραμμικής πολυπλοκότητας k σφαλμάτων.

Για την κατανόηση της σημασίας της διατήρησης υψηλής τιμής γραμμικής πολυπλοκότητας, αλλά και της επίπτωσης που έχει σε αυτή η αλλοίωση ψηφίων ακολουθίας που χρησιμοποιείται ως κλειδοροή, περιγράφηκαν μοντέλα επίθεσης που επιβεβαίωσαν τα παραπάνω.

Τέλος αναφέρθηκαν οι ακολουθίες de Bruijn και οι τεχνικές παραγωγής τους, καθώς και τα κριτήρια κρυπτογραφικής ισχύος τροποποιημένης ακολουθίας de Bruijn. Οι συγκεκριμένες ακολουθίες, αποτέλεσαν το δείγμα της πειραματικής προσέγγισης της έρευνας.

6.1.2 Πειραματική Προσέγγιση

Αντικείμενο της έρευνας στη μεταπτυχιακή διατριβή ήταν η εξέταση των κρυπτογραφικών χαρακτηριστικών που παρουσιάζουν οι ακολουθίες de Bruijn. Για τον λόγο αυτό επιλέχθηκαν ακολουθίες μεγέθους από 32 μέχρι 2048 bits. Ως μέσο για τη διεξαγωγή της έρευνας έγινε χρήση π[ρογραμμάτων σε γλώσσα προγραμματισμού C++, των οποίων ο πηγαίος κώδικας αποτυπώνεται στο Παράρτημα Β.

Ορισμένες από τις συγκεκριμένες ακολουθίες de Bruijn, ανά μέγεθος, δημιουργήθηκαν με καθορισμένη τεχνική παραγωγής, ενώ οι υπόλοιπες παρήχθησαν με παράγοντα τυχαιότητας ώστε να εξετασθεί αριθμός ακολουθιών που δεν παράγονται εγγυημένα από δεδομένους αλγόριθμους τεχνικών παραγωγής.

Για τη διεξαγωγή της έρευνας ορίστηκε δείκτης κρυπτογραφικής αξίας $P=2\cdot C_k(s)+k$ και περιγράφηκε η αναγκαιότητα υπολογισμού αναλογίας του πλήθους των τροποποιημένων ψηφίων k σε σχέση με την αρχική ακολουθία de Bruijn, όταν ο ανωτέρω δείκτης λάβει τιμή ίση με το μέγεθος της ακολουθίας.

Μετρήσεις κρυπτογραφικών μεγεθών από τις ακολουθίες de Bruijn που εξετάστηκαν και εξαγωγή αποτελεσμάτων έγινε με χρήση αρχικά του αλγόριθμου Stamp-Martin και στην συνέχεια με τον αλγόριθμο Lauder-Paterson, όπου υπολογίστηκαν τα κρίσιμα σημεία (Critical Points) και ο δείκτης κρυπτογραφικής αξίας για την κάθε ακολουθία.

Όταν ο δείκτης αποτίμησης κρυπτογραφικής αξίας ακολουθίας de Bruijn λάβει τιμή μικρότερη του μήκους της, εξετάζεται το πρώτο κρίσιμο σημείο στην κατάσταση αυτή. Βάσει των εξαχθέντων αποτελεσμάτων συμπληρώνονται πίνακες από τους οποίους προκύπτουν τα συμπεράσματα.

6.2 Συμπεράσματα

Οι ακολουθίες de Bruijn έχουν πολύ καλά κρυπτογραφικά κριτήρια, όπως η μέγιστη περίοδος ή η υψηλή τιμή γραμμικής πολυπλοκότητας, όμως δείχνουν να μην εμφανίζουν τη βέλτιστη συμπεριφορά στην περίπτωση μεταβολής κάποιων ψηφίων τους. Η συμπεριφορά δεν είναι κοινή για όλες τις de Bruijn ακολουθίας, αφού κάποιες φαίνεται να εμφανίζουν σαφώς καλύτερη συμπεριφορά από άλλες. Από τα αποτελέσματα της παρούσης έρευνας γίνεται φανερή η σπουδαιότητα του μεγέθους της γραμμικής πολυπλοκότητας k σφαλμάτων (k-error linear complexity).

Αύξηση στο πλήθος των αλλοιωμένων ψηφίων επιφέρει μείωση στην τιμή της γραμμικής πολυπλοκότητας ακολουθίας de Bruijn. Τέτοια ακολουθία χαρακτηρίζεται κρυπτογραφικά ασθενής όταν ο δείκτης αποτίμησης κρυπτογραφικής αξίας λάβει τιμή μικρότερη του μήκους της. Με βάση τα πειραματικά αποτελέσματα, φαίνεται ότι ο ως άνω δείκτης αναμένεται γενικά να λαμβάνει τιμή μικρότερη του μήκους της αν τροποποιηθούν περί το ¼ των bits της ακολουθίας.

Η τεχνική παραγωγής δεν φαίνεται να καθορίζει την συμπεριφορά ακολουθίας de Bruijn ως προς τη γραμμική πολυπλοκότητα k σφαλμάτων. Για την ίδια τεχνική, ανάλογα με το μέγεθος, μπορεί να παραχθεί - κρυπτογραφικά - ισχυρή ή μη ακολουθία.

Όσο αυξάνεται το μέγεθος ακολουθίας de Bruijn, τόσο φαίνεται ότι βελτιώνεται η κρυπτογραφική συμπεριφορά των τροποποιημένων αυτών ακολουθιών.

6.3 Περιορισμοί και Συστάσεις για Μελλοντική Έρευνα

Ο πρώτος περιορισμός της μεταπτυχιακής αυτής διατριβής, λόγω οικονομίας χώρου, είναι ότι χρησιμοποιήθηκαν πεπερασμένα μεγέθη ακολουθιών de Bruijn μεγέθους από 2^5 (32 bits) έως 2^{11} (2048 bits). Θεωρητικά, δεν υπάρχει όριο μεγέθους ακολουθιών de Bruijn προς εξέταση, πρακτικά όμως τα όρια καθορίζονται από την υπολογιστική ισχύ των μέσων. Υπάρχουν πολύ μεγαλύτερες ακολουθίες de Bruijn, των οποίων τα κρυπτογραφικά κριτήρια δεν εξετάσθηκαν στην παρούσα, αποτελούν όμως ερευνητικά μελλοντικά αντικείμενα.

Ο δεύτερος περιορισμός αφορά στην επιλογή του πλήθους των εξεταζομένων ακολουθιών ανά μέγεθος. Μέγιστο πλήθος των ακολουθιών de Bruijn που εξετάσθηκαν ανά μέγεθος ήταν 12, υπό την έννοια ότι δεν αντιπροσωπεύθηκε μεγάλο δείγμα ακολουθιών, λόγω οικονομίας χώρου. Προτείνεται διεξαγωγή περαιτέρω έρευνας σε διευρυμένο πλήθος μεγεθών των ακολουθιών.

Ο τρίτος περιορισμός αφορά στο γεγονός ότι δεν επιχειρήθηκε ευρεία αναζήτηση μεθόδων κατασκευής ακολουθιών de Bruijn, λόγω του μεγάλου πλήθους αυτών. Έτσι, δεν έχουν εξεταστεί ακολουθίες που παράγονται από όλες τις τεχνικές κατασκευής ακολουθιών de Bruijn. Προτείνεται προς άρση του συγκεκριμένου περιορισμού, περαιτέρω έρευνα διευρυμένη κατά τα προαναφερθέντα κρυπτογραφικά κριτήρια και σύγκριση των αποτελεσμάτων ανάμεσα σε ακολουθίες παραγόμενες από περισσότερους αλγόριθμους τεχνικής κατασκευής τους.

Τέλος, προτείνεται η εξέταση των κρυπτογραφικών κριτηρίων των συναρτήσεων που παράγουν τις τροποποιημένες ακολουθίες de Bruijn, ώστε να διαπιστωθούν συγκεκριμένα κριτηρία που πρέπει να πληροί η αρχική de Bruijn ακολουθία έτσι ώστε, ακόμα και αν συμβούν μεταβολές σε λίγα ψηφία αυτής, να μη γίνεται απώλεια των κρυπτογραφικών της ιδιοτήτων. Επίσης, θα πρέπει να μελετηθούν συγκεκριμένες κρυπταναλυτικές τεχνικές ως προς το πώς μπορούν να ισχυροποιηθούν, αξιοποιώντας τυχόν πληροφορία αναφορικά με τη γραμμική πολυπλοκότητα k σφαλμάτων των ακολουθιών που υπεισέρχονται στη λειτουργία του κρυπτογραφικού αλγορίθμου.

Από την παρούσα έρευνα έγινε εμφανής η σπουδαιότητα του μεγέθους της γραμμικής πολυπλοκότητας k σφαλμάτων (k-error linear complexity) καθώς και η επίδραση που έχει η αλλοίωση ορισμένων ψηφίων ακολουθιών de Bruijn στην τιμή της γραμμικής πολυπλοκότητας, επηρεάζοντας την κρυπτογραφική συμπεριφορά τους. Ως σήμερα, στο συγκεκριμένο πεδίο, δεν έχει πραγματοποιηθεί έρευνα επί των ακολουθιών de Bruijn, δημιουργείται όμως η ανάγκη για περαιτέρω μελέτη και ανάλυση σχετικά με το μέγεθος της γραμμικής πολυπλοκότητας k σφαλμάτων (k-error linear complexity) σε de Bruijn ακολουθίες.

Η παρούσα έρευνα και τα αποτελέσματά της, θα είναι σε θέση να αποτελέσει αφετηρία μελλοντικής έρευνας και να εμφανίσει τη δέουσα συνεισφορά σε νέους ερευνητικούς δρόμους.

Βιβλιογραφία

[01] M. Bellare, P. Rogaway. «Introduction to Modern Cryptography». Course Notes. 2004.

[02] E. R. Berlekamp. «Algebraic coding theory». New York, McGraw-Hill. 1968

[03] G.R. Blakley, I. Borosh. «Rivest-Shamir-Adleman public key cryptosystems do not always conceal messages**».** Computers & Mathematics with Applications. Volume 5. No 3. Pages 169 – 178. 1979.

[04] H. G. Borisagar, P. R. Mishra, N. Gaba. «1-Error Linear Complexity Test for Binary Sequences». Advances in Intelligent Systems and Computing. Volume 259. Pages 225-235. 2014.

[05] A. J. Bromfield, F.C. Piper. «Linear Recursion Properties of Uncorrelated Binary Sequences». Discrete Applied Mathematics. Volume 27. Pages 187-193. 1990.

[06] A. H. Chan, R. A. Games, E. L. Key. «On the complexities of de Bruijn sequences». J. Combin. Theory, ser. A. vol. 33. pp. 233-246. 1982.

[07] Z. Chang, M. F. Ezerman, S. Ling, H. Wang. «Construction of de Bruijn sequences from product of two irreducible polynomials». Cryptography and Communications. Volume10. No 2. Pages251-275. 2018.

[08] R. Chikhi, A. Limasset, S. Jackman, J. T. Simpson, P. Medvedev. «On the representation of de Bruijn graphs».

[09] N. G. de Bruijn. «A combinatorial problem». Indagationes Mathematicae. Volume49. Pages 758-764. 1946.

[10] T. Denis. «Cryptography for Developers». Syngress Publishing, Inc. ISBN 1-59749-104-7. 2006.

[11] W. Diffie, M. Hellman. «New directions in cryptography». IEEE Transactions on Information Theory. Volume IT-22. No 6. Pages 644-654. 1976.

[12] C. Ding, G. Xiao, W. Shan. «The Stability Theory of Stream Ciphers». Lecture Notes in Computer Science. Springer-Verlag. Volume 561. 1991.

[13] L. Chen, G. Gong. «Communication Systems Security». Appendix A. Chapman and Hall/CRC Press. 2008.

[14] T. Etzion, A. Lempel. «Algorithms for the generation of full-length shift-register sequences». IEEE Transactions on Information Theory. Volume 30. No 3. Pages 480-484. 1984.

[15] T. Etzion, A. Lempel. «Construction of de Bruijn Sequences of Minimal Complexity». IEEE Transactions on Information Theory. Volume 30. No 5. Pages 705-708. 1984.

[16] H. Fredricksen. «Generation of the Ford sequence of length 2^n, n large». J. Combin. Theory, ser. A. Volume 12. Pages 153-154. 1972.

[17] H. Fredricksen. «A class of nonlinear de Bruijn cycles». J. Combin. Theory, ser. A. Volume 19, Issue 2. Pages 192-199. 1975.

[18] H. Fredricksen, I. Kessler. «Lexicographic compositions and de Bruijn sequences». J. Combin. Theory, ser. A. Volume 22, Issue 1. Pages 17-30. 1977.

[19] H. Fredricksen, J. Maiorana. «Necklaces of beads in k colors and k-ary de Bruijn sequences». Discrete Mathimatics. Volume 23. Pages 207-210. 1978.

[20] D. Gabric, J. Sawada, A. Williams, D. Wong. «A framework for constructing de Bruijn sequences via simple successor rules». Discrete Mathimatics. Volume 341. Pages 2977-2987. 2018.

[21] R. Games, A. Chan. «A fast algorithm for determining the complexity of a binary sequence with period2^n». IEEE Transactions on Information Theory. Volume 29. No 1. Pages 144-146. 1983.

[22] S. Goldwasser, S. Micali. «Probabilistic encryption». Journal of Computer and System Sciences. Volume 28. Issue 2. Pages 270-299. 1984.

[23] S. W. Golomb. «Shift-Register Sequences and Spread-Spectrum Communications». Third International Symposium on Spread Spectrum Techniques & Applications. Pages 14-15. 1994.

[24] S. W. Golomb. «Shift Register Sequences». Holden-Day, Inc., San Francisco, 1967. revised edition, Aegean Park Press, Laguna Hills, CA. ISBN:0894120484. 1981

[25] G. Gong. «Randomness and Representation of Span n Sequences». in Proceedings of the 2007 International Conference on Sequences, Subsequences, and Consequences. Spinger, Heidelberg. Pages 192-203. 2007.

[26] O. Gordreich. «Foundations of Cryptography, basic tools». Cambridge University Press. 2001.

[27] E. R. Hauge, T. Helleseth. «De Bruijn sequences, irreducible codes and cyclotomy». Discrete Mathematics. Volume 159. Pages 143-154. 1996.

[28] Y. Huang. «A new algorithm for the generation of binary de Bruijn sequences». Journal of Algorithms. Volume 11. Issue 1. Pages 44-51. 1990.

[29] A. Kerckhoffs. «cryptographie militaire». Journal des sciences militaires, Volume IX. Pages 161-191. 1883.

[30] E. Key. «An analysis of the structure and complexity of nonlinear binary sequence generators». IEEE Transactions on Information Theory. Volume 22. No 6. Pages 732-736. 2006.

[31] K. Kurosawa, F. Sato, T. Sakata, W. Kishimoto. «A relationship between linear complexity and k-error linear complexity». IEEE Transactions on Information Theory. Volume 46. Pages 694-698. 2000.

[32] A. Lauder, K. Paterson. «Computing the error linear complexity spectrum of a binary sequence of period 2^n». IEEE Transactions on Information Theory. Volume 49. Pages 273-280. 2003.

[33] K. Lek, N.Rajapakse. «: Cryptography : Protocols, Design, and Applications». Nova Publishers New York. ISBN 9781621007791. 2012.

[34] M. Liao, W. He, D. Lu, X. Peng. «Ciphertext-only attack on optical cryprosystem with spatially incoherent illumination : from the view of imaging through scattering medium». Scientific Reports. volume7. No 41789. 2017.

[35] K. Limniotis, N. Kolokotronis, D. Kotanidis. «De Bruijn Sequences and Suffix Arrays: Analysis and Constructions». Modern Discrete Mathematics and Analysis. Springer International Publishing. ISBN 978-3-319-74325-7. 2018.

[36] J.L. Massey. «Shift-register synthesis and BCH decoding». IEEE Transactions on Information Theory. Volume 15. No 1. Pages 81-92. 1969.

[37] A. J. Menezes, P. C. van Oorschot, S. A. Vanstone. «Handbook of Applied Cryptography». CRC Press. ISBN 0-8493-8523-7. 1996.

[38] C. Paar, J. Pelzl. «Understanding Cryptography». Springer Heidelberg. ISBN 978-3-642-04100-6. 2009.

[39] D. Pointcheval. «Number Theory and Public-Key Cryptography». Combinatorial and Computational Mathematics. 2000.

[40] W. Qin, X. Peng, X. Meng, B. Z. Gao. «Vulnerability to chosen-plaintext attack of optoelectronic information encryption with phase-shifting interferometry». Optical engineering (Redondo Beach, Calif.). Volume 50. No 6. 2011.

[41] T. Rachwalik, J. Szmidt, R. Wicik, J. Zabłocki. «Generation of Nonlinear Feedback Shift Registers with special-purpose hardware». Military, Communications and Information Systems Conference (MCC). 2012.

[42] A. Ralston. «A new memoryless algorithm for de Bruijn sequences». Journal of Algorithms. Volume 2. Issue 1. Pages 50-62. 1981.

[43] R.A. Rueppel. «Analysis and Design of Stream Ciphers». Springer-Verlag. ISBN 978-3-642-82865-2. 1986.

[44] J. Sawada, B. Stevens, A. Williams. «De Bruijn sequences for the binary strings with maximum density». Chapter 19 of Algorithms and Computation, ISBN 9783642190933, pages 182–190. Springer. 2011.

[45] J. Sawada, B. Stevens, D. Wong. «A surprisingly simple de Bruijn sequence construction». Discrete Mathematics. Volume 339. Pages 127-131. 2016.

[46] J. Sawada, B. Stevens, D. Wong. «Universal cycles for weight-range binary strings». In Proceedings of 24th International Workshop on Combinatorial Algorithms. Springer. Pages 388-401. 2013.

[47] C. F. Shannon. «Communication theory of secrecy systems». Bell Systems Technical Journal. Volume 28. Pages 656-715. 1949.

[48] N. Smart. «Cryptography: An Introduction». McGraw-Hill College. ISBN 978-0077099879. 2004.

[49] H. Y. Song. «Feedback Shift Register Sequences». Wiley Series in Telecommunications and Signal Processing. John Wiley & Sons, Inc. 2003.

[50] M. Stamp, C.F. Martin. «An algorithm for the k-error linear complexity of binary sequences with period 2^n». IEEE Transactions on Information Theory. Volume 39. No 4. Pages 1398-1401. 1993.

[51] G. S. Vernam. «Cipher Printing Telegraph Systems for Secret Wire and Radio Telegraphic Communications». Journal of the American Institute of Electrical Engineers. Volume 45. Pages 109-115. 1926.

[52] D. Wong. «New successor rules for constructing de Bruijn sequences». Northwest Missouri State University. 2017.

[53] Π. Γροντάς. «Μοντέλα και Αποδείξεις Ασφάλειας στην Κρυπτογραφία». Διαφάνειες μαθήματος ΕΜΠ-Κρυπτογραφία. 2015.

[54] Ε. Ζάχος, Α. Παγουρτζής, Π. Γροντάς. «Αλγόριθμοι στην Κρυπτογραφία». Κεφ. Συγγράμματος *Υπολογιστική Κρυπτογραφία*. ΣΥΝΔΕΣΜΟΣ ΕΛΛΗΝΙΚΩΝ ΑΚΑΔΗΜΑΪΚΩΝ ΒΙΒΛΙΟΘΗΚΩΝ. ISBN 978-960-603-276-9. 2015.

[55] Β. Κάτος, Γ. Στεφανίδης. «Τεχνικές Κρυπτογραφίας και Κρυπτανάλυσης». ΖΥΓΟΣ, ISBN 960-8065-40-2. 2003.

