Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακή Διατριβή
στα Πληροφοριακά και Επικοινωνιακά Συστήματα

Τεχνικές παραγωγής ψευδοτυχαίων ακολουθιών: Ακολουθίες De Bruijn

Κοτανίδης Π. Δημήτριος

Επιβλέπων Καθηγητής
Δρ. Κωνσταντίνος Λιμνιώτης

Σεπτέμβριος 2015

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Τεχνικές παραγωγής ψευδοτυχαίων ακολουθιών: Ακολουθίες De Bruijn

Κοτανίδης Π. Δημήτριος

Επιβλέπων Καθηγητής
Δρ. Κωνσταντίνος Λιμνιώτης

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

μεταπτυχιακού τίτλου σπουδών
στα Πληροφοριακά Συστήματα

από τη Σχολή Θετικών και Εφαρμοσμένων Επιστημών
του Ανοικτού Πανεπιστημίου Κύπρου

Σεπτέμβριος 2015

Περίληψη

Οι συμμετρικοί κρυπτογραφικοί αλγόριθμοι βασίζουν την ασφάλειά τους, σε μεγάλο βαθμό, στα χαρακτηριστικά τυχαιότητας που εμφανίζουν οι υπεισερχόμενες ακολουθίες. Στην παρούσα διατριβή θα γίνει μία επισκόπηση των υπαρχουσών τεχνικών για την παραγωγή ψευδοτυχαίων ακολουθιών. Ακολούθως θα δοθεί έμφαση στους τρόπους παραγωγής των ακολουθιών από NLFSR, όπου με τη βοήθεια των πρόσφατων ερευνητικών αποτελεσμάτων θα επιχειρηθεί να υπάρξουν νέες κατασκευές για γεννήτριες ακολουθιών. Απώτερος στόχος είναι η θεμελίωση τεχνικών που να διασφαλίζουν την παραγωγή ακολουθιών με καλά χαρακτηριστικά τυχαιότητας.

Για την παραγωγή ψευδοτυχαίων ακολουθιών (δηλ. ακολουθιών που προσομοιάζουν, κατά το δυνατόν, τις απολύτως τυχαίες) υπάρχουν ήδη πολλές γνωστές τεχνικές. Η πλειοψηφία των τεχνικών αυτών στηρίζεται στη χρήση γραμμικών καταχωρητών ολίσθησης με ανάδραση (LFSR). Ωστόσο, τα τελευταία χρόνια είναι έντονο το ενδιαφέρον χρήσης μη γραμμικών καταχωρητών (NLFSRs) για την παραγωγή τέτοιων ακολουθιών, διότι εμφανίζουν ένα σύνολο σημαντικών πλεονεκτημάτων τέτοιων ώστε να αποτρέπονται γνωστές επιθέσεις. Παρόλα αυτά, δεν είναι ακόμα γνωστό πώς μπορεί να επιλεγεί ένας NLFSR που να παράγει ακολουθίες με εγγυημένα καλά κρυπτογραφικά χαρακτηριστικά (π.χ. μέγιστη περίοδο). Το πρόβλημα αυτό είναι στενά συνυφασμένο με την κατασκευή De Bruijn ακολουθιών, το οποίο ακόμα δεν έχει αντιμετωπιστεί πλήρως, και στο οποίο εστιάζει ιδίως η παρούσα διατριβή.

Η παρούσα εργασία αντιμετωπίζει ένα αντικείμενο με έντονο ερευνητικό ενδιαφέρον, διότι προάγει τις τεχνικές κατασκευής κρυπτογραφικών ακολουθιών, που με τη σειρά τους αποτελούν αναπόσπαστο συστατικό για την ανάπτυξη ισχυρών κρυπτογραφικών αλγορίθμων.

Στη διατριβή αυτή πραγματοποιείται αρχικά μία επισκόπηση των γνωστών κριτηρίων τυχαιότητας για τις κρυπτογραφικές ακολουθίες. Στη συνέχεια, παρουσιάζονται πρόσφατα ερευνητικά αποτελέσματα αναφορικά με τη χρήση μη γραμμικών καταχωρητών ολίσθησης με ανάδραση (NLFSR) για την παραγωγή ακολουθιών μεγίστης περιόδου (ακολουθίες De Bruijn). Ακολούθως, παρουσιάζεται για πρώτη φορά μία νέα κατεύθυνση για την κατασκευή ακολουθιών De Bruijn, χρησιμοποιώντας τη δομή των πινάκων επιθεμάτων (suffix arrays). Η ανάλυσή μας καταδεικνύει ότι είναι εφικτή η ανάπτυξη καινούριας συστηματικής μεθόδου για την κατασκευή αυτών των ακολουθιών.

Summary

The security of symmetric cryptographic algorithms is stongly contingent on the (pseudo)randomness of the cryptographic sequences (keystreams). In this thesis, current techniques for generating pseudorandom sequences are studied. Emphasis is given on generating sequences via NLFSRs, focusing on recent research results in the field. The ultimate goal is the development of new design techniques for generating sequences with nice cryptographic properties.

To produce pseudorandom sequences (ie. sequences that resemble , absolutely random) there are many known techniques. The majority of these techniques relies upon the use of Linear Feedback Shift Registers (LFSR). However, in recent years there has been an intense interest in using Non-linear Feedback Shift Registers (NLFSRs) to produce such sequences, because they seem to prevent known attacks. However, it is not yet known how to select a NLFSR that produces sequences with guaranteed good cryptographic features (such as as maximum period). This problem is closely related to the construction of De Bruijn sequences, which still has not been fully addressed.

This thesis copes with an object of intense research interest, since it promotes the construction techniques of cryptographic sequences, which in turn form an integral component for the development of strong cryptographic algorithms.

This thesis studies the known criteria of randomness for cryptographic sequences. Afterward, recent reseach work regarding the use of Non-linear Feedback Shift Registers to produce maximal period sequences (De Bruijn sequences) is presented. Subsequently, we present a new method to generate such sequences, based on suffix arrays. Our analysis shows that this new approach suffices to ensure the generation of De Bruijn sequences in a systematic way.

Ευχαριστίες

Ως ελάχιστη ένδειξη της βαθιάς μου ευγνωμοσύνης, αισθάνομαι μέγιστη υποχρέωση μου να εκφράσω τις θερμές μου ευχαριστίες, σε όλους όσους συνέβαλαν έστω και ελάχιστα, άμεσα ή έμμεσα, στην ολοκλήρωση της παρούσας μεταπτυχιακής διατριβής.

Πρώτιστα στον επιβλέποντα Καθηγητή μου κ. Λιμνιώτη Κωσταντίνο, που αφιέρωσε άπλετο χρόνο και έδινε πάντα εύστοχες υποδείξεις και επισημάνσεις. Ο σπάνιος συνδυασμός γνώσεων και εμπειριών που τον διακρίνουν, διαδραμάτισαν καίριο και καθοριστικό ρόλο στην διαδικασία εκπόνησης της μεταπτυχιακής μου διατριβής και στην επιλογή ενός τόσου ωραίου θέματος διατριβής, που ενέπνευσε τις επιλογές μου.

Ιδιαίτερα, αισθάνομαι την ανάγκη να αποτίσω έναν ελάχιστο φόρο τιμής στην μάνα μου - Ζωή, τον πατέρα μου - Παναγιώτη και τα αδέλφια μου Μαγδαληνή και Νικόλαο, που με στήριξαν όλο αυτό το διάστημα.

Συγκεκριμένα όμως θέλω να ευχαριστήσω την γυναίκα μου Ευφροσύνη Παπαντωνίου, που κατά τη διάρκεια της διατριβής μου χάρισε το πολυτιμότερο δώρο της φύσης, τον γιό μου Παναγιώτη, ο οποίος μου έδωσε δύναμη να ολοκληρώσω τη διατριβή μου και να έχω πλέον περισσότερο χρόνο, ώστε να ασχολούμαι αδιάκοπα μαζί του.

...στο νεογέννητο γιό μου
Παναγιώτη και στην μνήμη της
γιαγιάς μου Μαγδαληνής

Περιεχόμενα

Κεφάλαιο 1
Εισαγωγή

Η ραγδαία αύξηση στην καθημερινή χρήση των επικοινωνιών οποιουδήποτε είδους (διαδικτυακών, ασύρματων κτλ.), που ιδιαίτερα στα τελευταία χρόνια συντελείται με ακόμα πιο γοργούς ρυθμούς, είναι στενά συνυφασμένη με τις νέες εξελίξεις στις τεχνολογίες των επικοινωνιών, οι οποίες με τη σειρά τους μας φέρουν αντιμέτωπους με πλήθος νέων προβλημάτων ασφάλειας στα Πληροφοριακά και Επικοινωνιακά Συστήματα. Ως εκ τούτου, τόσο ο κλάδος των επιχειρήσεων όσο και ο ακαδημαϊκός χώρος έχουν στραμμένες τις προσπάθειές τους στην παροχή ασφάλειας και αξιοπιστίας στις επικοινωνίες, μέσα σε αυτό το διαρκώς αυξανόμενο πλήθος παρεχόμενων υπηρεσιών, χωρίς να θυσιάζεται η απόδοση. Ειδικότερα, για την επίτευξη της ασφάλειας είναι κρίσιμη η «οχύρωση» των δεδομένων, απέναντι σε υποκλοπές, αλλοιώσεις και κάθε άλλου τύπου κακόβουλες επιθέσεις.

Οι βασικές τεχνικές που χρησιμοποιούνται για τη διασφάλιση των δεδομένων στα Πληροφοριακά Συστήματα είναι οι κρυπτογραφικοί αλγόριθμοι ή γενικά τα κρυπτογραφικά συστήματα. Κατά την ανάπτυξη κρυπτογραφικών αλγορίθμων, απώτερος στόχος ιδανικά παραμένει η ανακάλυψη ενός τέλειου κρυπτογραφικού συστήματος, που θα είναι αδιάβλητο και απόλυτα ασφαλές. Ο Claude Shannon, ως θεμελιωτής της Θεωρίας της Πληροφορίας, ήταν ο πρώτος που περιέγραψε τις βασικές ιδιότητες που πρέπει να φέρει ένα τέλειο κρυπτογραφικό

σύστημα, για το οποίο είχε αναφέρει με απλά λόγια ότι: *«Η γνώση του κρυπτοκειμένου δεν αποκαλύπτει σε καμία περίπτωση για το ποιο μπορεί να είναι το αρχικό μήνυμα».* Ως κρυπτοκείμενο χαρακτηρίζουμε τα μετασχηματισμένα δεδομένα, τα οποία «ταξιδεύουν» στο μέσο επικοινωνίας και αποστέλλονται από ένα Πληροφοριακό Σύστημα σε ένα άλλο, έτσι ώστε να «κρύβεται» η υποκείμενη γνήσια πληροφορία κατά τρόπο τέτοιο ώστε όποιος τρίτος αποκτά πρόσβαση στα μετασχηματισμένα (κρυπτογραφημένα) αυτά δεδομένα να μην μπορεί να ανακτήσει τα γνήσια. Ο Shannon δεν πρότεινε ένα σύστημα τέλειας μυστικότητας το οποίο να μπορεί πρακτικά να υλοποιηθεί, περιέγραψε όμως βασικές ιδιότητες που πρέπει να έχουν τα κρυπτογραφικά συστήματα, όπως:

Διάχυση (Diffusion): κάθε ψηφίο (bit) του αρχικού μηνύματος, πρέπει να επηρεάζει όσο γίνεται περισσότερα ψηφία του κρυπτοκειμένου.

Σύγχυση (Confusion): Η σχέση μεταξύ του κρυπτοκειμένου και του μυστικού κλειδιού πρέπει να είναι σύνθετη, έτσι ώστε ακόμη και από τα στατιστικά χαρακτηριστικά του κρυπτοκειμένου να μην είναι εφικτή η ανάκτηση του κλειδιού λόγω ακριβώς του σύνθετου τρόπου με τον οποίο επέδρασσε το κλειδί κατά την παραγωγή του κρυπτοκειμένου.

Το μυστικό κλειδί που αναφέρεται ανωτέρω δεν είναι τίποτα άλλο παρά μία μυστική ποσότητα, που γνωρίζουν μόνο τα δύο «άκρα» της επικοινωνίας (οι «συνδιαλεγομένοι»), και η οποία καθορίζει το αποτέλεσμα της κρυπτογράφησης: αν δύο ίδια μηνύματα κρυπτογραφηθούν δύο διαφορετικές φορές με τον ίδιο κρυπτογραφικό αλγόριθμο αλλά με διαφορετικό κλειδί κρυπτογράφησης, θα παραχθούν δύο διαφορετικά κρυπτοκείμενα. Με άλλα λόγια, οι αλγόριθμοι κρυπτογράφησης και αποκρυπτογράφησης μπορούν να θεωρηθούν ως συναρτήσεις δύο μεταβλητών, όπου η μία είναι το κλειδί και η άλλη το αρχικό μήνυμα ή το κρυπτοκείμενο αντίστοιχα (βλ. Κεφάλαιο 2).

Οι ανωτέρω θεμελιώδεις αρχές του Shannon εφαρμόζονται σήμερα στην πράξη, αφού είναι απαραίτητο συστατικό για την κατασκευή κρυπτογραφικών αλγόριθμων και ο ακρογωνιαίος λίθος στη σχεδίαση ισχυρών κρυπτογραφικών συστημάτων με στόχο την διασφάλιση των δεδομένων στα Πληροφοριακά Συστήματα.

Όμως ας δούμε τι εννοούμε με τον όρο ασφάλεια;

Η ασφάλεια των κρυπτογραφικών αλγορίθμων αποτιμάται με δύο έννοιες:

Απεριόριστη ασφάλεια, που χαρακτηρίζει ένα κρυπτογραφικό σύστημα, το οποίο ανεξάρτητα από το πόσο μεγάλο τμήμα του κρυπτοκειμένου είναι γνωστό, δεν υπάρχει αρκετή πληροφορία για ανάκτηση του αρχικού μηνύματος ακόμα και με άπειρη υπολογιστή ισχύ. Ένα σύστημα επιτυγχάνει την τέλεια μυστικότητα κατά Shannon αν και μόνο αν είναι απεριόριστα ασφαλές.

Υπολογιστική ασφάλεια, χαρακτηρίζει ένα κρυπτογραφικό σύστημα, το οποίο είναι υπολογιστικά αδύνατο να παρουσιάσει ευπάθεια, με απλά λόγια να «σπάσει». Έτσι, μπορούμε να με βεβαιότητα να πούμε ότι είναι αδύνατον με τους υπάρχοντες υπολογιστικούς πόρους να ανακτήσει ένας υποκλοπέας το αρχικό μήνυμα, αν γνωρίζει το κρυπτοκείμενο.

Δεδομένου ότι στην πράξη δεν μπορούμε να πετύχουμε απεριόριστη ασφάλεια, θέτουμε ως στόχο στα σύγχρονα κρυπτογραφικά συστήματα την επίτευξη της υπολογιστικής ασφάλειας. Βασικός πυλώνας για την ασφάλεια των κρυπτογραφικών αλγορίθμων είναι η μυστικότητα του κλειδιού, υπό την έννοια ότι δεν πρέπει να είναι προβλέψιμο: κατά συνέπεια, υπάρχει ακόμα και σήμερα έντονο ερευνητικό ενδιαφέρον για την παραγωγή ακολουθιών με καλά χαρακτηριστικά τυχαιότητας, τέτοιων ώστε να μπορούν να χρησιμοποιηθούν εν είδη μυστικού κλειδιού σε κρυπτογραφικές εφαρμογές.

1.1 Σκοπός έρευνας

Η παρούσα διπλωματική διατριβή πραγματεύεται τις υπάρχουσες τεχνικές παραγωγής ψευδοτυχαίων ακολουθιών, στις οποίες βασίζουν την ασφάλειά τους οι κρυπτογραφικοί αλγόριθμοι και τα σύγχρονα κρυπτογραφικά συστήματα. Μελετώνται οι βασικότερες εξ αυτών, αναλύοντας τις πιο χαρακτηριστικές τεχνικές παραγωγής ψευδοτυχαίων ακολουθιών που διασφαλίζουν την παραγωγή ακολουθιών με καλά χαρακτηριστικά τυχαιότητας, τα οποία και περιγράφονται εκτενώς. Έμφαση δίνεται σε μια ειδική κατηγορία ακολουθιών, τις λεγόμενες ακολουθίες De Bruijn, οι οποίες παρουσιάζουν ξεχωριστό ενδιαφέρον – με εφαρμογές όχι μόνο στην κρυπτογραφία – όπου περιγράφεται η σπουδαιότητά τους, οι γνωστές μέθοδοι κατασκευής τους καθώς τα ανοιχτά σχετικά ερευνητικά προβλήματα: στο πλαίσιο αυτό, στην παρούσα εργασία παρουσιάζεται μία νέα προσέγγιση, προκειμένου να θεμελιώσουμε καινούριες τεχνικές παραγωγής ψευδοτυχαίων ακολουθιών.

1.2 Βασικά ερευνητικά ερωτήματα

Για την παραγωγή ψευδοτυχαίων ακολουθιών δηλαδή ακολουθιών που προσομοιάζουν, κατά το δυνατόν, τις απολύτως τυχαίες, υπάρχουν ήδη πολλές γνωστές τεχνικές. Η πλειοψηφία των τεχνικών αυτών στηρίζεται στη χρήση γραμμικών καταχωρητών ολίσθησης με ανάδραση (LFSR). Ωστόσο, τα τελευταία χρόνια είναι έντονο το ενδιαφέρον χρήσης μη γραμμικών καταχωρητών (NLFSR) για την παραγωγή τέτοιων ακολουθιών, διότι εμφανίζουν ένα σύνολο σημαντικών πλεονεκτημάτων τέτοιων ώστε να αποτρέπονται γνωστές επιθέσεις. Ωστόσο, δεν είναι ακόμα γνωστό πώς μπορεί να επιλεγεί ένας NLFSR που να παράγει ακολουθίες με εγγυημένα καλά κρυπτογραφικά χαρακτηριστικά – για παράδειγμα, δε γνωρίζουμε τεχνικές κατασκευής NLFSR που να παράγουν ακολουθίες με τη μέγιστη δυνατή περίοδο. Το πρόβλημα αυτό είναι στενά συνυφασμένο με την κατασκευή De Bruijn ακολουθιών, οι οποίες παράγονται με τη χρήση μη γραμμικών καταχωρητών ολίσθησης με ανάδραση NFLSR με βασικό χαρακτηριστικό την παραγωγή ακολουθιών με μέγιστη δυνατή περίοδο.

Οι ακολουθίες De Bruijn, λόγω ξεχωριστού ενδιαφέροντος που παρουσιάζουν σε πληθώρα εφαρμογών, έχουν μελετηθεί εκτενώς επί δεκαετίες. Παρόλα αυτά, πολλά ερευνητικά ζητήματα παραμένουν ανοιχτά, με κύριο την εύρεση μεθόδων κατασκευής τους – ιδανικά, απώτερος στόχος είναι η εύρεση τεχνικής που εγγυημένα να μπορεί να οδηγήσει στην κατασκευή όλων των ακολουθιών De Bruijn δοθείσης περιόδου.

1.3 Στόχος της έρευνας

Στόχος της παρούσας διατριβής είναι η καταγραφή και συγκριτική μελέτη των πλέον συχνά χρησιμοποιούμενων κατασκευών ψευδοτυχαίων ακολουθιών, των πρόσφατων ερευνητικών αποτελεσμάτων αναφορικά με τη κατασκευή NLFSR που παράγουν εγγυημένα ακολουθίες De Bruijn, με απώτερη επιδίωξη την ανάπτυξη νέων μεθόδων κατασκευής τέτοιων ακολουθιών χρησιμοποιώντας τους λεγόμενους *πίνακες επιθεμάτων* (*suffix arrays)*. Οι πίνακες αυτοί αποτελούν μία γνωστή δομή για τη δεικτοδότηση ακολουθιών κατά τρόπο τέτοιο ώστε να τις περιγράφουν επαρκώς, ούτως ώστε μέσω αυτής της δομής να μπορούν να απαντηθούν ερωτήματα επί ιδιοτήτων της ακολουθίας (π.χ. ποιο τμήμα αυτής εμφανίζεται περισσότερο από μία φορά κτλ.). Δεδομένων συγκεκριμένων ιδιοτήτων των ακολουθιών De Bruijn, οι οποίες «αντανακλώνται» στον πίνακα επιθεμάτων όπως καταδεικνύουμε στην παρούσα διατριβή, επιχειρούμε την κατασκευή κατάλληλων τέτοιων πινάκων οι οποίοι να αντιστοιχούν σε ακολουθίες De Bruijn. Με αυτόν τον τρόπο, στην παρούσα διατριβή αποτυπώνεται για πρώτη φορά η δυνατότητα παραγωγής ακολουθιών De Bruijn μέσω χρήσης πινάκων επιθεμάτων –

ακριβώς τη στιγμή που η ερευνητική κοινότητα δείχνει να ξεκινά τη διερεύνηση συσχετίσεων μεταξύ παρεμφερών εννοιών [02].

1.4 Δομή της Μεταπτυχιακής Διατριβής

Η δομή της διατριβής οργανώθηκε στα παρακάτω κεφάλαια:

1ο Κεφάλαιο: Παρουσιάζεται το αντικείμενο μελέτης της διατριβής, ο σκοπός, τα ερευνητικά ερωτήματα της έρευνας, η αναγκαιότητα και σπουδαιότητα της έρευνας και η δομή της μεταπτυχιακής διατριβής.

2ο Κεφάλαιο: Παρουσιάζεται το θεωρητικό πλαίσιο της κρυπτογραφίας και των κρυπτογραφικών αλγόριθμων, κάνοντας μία ανασκόπηση της υπάρχουσας βιβλιογραφίας.

3ο Κεφάλαιο: Παρουσιάζεται το θεωρητικό υπόβαθρο των ψευδοτυχαίων ακολουθιών, των γεννητριών παραγωγής ψευδοτυχαίων ακολουθιών, με έμφαση στις γεννήτριες που βασίζονται σε γραμμικούς καταχωρητές ολίσθησης με ανάδραση (LFSR). Επίσης περιγράφονται και οι μη γραμμικοί καταχωρητές ολίσθησης (NLFSR), οι οποίοι γενικά έχουν μελετηθεί πολύ λιγότερο στη διεθνή βιβλιογραφία από ό,τι οι LFSR, κάνοντας μία ανασκόπηση της υπάρχουσας βιβλιογραφίας, ώστε να γίνει κατανοητό γιατί τελικά επιθυμούμε ακολουθίες μεγίστης περιόδου (De Bruijn) από μη γραμμικά συστήματα.

4ο Κεφάλαιο: Παρουσιάζεται η μεθοδολογία της έρευνας, επεξηγώντας τις έννοιες των De Bruijn ακολουθιών και των NLFSR που παράγουν δυαδικές De Bruijn ακολουθίες, κάνοντας μία ανασκόπηση της υπάρχουσας βιβλιογραφίας και των πλέον σύγχρονων επιστημονικών αποτελεσμάτων.

5ο Κεφάλαιο: Επεξηγούνται σαφώς οι έννοιες των δέντρων επιθεμάτων (suffix trees) και πινάκων επιθεμάτων (suffix arrays), έννοιες που αποτελούν εργαλεία για την έρευνα και την συλλογή των δεδομένων.

6ο Κεφάλαιο: Παρουσιάζεται ο σχεδιασμός, οι μέθοδοι και τα μέσα συλλογής δεδομένων της έρευνάς μας. Επίσης, παρουσιάζονται τα ευρήματα της έρευνας (περιγραφή της νέας μεθόδου κατασκευής ακολουθιών De Bruijn) και γίνεται εφαρμογή των αποτελεσμάτων της έρευνας (σχετικά παραδείγματα) προσεγγίζοντας νέες λύσεις και τέλος γίνεται μια αποτίμηση των αποτελεσμάτων.

7ο Κεφάλαιο: Σε αυτό το κεφάλαιο γίνεται μία ανασκόπηση της διατριβής, παρατίθενται τα συμπεράσματα της έρευνας και προτάσεις για μελλοντική έρευνα.

Κεφάλαιο 2
Κρυπτογραφικοί αλγόριθμοι

2.1 Εισαγωγή

Η κρυπτογραφία μελετά τεχνικές με τις οποίες ένα μήνυμα μπορεί να μετασχηματιστεί σε μία ακατάληπτη μορφή, με κύριο στόχο την εμπιστευτικότητα του μηνύματος, έτσι ώστε ακόμα και αν υποκλαπεί από κάποιον, να μη μπορεί να το αναγνώσει. Η διαδικασία μετατροπής ενός μηνύματος σε ακατάληπτη μορφή ονομάζεται κρυπτογράφηση, ενώ η αντίστροφη διαδικασία ονομάζεται αποκρυπτογράφηση. Σήμερα, ο όρος «κρυπτογραφία» είναι πολύ πιο ευρύς: Με την κρυπτογραφία αναφερόμαστε στη μελέτη μαθηματικών τεχνικών που στοχεύουν στην διασφάλιση διαφορών ζητημάτων που άπτονται της ασφάλειας της πληροφορίας, όπως:

- Εμπιστευτικότητα (confidentiality)
- Πιστοποίηση ταυτότητας του αποστολέα (authentication)
- Διασφάλιση του αδιάβλητου (ακεραιότητας) της πληροφορίας (integrity)

Στην πράξη συναντώνται δύο μεγάλες οικογένειες κρυπτογραφικών αλγορίθμων, όπως θα δούμε παρακάτω στο κεφάλαιο. Αυτές οι οικογένειες χωρίζονται σε επιμέρους κατηγορίες και η

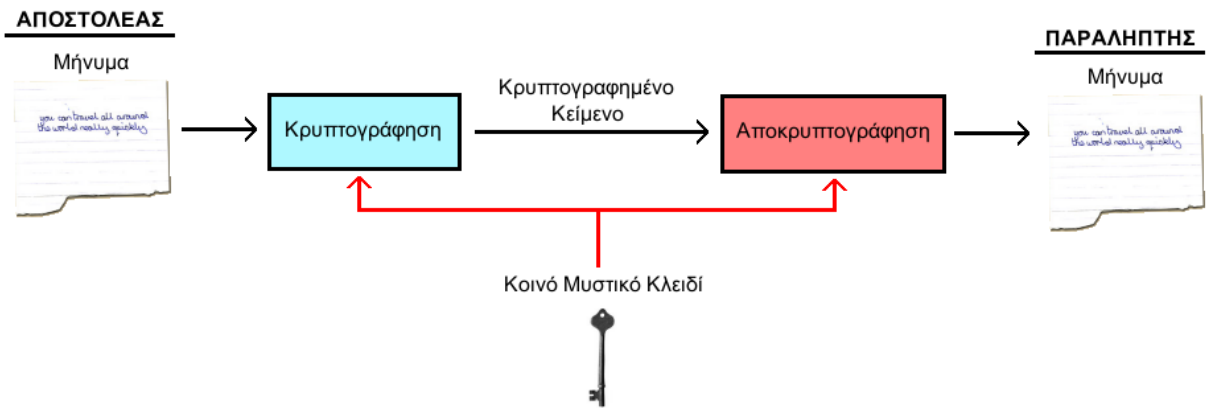
χρήση τους εξαρτάται κάθε φορά από μία σειρά απαιτήσεων που πρέπει να καλύπτουν, τόσο ως προς το υλικό και την απόδοση, όσο και ως προς το επίπεδο ασφάλειας.

2.2 Κατηγορίες κρυπτογραφικών αλγόριθμων

Υπάρχουν δύο κατηγορίες κρυπτογραφικών αλγορίθμων:

- Συμμετρικού κλειδιού (Symmetric Cryptography)
- Δημόσιου κλειδιού (Public Key Cryptography) ή ασύμμετρου κλειδιού (Asymmetric Cryptography)

Η κρυπτογράφηση **συμμετρικού κλειδιού (Symmetric Cryptography**) βασίζεται στην ύπαρξη ενός και μόνο κλειδιού, το οποίο χρησιμοποιείται τόσο στην κρυπτογράφηση όσο και στην αποκρυπτογράφηση του μηνύματος. Το κλειδί αυτό θα πρέπει να είναι γνωστό μόνο στα συναλλασσόμενα μέρη. Η διαδικασία της κρυπτογράφησης και αποκρυπτογράφησης φαίνεται πιο παραστατικά στην Εικόνα 2-1, που ακολουθεί [Web01]:



Εικόνα 2-1: Η διαδικασία κρυπτογράφησης συμμετρικού κλειδιού.

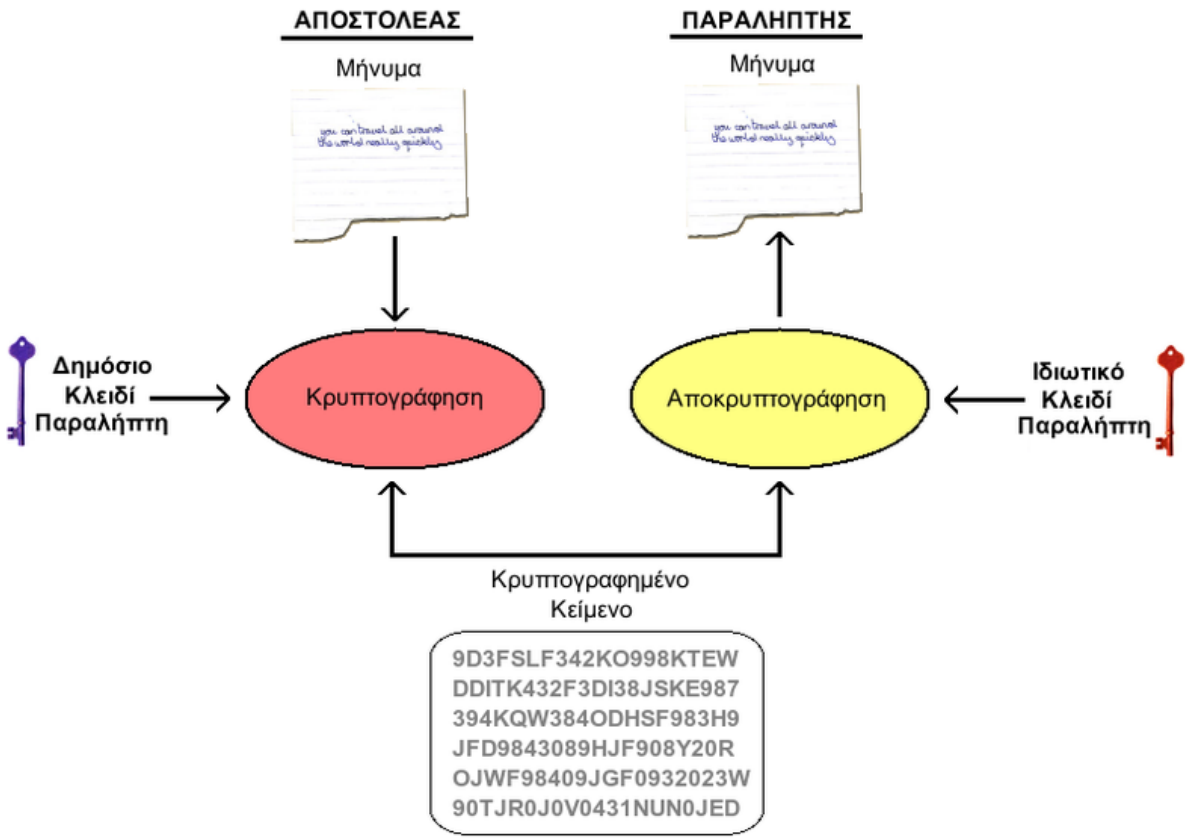
Οι αλγόριθμοι συμμετρικού κλειδιού συναντώνται σήμερα σε διάφορες εφαρμογές και διακρίνονται σε δύο μεγάλες κατηγορίες:

- **Αλγόριθμοι τμήματος (Block Cipher)**, όπου το αρχικό μήνυμα χωρίζεται σε τμήματα (blocks) και το κάθε τμήμα κρυπτογραφείται ξεχωριστά

- **Αλγόριθμοι ροής (Stream Cipher)**, όπου το αρχικό μήνυμα κρυπτογραφείται χαρακτήρα - χαρακτήρα (στην πράξη: ξεχωριστή κρυπτογράφηση για κάθε bit ή byte)

Η κρυπτογράφηση **δημοσίου κλειδιού (Public Key Cryptography) ή ασύμμετρου κλειδιού (Asymmetric Cryptography)** παρέχει ένα εντελώς διαφορετικό μοντέλο διαχείρισης των κλειδιών κρυπτογράφησης από την προγενέστερη κρυπτογράφηση συμμετρικού κλειδιού. Η βασική ιδέα είναι ότι ο αποστολέας και ο παραλήπτης δεν μοιράζονται ένα κοινό μυστικό κλειδί όπως στην περίπτωση της κρυπτογράφησης συμμετρικού κλειδιού, αλλά διαθέτουν διαφορετικά κλειδιά για διαφορετικές λειτουργίες.

Συγκεκριμένα κάθε χρήστης διαθέτει δύο κλειδιά κρυπτογράφησης: το ένα ονομάζεται **ιδιωτικό κλειδί (private key)** και το άλλο **δημόσιο κλειδί (public key)**. Το ιδιωτικό κλειδί θα πρέπει ο κάθε χρήστης να το προφυλάσσει και να το κρατάει κρυφό, ενώ αντιθέτως το δημόσιο κλειδί μπορεί να το ανακοινώνει σε όλη τη διαδικτυακή κοινότητα ή σε συγκεκριμένους παραλήπτες. Η διαδικασία της κρυπτογράφησης και αποκρυπτογράφησης φαίνεται πιο παραστατικά στην Εικόνα 2-2, που ακολουθεί [Web02]:



Εικόνα 2-2: Η διαδικασία κρυπτογράφησης δημοσίου κλειδιού.

Στην παρούσα διπλωματική δεν θα εμβαθύνουμε σε πιο λεπτομερείς έννοιες για τους τύπους των κρυπτογραφικών αλγόριθμων, επειδή εκφεύγει του σκοπού μας: ωστόσο, πρέπει να σημειωθεί ότι για τους κρυπτογραφικούς αλγόριθμους της κατηγορίας του συμμετρικού κλειδιού και συγκεκριμένα τους αλγόριθμους ροής, η ασφάλειά τους βασίζεται σε μεγάλο βαθμό στην τυχαιότητα της ακολουθίας του κλειδιού – ως εκ τούτου, ανακύπτει το ζήτημα της παραγωγής ακολουθιών με καλά χαρακτηριστικά τυχαιότητας, το οποίο αποτελεί και το κύριο αντικείμενο της διατριβής.

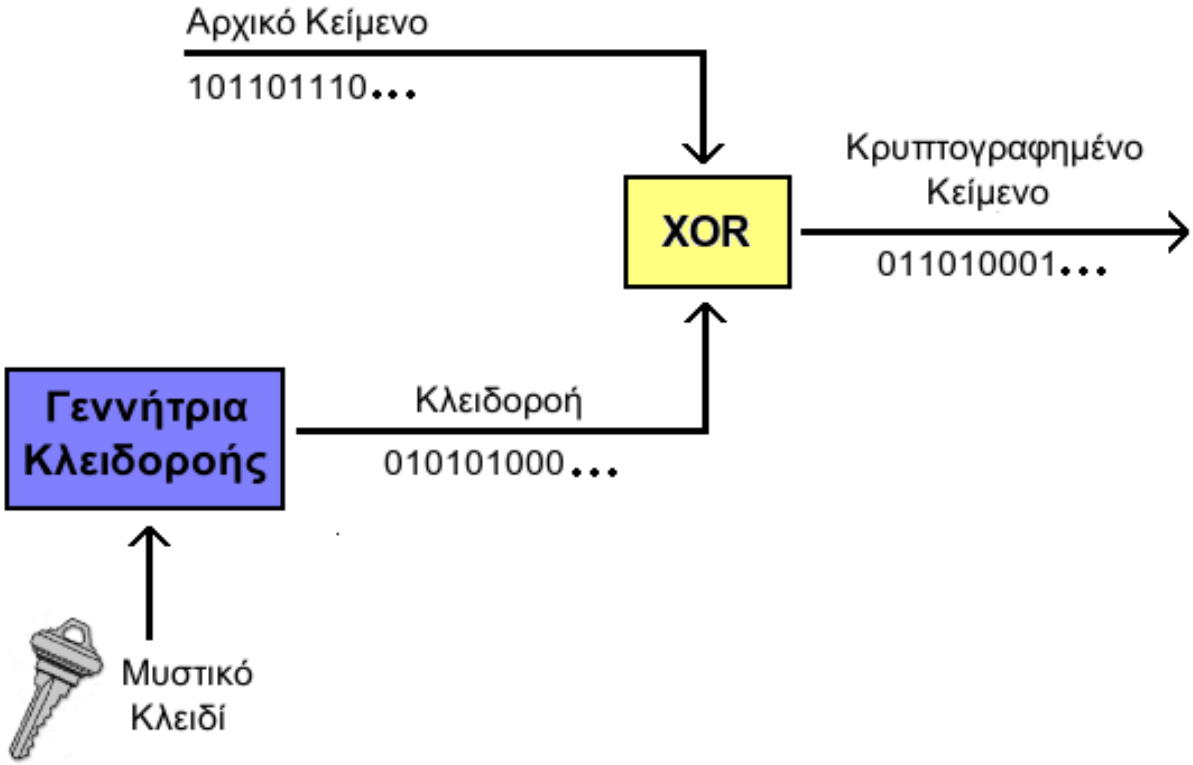
2.3 Κρυπταλγόριθμοι ροής

Οι κρυπταλγόριθμοι ροής (stream ciphers) είναι μία σημαντική κατηγορία κρυπτογραφικών αλγορίθμων με πληθώρα εφαρμογών (Διαδίκτυο, ασύρματες / κινητές επικοινωνίες κ.α.). Γενικότερα, χρησιμοποιούνται ευρέως σε εφαρμογές με απαιτήσεις για υψηλή ταχύτητα και χαμηλή κατανάλωση ισχύος, ενώ επίσης έχουν το πλεονέκτημα της χαμηλής πολυπλοκότητας των κυκλωμάτων τους. Σε πολλές περιπτώσεις η χρήση τους είναι υποχρεωτική, όπως σε συστήματα και εφαρμογές επικοινωνιών, όπου η δυνατότητα χρήσης μνήμης είναι περιορισμένη ή πρέπει τα λαμβανόμενα σύμβολα να υπόκεινται σε ανεξάρτητη επεξεργασία κατά την μετάδοση ή την λήψη τους. Σε σχέση με τους κρυπταλγόριθμους τμήματος, οι κρυπταλγόριθμοι ροής θεωρούνται λιγότερο ασφαλείς και αυτό οφείλεται εν μέρη στην απλότητά του σχεδιασμού: για αυτό εξάλλου οι λεπτομέρειες πολλών απ΄ αυτών που χρησιμοποιούνται σε εμπορικές εφαρμογές παραμένουν εμπιστευτικές. Παρόλα αυτά λόγω των σημαντικών πλεονεκτημάτων τους οι κρυπταλγόριθμοι ροής χρησιμοποιούνται ευρέως σήμερα [03], ενώ υπάρχουν αλγόριθμοι της κατηγορίας αυτής για τους οποίους δεν έχουν εντοπιστεί ευπάθειες. Χαρακτηριστικά παραδείγματα εφαρμογής των κρυπταλγορίθμων είναι:

- Ασύρματα δίκτυα
- Κινητές επικοινωνίες (GSM, 3G)
- Πρωτόκολλα Bluetooth
- Δίκτυα RFID

Παράλληλα η χρήση του συναντάται στο διαδίκτυο σε πολλά κρυπτογραφικά πρωτόκολλα ασφάλειας όπως είναι το SSL, TLS (αλγόριθμος RC4).

Οι κρυπτογραφικοί αλγόριθμοι ροής (stream ciphers) χρησιμοποιούνται για την κρυπτογράφηση μίας συνεχούς ροής δεδομένων (data stream). Για την κρυπτογράφηση επιλέγεται αρχικά μία *γεννήτρια κλειδοροής (keystream generator)*, η οποία δέχεται ως είσοδο το μυστικό κλειδί και παράγει στην έξοδό της μία ψευδοτυχαία ακολουθία από δυαδικά ψηφία (bits), η οποία ονομάζεται κλειδοροή (keystream). Στην συνέχεια εφαρμόζεται η συνάρτηση XOR ανάμεσα στο αρχικό κείμενο και στην κλειδοροή και το αποτέλεσμα της συνάρτησης είναι η τελική κρυπτογραφημένη ροή δεδομένων. Η διαδικασία που μόλις περιγράφηκε φαίνεται πιο καθαρά στην Εικόνα 2-3, που παρατίθεται [Web03].



Εικόνα 2-3: Σχηματικό διάγραμμα του τρόπου λειτουργίας των κρυπτογραφικών αλγορίθμων ροής.

Η αποκρυπτογράφηση γίνεται με την ακριβώς αντίστροφη διαδικασία. Εάν χρησιμοποιηθεί το ίδιο κλειδί ως είσοδο στην γεννήτρια κλειδοροής, τότε η δεύτερη θα παράγει ακριβώς την ίδια ακολουθία bits (κλειδοροή) όπως και προηγουμένως κατά την διαδικασία της κρυπτογράφησης. Εφαρμόζοντας την συνάρτηση XOR ανάμεσα στην κρυπτογραφημένη ακολουθία δεδομένων και την κλειδοροή παράγεται τελικά το αρχικό κείμενο.

2.4 Περιγραφή κρυπταλγόριθμων ροής

Η κρυπτογράφηση γίνεται πάνω σε μία ροή από bits (ή bytes) και για το σκοπό αυτό χρησιμοποιείται μια γεννήτρια ψευδοτυχαίας ακολουθίας bits , η οποία παράγει μια ακολουθία από bits που ονομάζεται «κλειδοροή». Οι ακολουθίες αυτές ονομάζονται ψευδοτυχαίες και προσομοιάζουν το αποτέλεσμα ανεξάρτητων επαναλήψεων ενός τυχαίου πειράματος του οποίου οι πιθανές εκβάσεις είναι τα εν δυνάμει σύμβολα της ακολουθίας. Η παραγόμενη κλειδοροή είναι το μόνο χαρακτηριστικό που πρέπει να μείνει κρυφό σύμφωνα με την κλασική πλέον αρχή του Kerchhoff, βάσει της οποίας η ασφάλεια πρέπει να έγκειται μόνο στο μυστικό κλειδί [14]. Επιπλέον πρέπει να διασφαλιστεί η αρχική κατάσταση από την οποία πηγάζει αυτή η κλειδοροή, ώστε να αυξηθεί η ασφάλεια του κρυπτοκειμένου απέναντι σε κρυπτογραφικές επιθέσεις, που ποικίλουν – και που συνεχώς εμφανίζονται καινούριες.

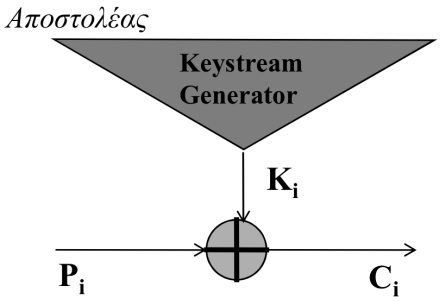
Τα bits της κλειδοροής προστίθενται με πράξη XOR (exclusive OR) με τα bits του αρχικού κειμένου (μηνύματος), για να προκύψει το κρυπτοκείμενο. Ως εκ τούτου καταλαβαίνουμε ακόμα καλύτερα τη δημοφιλή χρήση τους, εφόσον με μία απλή πράξη XOR συντελείται η κρυπτογράφηση, αποφεύγοντας πολύπλοκους υπολογισμούς. Η πράξη XOR περιγράφεται στο παρακάτω πίνακα αληθείας:

X_i	S_i	ΕΞΟΔΟΣ
0	0	0
0	1	1
1	0	1
1	1	0

Πίνακας 2-1: Πίνακας αληθείας της πράξης XOR.

Η κρυπτογράφηση και αποκρυπτογράφηση του αρχικού κειμένου περιγράφεται συνοπτικά από το παρακάτω κείμενο:

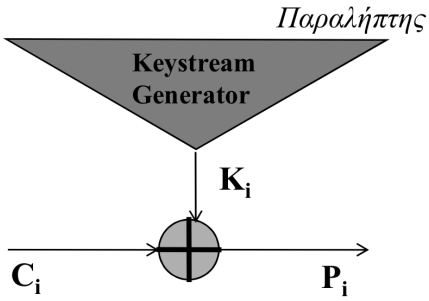
- **Κρυπτογράφηση:** $C_i = P_i + K_i$ όπου C_i είναι το κρυπτοκείμενο , P_i το αρχικό μήνυμα και K_i η κλειδοροή



Εικόνα 2-4: Σχηματική παρουσίαση κρυπτογράφησης με τη χρήση γεννήτριας κλειδορόης.

Για την αποκρυπτογράφηση έχουμε την αντίστοιχη διαδικασία.

- **Αποκρυπτογράφηση**: $P_i = C_i + K_i$ όπου C_i είναι το κρυπτοκείμενο , P_i το αρχικό μήνυμα και K_i η κλειδοροή



Εικόνα 2-5: Σχηματική παρουσίαση αποκρυπτογράφησης με τη χρήση γεννήτριας κλειδορόης.

Γιατί όμως η πράξη XOR είναι τόσο βασική στους αλγόριθμους ροής και δεν χρησιμοποιούμε, για παράδειγμα, την πράξη AND;

Θεωρούμε ότι θέλουμε να κρυπτογραφήσουμε το αρχικό μήνυμα $X_i = 0$. Αν κοιτάξουμε στον πίνακα αληθείας (Πίνακας 2-1), παρατηρούμε ότι με τούτο το αρχικό μήνυμα προσανατολιζόμαστε στην 1η ή στην 2η γραμμή του πίνακα αληθείας. Παρατηρούμε ότι η έξοδος, δηλαδή το κρυπτοκείμενο, εξαρτάται από το κλειδί S_i. Αν το κλειδί είναι το 0, τότε η έξοδος είναι 0, σε κάθε άλλη περίπτωση, το κλειδί είναι το 1 και η έξοδος, δηλαδή το κρυπτοκείμενο, είναι 1. Αν το κλειδί έχει απόλυτα τυχαία συμπεριφορά, τότε η έξοδος είναι μη προβλέψιμη και κατά 50% θα είναι ίση με 0 ή 1. Όμοια είναι όταν θέλουμε να κρυπτογραφήσουμε το αρχικό μήνυμα $X_i = 1$. Εκεί θα προσανατολιστούμε στην 3η και 4η γραμμή του πίνακα αλήθειας (Πίνακας 2-1) και ομοίως θα πάρουμε έξοδο, δηλαδή κρυπτοκείμενο, ίσο με 0 ή 1 με την ίδια πιθανότητα, και απόλυτα εξαρτώμενη από το κλειδί.

Παρατηρούμε δηλαδή ότι η πράξη XOR είναι απόλυτα ισορροπημένη και η έξοδός της έχει ακριβώς την ίδια πιθανότητα να εμφανίσει 0 ή 1 για κάθε τιμή εισόδου. Αυτό το γεγονός την διακρίνει ως πιο χρήσιμη πράξη, σε αντίθεση με άλλες πράξεις, όπως το AND ή OR [15].

2.5 Ο «τέλειος» κρυπταλγόριθμος ροής – «σημειωματάριο» μιας χρήσης

Ο μηχανικός Gilbert Vernam [Web10] είχε προτείνει από το 1918 ένα σύστημα όπως οι σημερινοί κρυπταλγόριθμοι ροής, με μόνη κύρια απαίτηση το πολύ μεγάλο μέγεθος της κλειδοροής, έτσι ώστε να μην επαναλαμβάνεται ποτέ: αυτό όμως ερχόταν σε αντίθεση με τη λογική του ότι όσο μεγάλο μέγεθος και αν θέσουμε για το κλειδί εκ των προτέρων, πάντοτε μπορεί το αρχικό κείμενο (μήνυμα) να έχει μεγαλύτερο μέγεθος, γεγονός που καθιστούσε μη εφικτή την πρακτική εφαρμογή του.

Το «σημειωματάριο» μίας χρήσης (one time pad) [Web10] είναι η γενίκευση του αλγόριθμου Vernam, και αποκαλείται εκείνο το κρυπτοσύστημα, όπου η κλειδοροή είναι μια απόλυτα τυχαία ακολουθία από bits, μεγέθους όσο και το αρχικό κείμενο (μήνυμα), μη περιοδική, και δε χρησιμοποιεί ποτέ το ίδιο κλειδί εκ νέου, οπότε κάθε μήνυμα κρυπτογραφείται με διαφορετικό κλειδί. Αποδεικνύεται ότι το σημειωματάριο μίας χρήσης είναι απόλυτα ασφαλές κατά Shannon [17], αλλά δυστυχώς στην πράξη δεν μπορεί να υλοποιηθεί, γιατί αφενός δεν μπορούμε να έχουμε κλειδοροή ίδιου μεγέθους με το μήνυμα στη γενική περίπτωση (δεν μπορεί να παραχθεί ακολουθία άπειρου μήκους, μη περιοδική - υποχρεωτικά η κλειδοροή θα έχει κάποια επανάληψη, έστω και έχει πολύ μεγάλη περίοδο), και αφετέρου γιατί δεν μπορεί από μία ντετερμινιστική μηχανή, όπως είναι ένας Η/Υ, να έχουμε παραγωγή απολύτως τυχαίας ακολουθίας.

Επομένως καταλήγουμε στο συμπέρασμα, ότι στόχος των κρυπταλγόριθμων ροής είναι να προσομοιάσουν, κατά το δυνατόν, το σημειωματάριο μιας χρήσης, δηλαδή να βρεθούνε τεχνικές παραγωγής ακολουθιών πολύ μεγάλης περιόδου, που εμφανίζουν καλά χαρακτηριστικά τυχαιότητας. Τεχνικές κατασκευής ακολουθιών με καλά χαρακτηριστικά τυχαιότητας (ήτοι ψευδοτυχαίες ακολουθίες) περιγράφονται στο επόμενο κεφάλαιο.

Κεφάλαιο 3
Ψευδοτυχαίες ακολουθίες και τεχνικές κατασκευής τους

3.1 Εισαγωγή

Το κυνήγι των τυχαίων αριθμών και γενικότερα των τυχαίων ακολουθιών θεωρείται μια από τις μεγαλύτερες προκλήσεις των θετικών επιστημών. Ιδιαίτερα για την ασφάλεια των κρυπτοσυστημάτων, η ικανότητα επιλογής τυχαίων αριθμών είναι ένα κρίσιμο χαρακτηριστικό. Η δημιουργία των κλειδιών περιλαμβάνει τυχαίες επιλογές από σύνολα στοιχείων τα οποία μπορεί να είναι τα ίδια τα κλειδιά, ή ποσότητες που καθορίζουν τα κλειδιά αυτά (όπως για παράδειγμα η τυχαία επιλογή ενός πρώτου αριθμού). Θα δούμε σε αυτό το κεφάλαιο την έννοια της τυχαιότητας και τι ορίζουμε τυχαίο και ποια η διαφορά από το ψευδοτυχαίο. Επίσης, θα παρουσιάσουμε γεννήτριες που παράγουν ψευδοτυχαίους αριθμούς και τα χαρακτηριστικά αυτών των γεννητριών. Οι πιο κλασικές γεννήτριες είναι οι γραμμικοί καταχωρητές ολίσθησης με ανάδραση η αλλιώς LFSR (Linear Feedback Shift Registers). Θα γίνει αναφορά σε αυτές και στα χαρακτηριστικά τους. Ένα από τα βασικότερα χαρακτηριστικά για την κρυπτογραφική ισχύ μιας κλειδοροής είναι και η γραμμική πολυπλοκότητα, στην οποία θα αναφερθούμε, για να καλύψουμε βασικές ανάγκες κατανόησης εννοιών στη συνέχεια της διατριβής.