[56] Α. Παγουρτζής, Ε. Ζάχος. «Συμμετρικά κρυπτοσυστήματα». Κεφ. Συγγράμματος *Υπολογιστική Κρυπτογραφία*. ΣΥΝΔΕΣΜΟΣ ΕΛΛΗΝΙΚΩΝ ΑΚΑΔΗΜΑΪΚΩΝ ΒΙΒΛΙΟΘΗΚΩΝ. ISBN 978-960-603-276-9. 2015.

[57] Θ. Τσιάκης. «Η εφαρμοσμένη κρυπτογραφία ως τυπική μέθοδος και μοντέλο για την ασφάλεια των ηλεκτρονικών συναλλαγών». Διδακτορική Διατριβή στο Πανεπιστήμιο Μακεδονίας Οικονομικών και Κοινωνικών Επιστημών. Τμήμα Εφαρμοσμένης Πληροφορικής. 2005.

[58] M. Klein. «Securing Record Communications: The TSEC/KW-26».

[59] NIST SP 800-22 . «A Statistical Test Suite for the Validation of Random Number Generators and Pseudo Random Number Generators for Cryptographic Applications».

[60] NIST SP 800-90A. «Recommendation for Random Number Generation Using Deterministic Random Bit Generators».

[61] M. Green. «A few Thoughts on Cryptographic Engineering». 2018. https://blog.cryptographyengineering.com/2018/04/21/wonk-post-chosen-ciphertext-security-in-public-key-encryption-part-1/.

[62] http://www.circuitsgallery.com/

[63] http://www.crypto-it.net/eng/theory/kerckhoffs.html

[64] http://www.tech-faq.com/known-plaintext-attack.html

[65] https://el.wikipedia.org/wiki/Ασφάλεια_πληροφοριακών_συστημάτων

[66] https://el.wikipedia.org/wiki/Κρυπτογραφία

[67] https://en.wikipedia.org/wiki/Blaise_de_Vigenère

[68] https://en.wikipedia.org/wiki/De_Bruijn_sequence

[69] https://en.wikipedia.org/wiki/Gilbert_Vernam

[70] https://en.wikipedia.org/wiki/Kerckhoffs%27s_principle

[71] https://www.cs.mcgill.ca/~rwest/wikispeedia/wpcd/wp/c/Caesar_cipher.htm

Παράρτημα Α
Αποτελέσματα Εκτέλεσης Προγραμμάτων

Στο παράρτημα Α αποτυπώνονται τα αποτελέσματα εκτέλεσης των απαιτούμενων για την έρευνα προγραμμάτων.

A.1 Δημιουργία Ακολουθιών de Bruijn

Τα αναφερθέντα στην παράγραφο 5.1 αρχεία κειμένου (.txt) περιέχουν μία περίοδο από τις ακολουθίες de Bruijn, που θα αποτελέσουν τα δοκίμια της έρευνας για την παρούσα μελέτη διατριβής. Ακολούθως οι ακολουθίες παρουσιάζονται σε τμήματα των 4άρων ψηφίων ώστε να είναι ευανάγνωστες. Ακόμα παρουσιάζονται οι ακολουθίες μέχρι n=8 για οικονομία χώρου.

1. Για **n=5**, δηλαδή ακολουθίες παραγόμενες από NLFSR πέντε (5) βαθμίδων και περιόδου 32 bits, στο αρχείο De_Bruijn_5.txt υπάρχουν οι ακολουθίες :

0000 0100 0110 0101 0011 1010 1101 1111
0000 0100 1010 0011 0101 1001 1101 1111
0000 0111 1101 1100 1100 0101 1010 1001
0000 0111 1101 1010 1001 0111 0011 0001
0000 0111 1100 1001 1011 0001 0111 0101
0000 0100 1101 1001 0101 1101 0001 1111

2. Για **n=6**, δηλαδή ακολουθίες παραγόμενες από NLFSR έξι (6) βαθμίδων και περιόδου 64 bits, στο αρχείο De_Bruijn_6.txt υπάρχουν οι ακολουθίες :

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

3. Για **n=7**, δηλαδή ακολουθίες παραγόμενες από NLFSR επτά (7) βαθμίδων και περιόδου 128 bits, στο αρχείο De_Bruijn_7.txt υπάρχουν οι ακολουθίες :

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

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

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

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

0000 0001 1111 1100 0110 0111 0011 0001 0101 1101 0101 0001 0001 1101 1100 0011
0111 1001 0100 1101 0110 0100 1011 0110 1001 0000 1001 1110 1100 0001 0111 1101

0000 0001 1111 1100 0100 0111 0111 0000 1001 1110 1100 1010 0110 1011 0001 1001
1100 1100 0001 0111 110 1001 0010 1101 1010 0010 1011 1010 1010 0001 1011 11001

0000 0001 0001 1101 1100 0100 1011 0110 1001 0011 1101 1000 0101 0011 0101 1001
0101 0111 0101 0001 0111 1101 0000 0110 0111 0011 0001 1011 1100 1000 0111 1111

4. Για **n=8**, δηλαδή ακολουθίες παραγόμενες από NLFSR οκτώ (8) βαθμίδων και περιόδου 256 bits, στο αρχείο De_Bruijn_8.txt υπάρχουν οι ακολουθίες :

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

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

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

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

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

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

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

A.2 Πρόγραμμα Stamp-Martin

Ακολούθως παρατίθενται τα αποτελέσματα του προγράμματος το οποίο υπολογίζει την τιμή της γραμμικής πολυπλοκότητας ck(s) όταν αυξάνει η τιμή των αλλοιωμένων ψηφίων k, χρησιμοποιώντας τον αλγόριθμο Stamp-Martin.

A.2.1 Ακολουθίες de Bruijn μεγέθους 32 bits

Για τις ακολουθίες προς δοκιμή που βρίσκονται στο αρχείο De_Bruijn_5.txt και είναι μεγέθους 32 bits το πρόγραμμα Stamp-Martin υπολόγισε τα ακόλουθα αποτελέσματα. Τα συγκεκριμένα αποτελέσματα βρίσκονται αποθηκευμένα στο αρχείο **StampMartin_5.csv** .

de Bruijn Sequence Size 32						
	No 1	No 2	No 3	No 4	No 5	No 6
k	Ck(s)	Ck(s)	Ck(s)	Ck(s)	Ck(s)	Ck(s)
0	31	31	31	31	30	31
1	31	31	31	31	30	31
2	22	22	25	25	21	25
3	22	22	25	25	21	25
4	18	18	25	25	21	25
5	18	18	25	25	21	25
6	17	17	18	18	17	18
7	17	17	18	18	17	18
8	17	17	9	9	17	7
9	17	17	9	9	17	7
10	9	9	6	6	11	2
11	9	9	6	6	11	2
12	9	9	3	3	2	2
13	9	9	3	3	2	2
14	2	2	2	2	2	2
15	2	2	2	2	2	2
16	0	0	0	0	0	0

Πίνακας Α.1 : Αποτελέσματα Προγράμματος Stamp-Martin για Ακολουθίες de Bruijn Μεγέθους 32 bits.

A.2.2 Ακολουθίες de Bruijn μεγέθους 64 bits

Για τις ακολουθίες προς δοκιμή που βρίσκονται στο αρχείο De_Bruijn_6.txt και είναι μεγέθους 64 bits το πρόγραμμα Stamp-Martin υπολόγισε τα ακόλουθα αποτελέσματα. Τα συγκεκριμένα αποτελέσματα βρίσκονται αποθηκευμένα στο αρχείο **StampMartin_6.csv** .

De Bruijn Sequence Size 64							
	No 1	No 2	No 3	No 4	No 5	No 6	No 7
k	Ck(s)	Ck(s)	Ck(s)	Ck(s)	Ck(s)	Ck(s)	Ck(s)
0	62	61	59	62	61	61	61
1	62	61	59	62	61	61	61
2	59	61	59	59	61	61	61
3	59	61	59	59	61	61	61
4	45	51	53	53	50	50	50
5	45	51	53	53	50	50	50
6	41	51	53	49	50	50	50
7	41	51	53	49	50	50	50
8	41	42	41	49	34	43	43
9	41	42	41	49	34	43	43
10	37	36	41	41	34	34	34
11	37	36	41	41	34	34	34
12	33	33	34	35	25	34	34
13	33	33	34	35	25	34	34
14	33	33	34	21	25	33	33
15	33	33	34	21	25	33	33
16	33	33	17	17	6	33	33
17	33	33	17	17	6	33	33
18	33	33	17	17	6	6	6
19	33	33	17	17	6	6	6
20	18	17	17	11	3	3	3
21	18	17	17	11	3	3	3
22	18	17	17	11	3	3	3
23	18	17	17	11	3	3	3
24	9	10	9	2	3	3	3
25	9	10	9	2	3	3	3
26	3	5	9	2	3	3	3
27	3	5	9	2	3	3	3
28	3	3	2	2	3	3	3
29	3	3	2	2	3	3	3
30	3	3	2	2	3	3	3
31	3	3	2	2	3	3	3
32	0	0	0	0	0	0	0

Πίνακας Α.2 : Αποτελέσματα Προγράμματος Stamp-Martin για Ακολουθίες de Bruijn Μεγέθους 64 bits.

A.3 Πρόγραμμα Lauder-Paterson

Ακολούθως παρατίθενται τα αποτελέσματα του προγράμματος το οποίο υπολογίζει τις τιμές των κρίσιμων σημείων για ακολουθίες de Bruijn ως ζεύγη τιμών <k, $C_k(s)$>, με χρήση του αλγόριθμου Lauder-Paterson.

A.3.1 Ακολουθίες de Bruijn μεγέθους 32 bits

Για τις ακολουθίες προς δοκιμή που βρίσκονται στο αρχείο De_Bruijn_5.txt, μεγέθους 32 bits, το πρόγραμμα Lauder-Paterson υπολόγισε τα ακόλουθα αποτελέσματα που βρίσκονται αποθηκευμένα στο αρχείο **LauderPaterson _5.csv** .

No 1	No 2	No 3	No 4	No 5	No 6
<0-31>	<0-31>	<0-31>	<0-31>	<0-30>	<0-31>
<2-22>	<2-22>	<2-25>	<2-25>	<2-21>	<2-25>
<4-18>	<4-18>	<6-18>	<6-18>	<6-17>	<6-18>
<6-17>	<6-17>	<8-9>	<8-9>	<10-11>	<8-7>
<10-9>	<10-9>	<10-6>	<10-6>	<12-2>	<10-2>
<14-2>	<14-2>	<12-3>	<12-3>	<16-0>	<16-0>
<16-0>	<16-0>	<14-2>	<14-2>		
		<16-0>	<16-0>		

Πίνακας A.3 : Αποτελέσματα Προγράμματος Lauder-Paterson για Ακολουθίες de Bruijn Μεγέθους 32 bits.

A.3.2 Ακολουθίες de Bruijn μεγέθους 64 bits

Για τις ακολουθίες προς δοκιμή που βρίσκονται στο αρχείο De_Bruijn_6.txt, μεγέθους 64 bits, το πρόγραμμα Lauder-Paterson υπολόγισε τα ακόλουθα αποτελέσματα που βρίσκονται αποθηκευμένα στο αρχείο **LauderPaterson _6.csv** .

No 1	No 2	No 3	No 4	No 5	No 6	No 7
<0-62>	<0-61>	<0-59>	<0-62>	<0-61>	<0-61>	<0-61>
<2-59>	<4-51>	<4-53>	<2-59>	<4-50>	<4-50>	<4-50>
<4-45>	<8-42>	<8-41>	<4-53>	<8-34>	<8-43>	<8-43>
<6-41>	<10-36>	<12-34>	<6-49>	<12-25>	<10-34>	<10-34>
<10-37>	<12-33>	<16-17>	<10-41>	<16-6>	<14-33>	<14-33>
<12-33>	<20-17>	<24-9>	<12-35>	<20-3>	<18-6>	<18-6>
<20-18>	<24-10>	<28-2>	<14-21>	<32-0>	<20-3>	<20-3>
<24-9>	<26-5>	<32-0>	<16-17>		<32-0>	<32-0>
<26-3>	<28-3>		<20-11>			

No 1	No 2	No 3	No 4	No 5	No 6	No 7
<32-0>	<32-0>		<24-2>			
			<32-0>			

Πίνακας Α.4 : Αποτελέσματα Προγράμματος Lauder-Paterson για Ακολουθίες de Bruijn Μεγέθους 64 bits.

Α.3.3 Ακολουθίες de Bruijn μεγέθους 128 bits

Για τις ακολουθίες προς δοκιμή που βρίσκονται στο αρχείο De_Bruijn_7.txt, μεγέθους 128 bits, το πρόγραμμα Lauder-Paterson υπολόγισε τα ακόλουθα αποτελέσματα που βρίσκονται αποθηκευμένα στο αρχείο **LauderPaterson _7.csv** .

No 1	No 2	No 3	No 4	No 5	No 6
<0-127>	<0-127>	<0-127>	<0-126>	<0-125>	<0-125>
<2-117>	<2-121>	<2-121>	<2-121>	<4-115>	<4-116>
<6-113>	<6-106>	<6-105>	<6-107>	<8-106>	<6-113>
<10-105>	<8-103>	<10-101>	<8-103>	<10-97>	<10-105>
<14-98>	<10-101>	<14-97>	<10-101>	<22-73>	<12-98>
<16-81>	<12-99>	<18-84>	<12-99>	<26-68>	<16-83>
<22-76>	<14-98>	<20-81>	<14-97>	<28-65>	<18-81>
<24-70>	<16-89>	<24-73>	<18-89>	<36-37>	<20-75>
<26-65>	<18-72>	<26-70>	<20-82>	<38-33>	<22-74>
<38-37>	<20-68>	<28-67>	<22-72>	<40-25>	<24-70>
<40-34>	<24-66>	<30-65>	<24-66>	<42-21>	<26-68>
<42-33>	<26-65>	<34-38>	<32-49>	<46-19>	<28-65>
<44-18>	<38-37>	<36-34>	<36-35>	<48-2>	<36-21>
<46-17>	<40-33>	<38-33>	<38-33>	<64-0>	<40-19>
<50-12>	<46-21>	<42-22>	<40-20>		<42-13>
<52-7>	<48-19>	<44-19>	<42-18>		<46-11>
<54-6>	<50-17>	<46-18>	<44-17>		<48-10>
<56-3>	<54-9>	<48-17>	<48-9>		<50-7>
<58-2>	<58-6>	<50-9>	<58-7>		<52-2>
<64-0>	<60-3>	<56-6>	<60-2>		<64-0>
	<62-2>	<58-5>	<64-0>		
	<64-0>	<60-3>			
		<62-2>			
		<64-0>			

Πίνακας Α.5.1 : Αποτελέσματα Προγράμματος Lauder-Paterson για τις Πρώτες 6 Ακολουθίες de Bruijn Μεγέθους 128 bits.

Ακολουθούν αποτελέσματα για τις επόμενες έξι (6) ακολουθίες de Bruijn :

No 7	No 8	No 9	No 10	No 11	No 12
<0-126>	<0-126>	<0-118>	<0-126>	<0-127>	<0-124>
<2-123>	<2-113>	<4-115>	<2-123>	<2-122>	<2-119>

No 7	No 8	No 9	No 10	No 11	No 12
<4-115>	<14-99>	<8-105>	<4-115>	<4-115>	<4-117>
<6-109>	<16-79>	<16-82>	<6-106>	<6-114>	<6-109>
<8-101>	<18-70>	<20-75>	<8-105>	<8-109>	<8-106>
<12-98>	<22-69>	<22-73>	<10-102>	<10-98>	<10-98>
<16-81>	<24-68>	<26-67>	<12-99>	<16-85>	<14-91>
<26-73>	<26-65>	<30-65>	<14-97>	<18-82>	<16-72>
<28-69>	<38-49>	<34-34>	<18-85>	<20-77>	<18-71>
<30-41>	<40-33>	<40-33>	<20-81>	<22-73>	<20-67>
<36-35>	<46-11>	<42-25>	<24-73>	<26-69>	<24-66>
<38-21>	<54-3>	<44-22>	<26-69>	<28-65>	<28-43>
<40-19>	<56-2>	<46-19>	<28-67>	<36-34>	<30-34>
<44-14>	<64-0>	<48-17>	<30-49>	<40-21>	<34-33>
<46-7>		<50-11>	<32-41>	<42-18>	<44-18>
<48-2>		<52-6>	<34-36>	<46-17>	<48-9>
<64-0>		<56-3>	<36-26>	<48-13>	<60-3>
		<60-2>	<38-21>	<50-11>	<64-0>
		<64-0>	<42-18>	<52-5>	
			<44-17>	<56-3>	
			<46-10>	<62-2>	
			<48-9>	<64-0>	
			<50-5>		
			<62-3>		
			<64-0>		

Πίνακας Α.5.2 : Αποτελέσματα Προγράμματος Lauder-Paterson για τις Επόμενες 6 Ακολουθίες de Bruijn Μεγέθους 128 bits.

Α.3.4 Ακολουθίες de Bruijn μεγέθους 256 bits

Για τις ακολουθίες προς δοκιμή που βρίσκονται στο αρχείο De_Bruijn_8.txt, μεγέθους 256 bits, το πρόγραμμα Lauder-Paterson υπολόγισε τα ακόλουθα αποτελέσματα που βρίσκονται αποθηκευμένα στο αρχείο **LauderPaterson _8.csv** .

No 1	No 2	No 3	No 4	No 5	No 6
<0-254>	<0-251>	<0-255>	<0-255>	<0-248>	<0-247>
<2-251>	<4-241>	<2-246>	<2-250>	<2-239>	<4-230>
<4-243>	<12-230>	<4-241>	<4-242>	<4-228>	<8-199>
<6-234>	<14-210>	<12-226>	<6-241>	<8-214>	<12-198>
<8-233>	<16-209>	<14-211>	<10-233>	<12-213>	<16-165>
<10-229>	<18-206>	<16-209>	<12-227>	<14-204>	<24-135>
<14-213>	<20-202>	<18-205>	<14-213>	<16-194>	<28-134>
<16-211>	<22-199>	<22-202>	<16-203>	<32-164>	<32-132>
<18-203>	<24-198>	<24-194>	<20-201>	<34-143>	<44-71>

No 1	No 2	No 3	No 4	No 5	No 6
<20-198>	<26-194>	<30-193>	<22-199>	<36-130>	<48-69>
<22-196>	<30-193>	<34-162>	<24-195>	<52-55>	<52-37>
<24-195>	<34-165>	<36-146>	<26-194>	<54-9>	<60-23>
<26-193>	<38-162>	<38-140>	<32-178>	<128-0>	<64-9>
<38-177>	<42-147>	<40-135>	<34-167>		<128-0>
<40-163>	<44-142>	<48-113>	<36-157>		
<42-162>	<46-139>	<50-105>	<38-147>		
<44-149>	<48-137>	<52-97>	<40-146>		
<46-147>	<54-131>	<56-81>	<42-134>		
<48-145>	<58-99>	<58-77>	<50-131>		
<50-134>	<60-97>	<60-71>	<58-130>		
<52-133>	<62-75>	<62-68>	<60-101>		
<54-131>	<64-69>	<64-66>	<62-77>		
<60-98>	<72-67>	<68-65>	<64-69>		
<62-89>	<74-66>	<74-53>	<68-67>		
<64-81>	<76-39>	<76-50>	<72-43>		
<66-69>	<80-37>	<78-43>	<74-38>		
<68-68>	<82-20>	<80-38>	<76-37>		
<70-67>	<86-7>	<82-29>	<80-20>		
<72-66>	<100-3>	<84-7>	<82-15>		
<74-46>	<104-2>	<100-6>	<84-13>		
<76-42>	<128-0>	<102-4>	<90-5>		
<78-36>		<104-3>	<102-4>		
<80-20>		<106-2>	<106-2>		
<84-12>		<128-0>	<128-0>		
<88-11>					
<92-8>					
<94-4>					
<98-3>					
<104-2>					
<128-0>					

Πίνακας Α.6.1 : Αποτελέσματα Προγράμματος Lauder-Paterson για τις Πρώτες 6 Ακολουθίες de Bruijn Μεγέθους 256 bits.