Στους κρυπταλγορίθμους ροής, κρίσιμο χαρακτηριστικό για την ασφάλειά τους είναι το κατά πόσον η κλειδοροή μπορεί να προβλεφθεί – και αυτό γιατί γνώση της κλειδοροής στον επιτιθέμενο θα του επιτρέψει, προφανώς, την πλήρη αποκρυπτογράφηση του μηνύματος. Η παραγόμενη κλειδοροή εξαρτάται από την αρχική κατάσταση της γεννήτριας. Ωστόσο η αρχική κατάσταση της γεννήτριας κλειδοροής καθορίζεται από το μυστικό κλειδί της κρυπτογράφησης / αποκρυπτογράφησης. Με άλλα λόγια, η «καρδιά» κάθε τέτοιου αλγόριθμου έγκειται στη γεννήτρια κλειδοροής, η οποία πρέπει να διασφαλίζει την παραγωγή ακολουθιών με χαρακτηριστικά τυχαιότητας, δηλαδή ακολουθίες μη προβλέψιμες. Δεν είναι εύκολη η διασφάλιση όλων των κριτηρίων τυχαιότητας, ωστόσο η υψηλή γραμμική πολυπλοκότητα – που αναφέρθηκε παραπάνω και θα μελετηθεί στη συνέχεια - πρέπει να είναι ευαπόδεικτη. Στη συνέχεια θα μελετήσουμε τεχνικές κατασκευής γεννητριών κλειδοροής, που βασίζονται σε χρήση μη γραμμικών λογικών συναρτήσεων.

3.2 Η έννοια της τυχαιότητας μίας ακολουθίας

Οι κρυπτογραφικοί αλγόριθμοι βασίζουν την ασφάλειά τους σε έναν μεγάλο βαθμό, στα χαρακτηριστικά τυχαιότητας που εμφανίζουν οι υπεισερχόμενες ακολουθίες. Ακριβώς γι΄ αυτό τον λόγο πρέπει να χρησιμοποιούνται ακολουθίες ως κλειδοροές σε αλγόριθμους ροής που είναι τυχαίες.

Τι εννοούμε όμως με την λέξη «τυχαία» όταν αναφερόμαστε σε μία δυαδική ακολουθία; Είναι σαφής ορισμός;

Για παράδειγμα,

- είναι προφανές ότι η ακολουθία 11111111110 δεν μπορεί να θεωρηθεί τυχαία.
- Ομοίως, ούτε η ακολουθία 1010101010101 μπορεί να θεωρηθεί τυχαία

Μία τυχαία ακολουθία μπορούμε να πούμε ότι προσομοιώνει το αποτέλεσμα της ρίψης ενός αμερόληπτου δίκαιου νομίσματος του οποίου οι δύο πλευρές είναι συμβολισμένες με το μηδέν ή το ένα, η κάθε μία. Κάθε ρίψη έχει ακριβώς την ίδια πιθανότητα 50% να «φέρει» μηδέν ή ένα. Ακόμα περισσότερο η κάθε ρίψη είναι ανεξάρτητη από την προηγούμενη και δεν επηρεάζει την επόμενη. Το αμερόληπτο αυτό δίκαιο νόμισμα είναι μία γεννήτρια παραγωγής μίας τελείως τυχαίας ακολουθίας, εφόσον οι τιμές μηδέν και ένα διανέμονται απόλυτα τυχαία και δεν είναι προβλέψιμες. Είναι προφανές όμως ότι η ρίψη ενός τέτοιου νομίσματος δεν είναι καθόλου καλή

και εφαρμόσιμη πρακτική για το σκοπό της παραγωγής ακολουθιών που χρειάζεται ένα ισχυρό κρυπτογραφικό σύστημα [16].

Ως εκ τούτου, στην κρυπτογραφία κάθε τυχαία ακολουθία παράγεται από μία ντετερμινιστική μηχανή, όπως προαναφέραμε, που είναι ένας Ηλεκτρονικός Υπολογιστής ή γενικά μια ηλεκτρονική συσκευή.

Επειδή η ακολουθία παράγεται από μια ντετερμινιστική μηχανή και όχι από την ίδια την φύση, όπως όταν ρίχνουμε ένα νόμισμα ή ένα ζάρι, μιλάμε για ψευδοτυχαία ακολουθία αντί για τυχαία ακολουθία [08].

Το εύλογο ερώτημα που προκύπτει είναι πως θα αποτιμήσουμε το μέτρο της τυχαιότητας της παραγόμενης ακολουθίας;

3.3 Κριτήρια τυχαιότητας ακολουθίας

Κάποια βασικά κριτήρια τυχαιότητας που προτάθηκαν και καθιερώθηκαν αρχικά για να αποτιμήσουνε το μέτρο της τυχαιότητας μιας ακολουθίας, είναι τα ακόλουθα:

- Μεγάλη περίοδος της ακολουθίας
- Ισοκατανεμημένο πλήθος 0 και 1 *(Balance property - R_1)*
- Κριτήριο «διαδρομών» 0 και 1 *(Run property - R_2)*
- Αυτοσυσχέτιση δύο τιμών *(two level autocorrelation property - R_3)*

Τα κριτήρια R_1, R_2 και R_3 είναι γνωστά **ως κριτήρια τυχαιότητας του Golomb** και είναι από τα πρώτα – και πιο γνωστά - κριτήρια που προτάθηκαν [13].

Για μια περιοδική δυαδική ακολουθία, το κριτήριο της ισοκατανομής 0 και 1 σημαίνει ότι ο αριθμός των 0 είναι προσεγγιστικά όμοιος με των αριθμό των 1 σε μία περίοδο αυτής και η ακολουθία καλείται ισοβαρής.

Το κριτήριο των διαδρομών των 0 και 1 bit περιγράφει ένα τμήμα της ακολουθίας που αποτελείται από μηδενικά ή μόνο από άσσους (και αμέσως πριν και μετά από αυτά βρίσκονται διαφορετικά bit από αυτά που απαρτίζουν το τμήμα) και εμφανίζουν διαδρομές με διάφορες

συχνότητες μηκών. Τότε, σε μία περίοδο οι μισές διαδρομές έχουν μήκος 1, το $^1/_4$ των διαδρομών έχουν μήκος 2, το $^1/_8$ μήκος 3 κ.ο.κ. Η ισχύς της συνθήκης εξετάζεται όσο ο αριθμός των διαδρομών είναι μεγαλύτερος ή ίσος από $2l$ όπου l το μήκος της διαδρομής.

Το κριτήριο της αυτοσυσχέτισης ανακλά τη διαφορά της ακολουθίας με την ολισθημένη ακολουθία κατά μια θέση σε κάθε επανάληψη της καταμέτρησης μετά από κυκλική ολίσθηση.

Η συνάρτηση αυτοσυσχέτισης μίας ακολουθίας $\alpha_0\alpha_1\ldots\alpha_{N-1}$ ορίζεται ως εξής:

$C(\tau) = \sum_{i=0}^{N-1}(-1)^{\alpha_i \oplus \alpha_{i+\tau}}$, η πράξη $\oplus$ είναι η γνωστή XOR (exclusive or).

Για να καταλάβουμε καλύτερα τα κριτήρια τυχαιότητας του Golomb, θα παρουσιάσουμε ένα παράδειγμα:

Έστω ότι έχουμε την δυαδική περιοδική ακολουθία $A = \{a_i\}$ που περιγράφεται από: 000100110101111000100110101111000100110101111

Παρατηρούμε τα ακόλουθα:

- Η περίοδος της ακολουθίας είναι ίση με 15, ενδεικτικά μία περίοδος της ακολουθίας είναι: 000100110101111 *(γραμμοσκιασμένο με κίτρινο χρώμα στην ακολουθία)*
- Έχουμε 7 μηδενικά και 8 άσσους σε μία περίοδο της ακολουθίας, άρα είναι ισοκατανεμημένα, οπότε ικανοποιείται το balance property.
    - Έχει 8 διαδρομές:
        - Οι μισές διαδρομές έχουν μήκος 1
        - Το $^1/_4$ των διαδρομών, δηλαδή οι δύο διαδρομές έχουνε μήκος 2
        - Το $^1/_8$ των διαδρομών, δηλαδή μία διαδρομή έχει μήκος 3
        - Για διαδρομή μήκους 4 δεν εξετάζουμε, μια που $8 < 2^4$
- Άρα ικανοποιείται και το run property.
- Για τ=0, η συνάρτηση αυτοσυσχέτισης ισούται με $\boldsymbol{C(0) = 15}$ και αποδεικνύεται με απλή αντικατάσταση στον τύπο, όπου τ=0 και N=15:

$$\boldsymbol{C(\tau) = \sum_{i=0}^{N-1} (-1)^{\alpha_i \oplus \alpha_{i+\tau}} \Rightarrow C(0) = \sum_{i=0}^{14} (-1)^{\alpha_1 \oplus \alpha_{i+\tau}} = (-1)^0 + (-1)^0 + \cdots + (-1)^0 = 15}$$

- Για τ=1, η συνάρτηση αυτοσυσχέτισης ισούται με:

Ακολουθία: 000100110101111

Ολισθημένη μια θέση: 001001101011110

XOR άθροισμα: 001101011110001

- Στο XOR άθροισμα έχουμε 7 μηδενικά και 8 άσσους
- Άρα, η συνάρτηση αυτοσυσχέτισης ισούται με -1 (κάθε ένας από τους άσσους συνεισφέρει με -1 στο άθροισμα που υπεισέρχεται στον υπολογισμό της C(1), ενώ κάθε μηδενικό συνεισφέρει με 1 στο άθροισμα).

Όμοια αποδεικνύεται ότι για τ=2 έως τ=14 η τιμή της συνάρτησης είναι πάντα ίση με -1.

- Άρα η ακολουθία πληροί όλα τα κριτήρια τυχαιότητας του Golomb και θεωρείται ότι προσομοιάζει μία τυχαία ακολουθία.

Βέβαια υπάρχει ένα μεγάλο πλήθος από διάφορους ελέγχους τυχαιότητας που μπορεί να υποβληθεί μία ακολουθία, για να προσπαθήσουμε να κατηγοριοποιήσουμε ή ακόμα καλύτερα να αξιολογήσουμε την ακολουθία ως προς την τυχαιότητά της. Ως εκ τούτου δημιουργήθηκαν πληθώρα από κριτήρια ελέγχου τυχαιότητας, καθένα από τα οποία μας δείχνει με το δικό του τρόπο το μέτρο της τυχαιότητας της ακολουθίας. Τα 15 πιο γνωστά κριτήρια τυχαιότητας, όπως αυτά περιγράφονται από τον οργανισμό προτυποποίησης NIST (National Institute for Standards and Technology) είναι [16]:

1. The Frequency (Monobit) Test
2. Frequency Test within a Block
3. The Runs Test
4. Tests for the Longest-Run-of-Ones in a Block
5. The Binary Matrix Rank Test
6. The Discrete Fourier Transform (Spectral) Test
7. The Non-overlapping Template Matching Test
8. The Overlapping Template Matching Test

9. Maurer's "Universal Statistical" Test
10. The Linear Complexity Test
11. The Serial Test
12. The Approximate Entropy Test
13. The Cumulative Sums (Cusums) Test
14. The Random Excursions Test
15. The Random Excursions Variant Test

Δε θα μπούμε σε λεπτομέρειες των κριτηρίων αυτών, γιατί δεν είναι σκοπός της παρούσας διπλωματικής διατριβής. Απαιτούν δε γνώσεις στατιστικής και μαθηματικών, που ξεφεύγει από τα όρια της ουσίας της παρούσας διπλωματικής, αλλά παρουσιάζουν με έναν απόλυτα μαθηματικό τρόπο την τυχαιότητα κάθε ακολουθίας. Θα μας απασχολήσει όμως στη συνέχεια ιδιαίτερα το κριτήριο της γραμμικής πολυπλοκότητας (linear complexity test), διότι καθόρισε εν πολλοίς τη λογική σχεδίασης γεννητριών ψευδοτυχαίων ακολουθιών.

3.4 Γεννήτριες ψευδοτυχαίων ακολουθιών

Ο βασικός στόχος των κρυπταλγόριθμων ροής είναι η αποδοτική παραγωγή μίας φαινομενικά τυχαίας ακολουθίας συμβόλων, που καλείται κλειδοροή. Οι ακολουθίες αυτές ονομάζονται ψευδοτυχαίες και προσομοιάζουν το αποτέλεσμα ανεξάρτητων επαναλήψεων ενός τυχαίου πειράματος του οποίου οι πιθανές εκβάσεις είναι τα εν δυνάμει σύμβολα της ακολουθίας. Σε αντίθεση με τις τυχαίες ακολουθίες, οι οποίες χαρακτηρίζονται από τις ακόλουθες ιδιότητες:

- πλήρης έλλειψη μοτίβων
- αδυναμία πρόβλεψης συμβόλων
- έλλειψη απλής περιγραφής

Οι ψευδοτυχαίες ακολουθίες διαθέτουν σχετικά απλή περιγραφή. Συνεπώς, η παραγόμενη κλειδοροή είναι κατάλληλη προς χρήση σε κρυπτογραφικές εφαρμογές μόνο εάν είναι δυσδιάκριτη ως προς την τυχαία ακολουθία, δηλαδή είναι υπολογίστηκα αδύνατη η εύρεση της αντίστοιχης απλής περιγραφής. Ο βαθμός τυχαιότητας δοθείσας κλειδοροής αποτιμάται μέσω της ικανοποίησης ενός μεγάλου αριθμού (πιθανώς αντικρουόμενων) κρυπτογραφικών κριτηρίων [03].

Για κάποια χρόνια θεωρήθηκε ότι κάθε ακολουθία που πληροί τα κριτήρια τυχαιότητας του Golomb, όπως τα προαναφέραμε , είναι μία κρυπτογραφικά ισχυρή ακολουθία, εφόσον βέβαια έχει μεγάλη περίοδο [22].

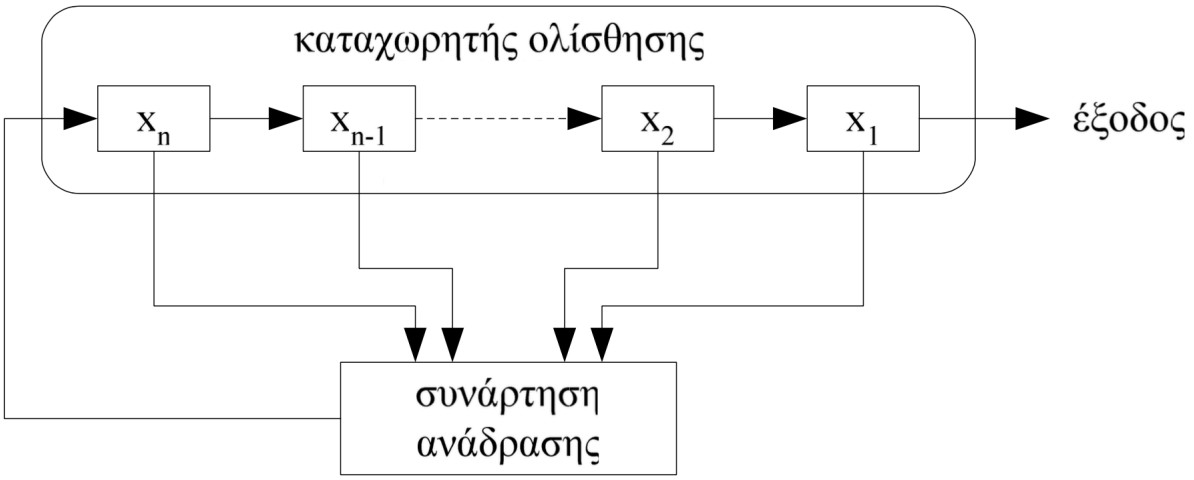
Γενικά, θα μπορούσαμε να πούμε ότι οι γεννήτριες ψευδοτυχαίων ακολουθιών είναι κάποιες συναρτήσεις οι οποίες παράγουν μεγάλες ακολουθίες από ψηφία (bits), ξεκινώντας από μία μικρή ακολουθία τυχαίων bits, η οποία καλείται φύτρο (random seed). Οι παραγόμενες ακολουθίες, φαινομενικά, έχουνε όλα τα απαραίτητα χαρακτηριστικά για να χαρακτηριστούν τυχαίες, αποκρύπτοντας το γεγονός ότι έχουν δημιουργηθεί με ένα σαφή, ντετερμινιστικό κανόνα, από ένα μικρό αριθμό τυχαίων bits [03].

Γνωρίζουμε ότι υπάρχουνε γεννήτριες ψευδοτυχαίων bits που παράγουν εγγυημένα ακολουθίες με μεγάλη περίοδο και καλά χαρακτηριστικά τυχαιότητας και είναι οι λεγόμενοι γραμμικοί καταχωρητές ολίσθησης με ανάδραση Linear Feedback Shift Registers - LFSRs. Οι LFSR έχουνε καλή μαθηματική περιγραφή που επιτρέπει την ανάλυσή τους ενώ επιπλέον υλοποιούνται εύκολα σε hardware.

3.4.1 Γραμμικοί καταχωρητές ολίσθησης με ανάδραση – LFSR

Οι καταχωρητές ολίσθησης (shift registers) έχουν μελετηθεί εκτενώς τις τελευταίες δεκαετίες. Ανήκουν στην κατηγορία των μηχανών πεπερασμένης κατάστασης (finite state machines). Μια μηχανή πεπερασμένης κατάστασης ορίζεται ως η συσκευή η ο οποία αποτελείται από έναν πεπερασμένο αριθμό καταστάσεων όπου η μεταπήδηση από τη μια κατάσταση στην άλλη ορίζεται από την υπάρχουσα κατάσταση και την είσοδο. Η είσοδος ορίζεται ως μια ακολουθία από ένα πεπερασμένο σύνολο στοιχειών. Η έξοδος μιας μηχανής πεπερασμένης κατάστασης είναι μια ακολουθία από ένα πεπερασμένο σύνολο στοιχείων. Οι ηλεκτρονικοί υπολογιστές είναι παραδείγματα μηχανών πεπερασμένης κατάστασης.

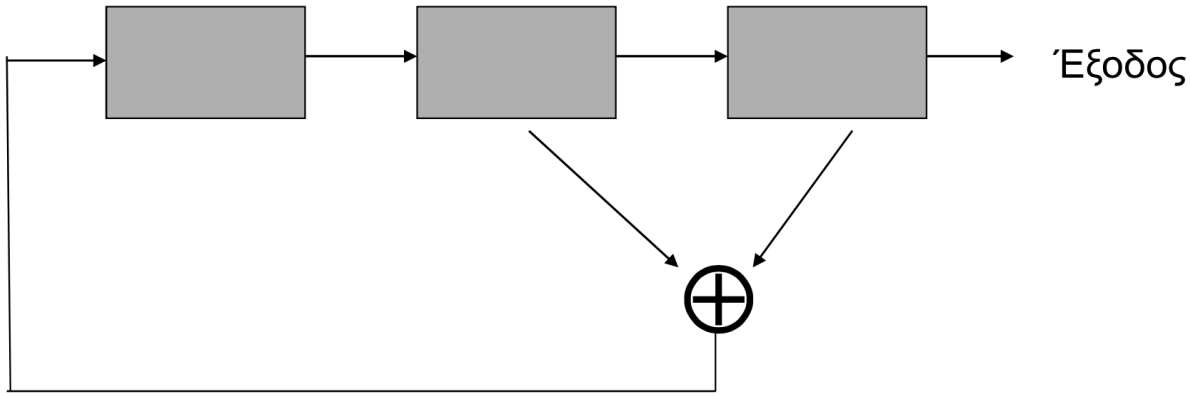
Οι καταχωρητές ολίσθησης με ανάδραση είναι αυτοί των οποίων η έξοδος τροφοδοτείται από μια συνάρτηση της οποίας το αποτέλεσμα τροφοδοτείται με τη σειρά του στην είσοδο του καταχωρητή, όπως φαίνεται στο παρακάτω σχήμα:



Εικόνα 3-1: Σχηματική αναπαράσταση του καταχωρητή ολίσθησης με ανάδραση – FSR (Feedback Shift Register).

Ουσιαστικά είναι ένα ψηφιακό κύκλωμα, το οποίο αποτελείται από N βαθμίδες ή αλλιώς N θέσεις μνήμης – όπου το N καθορίζει και το μέγεθος του FSR. Το περιεχόμενο της κάθε θέσης είναι το 0 ή το 1, δηλαδή το δυαδικό ψηφίο (bit). Εάν η συνάρτηση ανάδρασης είναι γραμμική, δηλαδή αν αποτελείται από απλά, μία πύλη XOR (πρόσθεση ⊕), η οποία έχει ως εισόδους κάποιες (οποιεσδήποτε) από τις βαθμίδες του FSR, τότε μιλάμε για γραμμικό καταχωρητή ολίσθησης με ανάδραση (LFSR). Το σύνολο των τιμών των βαθμίδων του LFSR ονομάζεται κατάσταση του (state) για την τρέχουσα χρονική στιγμή. Κάθε χρονική στιγμή η κατάστασή του μεταβάλλεται, η οποία ενεργοποιείται με τον παλμό του ρολογιού (clock). Σε κάθε νέα κατάσταση η τιμή της πρώτης βαθμίδας του LFSR προκύπτει από τις τιμές της προηγούμενης κατάστασης, βάσει της πρόσθεσης XOR που υλοποιεί ο LFSR, με απλά λόγια είναι το αποτέλεσμα της συνάρτησης ανάδρασης. Οι τιμές των υπόλοιπων βαθμίδων προκύπτουν από ολίσθηση προς τα δεξιά των τιμών των υπόλοιπων βαθμίδων της προηγούμενης κατάστασης του LFSR. Ως πρώτη βαθμίδα στην Εικόνα 3-1 είναι η τιμή x_n της πρώτης θέσης του LFSR από αριστερά προς δεξιά.

Στην παρακάτω Εικόνα 3-2, όπου περιγράφεται ένας απλός LFSR μεγέθους 3, αν η αρχική κατάσταση είναι 001, τότε το πρώτο bit της εξόδου είναι 1 (η δεξιότερη βαθμίδα). Την επόμενη χρονική στιγμή, η κατάσταση θα είναι 100 και η έξοδος 0. Το 100 προκύπτει ως εξής: το «1» είναι το XOR, που είχαν αρχικά η δεύτερη και η τρίτη βαθμίδα (που ήταν 0 και 1 αντίστοιχα), ενώ το «00» είναι απλά ολισθημένες οι τιμές που είχαν αρχικά η πρώτη με τη δεύτερη βαθμίδα.



Εικόνα 3-2: Σχηματική αναπαράσταση του γραμμικού καταχωρητή ολίσθησης με ανάδραση με 3 θέσεις μνήμης.

Οι διαδοχικές καταστάσεις από τις οποίες περνάει ο LFSR και η αντίστοιχη έξοδος του είναι οι εξής:

Κατάσταση	Έξοδος
001	1
100	0
010	0
101	1
110	0
111	1
011	1
001	1

Πίνακας 3-1: Πίνακας εξόδου του LFSR με 3 θέσεις μνήμης και αρχική κατάσταση την 001.

Η κατάσταση 001 ξαναεμφανίζεται στην όγδοη γραμμή του Πίνακα 3-1, άρα οι καταστάσεις επαναλαμβάνονται με την ίδια σειρά. Κατά συνέπεια ο συγκεκριμένος LFSR παράγει την ακολουθία 1001011, η οποία επαναλαμβάνεται περιοδικά.

3.4.2 Ιδιότητες γραμμικών καταχωρητών ολίσθησης με ανάδραση – LFSR

Ένας LFSR μήκους N μπορεί να περάσει από $2^N - 1$ διαφορετικές καταστάσεις (όλες πλην της μηδενικής, εφόσον δεν προστίθεται στην ανάδραση ο σταθερός όρος «1»), άρα μπορεί να γεννήσει ακολουθίες με μέγιστη περίοδο $2^N - 1$. Για να αποφύγουμε σε έναν LFSR να παράγει τη μηδενική ακολουθία, θα πρέπει να αποφύγουμε να έχουμε αρχική κατάσταση του καταχωρητή τη μηδενική.

Η ακολουθία εξόδου ενός LFSR εξαρτάται τόσο από την ανάδρασή του όσο και από την αρχική του κατάσταση.

Οι γραμμικοί καταχωρητές ολίσθησης με ανάδραση (LFSRs), που επιτυγχάνουν τη μέγιστη δυνατή περίοδο ονομάζονται πρωταρχικοί (primitive). Οι ακολουθίες που παράγονται από τέτοιους καταχωρητές ονομάζονται ακολουθίες μέγιστου μήκους (maximal-length sequences ή m-sequences). Ο LFSR της παραπάνω Εικόνας 3-2, παράγει προφανώς ακολουθία μέγιστου μήκους (έχει μήκος 3 και παράγει ακολουθία με μήκος 7). Σε πρωταρχικούς LFSRs η έξοδος έχει πάντα μέγιστη περίοδο ανεξάρτητα της αρχικής του κατάστασης. Το αν ένας LFSR είναι πρωταρχικός εξαρτάται αποκλειστικά από την ανάδρασή του και όχι από την αρχική κατάσταση.

Κάθε LFSR , ανάλογα με την ανάδρασή του, περιγράφεται μονοσήμαντα από ένα συγκεκριμένο πολυώνυμο μίας μεταβλητής με συγκεκριμένους συντελεστές 0 ή 1, που ονομάζεται χαρακτηριστικό πολυώνυμο LFSR.

Κάθε LFSR μεγέθους N είναι πρωταρχικός αν και μόνο αν το χαρακτηριστικό του πολυώνυμο είναι πρωταρχικό στο πεπερασμένο σώμα GF(2^n) [14].

Ευτυχώς γνωρίζουμε ποια πολυώνυμα είναι πρωταρχικά και άρα μπορούμε πολύ εύκολα να παράγουμε πρωταρχικούς LFSR για οποιοδήποτε N, επιλέγοντας την κατάλληλη ανάδραση έτσι ώστε να αντιστοιχεί σε πρωταρχικό πολυώνυμο. Δεν θα σταθούμε στην παρούσα διατριβή σε περισσότερες μαθηματικές λεπτομέρειες περί πρωταρχικών πολυωνύμων και πεπερασμένων σωμάτων – θα κρατήσουμε μόνο το ότι γνωρίζουμε πώς να κατασκευάζουμε LFSR μεγίστης περιόδου.

3.4.3 Ιδιότητες πρωταρχικών γραμμικών καταχωρητών ολίσθησης με ανάδραση

Οι πρωταρχικοί LFSR παράγουν ακολουθίες με τη μέγιστη δυνατή περίοδο και αυτό είναι πολύ θετικό διότι, όπως αναφέρθηκε και νωρίτερα, δεν επιθυμούμε η κλειδοροή να επαναλαμβάνεται. Επίσης έχει αποδειχτεί ότι οποιαδήποτε ακολουθία παράγεται από πρωταρχικό LFSR, δηλαδή **οποιαδήποτε m-sequence, ικανοποιεί τα κριτήρια τυχαιότητας του Golomb**, που προαναφέρθηκαν [14], γεγονός που αναδεικνύει τη σπουδαιότητα των LFSR ως γεννήτριες κλειδοροής.

Άρα, θα μπορούσαμε να υποθέσουμε σε αυτό το βήμα, ότι έχουμε βρει μία πάρα πολύ καλή γεννήτρια παραγωγής τυχαίας ακολουθίας, ή αλλιώς μία γεννήτρια κλειδοροής, επειδή είναι εύκολα υλοποιήσιμη, παράγει ακολουθία με καλά χαρακτηριστικά τυχαιότητας και επίσης η ακολουθία που παράγει έχει αποδεδειγμένα μεγάλη περίοδο, εφόσον βέβαια ο LFSR είναι πρωταρχικός.

Παρόλα αυτά η απευθείας χρήση LFSR χωρίς κάποια τροποποίηση θα καταστήσει τον κρυπτογραφικό αλγόριθμο ευάλωτο σε επιθέσεις τύπου γνωστού μηνύματος (known plaintext attack), επειδή σε έναν οποιονδήποτε αλγόριθμο ροής, αν ξέρουμε ένα τμήμα του μηνύματος, τότε στην ουσία ξέρουμε απευθείας το αντίστοιχο τμήμα της κλειδοροής. Αν θεωρήσουμε ότι η γεννήτρια κλειδοροής είναι γνωστή σε όλους (μια υπόθεση που ισχύει στα σύγχρονα κρυπτογραφικά συστήματα), τότε συμπεραίνουμε ότι, εάν η κλειδοροή παράγεται από έναν LFSR μεγέθους N, τότε αν ξέρουμε N bits του αρχικού μηνύματος αυτομάτως θα γνωρίζουμε το αντίστοιχο τμήμα της τρέχουσας κατάστασής του. Αυτό είναι άμεση απόρροια της χρήσης της πράξης XOR που είδαμε πιο πάνω. Συνεπώς, θα μπορούμε να υπολογίσουμε ολόκληρο το υπόλοιπο τμήμα της κλειδοροής.

Το συμπέρασμα είναι ότι ένας LFSR δεν μπορεί από μόνος του να χρησιμοποιηθεί ως γεννήτρια παραγωγής ψευδοτυχαίας κλειδοροής: θα πρέπει η γεννήτρια να έχει σύνθετη δομή που να μην επιτρέπει την εύκολη εύρεση της αρχικής κατάστασης με επιθέσεις τύπου γνωστού κειμένου (known plaintext).

Αν ωστόσο παραχθεί, με κάποια πιο σύνθετη δομή, μία ακολουθία με μεγάλη περίοδο που πληροί και τα κριτήρια του Golomb ή αν, π.χ., ο LFSR παραμένει μυστικός (κάτι που συνηθίζεται, για παράδειγμα, σε στρατιωτικές εφαρμογές), θα είναι το σύστημα ασφαλές;

Η απάντηση είναι ΌΧΙ, γιατί τα κριτήρια του Golomb από μόνα τους δεν αρκούν για να χαρακτηριστεί μία ακολουθία ως ψευδοτυχαία. Αυτό γίνεται φανερό στην επόμενη ενότητα.

3.4.4 Γραμμική πολυπλοκότητα ακολουθίας

Η γραμμική πολυπλοκότητα (linear complexity ή linear span) είναι ένα κρυπτογραφικό κριτήριο ακολουθιών που σχετίζεται με την παραγωγή αυτών από γραμμικούς καταχωρητές ολίσθησης. Ορίζεται ως το μέγεθος του μικρότερου LFSR ο οποίος παράγει την ακολουθία (αφού, κατά κανόνα, μία ακολουθία μπορεί να παραχθεί από πολλούς διαφορετικούς LFSR).

Είναι προφανές ότι η γραμμική πολυπλοκότητα ακολουθίας μεγίστης περιόδου $2^N - 1$ είναι N. Δηλαδή, δεν μπορεί να υπάρχει μικρότερος LFSR μεγέθους $K < N$ που παράγει την ακολουθία $2^N - 1$, γιατί η μέγιστη ακολουθία που μπορεί να παράγει θα είναι $2^K - 1$.

Όμως η γραμμική πολυπλοκότητα μίας ακολουθίας παρουσιάζει εξαιρετικό κρυπτογραφικό ενδιαφέρον για τους εξής λόγους:

Υπάρχει ένας πολύ γνωστός αλγόριθμος με την ονομασία **Berlekamp - Massey** (από τα ονόματα των δύο ερευνητών που εργάστηκαν ανεξάρτητα πάνω στο ζήτημα, με διαφορετική αφετηρία ο καθένας), που δοθείσας μίας ακολουθίας, υπολογίζει όχι μόνο τη γραμμική πολυπλοκότητα αλλά και τον μικρότερο σε μέγεθος LFSR που την παράγει. Είναι ένας πολύ γρήγορος αλγόριθμος, με πολυωνυμική υπολογιστική πολυπλοκότητα [14, Κεφ. 6].

Η εύλογη ερώτηση που θα μπορούσε να θέσει κανείς είναι, αν ο LFSR που υπολογίζει ο αλγόριθμος Berlrkamp - Massey είναι μοναδικός. Και η απάντηση είναι, Όχι, δεν είναι μοναδικός, αλλά ο αλγόριθμος βρίσκει έναν μόνο LFSR. Στην περίπτωση όμως που η περίοδος μίας ακολουθίας είναι N και η γραμμική την πολυπλοκότητα L, τέτοια ώστε $L < {}^{N}/_{2}$, τότε ο μικρότερος LFSR μήκους L που την παράγει είναι μοναδικός.

Το σπουδαίο στον αλγόριθμο Berlekamp - Massey είναι ότι αν η γραμμική πολυπλοκότητα μίας ακολουθίας είναι L, τότε ο αλγόριθμος χρειάζεται μόνο 2L διαδοχικά bits για να υπολογίσει τον LFSR ελαχίστου μήκους που την παράγει. Αν ο LFSR αυτός είναι μοναδικός, τότε ουσιαστικά έχουμε βρει μία γεννήτρια όλης της ακολουθίας. Αυτό με απλά λόγια σημαίνει ότι αν γνωρίζουμε 2L διαδοχικά bits της ακολουθίας τότε με την βοήθεια του αλγόριθμου Berlekamp - Massey μπορούμε να βρούμε ολόκληρη την ακολουθία, δηλαδή ολόκληρη την κλειδοροή, απλά εκτελώντας τον αλγόριθμο για το τμήμα 2L της ακολουθίας που γνωρίζουμε [22].

Άρα μπορούμε να αποφανθούμε αν τελικά οι ακολουθίες μεγίστου μήκους είναι ασφαλείς; Και η απάντηση είναι απλή. Μία ακολουθία μεγίστου μήκους έχει περίοδο $2^L - 1$ και γραμμική πολυπλοκότητα L, παρόλο όμως που έχει μεγάλη περίοδο, αν γνωρίζουμε 2L διαδοχικά bits αυτής, τότε την υπολογίζουμε ολόκληρη, οπότε μπορούμε να πούμε δεν είναι ασφαλής.

Επομένως, υπάρχει πλήθος κριτηρίων που πρέπει να ικανοποιούνται από μία ακολουθία προκειμένου να μην είναι προβλέψιμη. Ένα από τα βασικότερα κριτήρια συνεπώς είναι η γραμμική πολυπλοκότητα.

Τέλος, υπάρχει και το προφίλ γραμμικής πολυπλοκότητας που εκφράζει πώς μεταβάλλεται η τιμή της γραμμικής πολυπλοκότητας, καθώς «διατρέχουμε» bit προς bit την ακολουθία. Έχει αποδειχθεί ότι, για μία τυχαία ακολουθία, αναμένουμε το προφίλ της να είναι «κοντά» στην τιμή $^{n}/_{2}$, αλλά οι μεταβολές γύρω απ΄ αυτή την τιμή να είναι «ακανόνιστες» [14].

Λόγω ακριβώς της ανάγκης παραγωγής ακολουθιών υψηλής γραμμικής πολυπλοκότητας, για πολλές δεκαετίες οι γεννήτριες κλειδοροής βασίζονται σε LFSR (για τις λοιπές καλές κρυπτογραφικές τους ιδιότητες), στους οποίους όμως εφαρμόζονται κατάλληλα μη γραμμικές συναρτήσεις προκειμένου η παραγόμενη κλειδοροή να έχει υψηλή γραμμική πολυπλοκότητα. Αυτό είναι το αντικείμενο των επόμενων ενοτήτων.

3.4.5 Λογικές συναρτήσεις

Οι LFSR αποτελούν συστήματα με καλά χαρακτηριστικά, όπως εύκολη υλοποίηση σε hardware, απαιτούν μικρή κατανάλωση ισχύος, επιτυγχάνουν υψηλές ταχύτητες λειτουργίας και έχουν στέρεο μαθηματικό υπόβαθρο, δηλαδή γνωρίζουμε τις ιδιότητες των ακολουθιών που τις παράγουν.

Ωστόσο, οι LFSR από μόνοι τους, παράγουν ακολουθίες χαμηλής γραμμικής πολυπλοκότητας, γι' αυτό πρέπει να συνδυαστούν με πιο σύνθετες δομές, που να επιτυγχάνουν την παραγωγή ακολουθιών υψηλής γραμμικής πολυπλοκότητας. Τέτοιες δομές είναι η μη γραμμικές λογικές συναρτήσεις.

Για να γίνει σαφές το πιο πάνω ας σκεφτούμε έναν πρωταρχικό LFSR μεγέθους 128 bits που παράγει μία ακολουθία, δηλαδή μια κλειδοροή, μέγιστης περιόδου ίσης με $2^{128} - 1$. Ωστόσο η γραμμική πολυπλοκότητα αυτής είναι μόνο 128, που πρακτικά σημαίνει ότι αν κάποιος γνωρίζει 256 διαδοχικά bits της ακολουθίας αυτής, τότε μπορεί να βρει ολόκληρη την ακολουθίας, δηλαδή ολόκληρη την κλειδοροή, με την βοήθεια του αλγόριθμου Berkelamp - Massey.

Μία λογική συνάρτηση (Boolean function) είναι μία συνάρτηση N μεταβλητών, με τιμές στο σώμα $\{0,1\}$ και περιγράφονται από τον πίνακα αληθείας τους.

Μία λογική συνάρτηση μπορεί να αναπαρασταθεί με διάφορες μαθηματικές εκφράσεις, η πιο συνήθης για κρυπτογραφικές εφαρμογές είναι η αλγεβρική κανονική μορφή, που είναι το άθροισμα γινομένων μεταβλητών (XOR).

Η *Αλγεβρική Κανονική Μορφή* μίας συνάρτησης f, πηγάζει από τον πίνακα αληθείας της και έχει τη μορφή $f = x_1 \oplus x_2 \oplus x_3$, αν για παράδειγμα ο πίνακας είναι:

x_1	x_2	x_3	Έξοδος
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Πίνακας 3-2: Πίνακας αληθείας συνάρτησης $f = x_1 \oplus x_2 \oplus x_3$.

Η κατασκευή είναι απλή:

- Η f ισούται με 1 σε τέσσερις περιπτώσεις (001,010,100,111)
- Η 001 περιγράφεται με το γινόμενο μεταβλητών $(1 \oplus x_1)(1 \oplus x_2)x_3$
- Αντίστοιχα, η 010 περιγράφεται με το γινόμενο $(1 \oplus x_1)x_2(1 \oplus x_3)$
- Με αντίστοιχο τρόπο βρίσκουμε και τα γινόμενα των 100, 111

Αλγεβρική Κανονική Μορφή: Άθροισμα όλων των ανωτέρω: $(1 \oplus x_1)(1 \oplus x_2)x_3 + (1 \oplus x_1)x_2(1 \oplus x_3) + x_1(1 \oplus x_2)(1 \oplus x_3) + x_1x_2x_3 = x_1 \oplus x_2 \oplus x_3$

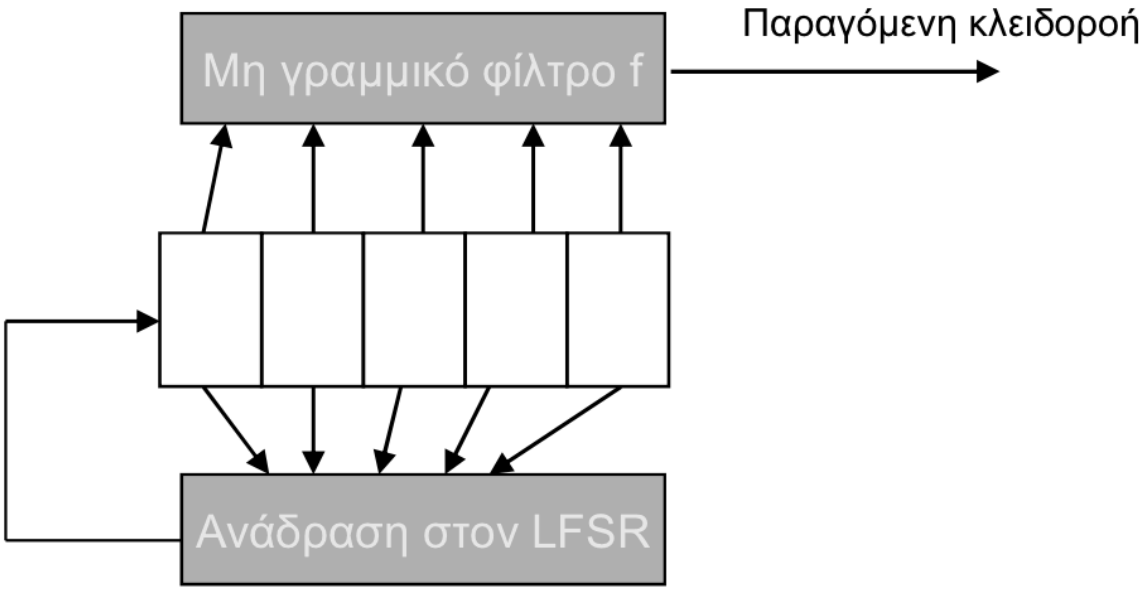
Οι ιδιότητες των λογικών συναρτήσεων είναι πολύ βασικές και εκείνες που χαρακτηρίζουν την συνάρτηση ως «καλή» ή «κακή» και είναι:

- Το Βάρος (weight) μίας συνάρτησης ($wt(f)$), που είναι το πλήθος των «1» στην έξοδο του πίνακα αληθείας της.
- Αν $wt(f) = 2^{(n-1)}$, τότε η συνάρτηση λέγεται ισοβαρής (balanced), που σημαίνει ότι έχει ισοκατενεμημένο πλήθος «0» και «1» στην έξοδό της.

- Ο βαθμός μίας συνάρτησης f $(deg(f))$, είναι το πλήθος των μεταβλητών που εμφανίζονται στο μεγαλύτερο γινόμενο στην *Αλγεβρική Κανονική Μορφή* της. Αν ο βαθμός μίας συνάρτησης είναι 1, τότε αυτή λέγεται γραμμική [22].

3.5 Γεννήτριες ψευδοτυχαίων ακολουθιών - Μη γραμμικά φίλτρα

Μια τεχνική που οδηγεί σε δυαδικές ακολουθίες μεγάλης γραμμικής πολυπλοκότητας είναι η εφαρμογή μιας μη γραμμικής λογικής συνάρτησης στις βαθμίδες ενός πρωταρχικού LFSR, όπως απεικονίζεται στην παρακάτω Εικόνα 3-3:



Εικόνα 3-3: Πρωταρχικός LFSR με μη γραμμικό φίλτρο.

Ο LFSR λειτουργεί κατά τον κλασικό του τρόπο λειτουργίας (μετάβαση από κατάσταση σε κατάσταση), αλλά η κλειδοροή δεν λαμβάνεται από την έξοδό του (δεξιότερο bit) αλλά από την συνάρτηση f. Η συνάρτηση f καλείται μη γραμμικό φίλτρο (non-linear filter function). Ιδανικά, το μη γραμμικό φίλτρο θα πρέπει να είναι ισοβαρής συνάρτηση, έτσι ώστε να εξασφαλίζεται ομοιόμορφη κατανομή των bits 0 ή 1 στην παραγόμενη κλειδοροή. Με πρωταρχικό LFSR, μπορεί να εξασφαλιστεί και η μέγιστη δυνατή περίοδος της κλειδοροής. Στη γενική περίπτωση, οι παραγόμενες ακολουθίες έχουν μεγάλη περίοδο και υψηλή γραμμική πολυπλοκότητα.

Αν $def(f) = d$ και το μέγεθος του LFSR είναι N, τότε η μέγιστη τιμή που μπορεί να έχει η γραμμική πολυπλοκότητα της κλειδοροής είναι:

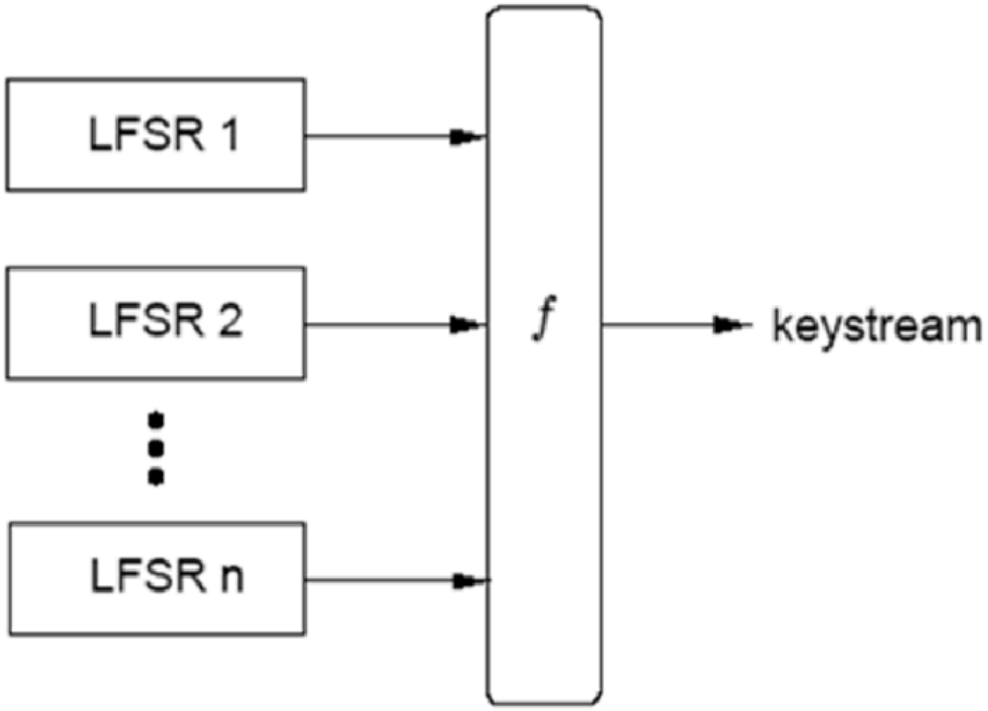
$$\sum_{i=1}^{d} \binom{N}{i}, \text{όπου } \binom{N}{i} = \frac{N!}{i \cdot (N-i)!}$$

Παρατηρούμε ότι η συνάρτηση f πρέπει να έχει υψηλό βαθμό d και ο LFSR να έχει αρκετά μεγάλο μέγεθος, ώστε να πετύχουμε υψηλή γραμμική πολυπλοκότητα.

Γενικά δεν μπορεί να προσδιοριστεί, δοθείσας f, η ακριβής τιμή της γραμμικής πολυπλοκότητας, υπάρχουνε ωστόσο συγκεκριμένες κατασκευές συναρτήσεων που, αν χρησιμοποιηθούν ως μη γραμμικά φίλτρα, παράγουν κλειδοροή εγγυημένα πολύ υψηλής γραμμικής πολυπλοκότητας. Δηλαδή αποδεικνύεται μαθηματικά ένα υψηλό κάτω φράγμα για την τιμή της γραμμικής πολυπλοκότητας.

3.6 Γεννήτριες ψευδοτυχαίων ακολουθιών - Μη γραμμικοί συνδυαστές

Μια δεύτερη τεχνική για την εξάλειψη της εγγενούς γραμμικότητας των LFSRs είναι ο συνδυασμός πολλών LFSRs, με τρόπο ώστε οι έξοδοί τους να τροφοδοτούν μια μη γραμμική λογική συνάρτηση, που καλείται συνάρτηση συνδυαστής, όπως φαίνεται στην παρακάτω Εικόνα 3-4.



Εικόνα 3-4: Πρωταρχικοί LFSRs με μη γραμμικό συνδυαστή.

Οι LFSRs που επιλέγονται για την κατασκευή τέτοιων συστημάτων είναι πρωταρχικοί, λόγω των καλών ιδιοτήτων και της υψηλής περιόδου που έχουν. Με κατάλληλη επιλογή της συνάρτησης f και των LFSRs, μπορεί να διασφαλιστεί μεγάλη περίοδος της κλειδοροής, που στην ιδανική περίπτωση είναι ίση με το ελάχιστο κοινό πολλαπλάσιο των περιόδων των ακολουθιών που παράγουν οι LFSR. Αυτό συμβαίνει αν τα μεγέθη των LFSR είναι πρώτα μεταξύ τους. Χαρακτηριστικό παράδειγμα είναι η γεννήτρια Geffe [09], όπου για παράδειγμα αν η συνάρτηση του γραμμικού συνδυαστή είναι η $f = x_1x_2 \oplus x_2x_3 \oplus x_3$ τότε η γραμμική πολυπλοκότητα της ακολουθίας είναι $L_1L_2 + L_2L_3 + L_3$, όπου $L_1, L_{2,}L_3$ είναι οι επιμέρους γραμμικές πολυπλοκότητες των πρωταρχικών LFSR που χρησιμοποιούνται στη γεννήτρια Geffe και είναι ανά δύο διαφορετικές μεταξύ τους. Επομένως και σε αυτήν την περίπτωση βλέπουμε ότι ο βαθμός της f πρέπει να είναι μεγάλος, ώστε να επιτυγχάνεται μεγάλη γραμμική πολυπλοκότητα.

3.7 Επιθέσεις στις γνωστές γεννήτριες παραγωγής ψευδοτυχαίων ακολουθιών

Ωστόσο οι γεννήτριες που έχουμε δει πιο πάνω δεν είναι άτρωτες. Με την πάροδο του χρόνου, έχουνε μελετηθεί σε βαθμό τέτοιο, ώστε να βρεθούνε αρκετές αδυναμίες με σκοπό να πετύχουνε

επιθέσεις σε αυτές σε ικανοποιητικό βαθμό. Η αποκάλυψη και μελέτη των επιθέσεων αυτών, παγίωσε το σκοπό της έρευνας για νέες λύσεις, πιο ανθεκτικές σε γνωστές επιθέσεις ίσως και άτρωτες.

Υπάρχουνε μεγάλες γνωστές οικογένειες επιθέσεων, που θα αναφέρουμε ενδεικτικά παρακάτω, χωρίς να κάνουμε λεπτομερή ανάλυση.