Ακολουθούν αποτελέσματα για τις επόμενες έξι (6) ακολουθίες de Bruijn :

No 7	No 8	No 9	No 10	No 11	No 12
<0-248>	<0-254>	<0-253>	<0-255>	<0-254>	<0-255>
<2-245>	<2-251>	<4-244>	<2-249>	<2-251>	<2-245>
<6-228>	<4-245>	<6-241>	<6-242>	<4-241>	<6-242>
<10-199>	<6-234>	<10-233>	<8-233>	<12-227>	<8-235>
<14-198>	<8-233>	<12-228>	<10-230>	<14-225>	<10-227>
<18-165>	<10-229>	<14-210>	<12-227>	<18-209>	<12-226>
<26-149>	<12-226>	<16-209>	<14-226>	<20-197>	<14-225>
<30-134>	<16-217>	<20-203>	<16-211>	<30-171>	<18-217>
<34-133>	<18-211>	<22-202>	<18-209>	<32-169>	<20-209>

No 7	No 8	No 9	No 10	No 11	No 12
<38-132>	<20-203>	<24-198>	<22-203>	<34-162>	<22-199>
<42-69>	<22-199>	<26-193>	<24-196>	<36-161>	<24-197>
<54-23>	<24-197>	<38-169>	<26-173>	<40-147>	<26-195>
<58-21>	<28-195>	<40-165>	<28-166>	<44-138>	<28-193>
<62-9>	<30-193>	<44-137>	<30-165>	<48-137>	<36-169>
<128-0>	<34-164>	<56-134>	<32-161>	<54-135>	<38-163>
	<36-155>	<58-132>	<42-148>	<56-133>	<40-147>
	<38-153>	<60-81>	<44-146>	<58-129>	<44-145>
	<40-145>	<64-72>	<48-139>	<70-97>	<48-141>
	<50-135>	<66-66>	<50-137>	<72-77>	<50-138>
	<52-131>	<70-53>	<54-134>	<74-70>	<52-132>
	<58-105>	<72-41>	<56-133>	<76-65>	<54-131>
	<60-101>	<80-37>	<58-130>	<92-41>	<58-130>
	<62-75>	<84-36>	<64-97>	<96-34>	<64-97>
	<64-73>	<86-33>	<68-85>	<98-25>	<66-81>
	<66-71>	<94-25>	<70-82>	<100-23>	<68-77>
	<68-66>	<96-24>	<72-73>	<102-18>	<70-75>
	<72-52>	<98-21>	<76-70>	<106-17>	<72-68>
	<74-51>	<102-6>	<78-65>	<108-13>	<74-65>
	<76-37>	<106-5>	<88-41>	<112-9>	<82-51>
	<82-34>	<114-4>	<90-37>	<114-7>	<84-39>
	<86-33>	<116-3>	<94-23>	<116-5>	<86-36>
	<92-27>	<124-2>	<96-21>	<120-2>	<88-34>
	<94-25>	<128-0>	<98-17>	<128-0>	<92-27>
	<98-20>		<108-13>		<94-25>
	<100-18>		<112-9>		<98-17>
	<104-10>		<116-5>		<108-15>
	<106-9>		<124-3>		<110-9>
	<108-8>		<126-2>		<118-2>
	<110-7>		<128-0>		<128-0>
	<112-5>				
	<118-3>				
	<120-2>				
	<128-0>				

Πίνακας Α.6.2 : Αποτελέσματα Προγράμματος Lauder-Paterson για τις Επόμενες 6 Ακολουθίες de Bruijn Μεγέθους 256 bits.

Α.3.5 Ακολουθίες de Bruijn μεγέθους 512 bits

Για τις ακολουθίες προς δοκιμή που βρίσκονται στο αρχείο De_Bruijn_9.txt, μεγέθους 512 bits το πρόγραμμα Lauder-Paterson υπολόγισε τα ακόλουθα αποτελέσματα που βρίσκονται αποθηκευμένα στο αρχείο **LauderPaterson _9.csv** .

No 1	No 2	No 3	No 4	No 5	No 6
<0-507>	<0-509>	<0-511>	<0-510>	<0-510>	<0-510>
<4-501>	<4-500>	<2-506>	<2-507>	<2-507>	<2-507>
<8-489>	<6-497>	<4-498>	<4-501>	<4-501>	<4-501>
<12-486>	<10-489>	<6-497>	<6-499>	<6-499>	<6-497>
<14-469>	<12-484>	<10-489>	<8-490>	<8-490>	<10-484>
<16-467>	<14-481>	<12-483>	<10-485>	<10-486>	<12-473>
<18-465>	<18-469>	<14-466>	<12-483>	<12-483>	<14-469>
<20-458>	<20-459>	<16-465>	<14-481>	<14-481>	<16-461>
<22-454>	<22-457>	<20-463>	<18-469>	<18-473>	<18-459>
<24-451>	<24-453>	<22-457>	<20-465>	<20-465>	<20-454>
<32-425>	<28-450>	<26-453>	<22-458>	<22-458>	<22-453>
<36-419>	<32-421>	<28-450>	<24-451>	<24-455>	<24-435>
<38-417>	<36-417>	<30-449>	<30-449>	<26-451>	<26-423>
<44-405>	<42-403>	<34-425>	<34-425>	<28-425>	<28-417>
<46-393>	<46-401>	<36-422>	<36-409>	<30-421>	<48-402>
<54-390>	<48-396>	<38-419>	<40-404>	<36-417>	<50-395>
<56-388>	<50-393>	<40-409>	<42-401>	<38-409>	<52-391>
<60-385>	<56-385>	<42-405>	<46-396>	<42-405>	<54-385>
<68-353>	<72-331>	<44-403>	<48-394>	<44-402>	<74-338>
<70-337>	<74-327>	<46-397>	<50-387>	<46-395>	<76-325>
<74-325>	<76-322>	<48-394>	<60-386>	<48-393>	<84-293>
<76-322>	<84-297>	<50-393>	<64-353>	<52-390>	<88-291>
<80-306>	<90-281>	<52-389>	<72-326>	<56-388>	<90-281>
<82-298>	<96-277>	<58-388>	<74-322>	<58-387>	<96-273>
<84-297>	<100-275>	<60-386>	<76-321>	<60-385>	<106-269>
<86-293>	<102-269>	<62-385>	<82-297>	<68-353>	<108-265>
<90-291>	<106-264>	<66-353>	<88-290>	<70-341>	<116-263>
<92-289>	<108-262>	<72-333>	<90-282>	<72-339>	<118-260>
<94-283>	<110-261>	<74-329>	<94-279>	<74-329>	<120-259>
<96-276>	<114-257>	<76-321>	<96-277>	<76-327>	<122-195>
<98-273>	<142-195>	<84-307>	<98-275>	<78-321>	<126-193>
<110-266>	<144-149>	<86-297>	<100-268>	<90-297>	<130-148>
<112-265>	<146-145>	<90-295>	<102-267>	<94-293>	<132-145>
<114-264>	<148-138>	<92-282>	<104-265>	<96-290>	<136-140>
<116-262>	<150-135>	<94-275>	<110-262>	<98-279>	<138-138>
<118-261>	<154-133>	<100-273>	<116-259>	<100-273>	<140-137>
<122-257>	<156-101>	<104-269>	<118-257>	<114-267>	<142-135>
<134-201>	<158-98>	<108-266>	<138-193>	<116-263>	<144-131>
<136-162>	<160-89>	<110-265>	<142-177>	<118-261>	<146-106>
<138-149>	<162-87>	<112-262>	<144-150>	<120-201>	<148-99>
<142-145>	<164-75>	<116-261>	<146-145>	<122-195>	<150-98>
<144-143>	<170-71>	<118-260>	<150-139>	<124-166>	<152-90>
<146-138>	<172-61>	<120-259>	<152-129>	<126-148>	<154-86>
<148-135>	<174-52>	<124-257>	<178-78>	<128-145>	<156-83>
<150-134>	<176-40>	<132-195>	<180-75>	<132-142>	<160-81>

No 1	No 2	No 3	No 4	No 5	No 6
<152-131>	<178-39>	<134-193>	<182-74>	<134-141>	<162-77>
<158-130>	<180-33>	<136-162>	<184-70>	<136-139>	<168-65>
<160-97>	<224-19>	<138-149>	<186-69>	<138-134>	<196-51>
<172-82>	<228-17>	<140-148>	<188-67>	<140-133>	<198-27>
<174-81>	<234-14>	<142-145>	<194-65>	<142-131>	<200-21>
<178-70>	<236-11>	<146-138>	<198-42>	<150-101>	<206-19>
<180-67>	<240-9>	<148-137>	<202-38>	<152-97>	<208-18>
<186-52>	<242-5>	<150-134>	<204-37>	<162-77>	<210-15>
<188-49>	<252-2>	<152-132>	<208-33>	<164-75>	<214-13>
<192-36>	<256-0>	<154-100>	<218-27>	<166-73>	<216-2>
<194-33>		<156-98>	<220-25>	<170-70>	<256-0>
<208-27>		<158-97>	<222-21>	<172-68>	
<210-25>		<162-89>	<224-19>	<174-65>	
<212-21>		<164-83>	<230-15>	<186-51>	
<218-17>		<166-74>	<232-11>	<188-49>	
<226-13>		<168-73>	<236-9>	<192-45>	
<230-11>		<170-71>	<238-7>	<194-42>	
<234-9>		<172-68>	<240-5>	<196-29>	
<238-7>		<174-66>	<248-4>	<198-25>	
<242-6>		<178-55>	<250-3>	<204-23>	
<244-2>		<180-51>	<252-2>	<206-21>	
<256-0>		<184-49>	<256-0>	<208-17>	
		<186-46>		<214-16>	
		<188-43>		<216-2>	
		<190-33>		<256-0>	
		<220-20>			
		<222-18>			
		<224-15>			
		<226-13>			
		<228-10>			
		<232-9>			
		<234-8>			
		<236-6>			
		<238-5>			
		<244-3>			
		<250-2>			
		<256-0>			

Πίνακας Α.7.1 : Αποτελέσματα Προγράμματος Lauder-Paterson για τις Πρώτες 6 Ακολουθίες de Bruijn Μεγέθους 512 bits.

Ακολουθούν αποτελέσματα για τις επόμενες έξι (6) ακολουθίες de Bruijn :

No 7	No 8	No 9	No 10	No 11	No 12
<0-511>	<0-511>	<0-509>	<0-511>	<0-511>	<0-511>
<2-506>	<2-500>	<4-499>	<2-506>	<2-505>	<2-506>
<4-498>	<4-497>	<8-489>	<4-501>	<6-497>	<4-501>

No 7	No 8	No 9	No 10	No 11	No 12
<6-493>	<12-489>	<12-484>	<6-489>	<10-482>	<6-498>
<8-489>	<14-481>	<14-469>	<12-483>	<16-463>	<8-484>
<10-485>	<18-465>	<18-465>	<14-468>	<18-457>	<10-483>
<12-483>	<20-461>	<20-459>	<16-466>	<24-454>	<12-481>
<14-481>	<24-457>	<22-457>	<18-457>	<26-450>	<20-469>
<18-469>	<26-452>	<24-453>	<24-455>	<32-435>	<22-453>
<20-465>	<28-450>	<26-452>	<26-452>	<34-423>	<26-450>
<24-457>	<30-425>	<28-449>	<28-450>	<36-417>	<32-427>
<28-453>	<32-422>	<36-433>	<30-449>	<44-403>	<34-425>
<30-449>	<34-418>	<38-421>	<34-433>	<46-401>	<36-417>
<34-425>	<36-409>	<40-419>	<36-422>	<50-398>	<40-410>
<36-419>	<40-403>	<42-405>	<38-409>	<52-393>	<42-406>
<38-406>	<44-399>	<44-402>	<40-406>	<56-391>	<44-402>
<40-403>	<46-397>	<46-395>	<42-403>	<58-387>	<48-391>
<44-402>	<48-386>	<50-393>	<44-402>	<60-339>	<52-387>
<46-399>	<62-385>	<54-389>	<46-395>	<62-332>	<56-385>
<48-393>	<66-355>	<58-387>	<48-394>	<64-329>	<72-333>
<56-389>	<68-341>	<62-338>	<50-388>	<68-327>	<74-331>
<58-342>	<70-331>	<64-337>	<52-386>	<70-325>	<76-325>
<60-333>	<74-326>	<66-329>	<62-357>	<72-322>	<78-323>
<62-331>	<76-325>	<70-326>	<64-353>	<76-309>	<80-322>
<64-326>	<78-323>	<72-321>	<66-337>	<78-306>	<84-301>
<68-324>	<80-322>	<82-306>	<70-327>	<80-301>	<86-293>
<70-310>	<82-298>	<84-302>	<72-324>	<82-294>	<90-289>
<72-307>	<84-294>	<86-297>	<74-301>	<84-293>	<96-279>
<74-301>	<86-291>	<92-290>	<76-297>	<86-292>	<98-277>
<76-297>	<92-283>	<96-275>	<80-285>	<88-289>	<100-276>
<80-293>	<94-278>	<100-273>	<82-283>	<94-275>	<102-271>
<84-291>	<96-273>	<104-267>	<84-277>	<100-270>	<104-267>
<86-283>	<108-270>	<108-266>	<92-275>	<102-269>	<106-266>
<88-281>	<110-267>	<110-265>	<94-274>	<104-264>	<108-264>
<90-277>	<114-261>	<112-263>	<96-269>	<106-262>	<110-263>
<92-271>	<120-259>	<114-260>	<98-268>	<110-260>	<112-259>
<94-265>	<122-258>	<116-225>	<100-263>	<116-257>	<124-258>
<114-260>	<128-201>	<118-193>	<102-262>	<140-197>	<126-257>
<118-257>	<130-193>	<122-178>	<106-260>	<142-164>	<130-193>
<138-209>	<136-163>	<124-163>	<110-201>	<144-155>	<134-162>
<142-197>	<138-147>	<128-154>	<112-196>	<146-149>	<136-147>
<144-195>	<140-145>	<130-151>	<114-193>	<148-140>	<140-145>
<146-163>	<146-141>	<132-146>	<118-177>	<150-137>	<146-135>
<148-149>	<148-138>	<134-139>	<122-169>	<158-133>	<148-105>
<150-141>	<150-131>	<136-137>	<124-164>	<164-130>	<150-98>
<152-139>	<158-129>	<138-133>	<126-149>	<168-97>	<152-97>
<154-100>	<160-103>	<154-101>	<128-147>	<172-85>	<158-89>
<156-97>	<162-98>	<158-98>	<130-142>	<174-75>	<160-85>

No 7	No 8	No 9	No 10	No 11	No 12
<164-75>	<164-97>	<160-89>	<132-139>	<176-71>	<162-84>
<168-69>	<166-90>	<162-82>	<134-134>	<178-66>	<164-81>
<178-65>	<168-85>	<164-77>	<136-133>	<190-65>	<168-73>
<184-52>	<170-82>	<166-75>	<144-131>	<192-45>	<172-71>
<186-49>	<172-77>	<168-69>	<150-105>	<194-42>	<174-70>
<194-35>	<174-73>	<170-65>	<152-102>	<196-38>	<176-65>
<198-27>	<178-70>	<188-51>	<154-99>	<198-35>	<186-51>
<200-23>	<180-67>	<192-50>	<156-98>	<206-33>	<190-41>
<202-20>	<184-66>	<194-45>	<158-81>	<210-27>	<200-33>
<204-16>	<186-50>	<196-35>	<164-71>	<212-25>	<210-27>
<206-2>	<188-42>	<206-34>	<166-67>	<214-21>	<212-21>
<256-0>	<190-39>	<208-25>	<170-66>	<216-18>	<216-18>
	<194-37>	<218-21>	<176-65>	<220-9>	<218-13>
	<196-29>	<220-18>	<178-55>	<242-7>	<222-10>
	<198-27>	<222-17>	<180-47>	<246-3>	<224-9>
	<202-22>	<224-9>	<182-43>	<254-2>	<232-6>
	<204-20>	<234-7>	<186-41>	<256-0>	<236-3>
	<208-17>	<240-3>	<190-39>		<250-2>
	<226-13>	<244-2>	<192-35>		<256-0>
	<230-7>	<256-0>	<196-34>		
	<232-6>		<200-33>		
	<234-4>		<206-29>		
	<238-3>		<208-22>		
	<242-2>		<210-21>		
	<256-0>		<214-13>		
			<220-10>		
			<224-9>		
			<230-8>		
			<232-6>		
			<234-3>		
			<250-2>		
			<256-0>		

Πίνακας Α.7.2 : Αποτελέσματα Προγράμματος Lauder-Paterson για τις Επόμενες 6 Ακολουθίες de Bruijn Μεγέθους 512 bits.

Α.3.6 Ακολουθίες de Bruijn μεγέθους 1024 bits

Για τις ακολουθίες προς δοκιμή που βρίσκονται στο αρχείο De_Bruijn_10.txt, μεγέθους 1024 bits το πρόγραμμα Lauder-Paterson υπολόγισε τα ακόλουθα αποτελέσματα βρίσκονται αποθηκευμένα στο αρχείο **LauderPaterson _10.csv** .