Επιθέσεις συσχέτισης:

Στο άρθρο "Cryptanalysts representasion of nonlinearly filtered m-sequences" του T. Siegenthaler [19] αναλύεται μια μέθοδος κρυπτανάλυσης που βασίζεται σε ιδιότητες της συνάρτησης ετεροσυσχέτισης (cross correlation function) ανάμεσα στην ακολουθία μέγιστου μήκους που παράγεται από τον LFSR και στην ακολουθία εξόδου του συστήματος. Με απλά λόγια, η έξοδος της συνάρτησης εξαρτάται από κάποιες συγκεκριμένες τιμές εισόδου, δηλαδή αν η έξοδος κάποιου από τους LFSR ταυτίζεται με πιθανότητα $p > 1/2$, τότε αν ένα ικανοποιητικά μεγάλο τμήμα της κλειδοροής είναι γνωστό, μπορεί να ευρεθεί η αρχική κατάσταση του εν λόγω καταχωρητή. Η τεχνική αυτή, που καλείται επίθεση συσχέτισης (correlation attack), βελτιώνεται περαιτέρω στο άρθρο "A fast correlation attack on nonlinearly feedforward filtered shift-register sequences" του R. Forre [07], όπου επιπροσθέτως αναδεικνύονται κάποιες ιδιότητες που πρέπει να έχουν τα συστήματα αυτά προκειμένου να είναι ανθεκτικά σε επιθέσεις συσχέτισης. Μεταξύ άλλων, ιδιότητες που πρέπει να πληρούνται είναι οι εξής:

- το χαρακτηριστικό πολυώνυμο του LFSR (δηλαδή η μαθηματική περιγραφή της συνάρτησης ανάδρασης) να αποτελείται από πολλούς όρους
- η μη γραμμικότητα του φίλτρου πρέπει να είναι υψηλή,
- δεν πρέπει να υπάρχουν πολλοί μηδενικοί όροι στο μετασχηματισμό Walsh του φίλτρου (ο μετασχηματισμός Walsh δεν θα μελετηθεί εδώ) [21].

Επιθέσεις προσεγγίσεων:

Η επίθεση προσέγγισης (approximation attack) είναι μια μέθοδος κρυπτανάλυσης και εφαρμόζεται αν η συνάρτηση f μπορεί να προσεγγιστεί ικανοποιητικά από μία συνάρτηση χαμηλότερου βαθμού. Η επίθεση αυτή αποτελεί γενίκευση της επίθεσης βέλτιστων γραμμικών προσεγγίσεων (best affine approximation attack).

Αλγεβρικές επιθέσεις:

Αν L το μήκος του LFSR, τότε κάθε bit της ακολουθίας κλειδιού μπορεί να γραφεί ως μια συνάρτηση των L bits της αρχικής κατάστασης. Συνεπώς, γνώση N στοιχείων της κλειδοροής επιτρέπει τον προσδιορισμό της αρχικής κατάστασης του LFSR μέσω επίλυσης ενός μη γραμμικού συστήματος N εξισώσεων με L αγνώστους. Τεχνικές που αποσκοπούν στην επίλυση τέτοιων συστημάτων καλούνται αλγεβρικές επιθέσεις (algebraic attacks).

Για να μειωθεί ο βαθμός των μη γραμμικών εξισώσεων του συστήματος, πρέπει η συνάρτηση φίλτρου g να ικανοποιεί κάποια απ' τις εξής ιδιότητες:

- υπάρχει συνάρτηση f χαμηλού βαθμού τέτοια ώστε $g * f = h$, όπου η h είναι χαμηλού βαθμού,
- υπάρχει συνάρτηση f χαμηλού βαθμού τέτοια ώστε $g * f = 0$,
- υπάρχει συνάρτηση f τέτοια ώστε $g * f = h$, όπου η h είναι χαμηλού βαθμού,

όπου τα f, g, h συμβολίζουν τους αντίστοιχους πίνακες αληθείας των συναρτήσεων, και το * υποδηλώνει το εσωτερικό γινόμενό τους. Έχει αποδειχθεί ότι οι παραπάνω τρεις ιδιότητες είναι ισοδύναμες με την εξής μία: δεν πρέπει να υπάρχει συνάρτηση χαμηλού βαθμού f τέτοια ώστε είτε $g * f = 0$ ή $(g + 1) * f = 0$. Κάθε τέτοια συνάρτηση f ονομάζεται εκμηδενιστής (annihilator) της g.

Κατά συνέπεια, οι αλγεβρικές επιθέσεις όρισαν ως βασικό κριτήριο σχεδίασης των μη γραμμικών φίλτρων το να μην έχουν εκμηδενιστές χαμηλού βαθμού. Από τη στιγμή που θα κατασκευαστεί ένα μη γραμμικό σύστημα χαμηλού βαθμού, μπορεί να επιλυθεί με διάφορες τεχνικές, οι κυριότερες εκ των οποίων στηρίζονται στις βάσεις Grobner. Οι αλγεβρικές επιθέσεις οδήγησαν στον ορισμό ενός νέου κρυπτογραφικού κριτηρίου για τις λογικές συναρτήσεις, της λεγόμενης αλγεβρικής ανθεκτικότητας (algebraic immunity ή annihilator immunity), η οποία ορίζεται ως ο ελάχιστος βαθμός από όλους τους μη μηδενικούς εκμηδενιστές της g ή της g + 1. Η αλγεβρική ανθεκτικότητα για κάθε συνάρτηση n μεταβλητών είναι μικρότερη ή ίση από $[n/2]$.

Κλασικό παράδειγμα αλγεβρικής επίθεσης αποτελεί ο αλγόριθμος Toyocrypt. Κατά τη διάρκεια του σχεδιασμού του αλγορίθμου αυτού, υπήρχε η πεποίθηση ότι μπορούσε να αντισταθεί σε όλες τις γνωστές επιθέσεις σε stream ciphers. Στον Toyocrypt, έχουμε έναν LFSR που αποτελείται από 128 bits και, συνεπώς, N = 128. Ο βαθμός της συνάρτησης Toyocrypt f, που είναι ένα μη

γραμμικό φίλτρο, είναι ίσος με 62. Ωστόσο, πολλαπλασιάζοντας την συνάρτηση Toyocrypt f με τη γραμμική συνάρτηση $g = 1 + x_{23}$, το αποτέλεσμα θα είναι μία συνάρτηση βαθμού μόλις 3. Το ίδιο ισχύει και με τον πολλαπλασιασμό της συνάρτησης f με την γραμμική συνάρτηση $k = 1 + x_{42}$. Άρα μπορούμε να πούμε ότι $f * g = h$, από όπου μπορεί κανείς να αποδείξει ότι $f(g + h) = 0$, οπότε η αλγεβρική ανθεκτικότητα της f είναι μόλις 3, ακριβώς όσος είναι ο βαθμός της g+h.

Γενικά μπορούμε να πούμε ότι για την σχεδίαση ενός κρυπταλγόριθμου ροής, η έμφαση δίνεται στη σχεδίαση της γεννήτριας κλειδοροής, ως εκ τούτου η γεννήτρια πρέπει να αποτελείται από μη γραμμικές λογικές συναρτήσεις, για να επιτυγχάνεται υψηλή γραμμική πολυπλοκότητα της παραγόμενης κλειδοροής, με αποτέλεσμα να μην είναι προβλέψιμη η κλειδοροή αυτή.

Όμως, από την άλλη πλευρά, οι λογικές συναρτήσεις πρέπει να πληρούν σύνολο άλλων κρυπτογραφικών κριτηρίων, ώστε να αποτρέπονται διάφορες επιθέσεις.

Τέτοια κριτήρια πρωτίστως, πέραν του υψηλού βαθμού, είναι:

- Ανθεκτικότητα σε συσχετίσεις (correlation-immunity)
- Υψηλή μη γραμμικότητα (nonlinearity)
- Αλγεβρική ανθεκτικότητα (algebraic immunity)

Συμπέρασμα είναι ότι η επίτευξη όλων των ανωτέρω κριτηρίων είναι δύσκολος σχεδιαστικός στόχος και γι' αυτό γεννιέται η επιτακτική ανάγκη εξεύρεσης νέων λύσεων, που παράγουν πιο ισχυρές κλειδοροές, που είναι πιο ανθεκτικές έναντι επιθέσεων.

3.8 Μη γραμμικοί καταχωρητές ολίσθησης με ανάδραση

Οι NLFSR είναι νέα τάση παραγωγής ψευδοτυχαίων ακολουθιών και χρησιμοποιούνται για την παραγωγή ψευδοτυχαίων ακολουθιών με «καλά» χαρακτηριστικά. Βέβαια δεν έχουνε μελετηθεί ακόμα επαρκώς, όπως έχουν μελετηθεί οι LFSR. *Γενικά, δεν είναι ακόμα γνωστό πως παράγουμε NLFSR με εγγυημένα μέγιστη περίοδο*. *Το μόνο σίγουρο είναι ότι παράγουν ακολουθίες με υψηλή γραμμική πολυπλοκότητα, μεγαλύτερη από εκείνη των LFSR συγκριτικά πάντα με τον ίδιο αριθμό θέσεων.*

Η χρήση μη γραμμικού FSR (NLFSR) γίνεται διότι η ανάδρασή του δεν είναι μία απλή XOR, αλλά έχει και μη γραμμικές πράξεις (AND), δηλαδή γινόμενα και έτσι μπορούνε να χρησιμοποιηθούνε σε οποιοδήποτε μοντέλο γεννήτριας (μη γραμμικού φίλτρου/ συνδυαστή). Γενική παραδοχή είναι ότι τελικά, δείχνουν να παρέχουν ασφάλεια έναντι των αλγεβρικών επιθέσεων, ωστόσο πολλά ερωτήματα παραμένουν ακόμα ανοικτά:

- Πως κατασκευάζουμε NLSFR που παράγουν ακολουθίες μέγιστης δυνατής περιόδου;
- Είναι ευάλωτοι σε άλλες επιθέσεις;

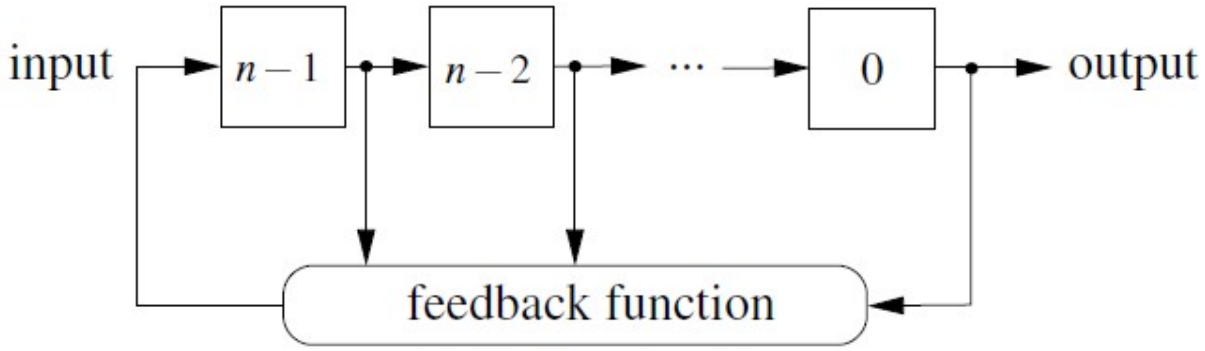
Γενικότερα, δεν υπάρχουν ασφαλείς μέθοδοι για την αποτίμηση της ασφάλειας σε κατασκευές που εμπεριέχουν NLFSR.

3.9 NLFSR που παράγουν ακολουθίες με περίοδο $2^n - 1$

Οι NLFSR είναι μία γενίκευση των LFSR, όπου η δεδομένη κατάσταση προέρχεται από μία μη γραμμική συνάρτηση της προηγούμενης κατάστασης. Ωστόσο οι LFSR έχουν μελετηθεί επαρκώς και γνωρίζουμε όλο το θεωρητικό υπόβαθρο αυτών. Αντίθετα, οι NLFSR δεν έχουν μελετηθεί επαρκώς και παρουσιάζουν ακόμα θεμελιώδη προβλήματα. Το βασικότερο εξ΄ αυτών είναι να βρούμε μία διαδικασία σταθερή που να παράγει με τη χρήση NLFSR, ακολουθίες με εγγυημένα μεγάλη περίοδο. Ωστόσο υπάρχουν αλγόριθμοι που εφαρμόζονται σε μικρούς NLFSR, αλλά επειδή οι NLFSR είναι μικρού μεγέθους, δεν έχουν καμία κρυπτογραφική εφαρμογή, διότι μπορεί η παραγόμενη κλειδοροή να προβλεφθεί εύκολα.

Μία καλή ανάλυση και μελέτη για τη χρήση NLFSR έχει γίνει από την E. Dubrova [05], την οποία θα παρουσιάσουμε πιο κάτω με παραγόμενη ακολουθία περιόδου $2^n - 1$ για $n < 25$. Θα παρουσιάσουμε συνοπτικά τα αποτελέσματα της μελέτης για τρεις διαφορετικές συναρτήσεις ανάδρασης βαθμού δύο.

Η γενική περιγραφή ενός NLFSR περιγράφεται από την παρακάτω Εικόνα 3-5:



Εικόνα 3-5: Γενική δομή ενός NLFSR με n θέσεις.

Οι τρεις τύποι συναρτήσεων ανάδρασης βαθμού δύο είναι:

1. $f(x_0, x_{1,} \dots, x_{n-1}) = x_0 \oplus x_a \oplus x_b \oplus x_c \cdot x_d$

2. $f(x_0, x_{1,} \dots, x_{n-1}) = x_0 \oplus x_a \oplus x_b x_c \oplus x_d \cdot x_e$

3. $f(x_0, x_{1,} \dots, x_{n-1}) = x_0 \oplus x_a \oplus x_b \oplus x_c \oplus x_d \oplus x_e \cdot x_h$

όπου $a, b, c, d, e, h \in \{1,2, \dots, n-1\}$, "$\oplus$" είναι πράξη XOR και " $\cdot$ " είναι πράξη AND.

Η μέγιστη δυνατή περίοδος για έναν n-bit NLFSR είναι 2^n. Αλλά για λόγους απλότητας του κυκλώματος δεν συμπεριλαμβάνεται η μηδενική αρχική κατάσταση, οπότε η περίοδος είναι $2^n - 1$. Βέβαια θα μπορούσε να συμπεριληφθεί και η μηδενική αρχική κατάσταση, προσθέτοντας στην συνάρτηση ανάδρασης f όρους γινόμενα $\bar{x}_{1,} \bar{x}_{2,} \dots, \bar{x}_{n,}$ όπου το $\bar{x}$ ορίζεται ως $\bar{x} = x \oplus 1$.

Ενδεικτικά θα παραθέσουμε μερικά αποτελέσματα των συναρτήσεων ανάδρασης ενός n θέσεων NLFSR που παράγουνε ακολουθίες μεγίστης περιόδου $2^n - 1$:

1. Για n=4, $f(x_0, x_{1,}, x_2, x_3) = x_0 \oplus x_1 \oplus x_2 \oplus x_1 \cdot x_2$

1. Για n=9, $f(x_0, x_{1,} x_{2,} x_{3,}, x_{4,} x_{5,} x_{6,} x_{7,} x_{8,}, x_{9,}) = x_0 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_7 \oplus x_3 \cdot x_7$

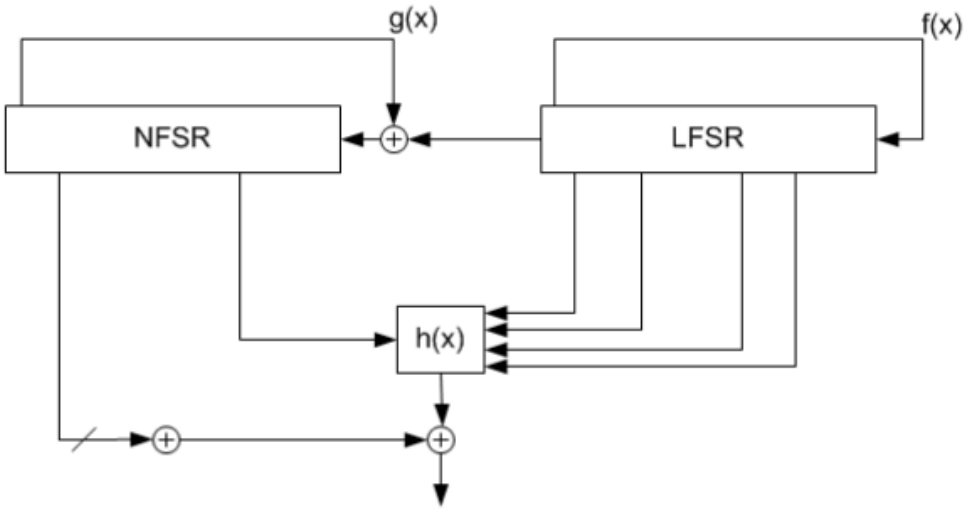
2. Για n=24, $f(x_0, x_{1,} x_{2,} x_{3,}, x_{4,} x_{5,} x_{6,} \dots, x_{23,}) = x_0 \oplus x_1 \oplus x_8 \oplus x_9 \oplus x_{15} \oplus x_7 \cdot x_{18}$

Από αυτή την έρευνα αντλούμε ότι μπορούμε να κατασκευάσουμε ακολουθίες μεγίστης περιόδου $2^n - 1$, με την χρήση NLFSR όμως για $n < 25$ θέσεις NLFSR. Από 25 θέσεις και πάνω δεν είμαστε βέβαιοι πως θα πάρουμε εγγυημένα ακολουθίες μεγίστης περιόδου. Αν συνυπολογίσουμε και τη γραμμική πολυπλοκότητα, καταλαβαίνουμε ότι ο αριθμός 25 είναι μικρός, ώστε να καθησυχάσουμε και να θεωρήσουμε την έρευνα ολοκληρωμένη και ασφαλή. Θέλουμε γεννήτριες που ναι μεν να παράγουμε εγγυημένα ακολουθίες μέγιστης περιόδου με χρήση NLFSR, αλλά να έχουν και μεγάλη γραμμική πολυπλοκότητα [05].

3.10 Πρακτική εφαρμογή NLFSR: Αλγόριθμος Grain

Μία χρήση των NLFSR έχει γίνει στην υλοποίηση του Αλγόριθμου Grain. Βασίστηκε στη λογική της χρήσης ενός γρήγορου, ασφαλούς και απλού κρυπτογραφικού συστήματος. Η χρήση του Grain ως κρυπταλγόριθμου ροής είναι ιδανική για ένα σύστημα με χαμηλή κατανάλωση ισχύος και μικρή διαθέσιμη μνήμη – είναι ένας από τους κρυπταλγορίθμους ροής που επελέγησαν στο πλαίσιο του ερευνητικού προγράμματος eStream [09][Web11], ως κατάλληλοι για χρήση σε ευρύ φάσμα εφαρμογών.

Η δομή κατασκευής του, όπως δείχνει και η Εικόνα 3-6, αποτελείται από:



Εικόνα 3-6: Δομή κατασκευής του αλγόριθμου Grain με χρήση NLFSR.

- Έναν LFSR μήκους 80 θέσεων (80 bit) λόγω της απλότητας και της ταχύτητας που διαθέτει και εγγυάται την ελάχιστη δυνατή περίοδο και όλα τα καλά κριτήρια τυχαιότητας όπως προαναφέραμε
- Έναν NLFSR μήκους 80 θέσεων (80 bit), ο οποίος μαζί με τη χρήση της μη γραμμικής συνάρτησης που διαθέτει, εισάγει μη γραμμικότητα στον κρυπταλγόριθμο.
- Μία μη γραμμική συνάρτηση h(x)

Η συνολική αρχική κατάσταση των καταχωρητών είναι μεγέθους 160 bit: 80bit του LFSR και 80bit του NLFSR. Σε κάθε κύκλο, μια είσοδος από το NFSR και τέσσερις από το LFSR συνδυάζονται από μία μη γραμμική συνάρτηση. Παράγεται μία ακολουθία με περίοδο τουλάχιστον $2^{80} - 1$.

Η είσοδος του NLFSR είναι συνδεδεμένη με την έξοδο του LFSR έτσι ώστε η εκάστοτε κατάσταση των bit του NLFSR να είναι ισοβαρής. Έτσι χρησιμοποιούμε τον NLFSR ως ένα πραγματικό φίλτρο και η γνώση που έχουμε για τους NLFSR δε χρησιμοποιείται άμεσα και εξαντλητικά σε αυτή την υλοποίηση. Αμέσως μετά την έξοδο της συνάρτησης h(x), προστίθενται 7 bit του NLFSR προκειμένου να παραχθεί το bit της κλειδοροής. Το μέγεθος του κλειδιού (από το οποίο δημιουργούνται με συγκεκριμένη διαδικασία οι αρχικές καταστάσεις των FSRs) είναι 80 bit. Η κατασκευή του αλγόριθμου είναι τέτοια που δεν επιτρέπει επίθεση γρηγορότερη από αυτή της εξαντλητικής αναζήτησης του κλειδιού. Επομένως η καλύτερη επίθεση θα χρειαζόταν μία ελάχιστη υπολογιστική ισχύ όχι μικρότερη από τον άμεσο υπολογισμό 2^{80} πιθανών κλειδιών [09].

Θέλουμε να τονίσουμε ότι ο αλγόριθμος Grain είναι ιδιαίτερα ανθεκτικός σε κρυπτανάλυση, αφού υπάρχει η συμβολή του NLFSR και θα αναλύσουμε παρακάτω την ανθεκτικότητά του έναντι δύο γνωστών μεθόδων κρυπτανάλυσης.

Επίθεση συσχέτισης

Είναι γνωστό από τις ακολουθίες μεγίστου μήκους που παράγονται από πρωταρχικούς LFSR, ότι τα bit των ακολουθιών αυτών είναι σχεδόν ισοκατανεμημένα. Αντίθετα, αυτό δε συμβαίνει με τις ακολουθίες που παράγονται από τους NLFSR, οι οποίοι καθοδηγούνται αυτόνομα και χωρίς μέτρο ισοκατανομής. Όπως και να έχει η ανάδραση της συνάρτησης g(x) γίνεται XOR με την εκάστοτε κατάσταση του LFSR και έτσι τα bits του των ακολουθιών που παράγονται από τους

NLSFR ισοκατανέμονται. Επιπρόσθετα, η συνάρτηση g(x) είναι ισοβαρής συνάρτηση, γι΄ αυτό μπορούμε να υποθέσουμε ότι τα bit του NFSR δεν είναι συσχετισμένα με αυτά του NLFSR.

Η συνάρτηση h(x) επιλέγεται έτσι ώστε να είναι ανθεκτική σε επιθέσεις συσχέτισης. Αυτό βέβαια δε σημαίνει ότι δεν αποκλείεται να υπάρχει συσχέτιση της εξόδου της συνάρτησης h(x) με το άθροισμα της εισόδου σε αυτήν. Επειδή η μία είσοδος στην h(x) προέρχεται από τον NLFSR και αμέσως μετά την έξοδο της συνάρτησης h(x), προστίθενται 7 bit του NLFSR, οι συσχετίσεις των bit εξόδου της γεννήτριας με τα bit του αθροίσματος του LFSR είναι λίγες και γι' αυτό αποτρέπεται οποιαδήποτε επίθεση συσχέτισης [09].

Αλγεβρική επίθεση

Μία γεννήτρια με ένα μη γραμμικό φίλτρο με βαθμό 3, θα ήταν από μόνη της αρκετά ευπαθής σε αλγεβρικές επιθέσεις. Αντίθετα, οι αλγεβρικές επιθέσεις δε βρίσκουν καμία ανταπόκριση στην πλήρη γεννήτρια Grain με αρχική κατάσταση 160 bit, επειδή η συνάρτηση ανατροφοδότησης του NLFSR δεν είναι γραμμική και η μετέπειτα κατάσταση των bit του NLFSR ως λειτουργία της αρχικής κατάστασης της γεννήτριας παρουσιάζει από μόνη της μεταβλητό αλλά συνάμα μεγάλο βαθμό αλγεβρικής ανθεκτικότητας. Αυτό γίνεται διότι η συνάρτηση h(x) δέχεται ως είσοδο την κατάσταση του NLFSR και η έξοδος της h(x) γίνεται XOR με τα 7 bit του NLFSR με αποτέλεσμα ο βαθμός της αλγεβρικής ανθεκτικότητας να είναι γενικά μεγάλος και μεταβλητός. Έτσι καταπολεμούνται οι αλγεβρικές επιθέσεις [09].

Παρατηρούμε σε μία γεννήτρια παραγωγής κλειδοροής με την βοήθεια NLFSR ότι περιορίζουμε και ίσως εξαλείφουμε μεγάλες οικογένειες γνωστών επιθέσεων. Αυτή και μόνο η ανακάλυψη μας ωθεί να αναζητήσουμε περισσότερα στοιχεία για τους NLFSR και τις ιδιότητές αυτών.

Στα επόμενα κεφάλαιο θα αναλύσουμε τα στοιχεία που γνωρίζουμε για τους NLFSR και θα εστιάσουμε στην έρευνα που έχουμε κάνει γι΄αυτούς.

Κεφάλαιο 4
De Bruijn ακολουθίες

4.1 Εισαγωγή

Οι μη γραμμικοί καταχωρητές ολίσθησης με ανάδραση ή NLFSR (Non Linear Feedback Shift Registers) είναι όπως αναφέραμε στο προηγούμενο Κεφάλαιο μία γενίκευση των LFSR, όπου η τρέχουσα κατάσταση του LFSR προέρχεται από την επεξεργασία της προηγούμενης κατάστασης μέσω μίας μη γραμμικής συνάρτησης. Ως γνωστό πλέον, όπως προαναφέραμε, ενώ η θεωρία των LFSR έχει καθιερωθεί, δεν μπορούμε να πούμε το ίδιο για τους NLFSR, όπου υπάρχουνε ακόμα θεμελιώδη προβλήματα τα οποία είναι ανοικτά. Ένα από τα πιο βασικά είναι η έρευση μίας συστηματικής διαδικασίας παραγωγής ακολουθιών με εγγυημένα μέγιστη δυνατή περίοδο. Αυτές οι ακολουθίες αποτελούν μία πολύ σημαντική κατηγορία ακολουθιών, τις λεγόμενες ακολουθίες De Bruijn, που θα παρουσιάσουμε στο Κεφάλαιο αυτό και θα δείξουμε την πρόοδο που έχει γίνει σε αυτό τον τομέα.

4.2 De Bruijn ακολουθίες

Οι De Bruijn ακολουθίες πήραν το όνομα τους από τον Δανό μαθηματικό Nicholas De Bruijn. Ο ορισμός της De Bruijn ακολουθίας μας λέει ότι σε μία ακολουθία που αποτελείται από ένα σύνολο τιμών από k-όρους με αριθμό ψηφίων n, κάθε υπακολουθία n τιμών υπάρχει ακριβώς μία φορά σε διάστημα μίας περιόδου της ακολουθίας. Το 1946, ανακάλυψε τον αριθμό των ακολουθιών De Bruijn που προκύπτουν από ένα σύνολο τιμών k-όρων με αριθμό ψηφίων n, και απέδειξε ότι ο τύπος δίνεται από τη σχέση: $((k-1)!)^{k^{n-1}} k^{k^{n-1}-n}$ ή $\frac{(k!)^{k^{n-1}}}{k^n}$ [20].

Οι De Bruijn ακολουθίες περιγράφονται από δύο παραμέτρους:

- k, ο αριθμός του συνόλου τιμών π.χ. {0,1,2,3,4,5,6,7,8,9} για k=10
- n, ο αριθμός των ψηφίων (το μήκος της υπακολουθίας), π.χ. n=4
- Η ακολουθία περιγράφεται ως: B(k,n)

Το βασικό χαρακτηριστικό των ακολουθιών De Bruijn είναι ότι κάθε υπακολουθία n, υπάρχει ακριβώς μία φορά σε μία περίοδο της ακολουθίας De Bruijn.

Για να καταλάβουμε καλύτερα τον τρόπο λειτουργίας των De Bruijn ακολουθιών, θα παραθέσουμε ένα παράδειγμα.

Φανταστείτε λοιπόν ότι έχουμε έναν 4-ψήφιο κωδικό από το σύνολο τιμών των γνωστών δεκαδικών αριθμών {0,1,2,3,4,5,6,7,8,9}. Για να «σπάσει» ο κωδικός αυτός, με τη χρήση της μεθόδου της εξαντλητικής αναζήτησης ή αλλιώς «brute force attack», δηλαδή όλων των δυνατών συνδυασμών, θα χρειαζόταν $10^4 = 10000$ πιθανούς συνδυασμούς, δηλαδή 40000 διαφορετικές πληκτρολογήσεις αριθμών. Θα ήταν όμως ευκολότερο και θα χρειαζόταν λιγότερο χρόνο και κόπο, αν υπήρχε τρόπος να διατρέξουμε ανά 4 ψηφία από μία De Bruijn ακολουθία και να δοκιμάζουμε κάθε φορά την τελευταία τετράδα με επικάλυψη [Web05].

Η περίοδος μίας ακολουθίας De Bruijn περιγράφεται από τη σχέση k^n, δηλαδή στο παράδειγμά μας θα έχουμε περίοδο $10^4 = 10000$.

Θα επανέλθουμε στο παράδειγμα, εφόσον επεξηγήσουμε κάποιες βασικές έννοιες που ακολουθούνε στο παρακάτω υποκεφάλαιο.

4.3 Δυαδικές ακολουθίες De Bruijn

Για τις περισσότερες εφαρμογές, όπως και για το δικό μας σκοπό, οι δυαδικές ακολουθίες De Bruijn, παρουσιάζουν το μεγαλύτερο ενδιαφέρον. Σύμφωνα με τον τύπο των De Bruijn ακολουθιών, αν αντικαταστήσουμε το k=2, θα έχουμε για δυαδικές ακολουθίες τον τύπο $2^{2^{n-1}-n}$, που μας λέει ότι τόσες είναι οι πιθανές ακολουθίες De Bruijn που μπορούν να παραχθούνε για μεταβλητό αριθμό n. Είναι προφανές ότι η περίοδος των δυαδικών ακολουθιών De Bruijn είναι 2^{n}.

Για παράδειγμα, μία De Bruijn ακολουθία για n=4 και με περίοδο $2^4 = 16$ μπορεί να είναι $\{s_t\} = 0000111101100101$ από το σύνολο $2^{2^{4-1}-4} = 16$ ακολουθιών. Κάθε δυαδική 4-άδα εμφανίζεται ακριβώς μία φορά μέσα σε μία περίοδο. Γενικά μπορούμε να πούμε ότι μια δυαδική ακολουθία De Bruijn είναι ισοβαρής, αυτό σημαίνει ότι ο αριθμός των «1» και «0» είναι ισοκατανεμημένος στην ακολουθία μέσα σε μία περίοδο. Επίσης, κάθε De Bruijn ακολουθία ικανοποιεί αρκετά κριτήρια τυχαιότητας. Χρησιμοποιούνται ως πηγή ψευδοτυχαίων ακολουθιών σε μία κλειδοροή στους αλγόριθμους ροής όπως θα δούμε παρακάτω [20].

Ας δούμε όμως ένα απλό αναλυτικό παράδειγμα De Bruijn ακολουθίας: B(2,2) πάντα στο σύνολο k={0,1}.

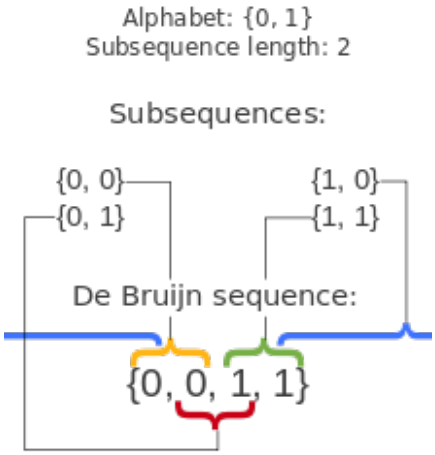
Θέλουμε να παράγουμε μία ακολουθία, η οποία να περιέχει όλες τις πιθανές υπακολουθίες των δύο δυαδικών ψηφίων {0,1}.

Μία λύση είναι η: 0011

Σημειώνουμε εδώ ότι άλλες προφανείς λύσεις De Bruijn ακολουθιών για το παράδειγμα μας μπορεί να είναι τέσσερις και είναι οι: 0011, 0110, 1100, 1001. Ως εκ τούτου θεωρούμε την ακολουθία ως κυκλικό βρόχο και έτσι δεν έχει σημασία ποιο στοιχείο του βρόχου θα θεωρήσουμε αρχικό. Αυτό δεν σημαίνει ότι οι De Bruijn ακολουθίες είναι μοναδικές. Αντιθέτως μπορεί να υπάρχουν πολλές De Bruijn ακολουθίες, αλλά είναι διαφορετικές από αυτές που δημιουργούνται με την επιλογή ενός διαφορετικού αρχικού στοιχείου σε έναν βρόχο μίας υπάρχουσας ακολουθίας.

Τα πρώτα δύο ψηφία μας δίνουν **00**, τα επόμενα δύο είναι **01** και τελευταία **11**. Για να πάρουμε και το **10** πρέπει να κάνουμε «περιτύλιξη», ώστε να πάρουμε το τελευταίο ψηφίο της

ακολουθίας και να το ενώσουμε με το πρώτο, όπως δείχνει η παρακάτω Εικόνα 4-1, και συγκεκριμένα η μπλε αγκύλη.



Εικόνα 4-1: De Bruijn ακολουθία για k=2 και n=2 [Web05].

Αν δεν είναι εφικτή η «περιτύλιξη», απλά «τοποθετούμε» το πρώτο ψηφίο της ακολουθίας μετά το τέλος της ακολουθίας και έχουμε την ακολουθία 00110. Γενικά ισχύει ως κανόνας, να ενώνουμε στο τέλος τα (n-1) αρχικά ψηφία της ακολουθίας, ώστε να παίρνουμε όλες τις πιθανές n-άδες (υπακολουθίες) της ακολουθίας.

Μία άλλη λύση, για το B(2,3) είναι: 00010111

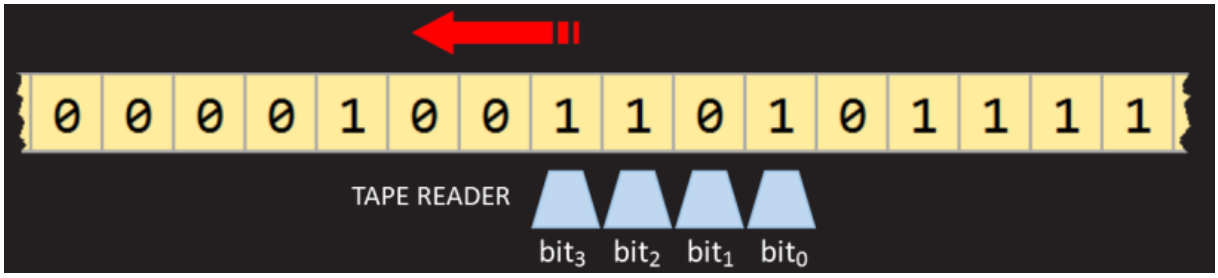
Αρχίζοντας από την αρχή, έχουμε τις 3-ψήφιες υπακολουθίες μέσα στην περίοδο ($2^3 = 8$), που είναι οι: 000, 001, 010, 101, 011, 111 και αν κάνουμε και την περιτύλιξη με τα πρώτα 2 ψηφία, έχουμε την **110** και τελικά την **100**. Όλες οι οκτώ 3-ψήφιες υπακολουθίες είναι παρών στην ακολουθία.

Και μία τελευταία λύση για το B(2,4): 0000100110101111

Μπορούμε να δούμε ότι έχουμε τις 4-ψήφιες υπακολουθίες μέσα στην περίοδο, όπως είναι: 0000, 0001, 0010, 0100, 1001,...

Και εδώ αρχίζουν τα δύσκολα. Όσο αυξάνει το n, δυσκολεύει η άντληση των n-άδων μέσα σε μία περίοδο της ακολουθίας.

Μία πρακτική λύση είναι να μπορούμε να φανταστούμε την De Bruijn ακολουθία γραμμένη σε μία ταινία, που διαπερνάει πάνω από έναν μηχανισμό ανάγνωσης με n κεφαλές, καθώς διατρέχει προς τα αριστερά, μέχρι να διαβαστούν όλες οι πιθανές n-άδες, όπως φαίνεται στην παρακάτω Εικόνα 4-2, για το παράδειγμά των 4-ψήφιων υπακολουθιών [Web04].



Εικόνα 4-2: Μηχανισμός ταινίας De Bruijn ακολουθία για k=2 και n=4.

Κάθε ολίσθηση της ταινίας κατά μία θέση αριστερά μας δίνει μία μοναδική έξοδο, που είναι η κάθε δυνατή υπακολουθία. Αυτή η ακολουθία λοιπόν περνάει από όλες τις καταστάσεις μεταξύ 0000 – 1111 των n-άδων που χρειαζόμαστε. Με απλά λόγια έχουμε περιγράψει πως αντλούμε όλες τις υπακολουθίες από μια δυαδική De Bruijn ακολουθία. Βέβαια για μεγαλύτερες ακολουθίες υπάρχουνε έτοιμα λογισμικά που κάνουν αυτή την διεργασία αυτόματα.

Οι De Bruijn ακολουθίες είναι σημαντικές επειδή:

- Έχουν περίοδο 2^n
- Ο αριθμός των «1» και «0» είναι ισοκατανεμημένος στην ακολουθία σε μία περίοδο
- Ικανοποιούν αρκετά κριτήρια τυχαιότητας
- Κάθε n-άδα (υπακολουθία) εμφανίζεται ακριβώς μία φορά μέσα σε μία περίοδο
- Η De Bruijn ακολουθία είναι αυτή που παράγεται από NLFSR που περνάει από όλες τις καταστάσεις, συμπεριλαμβανομένης της μηδενικής

Ας θεωρήσουμε έναν NLFSR μήκους 3 με την μη γραμμική συνάρτηση $f(x_1, x_2, x_3) = 1 \oplus x_2 \oplus x_3 \oplus x_1x_2$. Ο παρακάτω Πίνακας 4-1, παρουσιάζει τα περιεχόμενα των 3 σταδίων του NLFSR στο τέλος κάθε μονάδας του χρόνου όταν η αρχική κατάσταση είναι [0,0,0].

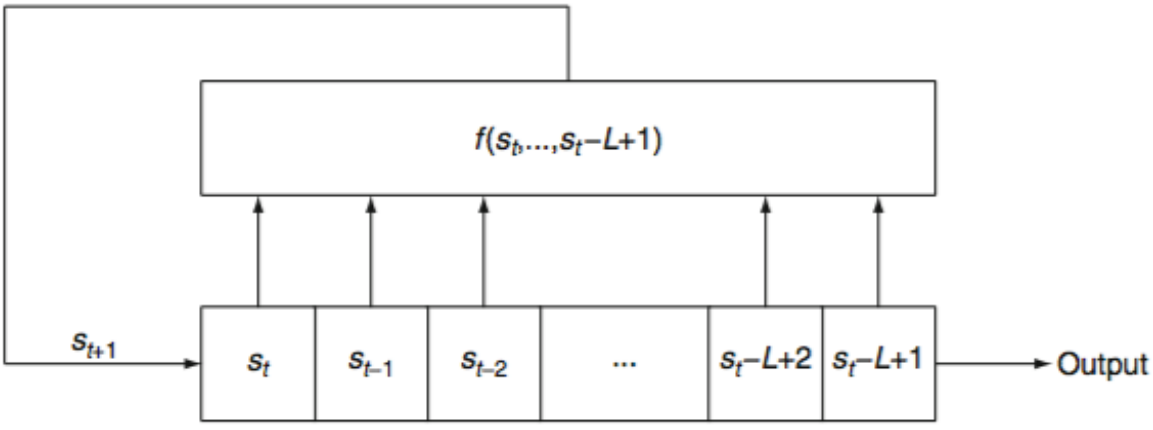
T	Στάδιο 2	Στάδιο 1	Στάδιο 0

0	0	0	0
1	1	0	0
2	1	1	0
3	1	1	1
4	0	1	1
5	1	0	1
6	0	1	0
7	0	0	1

Πίνακας 4-1: Πίνακας περιεχομένων των 3 σταδίων του NLFSR με μη γραμμικη συνάρτηση $f(x_1, x_2, x_3) = 1 \oplus x_2 \oplus x_3 \oplus x_1 x_2$.

Η παραγόμενη ακολουθία είναι με κύκλο 0001110100011101. Συνεπώς ο παραπάνω NLFSR παράγει ακολουθία De Bruijn εφόσον περνάει από όλες τις καταστάσεις και παράγει ακολουθία μεγίστης περιόδου $8=2^3$ [14][Web06].

Ας θεωρήσουμε έναν NLFSR της Εικόνα 4-3, με L=3 θέσεις και την μη γραμμική συνάρτηση ανάδρασης που δίνεται από τον τύπο $f(s_t, s_{t-1}, s_{t-2}) = s_t \cdot s_{t-1} \oplus s_t \oplus s_{t-2}$. Το bit ανάδρασης είναι $s_{t+1} = f(s_t, \dots, s_{t-L+1})$. Σε κάθε χρόνο ρολογιού υπάρχει μετατόπιση, η έξοδος είναι ίση με s_{t-L+1} και το νέο αριστερό bit είναι το s_{t+1}.



Εικόνα 4-3: Μη γραμμικός καταχωρήτης ολίσθησης με ανάδραση – NLFSR.

Ο πίνακας αληθείας της μη γραμμικής συνάρτησης f δίνεται παρακάτω, όπως δείχνει ο Πίνακας 4-2:

s_t	s_{t-1}	s_{t-2}	$f(s_t, s_{t-1}, s_{t-2})$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Πίνακας 4-2: Πίνακας αληθείας της συνάρτησης $f(s_t, s_{t-1}, s_{t-2}) = s_t \cdot s_{t-1} \oplus s_t \oplus s_{t-2}$.

Αν θεωρήσουμε ότι αρχική κατάσταση του NLFSR είναι $s_0 = 0, s_1 = 1, s_2 = 0$, τότε η παραγόμενη ακολουθία θα είναι 010011010011. Συνεπώς ο παραπάνω NLFSR δεν παράγει ακολουθία De Bruijn εφόσον δε περνάει από όλες τις καταστάσεις (περνάει απ΄ όλες εκτός τις 000 και 111) και δεν παράγει ακολουθία μεγίστης περιόδου 8=2^3, αλλά περίοδο 6 [20].

Για να επανέλθουμε στο παράδειγμα με τον 4 ψηφίο pin, έχουμε B(10,4) και ως γνωστό πλέον παράγεται μία ακολουθία με περίοδο 10000. Οπότε αν υπάρχει ένας απλουστευμένος μηχανισμός, όπως περιγράφηκε πιο πάνω με την υποτιθέμενη ταινία, θα έπρεπε η ταινία να διατρέξει πιθανές 4-άδες σε σύνολο 10003 *(διότι κάνουμε «περιτύλιξη» των n=4-1=3 αρχικών ψηφιών της ακολουθίας στο τέλος)* ψηφιών και να σταματήσει όταν βρεθεί ο 4 ψήφιος κωδικός. Η επικάλυψη είναι βασικό στοιχείο καθώς διατρέχει η ταινία, ώστε κάθε φορά να θεωρούνται δεδομένα μόνο τα τελευταία 4 ψηφία που διαβάστηκαν. Σε κάθε άλλη περίπτωση, δηλαδή της εξαντλητικής αναζήτησης, θα έπρεπε να πληκτρολογηθούν στην χειρότερη περίπτωση 40000 αριθμοί, ενώ με την ταινία στην χειρότερη περίπτωση θα είχαμε 10003 πληκτρολογήσεις, που είναι αποτέλεσμα πολύ καλύτερο.

Από τα παραπάνω γίνεται σαφές ότι δυαδικές De Bruijn ακολουθίες έχουν σημαντικό ρόλο στην παραγωγή ψευδοτυχαίων ακολουθιών με τη χρήση γεννητριών.

4.3 Παραγωγή δυαδικών De Bruijn ακολουθιών με χρήση NLFSR

Πρόσφατα, οι μη γραμμικοί καταχωρητές ολίσθησης με ανάδραση ή αλλιώς NLFSR (non-linear feedback shift registers) έχουν προσελκύσει το ερευνητικό ενδιαφέρον για το σχεδιασμό κρυπτογραφικών συστημάτων, όπως είναι οι γεννήτριες παραγωγής ψευδοτυχαίων ακολουθιών στους αλγόριθμους ροής. Ένα καλό παράδειγμα είναι η γεννήτρια Grain όπως έχουμε περιγράψει πιο πάνω στο Κεφάλαιο 3.

Το πρόβλημα όμως έγκειται στη θεωρία των NLFSR, που δεν έχει ολοκληρωθεί μέχρι σήμερα. Δεν έχουν μελετηθεί επαρκώς οι NLFSR. Δε γνωρίζουμε πώς να παράγουμε εγγυημένα ψευδοτυχαίες ακολουθίες με τη χρήση NLFSR με μέγιστη περίοδο. Το μόνο που ξέρουμε είναι ότι παράγουνε ακολουθίες με μεγάλη γραμμική πολυπλοκότητα. Για τη σχεδίαση ενός ασφαλούς κρυπτογραφικού συστήματος, όπως είναι μία γεννήτρια κλειδοροής στους αλγόριθμους ροής, η χρήση ενός αυθαίρετου NLFSR δεν ενδείκνυται, επειδή τα χαρακτηριστικά τυχαιότητας από την ακολουθία που παράγεται, δεν είναι πάντα γνωστά και δεν καθορίζονται εύκολα.

Μια κλασική προσέγγιση της χρήσης NLFSR σε γεννήτρια κλειδοροής είναι να τη συνδυάσουμε με έναν γραμμικό καταχωρητή ολίσθησης με ανάδραση (LFSR), του οποίου γνωρίζουμε τα χαρακτηριστικά και είμαστε βέβαιοι για την περίοδο της παραγόμενης ακολουθίας.

Η δυαδική De Bruijn ακολουθία έχει περίοδο 2^n μέσα στην οποία κάθε n-άδα εμφανίζεται ακριβώς μία φορά. Μία ακολουθία De Bruijn μπορεί να παραχθεί από έναν NLFSR μήκους n θέσεων και τότε η παραγόμενη ακολουθία θα έχει πλέον γνωστές ιδιότητες τυχαιότητας, όπως σίγουρα μεγάλη περίοδο, θα είναι ισοβαρής και η κάθε n-άδα θα εμφανίζεται ακριβώς μία φορά σε διάστημα μίας περιόδου, γνωστή ως ιδιότητα span n.

Ας επανέλθουμε στη γνωστή πλέον γραμμική πολυπλοκότητα μίας ακολουθίας, η οποία ουσιαστικά καθορίζει το μέγεθος του μικρότερου LFSR που μπορεί να παράγει την ακολουθία. Έχει αποδειχθεί ότι οι De Bruijn ακολουθίες έχουν γενικά μεγάλη γραμμική πολυπλοκότητα, η οποία γραμμική πολυπλοκότητα είναι μεγαλύτερη από το ήμισυ της περιόδου της ακολουθίας.

Όπως και να έχει όμως υπάρχει ένα παράδοξο. Αν αφαιρέσουμε ένα μηδενικό από ένα τμήμα με συνεχόμενα μηδενικά της n-υπακολουθίας, της ακολουθίας De Bruijn που παράγεται και έχει περίοδο 2^n, τότε θα πάρουμε μία ακολουθία που ονομάζεται τροποποιημένη ακολουθία De Bruijn ή span n ακολουθία, *(προσοχή: όχι span n ιδιότητα αλλά ακολουθία)* (modified De Bruijn sequence ή span n sequence). Ποιο είναι όμως το παράδοξο; Το παράδοξο είναι ότι η τροποποιημένη De Bruijn ακολουθία έχει τα ίδια χαρακτηριστικά με την αντίστοιχη αρχική De

Bruijn ακολουθία, είναι δηλαδή ισοβαρής και κάθε n-άδα (πλην της μηδενικής) εμφανίζεται ακριβώς μία φορά μέσα στην περίοδό της, αλλά δεν έχει πλέον την ίδια γραμμική πολυπλοκότητα με την αρχική ακολουθία De Bruijn. Η γραμμική πολυπλοκότητα μπορεί να μειωθεί δραματικά.

Ένα παράδειγμα αυτού του φαινομένου είναι μία m-sequence, δηλαδή η ακολουθία που παράγεται από πρωταρχικό LFSR. Η ακολουθία αυτή είναι της ίδιας κλάσης με μία τροποποιημένη ακολουθία De Bruijn, δηλαδή της τάξης μίας span n sequence. Μία πολύ απλή τεχνική παραγωγής De Bruijn ακολουθίας είναι η παραγωγή της από μία ακολουθία m-sequence. Η ακολουθία De Bruijn που θα παραχθεί με αυτό τον τρόπο, θα έχει γραμμική πολυπλοκότητα που είναι τουλάχιστον ίση με $2^{n-1} + n + 1$. Αν ωστόσο αφαιρέσουμε ένα μηδενικό από ένα τμήμα συνεχόμενων μηδενικών μήκους n αυτής της ακολουθίας De Bruijn, τότε προφανώς θα πάρουμε πάλι την m-sequence με γνωστή πλέον γραμμική πολυπλοκότητα ίση με n, όσες είναι και οι βαθμίδες του πρωταρχικού LFSR που την παράγουν. Έτσι το κάτω όριο της γραμμικής πολυπλοκότητας μίας De Bruijn ακολουθίας, από την οποία αφαιρούμε ένα μηδενικό από ένα τμήμα με συνεχόμενα μηδενικά μήκους n, μειώνεται σε βαθμό n. Αυτό δείχνει ότι η γραμμική πολυπλοκότητα μίας De Bruijn ακολουθίας δεν είναι επαρκές κριτήριο για να χαρακτηριστεί η ακολουθία ως προς τα «καλά» χαρακτηριστικά τυχαιότητας. Αντίθετα, θα πρέπει να λαμβάνεται υπόψιν μόνο η γραμμική πολυπλοκότητα της τροποποιημένης ακολουθίας De Bruijn, διότι έχουν ακριβώς μόνο ένα bit διαφορά. Η γραμμική πολυπλοκότητα μίας ακολουθίας ορίζεται το μήκος του μικρότερου LFSR που παράγει την ακολουθία και συγκριτικά για $n \geq 3$, παραθέτουμε τον παρακάτω Πίνακα 4-3:

Ακολουθία	Άνω όριο	Κάτω όριο
De Bruijn	$2^n - 1$	$2^{n-1} + n + 1$
N Span	$2^n - 2$	$> n$

Πίνακας 4-3: Πίνακας γραμμικής πολυπλοκότητας De Bruijn και n span ακολουθίας σε σύγκριση.

Υπάρχει μια «ένα-προς-ένα» αντιστοιχία μεταξύ των De Bruijn ακολουθιών και των τροποποιημένων De Bruijn (span n sequence) ακολουθίων, γιατί μία τροποποιημένη ακολουθία De Bruijn παράγεται άμεσα από μία De Bruijn ακολουθία (αφαιρώντας ένα μηδενικό από ένα τμήμα συνεχόμενων μηδενικών μήκους n της ακολουθίας) και αντίστροφα. Έχουν προταθεί αρκετές τεχνικές και μέθοδοι παραγωγής De Bruijn ακολουθιών από NLFSR, αλλά οι περισσότερες εξ' αυτών δεν είναι αποδοτικές και αποτελεσματικές για την παραγωγή De Bruijn ακολουθιών με περίοδο 2^n με $n \geq 30$ [12].