No 1	No 2	No 3	No 4	No 5	No 6
<0-1023>	<0-1023>	<0-1022>	<0-1023>	<0-1022>	<0-1021>
<2-1018>	<2-1017>	<2-1019>	<2-1018>	<2-1017>	<4-1012>
<4-1011>	<6-1001>	<4-1013>	<4-1013>	<6-1009>	<6-1003>
<6-1005>	<10-998>	<6-1011>	<6-1009>	<10-999>	<8-1001>
<8-1001>	<12-995>	<8-1001>	<10-997>	<12-994>	<10-996>
<10-997>	<14-982>	<12-995>	<12-985>	<16-977>	<12-993>
<12-993>	<16-979>	<14-981>	<14-979>	<22-968>	<20-981>
<20-979>	<18-978>	<18-977>	<16-973>	<24-963>	<22-977>
<22-966>	<20-968>	<20-972>	<18-971>	<30-961>	<24-970>
<24-962>	<22-964>	<22-969>	<20-969>	<34-938>	<26-945>
<30-937>	<24-963>	<26-962>	<22-965>	<36-931>	<28-935>
<32-930>	<26-962>	<30-961>	<24-963>	<38-930>	<30-930>
<34-922>	<28-947>	<34-937>	<30-962>	<40-929>	<34-919>
<36-917>	<30-938>	<36-933>	<32-938>	<42-915>	<38-914>
<42-916>	<32-934>	<38-930>	<34-934>	<44-913>	<42-913>
<44-913>	<34-919>	<42-915>	<36-929>	<46-905>	<46-907>
<48-909>	<36-914>	<46-913>	<48-914>	<58-903>	<48-898>
<50-905>	<40-913>	<50-910>	<50-913>	<60-899>	<64-865>
<54-902>	<44-909>	<52-906>	<52-906>	<62-867>	<66-847>
<56-900>	<48-904>	<54-905>	<56-901>	<64-853>	<68-842>
<58-899>	<50-900>	<56-899>	<60-897>	<66-849>	<72-839>
<60-898>	<52-897>	<62-897>	<68-857>	<68-842>	<74-837>
<62-869>	<76-849>	<66-869>	<70-849>	<70-836>	<80-833>
<64-852>	<78-845>	<68-849>	<72-839>	<72-833>	<86-809>
<66-849>	<80-841>	<70-844>	<74-836>	<84-810>	<88-805>
<70-839>	<82-834>	<72-841>	<76-818>	<88-804>	<92-801>
<72-838>	<88-817>	<76-837>	<78-810>	<90-791>	<96-794>
<74-834>	<90-809>	<78-833>	<82-809>	<92-790>	<98-790>
<78-817>	<92-807>	<82-818>	<84-805>	<94-784>	<100-786>
<80-810>	<94-802>	<86-811>	<90-801>	<96-778>	<104-785>
<82-806>	<98-790>	<88-801>	<94-793>	<102-777>	<108-779>
<84-803>	<100-787>	<104-793>	<96-789>	<112-776>	<110-778>
<88-792>	<104-778>	<106-789>	<98-782>	<114-770>	<114-774>
<90-787>	<108-777>	<108-779>	<100-774>	<118-709>	<118-773>
<94-786>	<112-770>	<110-778>	<106-773>	<120-705>	<120-771>
<96-785>	<126-705>	<112-777>	<108-772>	<122-679>	<124-705>
<102-781>	<130-689>	<114-775>	<114-770>	<126-660>	<128-679>
<106-777>	<132-677>	<116-771>	<116-707>	<128-655>	<130-667>
<116-773>	<134-658>	<122-770>	<118-705>	<130-653>	<132-653>
<118-772>	<136-651>	<126-709>	<126-681>	<134-650>	<136-647>
<120-769>	<140-649>	<130-677>	<128-657>	<138-642>	<138-642>
<136-709>	<142-647>	<132-663>	<140-647>	<146-641>	<154-641>
<140-677>	<144-644>	<134-659>	<142-645>	<166-613>	<162-621>
<142-674>	<146-642>	<136-655>	<146-617>	<168-609>	<164-603>
<144-673>	<152-641>	<138-653>	<148-610>	<170-593>	<166-589>

No 1	No 2	No 3	No 4	No 5	No 6
<146-658>	<154-618>	<140-650>	<150-609>	<178-585>	<168-586>
<148-651>	<156-613>	<142-647>	<154-602>	<180-578>	<170-580>
<152-650>	<162-610>	<144-645>	<156-597>	<186-566>	<172-578>
<154-645>	<164-596>	<150-643>	<158-595>	<188-556>	<178-562>
<160-643>	<166-593>	<156-613>	<160-585>	<190-553>	<180-554>
<162-642>	<172-589>	<158-611>	<170-581>	<196-525>	<182-534>
<164-613>	<174-588>	<160-594>	<172-578>	<206-522>	<186-514>
<166-610>	<176-581>	<164-593>	<174-577>	<212-521>	<214-513>
<168-609>	<180-580>	<168-587>	<178-569>	<214-518>	<298-417>
<170-597>	<182-577>	<170-582>	<180-561>	<216-514>	<300-385>
<172-594>	<194-557>	<174-577>	<186-557>	<244-421>	<304-321>
<174-589>	<196-554>	<192-561>	<188-552>	<246-389>	<318-305>
<176-582>	<198-553>	<200-554>	<190-549>	<250-338>	<320-293>
<178-581>	<200-549>	<202-549>	<196-540>	<254-321>	<322-291>
<180-578>	<202-547>	<206-546>	<198-537>	<276-299>	<324-273>
<182-569>	<206-545>	<208-535>	<202-532>	<278-290>	<332-266>
<184-565>	<216-540>	<210-533>	<206-526>	<284-289>	<334-262>
<186-562>	<218-534>	<214-530>	<208-403>	<292-278>	<336-209>
<188-550>	<220-533>	<218-527>	<210-395>	<294-277>	<338-201>
<190-548>	<222-530>	<220-525>	<212-391>	<296-274>	<340-198>
<192-547>	<224-520>	<224-521>	<214-338>	<298-265>	<342-169>
<198-546>	<226-518>	<242-519>	<218-335>	<308-261>	<346-164>
<200-541>	<234-517>	<244-518>	<220-330>	<312-258>	<348-149>
<202-539>	<238-513>	<246-513>	<224-327>	<318-213>	<350-141>
<204-534>	<274-401>	<266-393>	<226-321>	<320-194>	<356-137>
<206-531>	<278-389>	<268-353>	<244-313>	<322-193>	<360-133>
<214-530>	<282-323>	<270-329>	<246-306>	<330-185>	<364-130>
<218-526>	<284-305>	<272-325>	<248-297>	<332-169>	<370-117>
<220-523>	<286-301>	<274-322>	<250-295>	<334-162>	<372-115>
<224-516>	<288-297>	<278-298>	<252-279>	<340-153>	<374-102>
<228-513>	<292-292>	<280-294>	<254-278>	<346-149>	<376-99>
<284-385>	<294-289>	<282-293>	<256-276>	<348-146>	<380-98>
<292-322>	<300-269>	<286-282>	<258-272>	<350-138>	<382-74>
<296-321>	<306-262>	<288-272>	<260-258>	<352-113>	<390-50>
<302-305>	<312-260>	<290-266>	<300-227>	<354-105>	<394-42>
<304-290>	<314-257>	<296-260>	<302-205>	<356-97>	<396-34>
<306-282>	<346-225>	<298-257>	<304-201>	<358-86>	<402-30>
<308-274>	<348-195>	<354-209>	<306-193>	<360-85>	<404-3>
<310-269>	<352-193>	<356-197>	<316-177>	<362-84>	<512-0>
<312-268>	<354-165>	<358-195>	<318-164>	<364-81>	
<314-265>	<356-162>	<360-193>	<320-161>	<370-71>	
<320-260>	<358-161>	<362-163>	<332-155>	<372-69>	
<322-257>	<360-153>	<366-161>	<334-148>	<374-65>	
<338-198>	<362-146>	<368-148>	<336-146>	<390-3>	
<340-196>	<364-145>	<370-146>	<338-143>	<512-0>	

No 1	No 2	No 3	No 4	No 5	No 6
<342-193>	<366-141>	<374-140>	<340-140>		
<350-169>	<368-137>	<376-129>	<342-130>		
<352-157>	<374-132>	<410-109>	<354-129>		
<354-146>	<376-113>	<412-98>	<372-105>		
<356-142>	<378-102>	<416-83>	<374-100>		
<358-141>	<380-101>	<418-81>	<376-93>		
<360-135>	<382-99>	<424-74>	<378-89>		
<366-132>	<386-98>	<428-71>	<380-85>		
<368-131>	<388-90>	<430-69>	<382-81>		
<372-129>	<390-85>	<432-67>	<386-71>		
<392-106>	<394-81>	<434-65>	<390-69>		
<394-100>	<402-75>	<442-44>	<398-65>		
<396-86>	<404-74>	<444-33>	<416-51>		
<398-83>	<406-69>	<476-25>	<418-45>		
<400-78>	<412-55>	<478-19>	<420-40>		
<402-77>	<414-51>	<482-13>	<422-36>		
<404-68>	<416-45>	<488-11>	<424-31>		
<408-67>	<418-43>	<490-7>	<426-24>		
<412-65>	<420-39>	<494-3>	<428-23>		
<432-53>	<424-38>	<504-2>	<430-2>		
<436-51>	<426-33>	<512-0>	<512-0>		
<438-45>	<456-29>				
<440-41>	<458-26>				
<444-39>	<462-23>				
<446-37>	<464-18>				
<450-33>	<470-14>				
<462-24>	<472-13>				
<464-22>	<476-12>				
<466-20>	<478-11>				
<468-19>	<480-9>				
<472-17>	<486-8>				
<484-10>	<488-4>				
<490-9>	<490-2>				
<492-8>	<512-0>				
<494-6>					
<496-3>					
<506-2>					
<512-0>					

Πίνακας Α.8.1 : Αποτελέσματα Προγράμματος Lauder-Paterson για τις Πρώτες 6 Ακολουθίες de Bruijn Μεγέθους 1024 bits.

Ακολουθούν αποτελέσματα για τις επόμενες έξι (6) ακολουθίες de Bruijn :

No 7	No 8	No 9	No 10	No 11	No 12
<0-1016>	<0-1020>	<0-1021>	<0-1023>	<0-1022>	<0-1020>
<2-1013>	<2-1017>	<4-1011>	<2-1018>	<2-1019>	<2-1017>
<6-1009>	<6-1013>	<8-1002>	<4-1013>	<4-1011>	<6-1009>
<10-1001>	<8-1001>	<10-996>	<6-1010>	<6-1009>	<10-993>
<16-977>	<10-995>	<12-982>	<8-1003>	<10-1001>	<22-969>
<22-972>	<14-982>	<14-974>	<10-997>	<12-995>	<26-962>
<24-965>	<16-979>	<16-967>	<12-994>	<14-982>	<30-961>
<28-961>	<18-973>	<18-964>	<14-985>	<16-977>	<34-931>
<36-933>	<20-969>	<20-962>	<16-981>	<20-971>	<38-920>
<38-916>	<24-967>	<32-949>	<18-977>	<22-969>	<40-917>
<40-913>	<26-965>	<34-932>	<20-971>	<26-965>	<44-908>
<46-908>	<28-962>	<36-929>	<22-969>	<30-961>	<46-907>
<48-906>	<32-937>	<48-914>	<24-963>	<34-933>	<48-904>
<50-898>	<34-935>	<50-906>	<28-962>	<40-930>	<50-901>
<58-897>	<36-933>	<54-899>	<30-961>	<42-915>	<56-899>
<70-867>	<38-930>	<62-866>	<34-937>	<46-909>	<64-850>
<72-841>	<40-929>	<64-853>	<36-933>	<48-907>	<66-849>
<76-838>	<42-921>	<66-849>	<38-932>	<50-905>	<68-842>
<78-821>	<44-916>	<70-835>	<40-929>	<54-900>	<70-841>
<80-817>	<46-914>	<78-819>	<44-915>	<56-897>	<72-838>
<82-811>	<48-902>	<80-810>	<46-913>	<72-853>	<74-835>
<84-806>	<54-901>	<82-807>	<48-906>	<74-842>	<78-810>
<86-801>	<56-898>	<84-802>	<50-903>	<76-835>	<80-809>
<102-791>	<60-897>	<88-793>	<52-898>	<80-833>	<82-807>
<104-785>	<68-866>	<92-790>	<62-897>	<86-808>	<84-803>
<112-780>	<70-865>	<94-789>	<66-842>	<88-804>	<88-797>
<114-774>	<72-842>	<96-787>	<68-839>	<90-803>	<90-790>
<118-771>	<74-839>	<98-784>	<70-838>	<92-801>	<92-787>
<122-770>	<76-834>	<100-778>	<72-837>	<102-793>	<96-786>
<126-769>	<80-818>	<106-775>	<74-825>	<106-789>	<98-785>
<130-709>	<82-813>	<112-772>	<76-819>	<108-782>	<102-783>
<132-707>	<84-809>	<114-771>	<78-817>	<110-773>	<104-778>
<134-654>	<86-805>	<118-770>	<80-811>	<120-771>	<108-773>
<136-652>	<88-801>	<122-707>	<82-805>	<126-769>	<118-772>
<138-651>	<96-791>	<124-705>	<86-802>	<130-705>	<122-769>
<140-642>	<98-789>	<132-677>	<90-794>	<132-675>	<134-689>
<156-641>	<100-786>	<134-674>	<92-788>	<134-673>	<136-685>
<160-625>	<102-785>	<136-673>	<94-787>	<138-666>	<138-661>
<162-613>	<106-779>	<138-658>	<96-786>	<140-658>	<140-658>
<166-610>	<110-777>	<140-649>	<98-782>	<142-657>	<142-651>
<168-593>	<112-776>	<150-641>	<100-778>	<144-650>	<148-646>
<172-589>	<114-770>	<166-613>	<106-775>	<146-649>	<152-645>
<176-580>	<122-707>	<168-602>	<110-773>	<148-647>	<154-644>

No 7	No 8	No 9	No 10	No 11	No 12
<180-578>	<126-674>	<170-595>	<112-771>	<150-644>	<156-643>
<184-577>	<128-673>	<172-593>	<120-737>	<152-641>	<158-617>
<190-558>	<132-666>	<176-586>	<122-709>	<162-613>	<160-610>
<192-550>	<134-658>	<178-582>	<126-705>	<164-601>	<162-601>
<194-541>	<136-657>	<180-579>	<130-665>	<166-585>	<164-596>
<196-534>	<138-652>	<184-565>	<132-661>	<176-583>	<166-587>
<200-526>	<140-650>	<186-561>	<134-659>	<178-579>	<168-585>
<202-520>	<142-646>	<192-549>	<136-657>	<180-566>	<170-583>
<206-514>	<146-645>	<200-546>	<138-652>	<182-562>	<172-581>
<242-513>	<148-643>	<202-545>	<140-650>	<184-558>	<176-579>
<270-386>	<154-641>	<206-538>	<142-647>	<186-555>	<182-569>
<272-385>	<156-617>	<208-533>	<144-646>	<190-553>	<184-567>
<276-333>	<158-609>	<214-527>	<146-643>	<192-550>	<186-562>
<278-301>	<162-603>	<216-525>	<150-641>	<194-549>	<188-553>
<280-298>	<164-601>	<218-522>	<160-611>	<198-547>	<194-548>
<284-290>	<166-593>	<224-519>	<162-602>	<200-546>	<196-546>
<292-289>	<174-589>	<232-517>	<164-597>	<202-545>	<200-543>
<296-279>	<176-582>	<236-514>	<168-594>	<210-537>	<202-541>
<298-274>	<178-579>	<252-513>	<170-589>	<216-535>	<204-534>
<302-273>	<182-578>	<260-386>	<172-587>	<218-533>	<208-533>
<304-225>	<184-569>	<264-337>	<174-585>	<220-530>	<210-531>
<308-209>	<186-565>	<266-331>	<176-582>	<222-529>	<212-530>
<310-201>	<188-555>	<268-324>	<178-569>	<224-525>	<218-529>
<314-177>	<192-548>	<270-322>	<182-565>	<232-519>	<222-527>
<316-171>	<194-547>	<274-309>	<184-556>	<234-517>	<224-521>
<318-169>	<196-546>	<276-298>	<186-553>	<238-516>	<238-518>
<320-165>	<202-545>	<278-297>	<190-551>	<244-515>	<242-517>
<322-154>	<204-534>	<280-294>	<192-549>	<246-514>	<244-513>
<324-146>	<206-529>	<282-292>	<198-546>	<250-389>	<268-390>
<330-145>	<224-527>	<284-289>	<202-541>	<252-387>	<270-386>
<332-141>	<226-525>	<294-273>	<204-538>	<258-341>	<274-341>
<336-130>	<228-521>	<310-263>	<206-534>	<260-331>	<276-337>
<356-90>	<238-519>	<314-261>	<208-530>	<262-324>	<278-326>
<358-86>	<240-518>	<316-260>	<210-523>	<264-323>	<280-323>
<360-76>	<242-517>	<318-258>	<220-522>	<266-322>	<284-299>
<364-70>	<244-514>	<320-257>	<222-521>	<268-305>	<286-293>
<372-66>	<252-513>	<330-209>	<228-517>	<272-294>	<290-292>
<376-34>	<260-394>	<332-201>	<244-515>	<274-293>	<292-289>
<400-18>	<262-391>	<334-194>	<250-514>	<276-290>	<296-276>
<410-12>	<264-355>	<336-179>	<252-513>	<278-285>	<298-273>
<412-3>	<266-353>	<338-173>	<260-449>	<280-282>	<308-265>
<512-0>	<268-337>	<340-169>	<262-401>	<282-276>	<314-263>
	<270-325>	<342-161>	<264-387>	<284-273>	<316-260>
	<278-300>	<354-153>	<266-345>	<294-269>	<318-257>
	<280-293>	<356-147>	<268-330>	<298-267>	<328-226>

No 7	No 8	No 9	No 10	No 11	No 12
	<284-291>	<358-142>	<270-325>	<300-265>	<330-201>
	<288-290>	<360-135>	<274-323>	<302-262>	<332-198>
	<290-279>	<364-132>	<276-321>	<304-261>	<334-194>
	<292-274>	<366-130>	<280-305>	<306-260>	<336-193>
	<294-271>	<376-113>	<284-293>	<310-258>	<342-169>
	<296-265>	<378-99>	<288-290>	<314-257>	<346-166>
	<312-258>	<380-98>	<290-280>	<318-203>	<348-163>
	<322-217>	<382-91>	<292-273>	<320-197>	<350-149>
	<324-203>	<384-89>	<310-269>	<322-194>	<356-146>
	<326-196>	<388-86>	<312-267>	<326-193>	<358-141>
	<328-195>	<390-81>	<314-259>	<328-177>	<362-137>
	<330-177>	<396-70>	<322-257>	<330-169>	<364-136>
	<332-173>	<400-68>	<330-197>	<332-163>	<366-130>
	<334-169>	<402-67>	<332-193>	<334-154>	<376-129>
	<336-163>	<406-66>	<338-170>	<336-153>	<384-105>
	<340-157>	<410-65>	<340-165>	<338-146>	<386-98>
	<342-146>	<412-59>	<342-162>	<342-141>	<392-89>
	<346-145>	<414-54>	<344-161>	<348-131>	<394-82>
	<350-141>	<416-53>	<348-149>	<362-129>	<398-81>
	<354-137>	<418-46>	<350-145>	<374-113>	<400-72>
	<360-131>	<420-43>	<354-141>	<376-109>	<402-69>
	<372-105>	<424-41>	<356-135>	<378-101>	<410-67>
	<376-101>	<430-40>	<358-133>	<380-97>	<416-53>
	<378-97>	<432-34>	<364-130>	<388-85>	<418-50>
	<380-89>	<442-27>	<368-129>	<390-83>	<420-45>
	<386-83>	<444-25>	<372-114>	<392-82>	<424-43>
	<388-81>	<446-24>	<374-99>	<394-75>	<426-42>
	<392-79>	<448-22>	<378-98>	<398-73>	<428-41>
	<394-75>	<452-20>	<380-97>	<400-70>	<430-39>
	<396-74>	<456-12>	<382-89>	<402-65>	<432-36>
	<398-70>	<460-11>	<384-85>	<424-52>	<434-34>
	<402-68>	<468-10>	<388-83>	<426-51>	<436-31>
	<404-67>	<470-8>	<390-77>	<428-49>	<438-29>
	<406-66>	<472-7>	<392-69>	<434-44>	<440-23>
	<410-53>	<476-3>	<398-66>	<436-37>	<442-21>
	<414-50>	<492-2>	<412-65>	<446-35>	<446-17>
	<416-49>	<512-0>	<414-61>	<450-26>	<466-15>
	<418-41>		<416-58>	<452-23>	<468-14>
	<426-39>		<418-53>	<454-16>	<470-11>
	<428-37>		<420-51>	<456-14>	<476-9>
	<430-35>		<422-49>	<458-13>	<482-7>
	<432-30>		<428-43>	<462-10>	<486-5>
	<434-26>		<430-36>	<472-9>	<492-3>
	<436-24>		<434-35>	<474-8>	<508-2>
	<438-22>		<436-33>	<476-7>	<512-0>

No 7	No 8	No 9	No 10	No 11	No 12
	<442-19>		<454-27>	<482-6>	
	<450-18>		<456-21>	<484-2>	
	<456-12>		<462-18>	<512-0>	
	<458-11>		<464-15>		
	<464-10>		<466-11>		
	<466-9>		<470-10>		
	<474-7>		<474-9>		
	<476-6>		<486-6>		
	<478-5>		<488-4>		
	<496-4>		<494-2>		
	<500-2>		<512-0>		
	<512-0>				

Πίνακας Α.8.2 : Αποτελέσματα Προγράμματος Lauder-Paterson για τις Επόμενες 6 Ακολουθίες de Bruijn Μεγέθους 1024 bits.

Α.3.7 Ακολουθίες de Bruijn μεγέθους 2048 bits

Για τις ακολουθίες προς δοκιμή που βρίσκονται στο αρχείο De_Bruijn_11.txt, μεγέθους 2048 bits το πρόγραμμα Lauder-Paterson υπολόγισε τα ακόλουθα αποτελέσματα βρίσκονται αποθηκευμένα στο αρχείο **LauderPaterson _11.csv** .

No 1	No 2	No 3	No 4	No 5	No 6
<0-2047>	<0-2046>	<0-2047>	<0-2046>	<0-2047>	<0-2047>
<2-2042>	<2-2043>	<2-2042>	<2-2041>	<2-2041>	<2-2041>
<4-2037>	<4-2033>	<4-2037>	<6-2027>	<6-2024>	<6-2034>
<6-2034>	<12-2021>	<6-2034>	<8-2020>	<8-2019>	<8-2025>
<8-2026>	<14-2017>	<8-2022>	<10-2019>	<14-2018>	<10-2022>
<10-2021>	<18-2001>	<10-2019>	<12-2017>	<16-2003>	<12-2019>
<12-2004>	<20-1995>	<14-2018>	<20-2001>	<18-2002>	<14-2018>
<14-1996>	<22-1993>	<16-2003>	<26-1991>	<20-1997>	<16-2003>
<16-1993>	<26-1989>	<18-2002>	<28-1985>	<22-1993>	<18-1993>
<24-1987>	<30-1961>	<20-1997>	<36-1961>	<24-1990>	<24-1987>
<28-1969>	<32-1959>	<22-1994>	<38-1945>	<26-1988>	<30-1986>
<34-1956>	<34-1957>	<24-1989>	<40-1942>	<28-1965>	<32-1973>
<36-1942>	<36-1953>	<28-1986>	<42-1938>	<30-1961>	<34-1957>
<38-1939>	<40-1945>	<30-1969>	<44-1937>	<32-1956>	<36-1953>
<40-1937>	<42-1939>	<32-1962>	<46-1930>	<34-1954>	<44-1940>
<44-1933>	<46-1934>	<34-1961>	<50-1927>	<36-1941>	<46-1937>
<46-1932>	<48-1931>	<36-1941>	<56-1923>	<44-1932>	<52-1929>
<48-1929>	<50-1930>	<38-1938>	<58-1922>	<46-1929>	<56-1927>
<52-1923>	<52-1927>	<42-1934>	<62-1921>	<54-1926>	<58-1923>
<60-1922>	<54-1926>	<44-1929>	<66-1889>	<56-1922>	<62-1921>

No 1	No 2	No 3	No 4	No 5	No 6
<62-1921>	<56-1922>	<54-1926>	<68-1874>	<58-1921>	<66-1890>
<66-1893>	<60-1891>	<58-1923>	<70-1869>	<70-1889>	<68-1877>
<70-1874>	<62-1881>	<60-1922>	<72-1860>	<74-1873>	<70-1873>
<72-1866>	<64-1874>	<62-1893>	<74-1859>	<78-1865>	<72-1862>
<74-1863>	<66-1865>	<64-1891>	<76-1857>	<82-1841>	<74-1859>
<76-1858>	<68-1862>	<66-1877>	<82-1837>	<86-1829>	<78-1857>
<78-1842>	<72-1858>	<68-1862>	<84-1834>	<90-1819>	<84-1841>
<80-1834>	<74-1836>	<70-1859>	<86-1829>	<92-1817>	<86-1834>
<82-1821>	<76-1833>	<74-1858>	<88-1826>	<94-1813>	<88-1825>
<84-1813>	<82-1829>	<80-1841>	<92-1819>	<96-1812>	<100-1813>
<90-1812>	<84-1827>	<86-1837>	<96-1815>	<98-1807>	<102-1811>
<92-1810>	<86-1826>	<88-1829>	<98-1813>	<100-1803>	<104-1803>
<94-1809>	<88-1817>	<90-1825>	<100-1804>	<104-1802>	<106-1802>
<96-1807>	<90-1815>	<96-1818>	<102-1802>	<106-1799>	<108-1801>
<98-1805>	<92-1811>	<98-1817>	<104-1801>	<112-1798>	<112-1799>
<100-1802>	<94-1809>	<100-1813>	<110-1799>	<114-1737>	<116-1797>
<104-1799>	<106-1805>	<102-1809>	<114-1795>	<116-1732>	<120-1794>
<106-1797>	<110-1801>	<104-1807>	<120-1793>	<118-1729>	<126-1731>
<114-1796>	<112-1798>	<106-1802>	<136-1713>	<124-1698>	<128-1701>
<116-1793>	<118-1796>	<112-1797>	<138-1699>	<126-1697>	<130-1688>
<140-1737>	<120-1795>	<122-1794>	<142-1683>	<128-1677>	<132-1678>
<142-1732>	<122-1794>	<128-1737>	<144-1682>	<134-1673>	<134-1677>
<144-1705>	<126-1793>	<130-1713>	<146-1677>	<138-1671>	<136-1669>
<146-1698>	<130-1729>	<132-1683>	<148-1673>	<140-1669>	<148-1668>
<148-1675>	<140-1697>	<136-1681>	<154-1671>	<142-1665>	<150-1666>
<150-1673>	<144-1685>	<140-1675>	<156-1669>	<158-1650>	<154-1641>
<152-1668>	<146-1683>	<142-1673>	<158-1667>	<160-1633>	<156-1637>
<156-1665>	<148-1681>	<146-1670>	<162-1637>	<170-1613>	<158-1634>
<174-1641>	<150-1676>	<148-1669>	<164-1634>	<174-1610>	<160-1621>
<176-1633>	<152-1673>	<150-1665>	<166-1633>	<176-1609>	<162-1618>
<180-1621>	<156-1669>	<172-1637>	<168-1621>	<178-1604>	<164-1617>
<182-1617>	<160-1633>	<178-1611>	<172-1617>	<180-1603>	<166-1613>
<184-1611>	<164-1623>	<180-1610>	<174-1612>	<182-1586>	<168-1611>
<186-1607>	<166-1617>	<182-1607>	<176-1603>	<184-1578>	<170-1607>
<188-1603>	<174-1611>	<184-1605>	<184-1602>	<188-1577>	<172-1604>
<192-1589>	<176-1605>	<186-1602>	<186-1601>	<190-1573>	<174-1603>
<194-1586>	<178-1601>	<196-1587>	<192-1585>	<194-1570>	<176-1602>
<196-1563>	<188-1588>	<198-1579>	<198-1575>	<198-1569>	<180-1593>
<198-1562>	<190-1585>	<200-1570>	<200-1573>	<206-1565>	<182-1587>
<200-1558>	<196-1574>	<208-1557>	<202-1571>	<208-1563>	<184-1586>
<202-1555>	<198-1573>	<218-1553>	<204-1567>	<210-1561>	<186-1581>
<206-1554>	<200-1569>	<228-1547>	<206-1565>	<212-1559>	<188-1575>
<210-1551>	<212-1565>	<234-1545>	<208-1561>	<214-1557>	<190-1570>
<212-1549>	<214-1559>	<238-1540>	<214-1558>	<218-1555>	<200-1566>
<222-1547>	<216-1556>	<240-1539>	<216-1554>	<220-1554>	<202-1563>

No 1	No 2	No 3	No 4	No 5	No 6
<224-1546>	<218-1555>	<250-1538>	<222-1544>	<222-1550>	<206-1556>
<226-1545>	<220-1548>	<252-1537>	<224-1543>	<224-1548>	<210-1554>
<230-1542>	<224-1547>	<260-1412>	<228-1540>	<228-1544>	<214-1553>
<234-1538>	<226-1543>	<262-1410>	<234-1441>	<230-1541>	<228-1549>
<252-1537>	<232-1541>	<264-1409>	<236-1418>	<246-1539>	<230-1546>
<260-1411>	<246-1539>	<268-1381>	<238-1409>	<252-1414>	<232-1545>
<262-1361>	<248-1538>	<270-1322>	<248-1361>	<254-1411>	<238-1544>
<264-1353>	<252-1426>	<274-1317>	<250-1353>	<258-1363>	<240-1542>
<268-1349>	<254-1413>	<282-1315>	<254-1348>	<262-1361>	<244-1540>
<270-1347>	<258-1410>	<284-1313>	<256-1334>	<264-1351>	<246-1539>
<272-1321>	<262-1355>	<286-1307>	<258-1329>	<266-1331>	<248-1538>
<276-1319>	<264-1351>	<288-1303>	<264-1307>	<268-1330>	<254-1441>
<278-1313>	<266-1346>	<290-1301>	<266-1305>	<270-1321>	<256-1413>
<288-1305>	<270-1317>	<292-1299>	<268-1303>	<278-1314>	<258-1411>
<290-1297>	<274-1314>	<296-1293>	<270-1301>	<280-1313>	<264-1377>
<302-1291>	<278-1303>	<298-1291>	<272-1298>	<282-1307>	<266-1357>
<304-1290>	<280-1302>	<300-1290>	<276-1294>	<284-1306>	<268-1350>
<306-1286>	<282-1300>	<304-1285>	<278-1289>	<286-1301>	<270-1347>
<308-1285>	<284-1297>	<314-1283>	<292-1287>	<290-1297>	<272-1345>
<310-1281>	<296-1293>	<316-1281>	<294-1281>	<298-1294>	<280-1323>
<322-1253>	<298-1287>	<328-1234>	<316-1249>	<300-1291>	<282-1317>
<324-1241>	<300-1283>	<330-1225>	<318-1233>	<302-1289>	<284-1315>
<326-1222>	<308-1229>	<332-1221>	<320-1219>	<310-1287>	<286-1313>
<328-1219>	<310-1226>	<334-1219>	<322-1217>	<312-1285>	<290-1305>
<330-1217>	<314-1221>	<336-1193>	<326-1201>	<316-1281>	<292-1300>
<338-1202>	<316-1220>	<338-1185>	<328-1186>	<334-1219>	<294-1295>
<340-1201>	<318-1201>	<350-1179>	<336-1185>	<336-1217>	<296-1289>
<342-1189>	<320-1193>	<352-1171>	<338-1177>	<338-1201>	<312-1283>
<348-1178>	<322-1191>	<356-1169>	<340-1173>	<340-1186>	<318-1282>
<350-1174>	<324-1188>	<358-1164>	<342-1169>	<342-1172>	<320-1235>
<352-1170>	<326-1185>	<360-1161>	<348-1165>	<344-1167>	<322-1220>
<356-1159>	<334-1178>	<366-1129>	<352-1162>	<346-1163>	<324-1205>
<360-1158>	<336-1177>	<368-1126>	<354-1156>	<348-1162>	<326-1194>
<362-1153>	<338-1169>	<370-1123>	<356-1155>	<350-1161>	<328-1191>
<392-1137>	<348-1165>	<372-1121>	<364-1137>	<354-1159>	<330-1187>
<394-1129>	<350-1162>	<376-1115>	<368-1132>	<356-1157>	<334-1185>
<396-1123>	<352-1161>	<378-1104>	<370-1127>	<358-1129>	<340-1181>
<398-1109>	<354-1159>	<380-1101>	<372-1123>	<364-1126>	<342-1177>
<400-1107>	<356-1157>	<384-1099>	<374-1122>	<366-1117>	<344-1172>
<402-1102>	<360-1156>	<386-1089>	<378-1111>	<368-1113>	<346-1167>
<404-1098>	<362-1153>	<438-1076>	<380-1106>	<370-1108>	<348-1153>
<408-1095>	<372-1127>	<440-1068>	<382-1105>	<372-1105>	<398-1123>
<412-1094>	<374-1125>	<442-1065>	<384-1101>	<384-1101>	<400-1114>
<414-1091>	<376-1123>	<446-1062>	<386-1099>	<386-1096>	<402-1089>
<416-1077>	<378-1122>	<448-1058>	<390-1097>	<388-1092>	<460-1075>

No 1	No 2	No 3	No 4	No 5	No 6
<420-1066>	<380-1115>	<454-1050>	<392-1092>	<390-1089>	<464-1065>
<428-1062>	<384-1109>	<456-1049>	<394-1089>	<422-1081>	<466-1063>
<430-1057>	<388-1099>	<462-1043>	<416-1081>	<424-1078>	<468-1060>
<460-1051>	<394-1095>	<472-1036>	<420-1075>	<426-1074>	<470-1051>
<462-1046>	<396-1093>	<474-1034>	<422-1071>	<428-1073>	<472-1049>
<464-1043>	<404-1090>	<480-1031>	<424-1066>	<432-1071>	<476-1047>
<466-1042>	<416-1077>	<486-1027>	<426-1065>	<434-1068>	<478-1042>
<468-1034>	<418-1073>	<490-1025>	<428-1063>	<436-1064>	<480-1035>
<472-1025>	<424-1065>	<534-785>	<430-1051>	<438-1063>	<484-1034>
<552-773>	<432-1061>	<538-770>	<432-1050>	<440-1061>	<486-1025>
<558-673>	<436-1060>	<542-769>	<434-1049>	<446-1059>	<538-781>
<560-665>	<438-1057>	<546-659>	<438-1044>	<450-1058>	<540-775>
<562-646>	<444-1048>	<548-658>	<440-1041>	<452-1057>	<542-774>
<566-643>	<446-1046>	<550-650>	<460-1040>	<456-1053>	<544-769>
<572-595>	<450-1044>	<552-645>	<462-1036>	<458-1050>	<558-675>
<574-593>	<452-1040>	<560-644>	<464-1035>	<460-1044>	<560-658>
<578-589>	<454-1037>	<562-642>	<470-1033>	<462-1039>	<562-653>
<580-586>	<464-1033>	<564-641>	<478-1029>	<466-1038>	<564-649>
<582-582>	<484-1031>	<566-613>	<508-1027>	<468-1034>	<568-644>
<584-578>	<486-1029>	<568-609>	<510-1025>	<474-1032>	<570-642>
<586-569>	<496-1025>	<570-595>	<514-769>	<476-1031>	<572-589>
<588-566>	<528-897>	<572-583>	<544-677>	<480-1030>	<574-583>
<590-557>	<530-803>	<574-580>	<546-675>	<486-1029>	<578-580>
<592-553>	<532-788>	<576-578>	<548-673>	<488-1027>	<580-577>
<598-550>	<534-785>	<582-577>	<550-658>	<498-1025>	<598-567>
<600-549>	<538-781>	<586-563>	<552-651>	<526-801>	<600-556>
<602-548>	<540-773>	<588-554>	<554-649>	<532-777>	<602-554>
<604-542>	<546-771>	<590-550>	<558-645>	<538-661>	<604-552>
<606-539>	<548-770>	<592-541>	<562-641>	<542-648>	<606-549>
<608-537>	<550-705>	<594-537>	<564-609>	<544-641>	<610-547>
<610-534>	<552-673>	<600-535>	<570-595>	<566-609>	<612-546>
<612-530>	<556-659>	<602-527>	<572-593>	<568-597>	<616-529>
<618-529>	<558-649>	<604-525>	<574-587>	<570-593>	<634-526>
<620-527>	<560-645>	<606-513>	<576-581>	<572-589>	<636-522>
<624-526>	<562-643>	<692-393>	<582-579>	<574-577>	<638-521>
<626-523>	<564-642>	<694-390>	<584-558>	<592-563>	<644-517>
<628-513>	<566-610>	<696-385>	<586-550>	<594-554>	<650-515>
<688-405>	<568-597>	<708-345>	<588-549>	<596-553>	<652-514>
<690-403>	<570-582>	<710-337>	<590-547>	<600-547>	<654-396>
<692-389>	<572-569>	<712-330>	<598-546>	<602-537>	<656-393>
<696-385>	<574-565>	<714-329>	<600-545>	<608-533>	<662-391>
<708-357>	<576-557>	<716-310>	<602-535>	<610-532>	<664-355>
<710-354>	<578-555>	<718-306>	<606-530>	<612-529>	<666-337>
<712-338>	<580-553>	<720-305>	<622-522>	<618-523>	<670-333>
<714-327>	<582-550>	<724-298>	<624-518>	<622-519>	<672-329>

No 1	No 2	No 3	No 4	No 5	No 6
<716-325>	<584-549>	<726-293>	<626-517>	<624-516>	<676-324>
<718-323>	<588-546>	<728-292>	<638-514>	<632-419>	<678-311>
<720-315>	<590-535>	<730-287>	<642-513>	<634-404>	<680-300>
<722-309>	<592-524>	<732-279>	<662-421>	<636-401>	<682-298>
<724-298>	<598-513>	<734-278>	<664-395>	<642-393>	<686-297>
<728-291>	<702-417>	<736-275>	<666-393>	<644-391>	<688-293>
<734-283>	<704-394>	<740-273>	<668-387>	<648-369>	<698-291>
<736-277>	<706-389>	<748-265>	<670-385>	<650-361>	<700-281>
<738-273>	<710-369>	<764-261>	<678-353>	<652-353>	<708-277>
<752-267>	<712-338>	<770-257>	<684-337>	<654-341>	<712-273>
<754-263>	<714-331>	<782-209>	<686-326>	<656-330>	<724-269>
<758-257>	<716-329>	<788-201>	<688-321>	<658-327>	<728-268>
<790-225>	<718-323>	<790-198>	<700-298>	<660-324>	<730-233>
<798-203>	<722-321>	<792-195>	<702-291>	<662-299>	<732-226>
<800-198>	<726-306>	<794-194>	<708-289>	<666-297>	<734-225>
<802-195>	<728-294>	<796-170>	<710-282>	<670-293>	<736-199>
<808-178>	<730-289>	<798-165>	<712-277>	<674-291>	<740-195>
<810-170>	<746-282>	<806-162>	<714-275>	<676-287>	<744-193>
<812-169>	<748-281>	<808-161>	<718-267>	<678-281>	<756-177>
<814-163>	<750-277>	<810-153>	<724-265>	<686-279>	<758-173>
<816-154>	<752-276>	<812-149>	<726-263>	<688-278>	<760-170>
<818-145>	<754-273>	<814-146>	<730-258>	<690-270>	<762-167>
<842-141>	<760-266>	<816-141>	<740-225>	<692-269>	<764-162>
<844-137>	<766-257>	<820-139>	<742-217>	<694-257>	<772-161>
<848-134>	<794-214>	<822-136>	<744-210>	<778-205>	<774-147>
<850-131>	<796-207>	<824-134>	<746-209>	<780-201>	<780-146>
<852-130>	<798-202>	<826-129>	<750-203>	<786-195>	<782-143>
<856-113>	<800-201>	<860-114>	<752-197>	<788-193>	<784-139>
<858-103>	<802-199>	<862-113>	<758-185>	<792-178>	<790-135>
<862-98>	<804-193>	<864-101>	<760-173>	<794-177>	<792-133>
<868-90>	<820-177>	<870-99>	<762-166>	<796-170>	<804-129>
<870-85>	<822-169>	<872-87>	<764-165>	<798-167>	<832-114>
<872-83>	<824-165>	<874-85>	<766-151>	<800-165>	<834-106>
<876-82>	<826-162>	<876-84>	<770-149>	<802-164>	<836-102>
<878-78>	<828-152>	<878-82>	<772-148>	<804-161>	<838-97>
<880-77>	<830-145>	<882-75>	<774-147>	<810-149>	<848-90>
<882-76>	<846-139>	<888-72>	<778-142>	<814-147>	<850-85>
<884-73>	<848-135>	<890-71>	<780-135>	<816-145>	<852-84>
<892-71>	<852-131>	<892-68>	<784-132>	<820-141>	<854-77>
<894-69>	<856-130>	<896-67>	<792-129>	<824-139>	<856-73>
<896-67>	<858-107>	<898-65>	<858-115>	<826-138>	<864-70>
<904-58>	<860-101>	<928-57>	<860-109>	<828-133>	<866-69>
<906-50>	<866-97>	<930-49>	<862-100>	<832-132>	<870-55>
<910-45>	<870-87>	<936-42>	<864-99>	<834-119>	<874-54>
<914-43>	<872-84>	<938-41>	<866-97>	<836-109>	<876-52>

No 1	No 2	No 3	No 4	No 5	No 6
<918-39>	<874-82>	<940-37>	<878-85>	<838-107>	<878-45>
<920-37>	<876-81>	<948-35>	<880-83>	<840-100>	<884-30>
<924-35>	<882-74>	<950-33>	<884-76>	<844-99>	<886-2>
<926-33>	<888-73>	<952-27>	<886-68>	<846-97>	<1024-0>
<948-26>	<890-68>	<954-25>	<888-65>	<858-90>	
<950-23>	<892-67>	<960-23>	<928-51>	<860-87>	
<954-21>	<900-57>	<962-20>	<930-49>	<862-72>	
<958-20>	<904-53>	<964-17>	<932-33>	<864-54>	
<960-17>	<906-49>	<974-11>	<968-29>	<866-49>	
<972-11>	<914-44>	<980-7>	<970-25>	<892-43>	
<982-7>	<916-42>	<986-5>	<972-23>	<896-41>	
<986-5>	<918-41>	<1002-2>	<974-21>	<898-37>	
<1006-3>	<922-35>	<1024-0>	<980-18>	<904-35>	
<1014-2>	<926-34>		<982-14>	<906-32>	
<1024-0>	<930-33>		<984-12>	<908-29>	
	<938-28>		<988-11>	<914-2>	
	<940-23>		<990-9>	<1024-0>	
	<944-21>		<1004-6>		
	<952-18>		<1008-5>		
	<958-15>		<1016-4>		
	<960-12>		<1018-3>		
	<964-9>		<1020-2>		
	<986-7>		<1024-0>		
	<994-6>				
	<998-3>				
	<1008-2>				
	<1024-0>				

Πίνακας Α.9.1 : Αποτελέσματα Προγράμματος Lauder-Paterson για τις Πρώτες 6 Ακολουθίες de Bruijn Μεγέθους 2048 bits.