Ένας FSR με συνάρτηση ανάδρασης $f(s_j, s_{j-1}, \dots, s_{j-L})$ ονομάζεται μη ιδιάζων (non-singular), αν και μόνο αν η f είναι της μορφής $f = s_{j-L} \oplus g(s_{j-1}, s_{j-2}, \dots, s_{j-L+1})$ για μια Boolean συνάρτηση g. Η περίοδος της ακολουθίας εξόδου ενός μη ιδιάζοντος FSR μήκους L είναι το πολύ 2^L. Αν ωστόσο η περίοδος της ακολουθίας εξόδου για οποιαδήποτε αρχική κατάσταση ενός μη ιδιάζοντος FSR μήκους L είναι 2^L τότε ο FSR είναι NLFSR και η ακολουθία που παράγεται (ακολουθία εξόδου) είναι De Bruijn ακολουθία. Μία ακολουθία De Bruijn μπορεί να κατασκευαστεί από έναν NLFSR n θέσεων με αρχική κατάσταση $(s_0, s_1, \dots, s_{n-1})$ και μία μη γραμμική λογική συνάρτηση $f(z_0, z_1, \dots, z_{n-1})$ και θα παραχθεί η ακολουθία $s_{t+n} = f(s_t, s_{t+1}, \dots, s_{t+n-1}), t = 0,1,2, \dots$. Για παράδειγμα η δυαδική ακολουθία De Bruijn $\{s_t\} =$ 0000111101100101 με περίοδο $2^4 = 16$, μπορεί να παραχθεί από την $s_{t+4} = f(s_t, s_{t+1}, s_{t+2}, s_{t+3})$ χρησιμοποιώντας ως αρχική κατάσταση την (0000) και λογική συνάρτηση την $f(z_0, z_1, z_2, z_3) = 1 + z_0 + z_1 + z_1 z_2 z_3$ [20].

4.4 Τεχνικές κατασκευής δυαδικών De Bruijn ακολουθιών

Πέρα από την κρυπτογραφική αξία των ακολουθιών De Bruijn, που περιγράφηκε ανωτέρω, η συγκεκριμένη κατηγορία ακολουθιών έχει ιδιαίτερη σημασία και σε διάφορα άλλα επιστημονικά πεδία. Ειδικότερα, οι ακολουθίες αυτές έχουν ιδιαίτερη σημασία στη ρομποτική, καθώς επίσης και στην ιατρική (μελέτη ακολουθιών DNA). Ως εκ τούτου, έχουν επικεντρώσει για πολλές δεκαετίες το ερευνητικό ενδιαφέρον, ιδίως από τη σκοπιά της εύρεσης αποδοτικών μεθόδων κατασκευής τους.

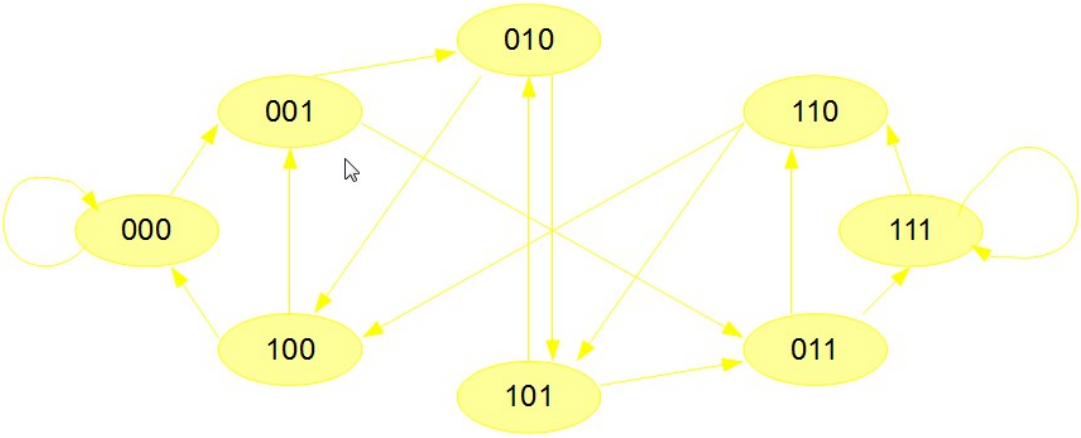
4.4.1 Κατασκευή δυαδικών De Bruijn ακολουθιών με Γράφους

Επειδή ασχολούμαστε καθαρά με δυαδικές ακολουθίες, θα αναλύσουμε την κατασκευή De Bruijn ακολουθιών με αλφάβητο {0,1}, γνωστές ως δυαδικές ακολουθίες De Bruijn.

Μία από τις βασικότερες τεχνικές είναι η κατασκευή ακολουθιών με την βοήθεια των γραφημάτων. Ειδικότερα, ορίζουμε για κάθε ακέραιο αριθμό n (όπου 2^n η περίοδος της ακολουθίας De Bruijn), τον εξής κατευθυνόμενο Γράφο, γνωστός και ως Γράφος De Bruijn: κάθε υπακολουθία μήκους n της ακολουθίας αντιστοιχεί σε έναν κόμβο του Γράφου, έτσι ώστε ο Γράφος να αποτελείται από 2^n κόμβους. Ένας κόμβος συνδέεται με όλους τους πιθανούς

κόμβους, βάσει του εξής κανόνα: ο κόμβος $A = \{a_1, a_2, a_3, \dots, a_n\}$ συνδέεται με τον κόμβο $B = \{b_1, b_2, b_3, \dots, b_n\}$, εφόσον ο κόμβος Β αντιστοιχεί στην υπακολουθία μήκους n, που είναι κατά μία θέση αριστερά ολισθημένη σε σχέση με την υπακολουθία του κόμβου Α ή αλλιώς $b_1 = a_2, b_2 = a_3, b_{n-1} = a_n$. Με απλά λόγια αυτό σημαίνει ότι πρέπει να υπάρχει επικάλυψη των συνδεόμενων κόμβων σε (n-1) bit της υπακολουθίας του κάθε κόμβου.

Ας θεωρήσουμε μία δυαδική ακολουθία De Bruijn με μήκος υπακολουθίας n=3. Αυτό πρακτικά σημαίνει ότι έχουμε μία ακολουθία με περίοδο 8 και όλες οι πιθανές υπακολουθίες της περιόδου, που θα εμφανίζονται μία φορά είναι οκτώ και θα είναι οι υπακολουθίες: 000, 001, 010, 011, 100, 101, 110, 111. Κάθε υπακολουθία αποτελεί και έναν κόμβο του Γραφήματος. Παρακάτω παραθέτουμε την Εικόνα 4-3 , που παρουσιάζει το ολοκληρωμένο κατευθυνόμενο γράφημα De Bruijn.



Εικόνα 4-3: Κατευθυνόμενος δυαδικός Γράφος De Bruijn με 8 κόμβους υπακολουθιών μήκους n=3.

Στο παραπάνω Γράφημα, δύο κόμβοι είναι συνδεδεμένοι ($A \rightarrow B$), μόνο αν τα τελευταία 2 bit της ακολουθίας του κόμβου Α είναι ίδια με τα πρώτα 2 bit της ακολουθίας του κόμβου Β. Αν γενικεύσουμε τον κανόνα θα πρέπει να πούμε ότι δύο κόμβοι συνδέονται μεταξύ τους, μόνο αν τα τελευταία (k-1) bit της ακολουθίας του κόμβου Α είναι ίδια με τα πρώτα (k-1) bit της ακολουθίας του κόμβου Β [Web07].

Ακολουθώντας την διαδρομή: $000 \rightarrow 001 \rightarrow 010 \rightarrow 101 \rightarrow 011 \rightarrow 111 \rightarrow 110 \rightarrow 100$, θα περάσουμε ακριβώς μία φορά από κάθε υπακολουθία μεγέθους n=3 και θα δημιουργήσουμε την ακολουθία De Bruijn (00010111), όπως δείχνουμε παρακάτω με επικαλύψεις:

0	0	0	1	0	1	1	1	0	0
	0	0	1						
		0	1	0					
			1	0	1				
				0	1	1			
					1	1	1		
						1	1	0	
							1	0	0

Η διαδρομή που διαπερνάει κάθε κόμβο του Γράφου De Bruijn ακριβώς μία φορά, παράγει σίγουρα De Bruijn ακολουθίες, όπως δείξαμε πιο πάνω, και είναι γνωστή από τη Θεωρία Γραφημάτων ως «κύκλος Hamilton».

Πρέπει να προσεχθεί όμως, ότι ακολουθώντας ένα «οποιοδήποτε» μονοπάτι ενός Γράφου De Bruijn δεν είναι σίγουρο ότι θα διατρέξουμε όλους τους κόμβους (και άρα, δε θα παραχθεί ακολουθία De Bruijn): Για παράδειγμα, ας θεωρήσουμε τη διαδρομή $001 \rightarrow 011 \rightarrow 110 \rightarrow 100 \rightarrow 001$. Συνεπώς, ο Γράφος De Bruijn είναι μεν βασικό εργαλείο, αλλά από μόνος του δεν αρκεί για την αποδοτική εύρεση De Bruijn ακολουθιών, αφού απαιτείται η εύρεση κύκλων Hamilton εντός του Γράφου. Σε έναν De Bruijn Γράφο που αποτελείται από 2^n κόμβους υπάρχουν $2^{2^{n-1}-n}$ κύκλοι Hamilton, που σημαίνει ουσιαστικά ότι υπάρχουνε ισάριθμες ακολουθίες De Bruijn.

Μία τεχνική κατασκευής De Bruijn ακολουθιών από έναν κατευθυνόμενο Γράφο είναι η συνένωση διαφόρων διαδρομών, «ξένων» μεταξύ τους (δηλ. χωρίς κοινούς κόμβους), ώστε τελικά να προκύψει μία ακολουθία De Bruijn. Η τεχνική αυτή είναι γνωστή ως «συνένωση κύκλων» ή "join cycles" και παράγει εγγυημένα ακολουθίες De Bruijn. Η τεχνική είναι ιδιαίτερα χρήσιμη σε μεγάλες ακολουθίες De Bruijn, τις οποίες παράγουμε αν συνενώσουμε διάφορες μικρότερες διαδρομές ή κύκλους, που διαπερνούν από μερικούς κόμβους, μέχρι να καταλήξουμε στην επιθυμητή De Bruijn ακολουθία.

Ειδικότερα, μπορούμε να θεωρήσουμε μία οποιαδήποτε λογική συνάρτηση $f(x_0, x_1, \dots, x_{n-1})$, τέτοια ώστε η αντιστοίχιση $(x_0, x_1, \dots, x_{n-1}) \rightarrow (x_1, \dots, x_{n-1}, f(x_0, x_1, \dots, x_{n-1}))$, να επιτελεί μία αντιμετάθεση (permutation) πάνω στο χώρο των n-άδων από δυαδικά ψηφία. Με άλλα

λόγια, ισχύει πάντα $f(x_0, x_1, \dots, x_{n-1}) \neq f(x_0', x_1, \dots, x_{n-1})$, όπου $x_0' = 1 + x_0$ το συμπληρωματικό του x_0. Κάθε τέτοια συνάρτηση f μπορεί να παράγει «ξένους» μεταξύ τους κύκλους σε έναν De Bruijn Γράφο, η ένωση των οποίων μπορεί να οδηγήσει σε έναν κύκλο Hamilton και, συνεπώς, σε ακολουθία De Bruijn. Για να ενώσουμε δύο κύκλους πρέπει να υπάρχει τέτοια υπακολουθία $(z_1, z_2, \dots, z_{n-1})$ σε μία διαδρομή, όπου $(z_1, z_2, \dots, z_{n-1}), f(z_1, z_2, \dots, z_{n-1})$ να ανήκει στην ίδια διαδρομή και $(z_1, z_2, \dots, z_{n-1}), (1 + f(z_1, z_2, \dots, z_{n-1}))$ να ανήκει σε διαφορετική διαδρομή. Κατόπιν θα γίνει η συνένωση των δύο διαδρομών συμπληρώνοντας η μία την άλλη [20].

Για παράδειγμα, ας θεωρήσουμε τον πίνακα αληθείας της συνάρτησης f:

x_0	x_1	x_2	$f(x_0, x_1, x_2)$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Πίνακας 4-4: Πίνακας αληθείας της συνάρτησης $f(x_0, x_1, x_2)$.

Αν θεωρήσουμε ως μία διαδρομή την:

$$010 \rightarrow 100 \rightarrow 000 \rightarrow 001 \rightarrow 010$$

και μία δεύτερη διαδρομή την:

$$101 \rightarrow 011 \rightarrow 111 \rightarrow 110 \rightarrow 101$$

και τις συνενώσουμε σε μία, θα πάρουμε την διαδρομή:

$$000 \rightarrow 001 \rightarrow 010 \rightarrow 101 \rightarrow 011 \rightarrow 111 \rightarrow 110 \rightarrow 100$$

που παράγει De Bruijn ακολουθία. (Ουσιαστικά, η συνένωση των δύο κύκλων ισοδυναμεί με την τροποποίηση του πίνακα αληθείας της f σε δύο σημεία και συγκεκριμένα στις θέσεις 010 και 110).

4.4.2 Κατασκευή δυαδικών De Bruijn ακολουθιών με αριθμητική υπολοίπων

Μία εναλλακτική τεχνική κατασκευής μόνο δυαδικών De Bruijn ακολουθιών είναι με την βοήθεια της αριθμητικής υπολοίπων (modular). Για δεδομένο μέγεθος της ακολουθίας μήκους n και αρχική τιμή $a_1 = 2^n - 1$, μπορούμε να παράγουμε μία ακολουθία De Bruijn αν κάνουμε επαναλαμβανόμενες αντικαταστάσεις του αριθμού που παράγεται από τον τύπο: $a_{i+1} = 2a_i \ (mod\ 2^n)$ για τους πρώτους n αριθμούς. Σε περίπτωση που ισχύει $i \leq j, a_i = 2a_j$, τότε ο τύπος δίνεται από τη σχέση: $a_{i+1} = 2a_i + 1 \ (mod\ 2^n)$. Αυτό πρακτικά σημαίνει ότι αν ο πρώτος τύπος δίνει έναν αριθμό που ήδη παράχθηκε, τότε προσθέτουμε ένα 1 σε αυτό τον αριθμό, ώστε να ληφθεί ο νέος αριθμός, με σκοπό να αποκτηθούν και οι επόμενοι αριθμοί. Οι παραγόμενοι αριθμοί, μετατρέπονται σε δυαδικοί μορφή (με κατάλληλη συμπλήρωση μηδενικών όπου απαιτείται) και στο τέλος συνενώνονται ώστε να παραχθεί η ακολουθία De Bruijn.

Ως παράδειγμα θεωρούμε ότι n=3 και θα έχουμε $mod\ 2^n = mod\ 2^3 = 8$. Οπότε:

Τύπος – Αριθμός	**Δυαδική μορφή**
$a_1 = 2^3 - 1 = 7 \ (mod\ 8) = 7$	111
$a_2 = 2 \cdot 7 = 14 \ (mod\ 8) = 6$	110
$a_3 = 2 \cdot 6 = 12 \ (mod\ 8) = 4$	100
$a_4 = 2 \cdot 4 = 8 \ (mod\ 8) = 0$	000
$a_5 = 2 \cdot 0 = 0 \ (mod\ 8) = 0, άρα\ a_5 = 0 + 1 \ (mod\ 8) = 1$	001
$a_6 = 2 \cdot 1 = 2 \ (mod\ 8) = 2$	010
$a_7 = 2 \cdot 2 = 4 \ (mod\ 8) = 4, άρα\ a_7 = 4 + 1 = 5 \ (mod\ 8) = 5$	101
$a_8 = 2 \cdot 5 = 10 \ (mod\ 8) = 2, άρα\ a_8 = 2 + 1 = 3 \ (mod\ 8) = 3$	011

Μπορούμε πλέον να σταματήσουμε, επειδή συγκεντρώσαμε τους πρώτους 2^n αριθμούς, που μας επιτρέπουν την δημιουργία της ακολουθίας De Bruijn, όπως ξέρουμε με το γνωστό τρόπο της επικάλυψης των 2 τελευταίων bit της υπακολουθίας με αυτά των 2 πρώτων bit της επόμενης υπακολουθίας και θα πάρουμε την 11100010 [18].

Κεφάλαιο 5
Δέντρα και Πίνακες Επιθεμάτων Ακολουθιών

5.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα ασχοληθούμε με τις έννοιες των δέντρων επιθεμάτων (suffix trees) και πινάκων επιθεμάτων (suffix arrays) – εφεξής, ST και SA αντίστοιχα. Αυτές οι δύο έννοιες ανήκουν στην κατηγορία των δομών δεικτοδότησης (indexing structures) και εξυπηρετούν στην αποδοτική αντιμετώπιση προβλημάτων που έχουν να κάνουν με συμβολοσειρές, όπως είναι η αναζήτηση του μεγαλύτερου υποτμήματος της ακολουθίας που εμφανίζεται δύο φορές, το οποίο με τη σειρά του έχει εφαρμογή τόσο π.χ. σε ανάλυση ακολουθιών DNA όσο και για τη συμπίεση μιας συμβολοσειράς [04]. Για να έχουμε καλύτερη απόδοση, όταν θέλουμε να επενεργήσουμε σε μια συμβολοσειρά, εργαζόμαστε πάνω στις δομές αυτές και όχι πάνω στην ίδια την συμβολοσειρά. Η έννοιες αυτές βρίσκουν μεγάλη εφαρμογή στην γενετική βιολογία, ωστόσο είναι εργαλείο που μπορεί να βοηθήσει και σε άλλους τομείς. Η ανάλυση που μπορεί να γίνει σε μία ακολουθία με τα εργαλεία αυτά εγείρει το ενδιαφέρον ότι μπορεί να ρίξει φως και στην παραγωγή De Bruijn ακολουθιών με απώτερο στόχο της δημιουργία ισχυρών κλειδοροών. Έτσι θα εστιάσουμε σε αυτή την πτυχή των δομών αυτών που θα μας επιτρέψουν να

πειραματιστούμε στο επόμενο Κεφάλαιο για την κατασκευή ακολουθιών De Bruijn με την βοήθεια αυτών των δομών.

5.2 Επιθέματα ακολουθιών

Πριν ορίσουμε τις δομές των δέντρων και πινάκων επιθεμάτων, είναι συνετό να ορίσουμε τον όρο επίθεμα. Η έννοια του επιθέματος δε θα πρέπει να συγχέεται με τη γενικότερη έννοια της υποσειμβολοσειράς, η οποία δεδομένης μίας συμβολοσειράς $T = \{t_1 t_2 \dots t_n\}$, είναι κάθε υπακολουθία $T' = \{t_{1+i} t_{2+i} \dots t_{m+i}\}, με\ i \geq 0\ και\ n \geq m + i$. Δηλαδή, αν θεωρήσουμε την συμβολοσειρά "banana", μια υποσυμβολοσειρά μπορεί να είναι η "nan". Ωστόσο, ως επίθεμα ορίζουμε οποιαδήποτε υποσυμβολοσειρά μίας συμβολοσειράς, η οποία είναι συνεχόμενη μέχρι και το τελευταίο σύμβολο της συμβολοσειράς – δηλαδή, με τον προηγούμενο συμβολισμό, οποιαδήποτε υποσυμβολοσειρά T' με $m + i = n$. Ως εκ τούτου, το "nana" είναι επίθεμα της συμβολοσειράς "banana", επειδή στην δεδομένη συνεχόμενη υποσυμβολοσειρά εμπεριέχεται το τελευταίο σύμβολο της συμβολοσειράς "banana" που είναι ο "a". Με απλά λόγια αν αφαιρέσουμε μηδέν ή περισσότερα σύμβολα από την αρχή της συμβολοσειράς θα έχουμε το επίθεμα αυτής.

Μία ακόμη υπαρκτή διαφορά που πρέπει να τονιστεί είναι ανάμεσα στην υποσυμβολοσειρά και την υποακολουθία μίας συμβολοσειράς. Μία υπακολουθία είναι μία γενίκευση της υποσυμβολοσειράς και για να γίνει πιο κατανοητό παραθέτουμε την συμβολοσειρά "It was the best of times", όπου μία υπακολουθία είναι "itwastimes". Δεν υπάρχει δηλαδή μία συνέχεια στα σύμβολα, όπως ισχύει στις υποσυμβολοσειρές [Web08].

5.3 Δέντρα Επιθεμάτων

Τα δέντρα επιθεμάτων (suffix trees - ST) είναι δομές δεδομένων με χρήση δεικτών, όπου το κάθε δέντρο κατασκευάζεται από μία συμβολοσειρά, η οποία διασπάται σε επιμέρους επιθέματα. Τα ST περιέχουν ολόκληρο το σύνολο των επιθεμάτων της αρχικής συμβολοσειράς. Η κατασκευή τους γίνεται σε γραμμικό χρόνο (ως προς το μέγεθος της συμβολοσειράς) [04, 06]. Έχουν μελετηθεί ευρέως και χρησιμοποιούνται σε σωρό εφαρμογών.

Το ST για την συμβολοσειρά της μορφής $T = \{t_1 t_2 \dots t_n\}$, που περιέχει έναν αριθμό n συμβόλων, είναι κατευθυνόμενο με n φύλλα, αριθμημένα από 1 έως n, και έχει μία ρίζα. Ένας ειδικός χαρακτήρας τερματισμού «$» τοποθετείται στο τέλος της συμβολοσειράς, πριν γίνει η

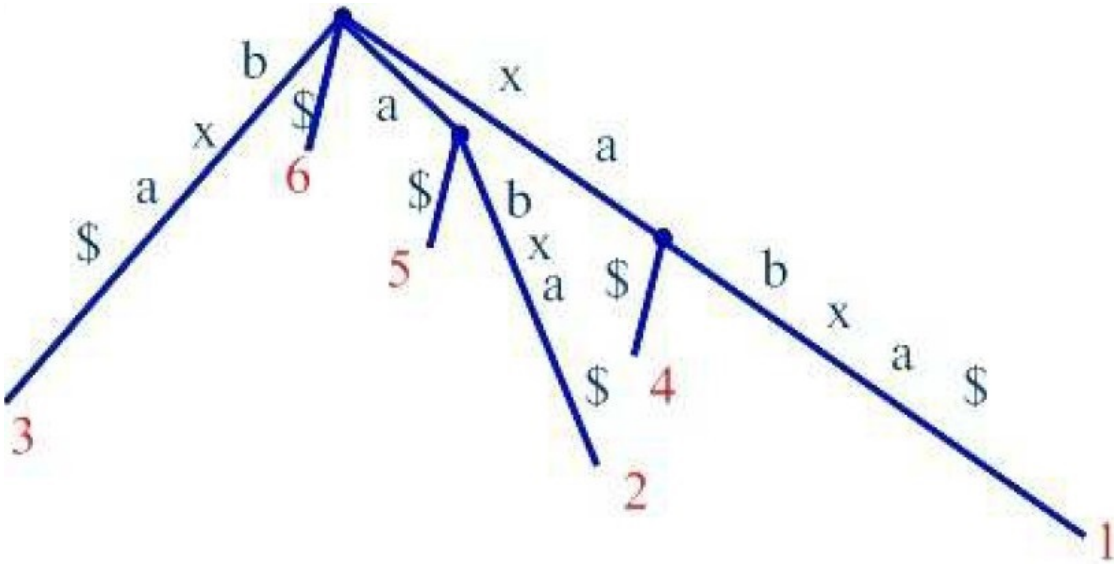
κατασκευή του ST, για να υπάρχει η εγγύηση ότι κάθε επίθεμα (suffix) τερματίζει σε ένα φύλλο του δέντρου. Κάθε εσωτερικός κόμβος του δέντρου, που δεν είναι ρίζα, έχει τουλάχιστον δύο «απόγονους - παιδιά». Κάθε φύλλο του δέντρου έχει την επιγραφή ενός μη κενού επιθέματος της συμβολοσειράς T και ποτέ δύο διαφορετικά φύλλα από τον ίδιο κόμβο δεν αρχίζουν με τον ίδιο χαρακτήρα. Η διαδρομή από την ρίζα μέχρι το φύλλο i ονομάζεται επίθεμα (suffix) $T[i..n]$ [04, 06, 13].

Ας θεωρήσουμε ως παράδειγμα την συμβολοσειρά $D = xabxa\$$, με n=6 σύμβολα (5 συν τον ειδικό χαρακτήρα τερματισμού «$»), οπότε θα έχουμε και 6 φύλλα στο suffix tree και τα αντίστοιχα επιθέματα θα είναι: $xabxa\$, abxa\$, bxa\$, xa\$, a\$ \kappa\alpha\iota\ \$$. Σε κάθε φύλλο αναγράφεται ο αντίστοιχος αριθμός του επιθέματος που ονομάζεται δείκτης. Ο παρακάτω Πίνακας 5-1 μας δείχνει τα επιθέματα δεικτοδοτημένα:

Δείκτης	Επίθεμα
1	$xabxa\$$
2	$abxa\$$
3	$bxa\$$
4	$xa\$$
5	$a\$$
6	$\$$

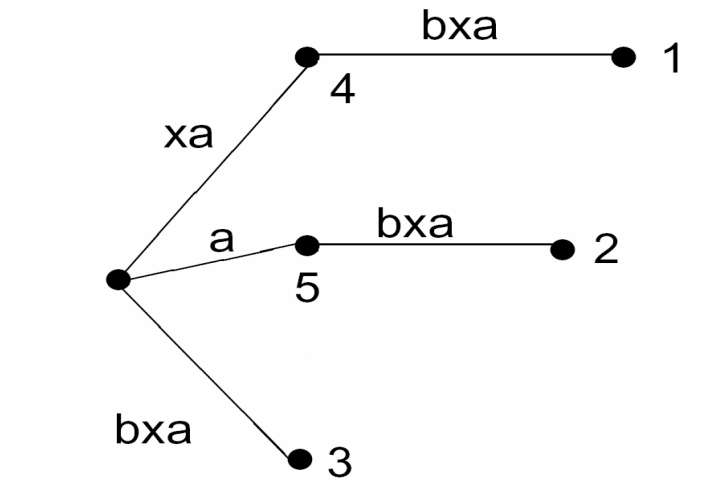
Πίνακας 5-1: Όλα τα επιθέματα (suffixes) της συμβολοσειράς $xabxa\$$.