Ακολουθούν αποτελέσματα για τις επόμενες έξι (6) ακολουθίες de Bruijn :

No 7	No 8	No 9	No 10	No 11	No 12
<0-2047>	<0-2046>	<0-2047>	<0-2047>	<0-2047>	<0-2047>
<2-2042>	<2-2035>	<2-2037>	<2-2035>	<2-2042>	<2-2036>
<4-2037>	<8-2029>	<6-2034>	<6-2034>	<4-2037>	<4-2034>
<6-2033>	<10-2025>	<8-2026>	<8-2022>	<6-2034>	<6-2033>
<10-2009>	<12-2019>	<10-2025>	<10-2021>	<8-2027>	<10-2029>
<12-2003>	<14-2009>	<12-2017>	<12-2018>	<10-2009>	<12-2019>
<14-2002>	<16-2005>	<20-2001>	<14-2017>	<12-2005>	<14-2018>
<16-1997>	<18-1994>	<24-1993>	<18-1998>	<16-2001>	<16-2001>
<20-1993>	<20-1991>	<30-1986>	<20-1993>	<18-1994>	<22-1997>
<22-1991>	<24-1987>	<32-1958>	<28-1986>	<20-1993>	<24-1990>
<24-1986>	<26-1973>	<34-1954>	<30-1985>	<22-1989>	<26-1985>
<30-1985>	<28-1969>	<36-1947>	<34-1969>	<30-1986>	<38-1961>

No 7	No 8	No 9	No 10	No 11	No 12
<34-1961>	<30-1953>	<38-1942>	<36-1957>	<32-1969>	<40-1955>
<40-1957>	<46-1939>	<40-1939>	<38-1953>	<34-1958>	<42-1953>
<42-1941>	<48-1937>	<44-1937>	<42-1946>	<36-1956>	<44-1937>
<44-1938>	<50-1933>	<46-1933>	<44-1935>	<38-1954>	<50-1931>
<46-1931>	<52-1931>	<50-1931>	<46-1931>	<42-1941>	<54-1929>
<50-1925>	<54-1929>	<52-1929>	<48-1930>	<44-1937>	<56-1922>
<54-1893>	<56-1925>	<54-1921>	<50-1926>	<48-1932>	<62-1921>
<56-1890>	<58-1924>	<74-1865>	<54-1923>	<50-1930>	<66-1889>
<58-1889>	<60-1889>	<76-1862>	<60-1889>	<52-1927>	<68-1873>
<60-1873>	<62-1876>	<78-1861>	<64-1875>	<54-1925>	<70-1867>
<62-1865>	<64-1870>	<80-1859>	<66-1869>	<58-1923>	<72-1838>
<68-1862>	<66-1869>	<82-1858>	<68-1861>	<60-1921>	<74-1835>
<70-1858>	<68-1861>	<84-1857>	<72-1859>	<68-1890>	<78-1833>
<74-1844>	<76-1858>	<86-1841>	<74-1857>	<70-1866>	<80-1829>
<76-1835>	<78-1841>	<88-1829>	<80-1843>	<72-1865>	<84-1827>
<78-1832>	<80-1837>	<94-1826>	<82-1834>	<74-1863>	<86-1826>
<80-1829>	<82-1831>	<96-1817>	<84-1833>	<76-1859>	<88-1818>
<88-1827>	<84-1827>	<100-1813>	<86-1829>	<80-1858>	<90-1813>
<90-1826>	<86-1819>	<106-1804>	<88-1825>	<82-1843>	<96-1805>
<92-1817>	<88-1816>	<108-1801>	<94-1815>	<84-1834>	<100-1802>
<96-1813>	<90-1812>	<114-1798>	<96-1814>	<86-1831>	<102-1798>
<98-1810>	<92-1810>	<118-1797>	<98-1813>	<88-1829>	<106-1796>
<102-1804>	<98-1806>	<120-1795>	<100-1810>	<90-1817>	<112-1794>
<104-1800>	<100-1803>	<124-1794>	<104-1809>	<94-1813>	<114-1793>
<106-1799>	<104-1800>	<126-1793>	<108-1806>	<96-1805>	<142-1730>
<108-1797>	<106-1799>	<130-1697>	<110-1802>	<102-1802>	<144-1713>
<116-1794>	<108-1797>	<138-1679>	<114-1794>	<104-1801>	<146-1681>
<126-1793>	<118-1794>	<140-1675>	<120-1729>	<108-1797>	<154-1675>
<130-1729>	<126-1793>	<142-1674>	<126-1713>	<116-1793>	<156-1674>
<138-1700>	<130-1733>	<144-1673>	<128-1698>	<140-1730>	<158-1666>
<140-1697>	<132-1730>	<146-1669>	<132-1689>	<142-1713>	<162-1637>
<150-1689>	<134-1729>	<150-1667>	<136-1683>	<146-1674>	<164-1621>
<152-1685>	<136-1700>	<154-1666>	<138-1679>	<148-1673>	<168-1613>
<154-1673>	<138-1698>	<156-1641>	<140-1676>	<150-1670>	<170-1610>
<160-1669>	<140-1686>	<160-1635>	<142-1674>	<154-1669>	<172-1609>
<164-1665>	<142-1683>	<162-1633>	<144-1673>	<158-1666>	<174-1606>
<168-1633>	<146-1673>	<164-1626>	<148-1669>	<162-1636>	<178-1602>
<170-1625>	<154-1637>	<166-1617>	<152-1665>	<164-1618>	<182-1589>
<172-1609>	<156-1635>	<174-1610>	<164-1635>	<166-1615>	<184-1581>
<174-1607>	<158-1626>	<176-1604>	<166-1633>	<168-1611>	<186-1577>
<176-1603>	<160-1620>	<180-1593>	<170-1625>	<170-1609>	<194-1573>
<180-1602>	<162-1618>	<182-1588>	<172-1618>	<178-1607>	<196-1572>
<182-1601>	<164-1615>	<184-1577>	<174-1610>	<180-1605>	<198-1562>
<186-1593>	<166-1610>	<192-1573>	<176-1609>	<182-1603>	<200-1561>
<188-1589>	<168-1609>	<196-1571>	<178-1605>	<186-1601>	<202-1557>

No 7	No 8	No 9	No 10	No 11	No 12
<190-1580>	<172-1605>	<200-1569>	<182-1603>	<190-1587>	<208-1553>
<192-1578>	<178-1603>	<202-1563>	<184-1587>	<194-1579>	<222-1550>
<194-1577>	<182-1602>	<204-1562>	<186-1585>	<198-1573>	<224-1548>
<196-1576>	<184-1586>	<206-1555>	<188-1581>	<204-1570>	<226-1547>
<198-1573>	<186-1578>	<214-1548>	<190-1579>	<208-1557>	<228-1546>
<204-1564>	<188-1574>	<216-1544>	<192-1571>	<212-1555>	<230-1542>
<206-1557>	<192-1573>	<218-1543>	<202-1566>	<218-1553>	<234-1540>
<216-1553>	<194-1571>	<224-1542>	<204-1563>	<222-1551>	<240-1539>
<226-1551>	<198-1569>	<226-1540>	<206-1562>	<224-1549>	<244-1537>
<228-1548>	<202-1558>	<232-1425>	<208-1553>	<226-1543>	<268-1417>
<230-1546>	<204-1555>	<236-1413>	<230-1550>	<232-1542>	<270-1410>
<232-1543>	<214-1550>	<238-1411>	<232-1549>	<234-1539>	<274-1377>
<234-1537>	<220-1547>	<240-1378>	<234-1543>	<248-1538>	<276-1348>
<278-1417>	<226-1544>	<242-1357>	<240-1539>	<250-1417>	<278-1346>
<280-1410>	<228-1543>	<246-1351>	<254-1537>	<252-1411>	<280-1345>
<282-1377>	<230-1542>	<248-1347>	<258-1425>	<254-1377>	<284-1321>
<284-1357>	<234-1541>	<252-1346>	<260-1411>	<258-1347>	<286-1318>
<286-1347>	<236-1538>	<256-1345>	<264-1409>	<262-1346>	<288-1313>
<288-1346>	<252-1537>	<258-1325>	<268-1361>	<264-1345>	<296-1301>
<290-1322>	<260-1416>	<262-1316>	<270-1353>	<266-1329>	<298-1297>
<292-1317>	<262-1413>	<266-1314>	<272-1350>	<268-1302>	<304-1293>
<294-1313>	<266-1411>	<270-1307>	<274-1329>	<270-1301>	<306-1291>
<304-1305>	<268-1409>	<272-1302>	<278-1327>	<276-1298>	<308-1285>
<306-1299>	<272-1353>	<274-1301>	<280-1319>	<278-1297>	<310-1284>
<308-1297>	<276-1349>	<276-1299>	<282-1315>	<286-1292>	<312-1283>
<310-1291>	<278-1330>	<278-1298>	<288-1314>	<288-1289>	<314-1282>
<316-1281>	<280-1322>	<280-1291>	<290-1313>	<298-1287>	<318-1281>
<344-1233>	<282-1321>	<286-1289>	<292-1300>	<300-1285>	<320-1229>
<346-1221>	<284-1318>	<288-1287>	<294-1299>	<302-1283>	<324-1225>
<348-1201>	<286-1315>	<290-1285>	<296-1294>	<308-1282>	<326-1221>
<352-1194>	<288-1301>	<298-1284>	<300-1292>	<310-1281>	<328-1218>
<354-1180>	<290-1300>	<302-1281>	<302-1290>	<312-1251>	<330-1217>
<356-1177>	<292-1297>	<312-1249>	<306-1287>	<314-1221>	<334-1201>
<362-1173>	<298-1293>	<316-1219>	<308-1233>	<316-1219>	<336-1193>
<364-1153>	<302-1291>	<320-1203>	<310-1225>	<320-1209>	<338-1187>
<426-1105>	<306-1284>	<322-1191>	<316-1221>	<322-1201>	<340-1186>
<428-1099>	<310-1282>	<324-1189>	<318-1194>	<326-1190>	<342-1173>
<430-1097>	<316-1281>	<326-1187>	<320-1189>	<330-1186>	<348-1172>
<434-1095>	<326-1250>	<328-1185>	<322-1186>	<332-1185>	<350-1158>
<436-1093>	<328-1233>	<334-1177>	<326-1177>	<338-1175>	<356-1156>
<438-1081>	<330-1226>	<336-1170>	<328-1171>	<340-1170>	<358-1155>
<440-1074>	<332-1219>	<342-1169>	<330-1169>	<342-1169>	<366-1153>
<444-1069>	<338-1217>	<344-1166>	<336-1166>	<350-1164>	<382-1130>
<446-1063>	<344-1193>	<346-1163>	<338-1165>	<352-1161>	<384-1125>
<452-1061>	<346-1186>	<348-1161>	<340-1163>	<362-1156>	<386-1122>

No 7	No 8	No 9	No 10	No 11	No 12
<454-1057>	<348-1175>	<352-1157>	<342-1161>	<364-1153>	<390-1121>
<470-1049>	<350-1173>	<358-1153>	<346-1157>	<378-1130>	<392-1109>
<474-1044>	<352-1171>	<370-1128>	<348-1155>	<380-1123>	<394-1106>
<476-1034>	<354-1165>	<372-1125>	<356-1154>	<382-1122>	<396-1100>
<484-1027>	<360-1158>	<376-1123>	<358-1153>	<384-1121>	<398-1098>
<500-1026>	<364-1157>	<378-1114>	<360-1121>	<386-1111>	<400-1091>
<510-789>	<366-1154>	<380-1111>	<380-1112>	<388-1107>	<414-1083>
<512-785>	<374-1139>	<382-1107>	<382-1106>	<392-1101>	<416-1081>
<514-769>	<376-1137>	<386-1103>	<386-1105>	<394-1099>	<418-1077>
<524-705>	<378-1129>	<388-1101>	<390-1099>	<398-1095>	<422-1073>
<526-673>	<380-1122>	<390-1098>	<392-1096>	<400-1093>	<428-1069>
<528-658>	<382-1121>	<396-1095>	<394-1094>	<402-1091>	<430-1065>
<530-657>	<386-1108>	<398-1094>	<398-1091>	<410-1076>	<432-1063>
<532-651>	<388-1107>	<400-1093>	<404-1089>	<412-1070>	<434-1061>
<534-641>	<390-1106>	<404-1089>	<412-1083>	<414-1067>	<436-1058>
<548-610>	<394-1105>	<424-1073>	<414-1077>	<416-1063>	<444-1057>
<550-599>	<396-1100>	<430-1069>	<416-1076>	<422-1060>	<448-1047>
<552-594>	<398-1092>	<432-1067>	<418-1073>	<424-1053>	<454-1045>
<554-586>	<402-1089>	<434-1063>	<422-1067>	<428-1051>	<456-1043>
<558-583>	<430-1077>	<436-1060>	<424-1064>	<430-1050>	<462-1042>
<560-579>	<432-1068>	<438-1059>	<426-1063>	<432-1042>	<466-1038>
<564-577>	<434-1066>	<440-1058>	<428-1061>	<440-1040>	<470-1034>
<578-561>	<436-1065>	<442-1053>	<434-1053>	<442-1037>	<472-1031>
<580-555>	<438-1062>	<444-1049>	<436-1050>	<452-1029>	<480-1030>
<582-551>	<444-1057>	<448-1047>	<438-1043>	<484-1026>	<484-1029>
<584-549>	<458-1049>	<450-1043>	<452-1042>	<510-785>	<488-801>
<586-545>	<464-1047>	<452-1042>	<454-1041>	<516-779>	<490-787>
<592-541>	<466-1044>	<458-1040>	<458-1040>	<518-777>	<492-785>
<594-537>	<468-1043>	<460-1037>	<460-1037>	<520-771>	<494-778>
<596-533>	<470-1039>	<468-1036>	<472-1034>	<524-769>	<496-777>
<598-530>	<472-1034>	<470-1034>	<476-1033>	<530-675>	<498-771>
<600-527>	<478-1033>	<472-1031>	<480-1031>	<532-657>	<502-706>
<602-522>	<484-1030>	<482-1027>	<482-1030>	<538-653>	<504-681>
<606-521>	<488-1029>	<498-1026>	<486-1027>	<540-649>	<506-673>
<608-519>	<506-1027>	<500-897>	<506-805>	<544-646>	<508-662>
<616-516>	<508-785>	<502-833>	<508-802>	<546-625>	<510-657>
<618-419>	<512-778>	<506-801>	<510-785>	<548-617>	<516-653>
<620-409>	<514-773>	<508-785>	<514-779>	<550-615>	<518-649>
<622-397>	<516-772>	<512-777>	<518-775>	<552-595>	<522-611>
<624-395>	<518-769>	<516-773>	<520-773>	<554-590>	<526-609>
<626-391>	<524-668>	<518-771>	<524-705>	<556-585>	<528-595>
<630-390>	<526-661>	<520-705>	<526-673>	<560-579>	<532-590>
<632-387>	<528-659>	<522-681>	<528-658>	<574-577>	<534-586>
<636-386>	<532-658>	<524-661>	<530-657>	<576-565>	<536-580>
<640-355>	<534-650>	<526-659>	<532-650>	<578-562>	<540-558>

No 7	No 8	No 9	No 10	No 11	No 12
<642-354>	<538-649>	<528-654>	<534-649>	<580-555>	<542-551>
<644-342>	<540-645>	<530-651>	<538-646>	<582-553>	<544-550>
<646-339>	<542-643>	<532-649>	<540-617>	<584-549>	<546-548>
<648-337>	<546-641>	<534-647>	<542-610>	<590-548>	<548-541>
<652-330>	<554-611>	<536-645>	<544-585>	<592-537>	<554-537>
<656-307>	<556-593>	<538-643>	<550-583>	<596-534>	<556-530>
<658-301>	<558-584>	<542-641>	<552-577>	<598-533>	<570-528>
<660-299>	<560-581>	<548-613>	<566-569>	<600-530>	<572-526>
<662-295>	<564-579>	<552-610>	<568-562>	<602-526>	<574-525>
<664-292>	<566-578>	<554-593>	<570-553>	<604-525>	<580-521>
<666-291>	<568-577>	<556-589>	<578-550>	<606-523>	<596-519>
<668-290>	<574-563>	<560-569>	<580-547>	<610-521>	<598-517>
<670-289>	<576-562>	<562-562>	<582-546>	<616-518>	<600-516>
<678-285>	<578-557>	<564-554>	<584-537>	<622-517>	<602-450>
<680-279>	<582-549>	<566-553>	<588-533>	<624-449>	<604-421>
<682-275>	<586-545>	<574-547>	<596-529>	<626-421>	<606-418>
<684-271>	<602-541>	<578-545>	<602-527>	<628-401>	<608-401>
<688-267>	<604-537>	<588-541>	<604-526>	<630-395>	<616-395>
<690-266>	<606-533>	<590-534>	<606-522>	<632-391>	<620-390>
<694-265>	<608-532>	<592-530>	<610-519>	<636-389>	<624-387>
<700-259>	<610-529>	<598-529>	<616-517>	<638-385>	<628-386>
<710-226>	<616-523>	<610-527>	<620-515>	<648-355>	<630-353>
<712-219>	<618-521>	<612-525>	<624-513>	<650-339>	<634-345>
<714-213>	<622-519>	<614-523>	<648-449>	<654-331>	<636-338>
<716-210>	<624-514>	<616-521>	<650-417>	<656-322>	<638-330>
<720-193>	<632-513>	<624-519>	<654-403>	<668-321>	<640-323>
<750-177>	<640-449>	<628-518>	<656-393>	<670-307>	<650-321>
<754-173>	<642-425>	<632-513>	<660-391>	<672-301>	<654-299>
<756-161>	<644-417>	<660-453>	<662-389>	<674-297>	<658-289>
<780-154>	<646-402>	<662-393>	<664-386>	<678-295>	<680-281>
<782-153>	<650-390>	<664-390>	<668-361>	<680-292>	<684-278>
<784-148>	<654-389>	<666-361>	<670-331>	<682-289>	<686-273>
<786-147>	<656-387>	<668-353>	<672-326>	<690-285>	<692-266>
<788-145>	<660-385>	<670-345>	<674-324>	<692-277>	<698-264>
<792-141>	<666-354>	<672-342>	<678-322>	<698-269>	<700-258>
<796-139>	<668-341>	<674-338>	<680-321>	<702-264>	<706-210>
<798-134>	<670-333>	<676-328>	<694-313>	<704-262>	<710-209>
<800-133>	<674-324>	<678-321>	<696-309>	<706-261>	<714-204>
<804-115>	<676-305>	<702-297>	<698-297>	<708-258>	<716-199>
<806-107>	<682-297>	<704-292>	<702-295>	<722-257>	<718-181>
<808-105>	<684-296>	<706-289>	<704-290>	<732-227>	<720-179>
<810-99>	<686-293>	<720-281>	<706-281>	<734-205>	<724-177>
<818-92>	<690-283>	<726-277>	<710-275>	<736-199>	<728-170>
<820-87>	<692-282>	<728-273>	<714-273>	<738-193>	<732-164>
<824-85>	<694-278>	<730-265>	<722-269>	<762-177>	<734-163>

No 7	No 8	No 9	No 10	No 11	No 12
<828-77>	<696-277>	<738-263>	<726-265>	<766-164>	<738-156>
<830-71>	<698-275>	<740-261>	<734-264>	<768-163>	<740-153>
<838-69>	<702-274>	<742-257>	<736-263>	<770-157>	<744-149>
<848-61>	<704-271>	<754-233>	<738-225>	<772-155>	<746-146>
<850-59>	<706-268>	<756-213>	<742-209>	<774-153>	<752-145>
<852-53>	<708-265>	<758-201>	<744-201>	<776-149>	<756-141>
<854-51>	<720-263>	<762-199>	<750-195>	<780-147>	<766-138>
<858-40>	<724-262>	<764-197>	<754-181>	<782-142>	<768-137>
<860-31>	<726-230>	<766-195>	<756-179>	<784-137>	<774-135>
<862-2>	<728-226>	<768-193>	<758-177>	<796-132>	<778-131>
<1024-0>	<730-213>	<772-178>	<760-171>	<802-130>	<792-130>
	<732-209>	<774-171>	<762-170>	<806-115>	<800-129>
	<736-202>	<776-167>	<764-166>	<808-113>	<806-118>
	<738-200>	<778-165>	<766-162>	<810-102>	<808-107>
	<740-195>	<782-164>	<772-155>	<814-100>	<810-105>
	<746-194>	<784-155>	<774-151>	<816-93>	<812-104>
	<748-185>	<786-150>	<776-149>	<818-90>	<814-101>
	<750-170>	<788-149>	<784-145>	<820-89>	<818-100>
	<754-167>	<790-145>	<792-141>	<822-83>	<820-92>
	<758-165>	<804-141>	<794-139>	<826-82>	<822-90>
	<760-164>	<806-139>	<798-137>	<828-81>	<824-86>
	<762-162>	<808-137>	<804-133>	<834-79>	<826-79>
	<766-161>	<812-131>	<812-131>	<836-77>	<828-77>
	<770-154>	<820-129>	<822-129>	<838-74>	<832-75>
	<772-150>	<836-117>	<826-107>	<842-70>	<836-74>
	<774-146>	<838-105>	<828-102>	<844-69>	<840-73>
	<776-145>	<842-103>	<830-99>	<858-68>	<842-70>
	<784-143>	<844-101>	<838-90>	<860-66>	<854-69>
	<786-141>	<846-99>	<840-87>	<862-54>	<856-68>
	<788-137>	<848-97>	<842-81>	<864-52>	<858-67>
	<794-135>	<854-84>	<854-78>	<866-51>	<862-65>
	<798-134>	<856-82>	<856-69>	<870-49>	<884-61>
	<800-131>	<860-81>	<868-66>	<878-47>	<886-57>
	<804-129>	<866-77>	<874-65>	<880-37>	<890-50>
	<820-106>	<868-72>	<878-59>	<904-35>	<892-45>
	<822-102>	<870-70>	<880-55>	<908-31>	<896-41>
	<824-97>	<874-68>	<882-50>	<910-26>	<908-37>
	<848-85>	<876-65>	<890-49>	<916-23>	<918-36>
	<850-83>	<904-59>	<892-45>	<918-18>	<920-34>
	<852-80>	<906-53>	<896-43>	<926-17>	<922-33>
	<854-77>	<908-47>	<900-41>	<952-14>	<926-27>
	<858-73>	<910-44>	<902-35>	<958-11>	<930-26>
	<864-70>	<912-42>	<910-29>	<976-5>	<932-25>
	<866-68>	<916-41>	<914-27>	<1018-3>	<934-22>
	<872-57>	<920-38>	<916-24>	<1022-2>	<936-21>

No 7	No 8	No 9	No 10	No 11	No 12
	<874-54>	<922-37>	<918-22>	<1024-0>	<942-19>
	<876-50>	<928-35>	<920-19>		<944-18>
	<878-46>	<936-33>	<928-17>		<946-17>
	<880-45>	<938-26>	<954-14>		<958-15>
	<882-41>	<942-24>	<956-9>		<962-11>
	<894-38>	<944-23>	<996-5>		<966-10>
	<896-37>	<946-21>	<1004-4>		<978-9>
	<904-35>	<950-18>	<1006-3>		<986-7>
	<912-27>	<954-17>	<1018-2>		<992-5>
	<916-21>	<970-14>	<1024-0>		<1010-3>
	<940-17>	<972-11>			<1014-2>
	<958-15>	<980-10>			<1024-0>
	<960-14>	<982-5>			
	<962-11>	<1012-3>			
	<970-7>	<1018-2>			
	<978-6>	<1024-0>			
	<982-5>				
	<984-2>				
	<1024-0>				

Πίνακας Α.9.2 : Αποτελέσματα Προγράμματος Lauder-Paterson για τις Επόμενες 6 Ακολουθίες de Bruijn Μεγέθους 2048 bits.

Παράρτημα Β
Πηγαίοι Κώδικες Προγραμμάτων

Στο παράρτημα Β αποτυπώνονται οι πηγαίοι κώδικες των προγραμμάτων σε γλώσσα προγραμματισμού C++ τα οποία χρησιμοποιήθηκαν κατά τη διάρκεια της έρευνας στη παρούσα μεταπτυχιακή διατριβή.

Β.1 Πρόγραμμα δημιουργίας ακολουθιών de Bruijn

Πρόγραμμα το οποίο δημιουργεί ακολουθίες de Bruijn περιόδου (μεγέθους) 2^n για $5 \leq n \leq 11$. Διαμορφώθηκε για τις ανάγκες της έρευνας την 28/3/2019 από τον Κωνσταντίνο Ρόζη, ερευνητή της παρούσας διατριβής, πάνω στο αρχικό πρόγραμμα που σχεδιάστηκε και γράφηκε από τους D. Gabric, J. Sawada, A. Williams, και D. Wong και δημοσιεύτηκε στο άρθρο «A framework for constructing de Bruijn sequences via simple successor rules» στις 5/7/2018 [20].

```
#include<stdio.h>
#define MAX 100

FILE *dbf;

// ====================
// Test if a[1... n] = 0^n
// ====================
int Zeros(int a[ ], int n)   {
        for (int i = 1; i <= n; i++) if (a[i] == 1) return 0;
        return 1;
}
// ============================
// Test if b[1... n] is a necklace
// ============================
int IsNecklace(int b[ ], int n)  {
        int i, p = 1;

        for (i = 2; i <= n; i++)     {
                if (b[i - p] > b[i]) return 0;
                if (b[i - p] < b[i]) p = i;
        }
        if (n % p != 0) return 0;
        return 1;
}
// ========================================
// Necklace Successor Rules
// ========================================
int Granddaddy(int a[ ], int n) {
        int i, j, b[MAX];

        j = 2;
        while (j <= n && a[j] == 1) j++;
        for (i = j; i <= n; i++) b[i - j + 1] = a[i];
        b[n - j + 2] = 0;
        for (i = 2; i<j; i++) b[n - j + i + 1] = a[i];
        if (IsNecklace(b, n)) return 1 - a[1];
        return a[1];
}
```

```
// ------------------------------
int Grandmama(int a[ ], int n) {
        int i, j, k, b[MAX];

        j = 1;
        while (j<n && a[n - j + 1] == 0) b[j++] = 0;
        b[j] = 1; k = 2;
        for (i = j + 1; i <= n; i++) b[i] = a[k++];
        if (IsNecklace(b, n)) return 1 - a[1];
        return a[1];
}
// ------------------------------
int PCR3(int a[ ], int n) {
        int i, b[MAX];

        for (i = 1; i<n; i++) b[i] = a[i + 1];
        b[n] = 1;
        if (IsNecklace(b, n)) return 1 - a[1];
        return a[1];
}
// ------------------------------
int PCR4(int a[ ], int n) {
        int i, b[MAX];

        b[1] = 0;
        for (i = 2; i <= n; i++) b[i] = a[i];
        if (IsNecklace(b, n)) return 1 - a[1];
        return a[1];
}
// ========================================
// Co-necklace Successor Rules
// ========================================
int CCR1(int a[ ], int n) {
        int i, j, b[MAX], c = 1;

        for (i = 2; i <= n; i++) if (a[i] == 0) break;
        for (j = i; j <= n; j++) b[c++] = a[j];
        b[c++] = 1;
        for (j = 2; j<i; j++) b[c++] = 1 - a[j];
        for (i = 1; i <= n; i++) b[n + i] = 1 - b[i];
        if (IsNecklace(b, 2 * n)) return a[1];
        return 1 - a[1];
}
```

```
// ------------------------------
int CCR2(int a[ ], int n) {
        int i, j, b[MAX], c = 1;

        i = n;
        while (a[i] == 0 && i >= 1) i--;
        if (i == 0) i = n;
        for (j = i + 1; j <= n; j++) b[c++] = 0;
        b[c++] = 1;
        for (j = 2; j <= i; j++) b[c++] = 1 - a[j];
        for (j = 1; j <= n; j++) b[n + j] = 1 - b[j];
        if (IsNecklace(b, 2 * n)) return a[1];
        return 1 - a[1];
}
// ------------------------------
int CCR3(int a[ ], int n) {
        int i, b[MAX];

        for (i = 1; i<n; i++) b[i] = a[i + 1];
        b[n] = 0;
        for (i = 1; i <= n; i++) b[n + i] = 1 - b[i];
        if (IsNecklace(b, 2 * n) && !Zeros(b, n)) return a[1];
        return 1 - a[1];
}
// ========================================================================
// Generate de Bruijn sequences by iteratively applying a successor rule
// ========================================================================
void DB(int seq, int n) {
        int i, new_bit, a[MAX];

        for (i = 1; i <= n; i++) a[i] = 0;  // First n bits
        do      {
                fprintf(dbf, "%d", a[1]);
                switch (seq)    {
                        case 1: new_bit = Granddaddy(a, n); break;
                        case 2: new_bit = Grandmama(a, n); break;
                        case 3: new_bit = PCR3(a, n); break;
                        case 4: new_bit = PCR4(a, n); break;
                        case 5: new_bit = CCR1(a, n); break;
                        case 6: new_bit = CCR2(a, n); break;
                        case 7: new_bit = CCR3(a, n); break;
                        default: break;   }
                for (i = 1; i <= n; i++) a[i] = a[i + 1];
                a[n] = new_bit;     }
        while (!Zeros(a, n));
}
```

```
// ========================================
int main( )  {
        int i, n;

        for (n = 5; n < 12; n++)  {
                switch (n) {
                        case 5: dbf = fopen("De_Bruijn_5.txt", "w");  break;
                        case 6: dbf = fopen("De_Bruijn_6.txt", "w");  break;
                        case 7: dbf = fopen("De_Bruijn_7.txt", "w");  break;
                        case 8: dbf = fopen("De_Bruijn_8.txt", "w");  break;
                        case 9: dbf = fopen("De_Bruijn_9.txt", "w");  break;
                        case 10: dbf = fopen("De_Bruijn_10.txt", "w");  break;
                        case 11: dbf = fopen("De_Bruijn_11.txt", "w");  break;
                       default: break;
                }
                for (i = 1; i <= 7; i++)  {
                        DB(i, n);
                        fprintf(dbf, "\n");
                }
        }
}
```

B.2 Πρόγραμμα Stamp-Martin

Πρόκειται για πρόγραμμα το οποίο δέχεται ακολουθίες de Bruijn περιόδου (μεγέθους) 2^n για $5 \leq n \leq 6$ από το αρχείο De_Bruijn_**X**.txt (X=5,6). Στην συνέχεια, για κάθε μία από τις ακολουθίες, υπολογίζει, χρησιμοποιώτας τον αλγόριθμο Stamp-Martin, την τιμή της k σφάλματων Γραμμικής Πολυπλοκότητας ck(s) για k=0, 1, 2, …, έως ότου ck(s) = 0. Τέλος, αποθηκεύει τα αποτελέσματα στα αρχεία StampMartin_**X**.csv.

Στο ακόλουθο πρόγραμμα, χάριν οικονομίας, δεν γίνεται έλεγχος ύπαρξης του αρχείου που χρησιμοποιείται για ανάγνωση δεδομένων, De_Bruijn_**X**.txt. Για την ομαλή εκτέλεση του προγράμματος θα πρέπει τα προαναφερθέντα αρχεία να βρίσκονται στον ίδιο φάκελο (Folder) με την εκτελέσιμη μορφή του.

Το πρόγραμμα δημιουργήθηκε για τις ανάγκες της έρευνας την 30/3/2019 από τον Κωνσταντίνο Ρόζη, ερευνητή της παρούσας διατριβής.

```
/* Program computes the Linear Complexity of k-error sequences  for k=0,1,…
   using the Stamp Martin algorithm.
   K. Rozis  30 March 2019  */

#include <iostream>
#include <fstream>
#include <stdio.h>

//
#define N 32
#define FileName "De_Bruijn_5.txt"
#define StampMartinExcelFile "StampMartin_5.csv"
using namespace std;

// Files
FILE * DBInputFile;
ofstream ExclFile;

// Arrays
int DBSequ[7][N];          // Initial De Bruijn Sequences
int sDB[N];                // Examine De Bruijn Sequences
int Results[7][(N/2)+1];   // Results Ck(s)=f(k)

//  Functions
```

```
void GetDBSequence( );
int StampMartin(int a[ ], int k);
int Min(int A, int B);
void SaveResults( );

// Main
int main( ) {
        int l, k, ck;

        GetDBSequence( ); // Routine for input the De Bruijn sequences, returns DBSequ[7][N]
        for (int j = 0; j < 7; j++) {
                k = 0;
                do {
                        ck = StampMartin(sDB, k); // Compute LC of k-error seq. for k=0,1,…
                        Results[j][k]=ck;
                        k++;
                } while (ck > 0); }
        SaveResults( );
        system("pause");
        return 0;
}

// Input the De Bruijn sequences, from De_Bruijn_X.txt file. Returns DBSequ[7][N]
void GetDBSequence( ) {
        int si, sj, sc;

        si = 0; sj = 0;
        DBInputFile = fopen(FileName, "r");
        do {
                sc = fgetc(DBInputFile);
                if (sc == 10) { // End Of Line
                        si++; sj = 0; }
                else {
                        if (sc == '1') DBSequ[si][sj] = 1;
                        if (sc == '0') DBSequ[si][sj] = 0;
                        sj++; }
        } while (sc != EOF);
        fclose(DBInputFile);
}

// StampMartin Algorithm
// Compute the k-error linear complexity
// Inputs sDB[N] and k. Output Ck(s)
int StampMartin(int s[ ], int ke) {
        int a[N], b[N], L[N], R[N], cost[N];
        int l, c, T;
        // initial values for Martin Stamp
        c = 0; l = N; T = 0;
        for (int i = 0; i < l; i++) {
                a[i] = s[i];
                cost[i] = 1; }
```

```
        // for k-error ke compute linear complexity c
        while (l>1) {
                l = l / 2;  T = 0;
                for (int i = 0; i < l; i++)   {
                        L[i] = a[i];
                        R[i] = a[i + l];
                        b[i] = L[i] ^ R[i];
                        T = T + b[i] * Min(cost[i], cost[i + l]); }
                if (T <= ke)       {
                        ke = ke - T;
                        for (int i = 0; i < l; i++)   {
                                if (b[i] == 1)     {
                                        if (cost[i] <= cost[i + l]) {
                                                L[i] = R[i];
                                                cost[i] = cost[i + l] - cost[i]; }
                                        else      cost[i] = cost[i] - cost[i + l]; }
                                else cost[i] = cost[i] + cost[i + l];  // (b[i] == 0)
                                a[i] = L[i]; } }
                else   {   //  (T > ke)
                        c = c + l;
                        for (int i = 0; i < l; i++)   {
                                a[i] = b[i];
                                cost[i] = Min(cost[i], cost[i + l]);
                        } } }
        if (a[0] == 1 && cost[0] > ke) c++;
        return c;
}

int Min(int A, int B) {
        if (A > B) return B;
        else return A;
}

// Save Results to Excel file
void SaveResults( ) {
        ExclFile.open(StampMartinExcelFile);
        ExclFile << "De Bruijn Sequence Size " << N << endl;
        ExclFile << " ," ;
        for (int i=0;i<7;i++)       ExclFile << "No "<<i+1<<"," ;
         ExclFile << endl<< "k ," ;
         for (int i=0;i<7;i++)       ExclFile << " Ck(s) ," ;
         ExclFile << endl ;
         for (int k=0; k<(N/2)+1;k++) {
                ExclFile <<k<<",";
                for (int j=0;j<7;j++)   ExclFile <<Results[j][k]<<",";
                ExclFile <<endl;     }
         ExclFile.close( );
}
```

B.3 Πρόγραμμα Lauder-Paterson

Πρόκειται για πρόγραμμα που, χρησιμοποιώντας τον αλγόριθμο Lauder–Paterson , υπολογίζει τα κρίσιμα σημεία (critical points) των ακολουθιών de Bruijn που εξετάζονται (εισάγονται από το αρχείο «**De_Bruijn_X.txt**» με X=5,6, …, 11) το οποίο βασίζεται στη συνάρτηση CELCS. Τα αποτελέσματα, μέσω της ρουτίνας **SaveResults**, σώζονται σε αρχείο τύπου Excel με ονομασία : «**LauderPaterson_X.csv**». Παράλληλα ελέγχει σε κάθε κρίσιμο σημείο εάν η τιμή $C_k(s)$ πέσει κάτω από τα όρια ελέγχου.

Στο ακόλουθο πρόγραμμα, χάριν οικονομίας, δεν γίνεται έλεγχος ύπαρξης του αρχείου που χρησιμοποιείται για ανάγνωση δεδομένων, De_Bruijn_**X**.txt. Για την ομαλή εκτέλεση του προγράμματος θα πρέπει τα προαναφερθέντα αρχεία να βρίσκονται στον ίδιο φάκελο (Folder) με την εκτελέσιμη μορφή του.

Το πρόγραμμα τροποποιήθηκε και διαμορφώθηκε για τις ανάγκες της έρευνας την 14/4/2019 από τον Κωνσταντίνο Ρόζη, ερευνητή της παρούσας διατριβής.

```
/* program to compute CELCS of costed binary sequences */
#include <stdio.h>
#include <fstream>

#define n 7        //  Length in power of 2
#define N 128   //    N is the period of the input sequence
#define H 12     //    H is the number of sequences
#define FileName "De_Bruijn_7.txt"    //  Input de Bruijn sequences
#define LauderPatersonExcelFile "LauderPaterson_7.csv" //   Output Results to Excel
using namespace std;

FILE * DBInputFile;
ofstream ExclFile;

void GetDBSequence( );
void SaveResults( );
void celcs(int *s, int *cost, int l, int tsf, int lim, int c);
void CheckResults(int k, int c);
int min(int a, int b);

int DBSequ[H][N];
int Results[H][2][(N / 2)];
int i, j, k, m;
```

```
int main( ) {
        int s[N];
        int cost[N];
        printf("Sequences Length N=%d (n=%d) \n", N, n);  // Print Bound Header
        GetDBSequence( );        //    Input the initial sequence of N bits
        for (j = 0; j < H; j++) {
                /*  setting s sequence for examine and all costs to 1 */
                for (i = 0; i< N; i++)  {
                        s[i] = DBSequ[j][i];
                        cost[i] = 1;  }
                m = 0;  // counter for Results
                printf("\nSeq No %d ", j + 1);          // Print Bound Header
                celcs(s, cost, N, 0, N, 0);    //  Now run the celcs algorithm
        }  // next j
        SaveResults( );  // Save to Excel
        system("pause");
        return 0;
}

void celcs(int *s, int *cost, int l, int tsf, int lim, int c)
{
        int i;
        int L[N], R[N], B[N];
        int Lcost[N], Bcost[N];
        int T = 0;

        if (l >1)  {

                /* calculate B(S) and L(S) */
                for (i = 0; i < (l / 2); i++) {
                        L[i] = s[i];
                        R[i] = s[i + (l / 2)];
                        B[i] = L[i] ^ R[i]; }

                /* calculate costs for B and L, and calculate T */
                for (i = 0; i < (l / 2); i++) {
                        Bcost[i] = min(cost[i], cost[i + (l / 2)]);
                        T += B[i] * Bcost[i];  }

                for (i = 0; i < (l / 2); i++) {
                        if (B[i] == 1)  {
                                if (cost[i] <= cost[i + (l / 2)])      {
                                        L[i] = R[i];
                                        Lcost[i] = cost[i + (l / 2)] - cost[i];  }
                                else      Lcost[i] = cost[i] - cost[i + (l / 2)];
                        }
                        else   Lcost[i] = cost[i] + cost[i + (l / 2)];
                }  // next i
```

```
                /* the main decision point in the algorithm */
                if (T > 0)   celcs(B, Bcost, l / 2, tsf, tsf + T - 1, c + (l / 2));
                if (tsf + T <= lim)  celcs(L, Lcost, l / 2, tsf + T, lim, c);

        }

        else {   // the case l=1
                /*  Compute the Critical Point*/
                if (s[0] == 0){
                        Results[j][0][m] = tsf;
                        Results[j][1][m] = c;
                         CheckResults(tsf, c);
                        m++;    }

                if ((s[0] == 1) && (cost[0] > 0)) {
                        Results[j][0][m] = tsf;
                        Results[j][1][m] = c+1;
                        CheckResults(tsf, c + 1);
                        m++;    }

                if ((s[0] == 1) && (tsf + cost[0] <= lim)) {
                        Results[j][0][m] = tsf + cost[0];
                        Results[j][1][m] = c;
                        CheckResults(tsf + cost[0], c);
                        m++;    }
        }
        return;
}

int min(int a, int b)  {
        if (a < b)  return(a);
        else        return(b);
}

void GetDBSequence( )  {
        int si, sj, sc;

        si = 0; sj = 0;
        DBInputFile = fopen(FileName, "r");
        do {
                sc = fgetc(DBInputFile);
                if (sc == 10)  {   // End Of Line
                        si++; sj = 0;       }
                else {
                        if (sc == '1') DBSequ[si][sj] = 1;
                        if (sc == '0') DBSequ[si][sj] = 0;
                        sj++;     }
        } while (sc != EOF);
        fclose(DBInputFile);
}
```

```
void SaveResults( ) {
        bool KeepGoing=true;
        bool LookSep;

        ExclFile.open(LauderPatersonExcelFile);
        /*  Excel Header */
        ExclFile << "De Bruijn sequnces size " << N << endl;
        for (int j = 0; j < H; j++)    ExclFile << "No " << (j + 1) << ",";
        ExclFile << endl;
        /* Critical Points */
        m = 0;
        do {
                KeepGoing = false;
                for (int j = 0; j < H; j++){
                        LookSep = true;
                        if (Results[j][0][m] == 0 && Results[j][1][m] == 0){
                                LookSep = false;
                                ExclFile << ",";  }
                        else{
                                ExclFile << "<" << Results[j][0][m] << "-" ;
                                ExclFile << Results[j][1][m] << ">,";  }
                        KeepGoing = KeepGoing | LookSep;
                }  // next j
                m++;
                ExclFile << endl;
        } while (KeepGoing);

        ExclFile.close( );
}

void CheckResults(int k, int c){
        int Pointer=2*c+k;

        if (Pointer < N) printf("*");
        if (c <= N)
                printf(DBOutputFile, "Ck=%d   k=%d   P =%d\n", c, k, Pointer);
        return;
}
```

Β.4 Πρόγραμμα Παραδείγματος Επίθεσης

Παρακάτω αποτυπώνονται οι πηγαίοι κώδικες προγραμμάτων, γραμμένα σε γλώσσα προγραμματισμού C++, τα οποία χρησιμοποιήθηκαν για την πραγματοποίηση επίθεσης γνωστού κειμένου. Και στα τρία προγράμματα, χάριν οικονομίας, δεν γίνεται έλεγχος ύπαρξης των αρχείων που χρησιμοποιούνται για ανάγνωση δεδομένων.

Οι πηγαίοι κώδικες των παρακάτω προγραμμάτων γράφηκαν από τον ερευνητή Κωνσταντίνο Ρόζη, στις 10 και 11 Απριλίου 2019, για τις ανάγκες της παρούσας μεταπτυχιακής διατριβής.

Β.4.1 Κωδικοποίηση και Αποκωδικοποίηση Κειμένου Παραδείγματος

Ο πηγαίος κώδικας σε γλώσσα προγραμματισμού C++ του προγράμματος που ακολουθεί, δέχεται ως είσοδο αρχείο με το κείμενο προς αναμετάδοση «Message.txt». Αρχικά, αφού το μετατρέψει σε δυαδική μορφή και αποθηκεύσει αυτή τη ροή δυαδικών ψηφίων του αρχικού μηνύματος στο αρχείο «OriginalPlainStream.txt», το κωδικοποιεί με την κλειδοροή, η οποία παράγεται από τον LFSR. Η κλειδοροή αυτή αποθηκεύεται στο αρχείο «LFSRKeyStream.txt». Η συνάρτηση που προσομοιάζει την έξοδο του LFSR είναι η «LFSROutBit». Το αποτέλεσμα της κωδικοποίησης είναι το κρυπτοκείμενο, σε μορφή ροής δυαδικών ψηφίων, το οποίο αποθηκεύεται στο αρχείο «CipherStream.txt».

```
#include <iostream>
#include<stdio.h>
using namespace std;

void MakeBin(int d);
int BinToDec(int Bin[8]);
int LFSROutBit( );
void InitLFSR( );

FILE *PlainText;  // Input file with plain text
FILE *LFSRKey, *MsgStream, *Crypto; // Output files

int A[8]; // Array for convert Bin to Dec
int Reg[9] = { 1, 0, 1, 0, 1, 0, 1, 0, 1 };  // LFSR Registers with init values
int FeedBack[9] = { 1, 0, 0, 0, 0, 0, 0, 1, 0 };  // FeedBack Function
int MaxReg=9; // No of total LFSR Stages

int main( ) {
        // Input Text from File, Make it Bin,
        // Save the stream to file MsgStream
```

```
PlainText = fopen("Message.txt", "r");
MsgStream = fopen("OriginalPlainStream.txt", "w+");
int PlainChar;
cout << "Transmit The Folow Message :" << endl;
do {
        PlainChar = fgetc(PlainText); // Gets each character in Ascii Code
        if (PlainChar != EOF){
                MakeBin(PlainChar);  // Fills A[8] of Binary for each character
                cout << char(BinToDec(A));
                for (int f = 0; f <8; f++)   fprintf(MsgStream, "%d", A[f]);
        }
} while (PlainChar != EOF);
fclose(PlainText);
fclose(MsgStream);
cout << endl;

// Input Binary Bits of plain text from file MsgStream,
// Encode with LFSR output,
// Save to file Crypto and to file LFSRKey
MsgStream = fopen("OriginalPlainStream.txt", "r");
Crypto = fopen("CipherStream.txt", "w+");
LFSRKey = fopen("LFSRKeyStream.txt", "w+");
int FileBit,PlainBit,CryptoBit,LOutBit;
InitLFSR( );  // Reset the LFSR
do {
        FileBit = fgetc(MsgStream);
        if (FileBit == '0') PlainBit = 0;
        if (FileBit == '1') PlainBit = 1;
        LOutBit = LFSROutBit( );
        fprintf(LFSRKey, "%d", LOutBit);
        CryptoBit = PlainBit^LOutBit;
        fprintf(Crypto, "%d", CryptoBit);
} while (FileBit != EOF);
fclose(MsgStream);
fclose(Crypto);
fclose(LFSRKey);

// Input Binary Bits of CipherStream from file Crypto
// Input Binary Bits of KeyStream from file LFSRKey
// Decode with two filew, Show Results
Crypto = fopen("CipherStream.txt", "r");
LFSRKey = fopen("LFSRKeyStream.txt", "r");
int CipherBit, i=0;
int KeyStreamBit,KeyBit;
cout << "The Received  Message Using KeyStream File is:" << endl;
do {
        CipherBit = fgetc(Crypto);
        KeyStreamBit = fgetc(LFSRKey);
        //cout << PlainChar;
        if (CipherBit == '0') CryptoBit = 0;
        if (CipherBit == '1') CryptoBit = 1;
```

```
                if (KeyStreamBit == '0') KeyBit = 0;
                if (KeyStreamBit == '1') KeyBit = 1;
                A[i] = CryptoBit ^ KeyBit;
                i++;
                if (i == 8){
                        cout << char(BinToDec(A));
                        i = 0;     }
        } while (CipherBit != EOF);
        fclose(Crypto);
        fclose(LFSRKey);
        cout << endl;

        // Input Binary Bits of CipherStream from file Crypto
        // Decode with LFSR output, Show Results
        InitLFSR( ); // Reset the LFSR
        Crypto = fopen("CipherStream.txt", "r");
        i = 0;
        cout << "The Received Mesage is:" << endl;
        do {
                CipherBit = fgetc(Crypto);
                KeyStreamBit = fgetc(LFSRKey);
                if (CipherBit == '0') CryptoBit = 0;
                if (CipherBit == '1') CryptoBit = 1;
                A[i] = CryptoBit^LFSROutBit();
                i++;
                if (i == 8){
                        cout << char(BinToDec(A));
                        i = 0;     }
        } while (CipherBit != EOF);
        fclose(Crypto);
        fclose(LFSRKey);
        cout << endl;

        system("pause");
        return 0;
}

void MakeBin(int d)  {
        int i = 7;
        for (int j = 0; j<8; j++)  A[j] = 0;
        while (d>0) {
                A[i] = d % 2;
                d = d / 2;
                i - - ;      }
}

int BinToDec(int Bin[8]) {
        int Dec = 0;
        for (int i = 0; i < 8; i++)    Dec = Dec + Bin[7 - i] * pow(2, i);
        return Dec;
}
```

```
int LFSROutBit( ) {
        int OutBit, FeedBackBit = 0, Bit;
        OutBit = Reg[0];
        for (int i = 0; i < MaxReg; i++){
                Bit = FeedBack[i] * Reg[i];
                FeedBackBit = FeedBackBit ^ Bit;  }
        for (int i = 0; i < MaxReg - 1; i++)    Reg[i] = Reg[(i + 1)];
        Reg[MaxReg - 1] = FeedBackBit;
        return OutBit;
}

void InitLFSR( ){
        for (int i = 0; i < MaxReg; i++)  Reg[i] = (i + 1) % 2;
}
```

Β.4.2 Πρώτο Μέρος Επίθεσης Παραδείγματος

Ο πηγαίος κώδικας σε γλώσσα προγραμματισμού C++ του προγράμματος που ακολουθεί, δέχεται ως είσοδο αρχείο κειμένου «KnownText.txt», το οποίο περιέχει τους χαρακτήρες με τους οποίους ο επιτιθέμενος πιστεύει ότι ξεκινά το αρχικό κείμενο και το αρχείο κειμένου «CipherStream.txt», το οποίο περιέχει το κρυπτοκείμενο σε μορφή ροής δυαδικών ψηφίων, τα περιεχόμενα του οποίου υποκλάπηκαν από το δημόσιο κανάλι επικοινωνίας. Αρχικά μετατρέπονται οι χαρακτήρες του περιεχομένου του αρχείου «KnownText.txt» σε μορφή ροής δυαδικών ψηφίων, με βάση την Ascii τιμή του κάθε γράμματος. Το αποτέλεσμα αποθηκεύεται στο αρχείο «KnownStream.txt». Στην συνέχεια, για κάθε bit του περιεχομένου του αρχείου «KnownStream.txt» εκτελείται η πράξη $m' \oplus c_2$, το αποτέλεσμα της οποίας αποθηκεύεται στο αρχείο «FakeKeyStream.txt».

```
#include <iostream>
#include<stdio.h>

using namespace std;

void MakeBin(int d);
int BinToDec(int Bin[8]);
int LFSROutBit();
void InitLFSR();

FILE *KnownPlainText, *Crypto;  // Input file with Known text and Cipher text
FILE *FakeLFSRKey, *KnownMsgStream; // Output files
```

```
int A[8]; // Array for convert Bin to Dec

int main( ) {

	// Input KnownText from file, Make it Bin,
	// Save the stream to file KnownMsgStream
	KnownPlainText = fopen("KnownText.txt", "r");
	KnownMsgStream = fopen("KnownStream.txt", "w+");
	int KnownChar;

	do {
		KnownChar = fgetc(KnownPlainText); // Gets each character in Ascii Code
		if (KnownChar != EOF){
			MakeBin(KnownChar);  // Fills A[8] of Binary for each character
			for (int f = 0; f <8; f++)
				fprintf(KnownMsgStream, "%d", A[f]);
		}
	} while (KnownChar != EOF);
	fclose(KnownPlainText);
	fclose(KnownMsgStream);
	cout << endl;

	// Input Binary Bits of Known text from fileKnownMsgStream,
	// Input Binary Bits of Cipher text from Crypto ,
	// Add them to find the Fake LFSRKey
	// Save results to FakeKeyStream.txt
	KnownMsgStream = fopen("KnownStream.txt", "r");
	Crypto = fopen("CipherStream.txt", "r");
	FakeLFSRKey = fopen("FakeKeyStream.txt", "w+");
	int KnownBit, PlainBit, CryptoBit, CipherBit;
	//	InitLFSR();
	do {
		KnownBit = fgetc(KnownMsgStream);
		CipherBit = fgetc(Crypto);
		if (KnownBit != EOF) {
			if (KnownBit == '0') PlainBit = 0;
			if (KnownBit == '1') PlainBit = 1;
			if (CipherBit == '0') CryptoBit = 0;
			if (CipherBit == '1') CryptoBit = 1;
			fprintf(FakeLFSRKey, "%d", (PlainBit^CryptoBit));
		}
	} while (KnownBit != EOF);
	fclose(KnownMsgStream);
	fclose(Crypto);
	fclose(FakeLFSRKey);

	system("pause");
	return 0;
}
```

Β.4.3 Δεύτερο Μέρος Επίθεσης Παραδείγματος

Το πρόγραμμα δέχεται είσοδο το αρχείο κειμένου «CipherStream.txt», το οποίο περιέχει το κρυπτοκείμενο σε μορφή ροής δυαδικών ψηφίων, τα περιεχόμενα του οποίου υποκλάπηκαν από το δημόσιο κανάλι επικοινωνίας. Επίσης χρησιμοποιεί το αρχείο «FakeKeyStream.txt» για να δώσει αρχική τιμή στον LFSR. Στην συνέχεια αποκωδικοποιεί την ακολουθία του κρυπτοκειμένου με την κλειδοροή η οποία παράγεται από τον LFSR με τα χαρακτηριστικά που έλαβε από τον αλγόριθμο Berlekamp–Massey. Η διαδικασία γίνεται για κάθε bit της ακολουθίας του κρυπτοκειμένου εκτελώντας την πράξη $m_i' = k_i' \oplus c_i$. Η προσομοίωση του LFSR παράγεται μέσω της συνάρτησης «FakeLFSR». Το αποτέλεσμα αποθηκεύεται σε μορφή καθαρού κειμένου στο αρχείο «FakePlainText.txt».

```
#include <iostream>
#include<stdio.h>
using namespace std;

int BinToDec(int Bin[8]);
int FakeLFSR( );  // Simulation of LFSR
void SetInitLFSR( );  // Initialization LFSR

FILE *FakeLFSRKey, *Crypto;  // Input file with Plain and Cipher text
FILE *FakeMsgStream, *FakeMsgText; // Output files

int A[8]; // Array for convert Bin to Dec
int Reg[16];// LFSR Registers
int FeedBack[16] = { 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0 }; // FeedBack Function
int MaxReg = 16; // No of total LFSR Stages

int main( ) {
        // Input Binary Bits of CipherStream from file Crypto,
        // Decode with FakeLFSR output,
        // Save to file FakeMsgText
        int CryptoBit, CipherBit;
        int LOutBit, i = 0;
        Crypto = fopen("CipherStream.txt", "r");
        FakeMsgStream = fopen("FakePlainStream.txt", "w+");
        FakeMsgText = fopen("FakePlainText.txt", "w+");
        SetInitLFSR( );
        do {
                CipherBit = fgetc(Crypto);
                if (CipherBit == '0') CryptoBit = 0;
                if (CipherBit == '1') CryptoBit = 1;
                if (CipherBit != EOF) {
                        LOutBit = FakeLFSR( );
                        fprintf(FakeMsgStream, "%d", CryptoBit^LOutBit);
```

```
                        A[i] = CryptoBit^LOutBit;
                        i++;
                        if(i == 8){
                                cout << char(BinToDec(A));
                                fprintf(FakeMsgText, "%c", char(BinToDec(A)));
                                i = 0;
                        }
                }
        } while (CipherBit != EOF);
        fclose(FakeMsgStream);
        fclose(Crypto);
        fclose(FakeMsgText);
        cout << endl;

        system("pause");
        return 0;
}

int BinToDec(int Bin[8])  {
        int Dec = 0;
        for (int i = 0; i < 8; i++)
                Dec = Dec + Bin[7 - i] * pow(2, i);
        return Dec;
}

int FakeLFSR( ) {
        int OutBit, FeedBackBit = 0, Bit;
        OutBit = Reg[0];
        for (int i = 0; i < MaxReg; i++){
                Bit = FeedBack[i] * Reg[i];
                FeedBackBit = FeedBackBit ^ Bit;
        }
        for (int i = 0; i < MaxReg - 1; i++)
                Reg[i] = Reg[(i + 1)];
        Reg[MaxReg - 1] = FeedBackBit;
        return OutBit;
}

void SetInitLFSR(){
        FakeLFSRKey = fopen("FakeKeyStream.txt", "r");
        int FakeLFSRBit, FakeLFSRStreamBit;

        for (int i = 0; i < MaxReg; i++)    {
                FakeLFSRStreamBit = fgetc(FakeLFSRKey);
                if (FakeLFSRStreamBit == '0') FakeLFSRBit = 0;
                if (FakeLFSRStreamBit == '1') FakeLFSRBit = 1;
                Reg[i] = FakeLFSRBit;
        }
        fclose(FakeLFSRKey);
}
```