Σύμφωνα με την περιγραφή, το ST θα είναι αυτό που μας δείχνει η παρακάτω Εικόνα 5-1:



Εικόνα 5-1: Δέντρο επιθεμάτων της συμβολοσειράς $xabxa\$$

Για την σωστή κατασκευή του ST είναι απαραίτητο να μην προηγείται κανένα επίθεμα μικρότερου μεγέθους ενός επιθέματος μεγαλύτερου μεγέθους. Γι' αυτό τοποθετείται ο ειδικός χαρακτήρας τερματισμού «$» στο τέλος κάθε επιθέματος, ώστε να αποφεύγουμε το φαινόμενο αυτό. Αν θεωρήσουμε ότι στο προηγούμενο παράδειγμα δεν είχαμε εισάγει τον ειδικό χαρακτήρα τερματισμού, θα είχαμε δημιουργήσει το δέντρο της παρακάτω Εικόνας 5-2:



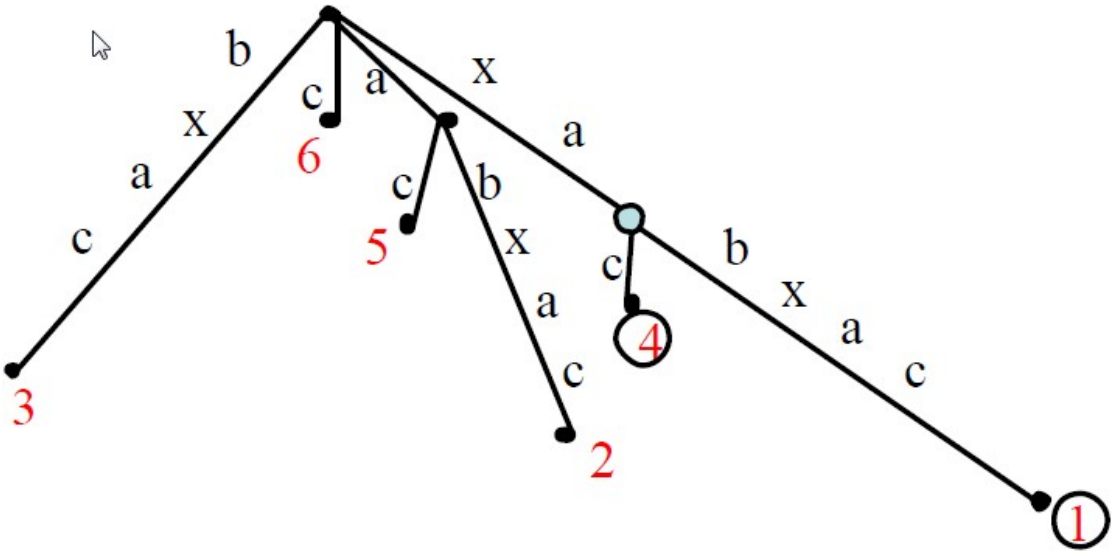
Εικόνα 5-2: Λανθασμένο δέντρο επιθεμάτων της συμβολοσειράς $xabxa$, χωρίς τη χρήση του ειδικού χαρακτήρα τερματισμού «$».

Παρατηρούμε εδώ ότι το 4ο και 5ο επίθεμα, βάσει του πιο πάνω Πίνακα 5-1, δεν παρουσιάζεται ως ανεξάρτητο φύλλο στο δέντρο και το καθένα αποτελείται από μικρότερο επίθεμα που προηγείται ενός μεγαλύτερου επιθέματος. Με απλά λόγια δεν τερματίζει το κάθε επίθεμα ανεξάρτητα. Με την εισαγωγή του ειδικού χαρακτήρα τερματισμού «$», το xa\$ δεν προηγείται του $xabxa$\$, όπως φαίνεται στην Εικόνα 5-1 πιο πάνω.

5.4 Χρήση των δέντρων επιθεμάτων για εύρεση προτύπων

Ιδιαίτερη χρησιμότητα αναδεικνύουν τα ST στην εύρεση προτύπων (patterns) μίας συμβολοσειράς. Αυτό πρακτικά σημαίνει ότι δεδομένης μίας συμβολοσειράς $S = S[1 \dots n]$ αναζητούμε όλα τα πρότυπα $P = P[1 \dots m]$, που εμπεριέχονται μέσα στην συμβολοσειρά S. Αν θεωρήσουμε την συμβολοσειρά $T = xabxac$\$ με τα επιθέματα (suffixes)

$xabxac\$, abxac\$, bxac\$, xac\$, ac\$, c\$ \kappa\alpha\iota\ \$$, θα εξετάσουμε τα πρότυπα $P_1 = xa\$$ και $P_2 = xb\$$ του ST που εμφανίζεται στην παρακάτω Εικόνα 5-3:



Εικόνα 5-3: Δέντρο επιθεμάτων της συμβολοσειράς $xabxac\$$.

Ο χρόνος αναζήτησης των προτύπων $P_1\ \kappa\alpha\iota\ P_2$ είναι γραμμικός επειδή η συμβολοσειρά είναι αρκετά μικρή. Ωστόσο για μεγάλες συμβολοσειρές αυτό δεν ισχύει. Στην πράξη τα ST «καταναλώνουν» πολύ μνήμη και αντικαθίστανται από τους πίνακες επιθεμάτων, οι οποίοι έχουν τις ίδιες ιδιότητες με τα ST και περιορίζουν το χρόνο αναζήτησης σημαντικά.

5.5 Πίνακες επιθεμάτων

Ένας πιο προσφιλής τρόπος εύρεσης προτύπων είναι οι πίνακες επιθεμάτων (suffix arrays - SA). Οι SA είναι μία στενά συνδεδεμένη δομή δεδομένων με δείκτες, που αποτελείται από έναν πίνακα ακεραίων με εύρος από 0 έως n που προσδιορίζει την αλφαβητική σειρά των n επιθεμάτων μίας συμβολοσειράς $T = \{t_1 t_2 \dots t_n\}$. Οι πληροφορίες και των δύο δομών (SA και ST) είναι ουσιαστικά ίδιες. Οι SA όμως είναι ένας εναλλακτικός τρόπος χρήσης των ST. Είναι πιο κατάλληλες δομές τα SA, επειδή χρησιμοποιούν αρκετά λιγότερη μνήμη. Γενικότερα, έχουν τα παρακάτω χαρακτηριστικά:

- Οι SA χρησιμοποιούν 3 έως 5 φορές λιγότερο μνήμη για την ίδια συμβολοσειρά σε σχέση με τα ST

- Οι SA μπορούν να επιλύσουν προβλήματα υποσυμβολοσειρών σχεδόν το ίδιο αποδοτικά όσο και τα ST
- Η χρήση των SA είναι πιο κατάλληλη, όταν το αλφάβητο της συμβολοσειράς είναι πολυπληθές

Υπάρχει δυνατότητα να μετατρέψουμε ένα ST σε ένα SA σε γραμμικό χρόνο[04, 06]. Για την κατασκευή ενός SA κάνουμε χρήση ενός από τους γνωστούς αποδοτικούς αλγόριθμους ταξινόμησης, όπως είναι ο αλγόριθμος της γρήγορης ταξινόμησης ή αλλιώς "quicksort". Ο αλγόριθμος αυτός χρησιμοποιείται για να αναζητεί κάθε ύπαρξη μίας υποσυμβολοσειράς μέσα στη δεδομένη συμβολοσειρά και να τις ταξινομεί σε αλφαβητική σειρά στον πίνακα. Η αναζήτηση της κάθε υποσυμβολοσειράς πρακτικά σημαίνει την εύρεση κάθε επιθέματος που αρχίζει με τον ίδιο χαρακτήρα της υποσυμβολοσειράς.

Ας θεωρήσουμε ως παράδειγμα μία συμβολοσειρά D με μήκος m (αυτό σημαίνει ότι θα έχουμε m επιθέματα – suffixes). Ένα SA είναι μία λίστα από ακέραιους αριθμούς με εύρος από 1 έως m που είναι οι δείκτες του SA, που αντιστοιχούν στην αλφαβητική σειρά των επιθεμάτων του D. Για $D = mississippi\$$ (με m=12) τα επιθέματα είναι όπως δείχνει ο παρακάτω Πίνακας 5-2:

Δείκτης ST	Επίθεμα
0	$mississippi\$$
1	$ississippi\$$
2	$ssissippi\$$
3	$sissippi\$$
4	$issippi\$$
5	$ssippi\$$
6	$sippi\$$
7	$ippi\$$
8	$ppi\$$
9	$pi\$$
10	$i\$$
11	$\$$

Πίνακας 5-2: Όλα τα επιθέματα (suffixes) της συμβολοσειράς $mississippi\$$

Μετά από την αλφαβητική ταξινόμηση θα έχουμε τον παρακάτω SA με προσαρτημένη τη στείλει που δείχνει και την σειρά των επιθεμάτων:

Σειρά	Δείκτης	Επίθεμα
1	10	$i\$$
2	7	$ippi\$$

3	4	$issippi\$$
4	1	$ississippi\$$
5	0	$mississippi\$$
6	9	$pi\$$
7	8	$ppi\$$
8	6	$sippi\$$
9	3	$sissippi\$$
10	5	$ssippi\$$
11	2	$ssissippi\$$
12	11	$\$$

Πίνακας 5-3: Suffix Array της συμβολοσειράς $mississippi\$$

Καταλήγοντας, αν θέλουμε να αναζητήσουμε το πρότυπο που ξεκινάει με τους χαρακτήρες "is" μέσα στη συμβολοσειρά "mississippi", η αναζήτηση θα γίνει με «έναν προς έναν» τους χαρακτήρες, μειώνοντας σημαντικά το εύρος της αναζήτησης, αφού εστιάζουμε στα γράμματα που μας ενδιαφέρουν.

Πρώτο γράμμα: i → εμφανίζεται στις σειρές από 1 έως 4 του SA.

Επομένως το πρότυπο θα ανήκει σε αυτό το εύρος των σειρών του SA.

Δεύτερο γράμμα: s → εμφανίζεται στις σειρές 3 και 4 του SA

Η τομή των δύο ευρών των σειρών είναι: η σειρά 3 και η σειρά 4, άρα:

Το αποτέλεσμα της αναζήτησης θα είναι αντίστοιχα: $issippi, ississippi$

5.6 Εφαρμογές των δέντρων και πινάκων επιθεμάτων

Η δομή των δέντρων και πινάκων επιθεμάτων βρίσκουν ευρεία εφαρμογή σε αλγόριθμους δεικτοδότησης κειμένου (text indexing) και σε αλγόριθμους ταύτισης συμβολοσειρών (String matching) [01]. Με τη χρήση τους επιτυγχάνονται υψηλοί ρυθμοί αναζήτησης κειμένων, φράσεων και ερωτήσεων μέσα σε βάσεις δεδομένων. Πρόσθετα, μπορούμε να πούμε ότι η επιτάχυνση οφείλεται σε μείωση της κατανάλωσης μνήμης, καθώς με τη δεικτοδότηση πετυχαίνουμε μία «καλή» ταξινόμηση των συμβόλων.

Οι δομές αυτές μελετήθηκαν στην παρούσα διατριβή, προκειμένου να αξιοποιηθούν για την ανάλυση των ακολουθιών De Bruijn: αυτό είναι και το αντικείμενο του επόμενου Κεφαλαίου.

Κεφάλαιο 6

Ανάλυση δυαδικών De Bruijn ακολουθιών με χρήση πινάκων επιθεμάτων

6.1 Εισαγωγή

Σε αυτό το Κεφάλαιο θα εισάγουμε μία νέα προσέγγιση για την ανάλυση διαφόρων δυαδικών De Bruijn ακολουθιών, η οποία θα βασίζεται στην αξιοποίηση της δομής των πινάκων επιθεμάτων. Απώτερος στόχος είναι η διερεύνηση του κατά πόσον είναι εφικτό να κατασκευάσει κανείς απευθείας ένα SA, το οποίο θα αντιστοιχεί σε μία ακολουθία De Bruijn: εάν κάτι τέτοιο είναι πράγματι εφικτό, τότε ουσιαστικά θα έχουμε τη δυνατότητα να αναπτύξουμε μία νέα μέθοδο κατασκευής ακολουθιών De Bruijn.

Στο πλαίσιο αυτό, θα καταδείξουμε αρχικά τις ιδιότητες που εμφανίζει το SA μιας De Bruijn ακολουθίας. Προς το σκοπό αυτό, αναπτύχθηκε κατάλληλη εφαρμογή λογισμικού η οποία, δοθείσης μίας οποιασδήποτε δυαδικής ακολουθίας, υπολογίζει το SA αυτής. Ως εκ τούτου, «τροφοδοτώντας» την εν λόγω εφαρμογή με μία ακολουθία De Bruijn – η οποία με τη σειρά της

παράγεται, για τους σκοπούς αυτών των πειραμάτων, με ειδική εφαρμογή λογισμικού – προκύπτει το SA αυτής. Ακολούθως θα γίνει ανάλυση, για τις διάφορες ακολουθίες De Bruijn, των SA αυτών. Η ανάλυση θα γίνει με τη χρήση λογισμικών που δημιουργήθηκαν για αυτόν το σκοπό και θα γίνει καταγραφή των ιδιοτήτων οι οποίες πηγάζουν από τους SA των ακολουθιών. Έτσι, θα εξετάσουμε αν, αξιοποιώντας τις ιδιότητες αυτές, μπορούμε να παράγουμε δυαδικές ακολουθίες De Bruijn αν «εμφυτεύσουμε» αυτές τις ιδιότητες σε έναν SA, με την αντίστροφη δηλαδή λογική. Με απλά λόγια, θα προσπαθήσουμε να κάνουμε τα πρώτα βήματα, ώστε να ανιχνεύσουμε έναν «σίγουρο» τρόπο παραγωγής δυαδικών ακολουθιών De Bruijn από την δομή των SA και κατ΄ επέκταση ισχυρών κλειδοροών ή αλλιώς ψευδοτυχαίων ακολουθιών με καλά χαρακτηριστικά τυχαιότητας, που θα παρέχουν μεγάλο επίπεδο ασφάλειας στους κρυπτογραφικούς αλγόριθμους και θα παράγονται με νέες μεθόδους.

6.2 Πίνακες επιθεμάτων για ακολουθίες De Bruijn

Για το σκοπό της ανάλυσης των δυαδικών De Bruijn ακολουθιών, αναπτύχθηκαν δύο λογισμικά, με τις παρακάτω ιδιότητες:

- Το πρώτο λογισμικό με όνομα *ConstructDeBruijn* αναπτύχθηκε σε γλώσσα Python και δημιουργεί μία δυαδική De Bruijn ακολουθία περιόδου 2^n, εφόσον του δώσουμε την «δύναμη» του n και αυτομάτως παράγει ένα αρχείο newfile.txt, το οποίο αρχείο εμπεριέχει την δυαδική ακολουθία De Bruijn. Για λόγους απλότητας, και χωρίς βλάβη της γενικότητας, η n-άδα από άσσους εμφανίζεται στο τέλος της περιόδου της ακολουθίας.
- Το δεύτερο λογισμικό με όνομα *ConstructSuffixArray* αναπτύχθηκε σε γλώσσα C++ και παράγει την δομή του SA, εφόσον αντλεί την δυαδική ακολουθία De Bruijn από το αρχείο newfile.txt που δημιουργήθηκε από το προηγούμενο λογισμικό. Εμφανίζει ως έξοδο ολόκληρη την ακολουθία και τα επιμέρους επιθέματα με τον αντίστοιχο δείκτη σε αύξουσα σειρά, δηλαδή από το μικρότερο στο μεγαλύτερο επίθεμα καθώς διατρέχουμε τον πίνακα από πάνω προς τα κάτω. Να σημειωθεί εδώ ότι το bit 1 εκλαμβάνεται ως αλφαβητικά μεγαλύτερο του bit 0 και καθώς διατρέχουμε δύο δυαδικές ακολουθίες και συγκρίνουμε τα bit «ένα-προς-ένα», η ακολουθία εκείνη που το bit αυτής έχει την τιμή 1 στην πρώτη θέση που διαφέρουν, θα είναι και η μεγαλύτερη, δηλαδή θα βρίσκεται πιο κάτω στο πίνακα της δομής του SA. Η ιδιαιτερότητα έγκειται όταν έχουμε να συγκρίνουμε δύο ακολουθίες που ταυτίζονται σε όλα τα bit και η μία ακολουθία είναι μεγαλύτερη σε μήκος από την άλλη. Σε αυτή την περίπτωση η μεγαλύτερη σε μήκος ακολουθία είναι και η μεγαλύτερη ακολουθία

στο πίνακα της δομής του SA. Κατόπιν μίας γενικής μελέτης ταξινόμησης επιθεμάτων σε SA, έχει αποδειχθεί ότι ο χρόνος εκτέλεσης του αλγορίθμου δημιουργίας SA είναι γραμμικός, δηλαδή με γραμμική υπολογιστική πολυπλοκότητα $O(n)$[10].

Παρακάτω θα παρουσιάσουμε τους παραγόμενους SA των δυαδικών ακολουθιών De Bruijn για μικρούς αριθμούς n, ώστε να μην έχουμε πολλούς βαθμούς ελευθερίας και να εξάγουμε «σίγουρα» συμπεράσματα. Να υπενθυμίσουμε ότι ο δείκτης του πρώτου επιθέματος κάθε δυαδικής ακολουθίας De Bruijn θα είναι το μηδέν (0). Επίσης, για την ανάλυση των ακολουθιών De Bruijn που επιχειρούμε, δεν χρειάζεται να τοποθετούμε τον ειδικό χαρακτήρα $ στο τέλος της ακολουθίας.

Για n=2, θα έχουμε την ακολουθία: 0011 και τον SA:

Σειρά	Δείκτης	Επίθεμα
1	0	0011
2	1	011
3	3	1
4	2	11

Πίνακας 6-1: Πίνακας Επιθεμάτων της δυαδικής De Bruijn ακολουθίας 0011 για n=2.

Για n=3, θα έχουμε την ακολουθία: 00010111και τον SA:

Σειρά	Δείκτης	Επίθεμα
1	0	00010111
2	1	0010111
3	2	010111
4	4	0111
5	7	1
6	3	10111
7	6	11
8	5	111

Πίνακας 6-2: Πίνακας Επιθεμάτων της δυαδικής De Bruijn ακολουθίας 00010111 για n=3.

Για n=4, θα έχουμε την ακολουθία: 0000100110101111τον SA:

Σειρά	Δείκτης	Επίθεμα
1	0	0000100110101111
2	1	000100110101111
3	2	00100110101111
4	5	00110101111

5	3	0100110101111
6	9	0101111
7	6	0110101111
8	11	01111
9	15	1
10	4	100110101111
11	8	10101111
12	10	101111
13	14	11
14	7	110101111
15	13	111
16	12	1111

Πίνακας 6-3: Πίνακας Επιθεμάτων (Suffix Array) της δυαδικής De Bruijn ακολουθίας 0000100110101111 για n=4.

Για n=5, θα έχουμε την ακολουθία: 00000100011001010011101011011111 SA:

Σειρά	Δείκτης	Επίθεμα
1	0	00000100011001010011101011011111
2	1	0000100011001010011101011011111
3	2	000100011001010011101011011111
4	6	00011001010011101011011111
5	3	001000110010100111010110111111
6	11	00101001110101101111
7	7	001100101001110101101111
8	16	0011101011011111
9	4	010001100101001110101101111
10	14	010011101011011111
11	12	01010011101011011111
12	21	01011011111
13	8	011001010011101011011111
14	23	011011111
15	17	0111010110111111
16	26	011111
17	31	1
18	5	100011001010011101011011111
19	10	10010100111010110111111
20	15	10011101011011111
21	13	10100111010110111111
22	20	101011011111
23	22	1011011111
24	25	1011111
25	30	11
26	9	110010100111010110111111
27	19	1101011011111
28	24	11011111

29	29	111
30	18	11101011011111
31	28	1111
32	27	11111

Πίνακας 6-4: Πίνακας Επιθεμάτων της δυαδικής De Bruijn ακολουθίας 00000100011001010011101011011111 για n=5.

Για n=6, θα έχουμε την ακολουθία: 0000001000011000101000111001001011001101001111010101110110111111 SA:

Σειρά	Δείκτης	Επίθεμα
1	0	0000001000011000101000111001001011001101001111010101110110111111
2	1	000001000011000101000111001001011001101001111010101110110111111
3	2	00001000011000101000111001001011001101001111010101110110111111
4	7	00001100010100011100100101100110100111101010111011011111
5	3	00010000110001010001110010010110011010011110101011101110111111
6	13	00010100011100100101100110100111101010111011111
7	8	00011000101000111001001011001101001111010101110110111111
8	19	00011100100101100110100111101010111011111
9	4	00100001100010100011100100101100110100111101010111011011111
10	25	00100101100110100111101010111011111
11	14	001010001110010010110011010011110101011101111
12	28	00101100110100111101010111011111
13	9	001100010100011100100101100110100111101010111011011111
14	34	00110100111101010111011111
15	20	001110010010110011010011110101011101111
16	40	00111101010111011111
17	5	010000110001010001110010010110011010011110101011101101111111
18	17	01000111001001011001101001111010101110111111
19	26	0100101100110100111101010111011111
20	38	010011110101011101111
21	15	0101000111001001011001101001111010101110111111
22	46	010101110111111
23	29	01011001101001111010101110111111
24	48	0101110111111
25	10	011000101000111001001011001101001111010101110110111111

26	31	01100110100111101010111010111111
27	35	0110100111101010111010111111
28	54	0110111111
29	21	011100100101100110100111101010111010111111
30	50	01110110111111
31	41	011110101011101101111111
32	57	0111111
33	63	1
34	6	1000011000101000111001001011001101001111010101110110111111
35	12	100010100011100100101100110100111101010111010111111
36	18	100011100100101100110100111101010111010111111
37	24	100100101100110100111101010111010111111
38	27	100101100110100111101010111010111111
39	33	100110100111101010111010111111
40	39	100111101010111010111111
41	16	101000111001001011001101001111010101110110111111
42	37	101001111010101110110111111
43	45	1010101110110111111
44	47	10101110110111111
45	30	101100110100111101010111010111111
46	53	10110111111
47	49	101110110111111
48	56	10111111
49	62	11
50	11	110001010001110010010110011010011111010101110110111111
51	23	11001001011001101001111010101110110111111
52	32	110011010011110101011101101111111
53	36	11010011110101011101101111111
54	44	11010101110110111111
55	52	110110111111
56	55	110111111
57	61	111
58	22	111001001011001101001111010101110110111111
59	43	11101010111010111111
60	51	1110110111111
61	60	1111
62	42	11110101011101101111111
63	59	11111
64	58	111111

Πίνακας 6-5: Πίνακας Επιθεμάτων της δυαδικής De Bruijn ακολουθίας 0000001000011000101000111001001011001101001111010101110110111111 για n=6.

6.3 Ιδιότητες των πινάκων επιθεμάτων των ακολουθιών De Bruijn

Θα προσπαθήσουμε να καταγράψουμε κοινές ιδιότητες που εμφανίζονται σε όλους τους πίνακες επιθεμάτων για τους ποικίλους βαθμούς δυνάμεων n των ακολουθιών De Bruijn. Αξίζει να υπενθυμίσουμε ότι, χωρίς βλάβη της γενικότητας, οι δυαδικές ακολουθίες De Bruijn δημιουργήθηκαν έτσι ώστε η κάθε εκάστοτε ακολουθία να λήγει με την n-άδα των άσσων. Οι ιδιότητες που καταγράψαμε μετά από τα αποτελέσματα που μας παρουσιάστηκαν από τα λογισμικά, και η ορθότητα των οποίων για τη γενική περίπτωση μπορεί εύκολα να αποδειχθεί, περιγράφονται στη συνέχεια.

Μία πρώτη βασική παρατήρηση – ιδιότητα είναι:

- Τα επιθέματα στο SA είναι τοποθετημένα πάντα σε αύξουσα σειρά με τέτοιο τρόπο, ώστε κάθε επίθεμα να αρχίζει με την μοναδική n-άδα που υπάρχει σε μία περίοδο κάθε De Bruijn ακολουθίας 2^n . Οι n-άδες αυτές για κάθε πιθανό n ακολουθούν πάντα την ίδια σειρά στον SA, ανεξάρτητα με την De Bruijn ακολουθία. Εκεί όπου δεν είναι συμπληρωμένη n-άδα, επειδή δεν γίνεται η κυκλική συνένωση στην αρχική ακολουθία (γνωστή τεχνική με την ονομασία περιτύλιξη ή «wrapping»), τοποθετούμε τα αντίστοιχα (n-λ) πρώτα bits της αρχικής ακολουθίας, ώστε να συμπληρωθεί η n-άδα.

Για να εξηγήσουμε την προηγούμενη ιδιότητα, θα επικεντρωθούμε σε μία De Bruijn ακολουθία με περίοδο $2^3 = 8$. Η ακολουθία αυτή σίγουρα θα περιέχει όλες τις 3-άδες, όπως 000, 001, 010, 011, 100, 101, 110, 111. Αυτές οι 3-άδες, ανεξάρτητα από την ακολουθία, θα είναι πάντα τοποθετημένες με την ίδια σειρά στον SA. Κατ΄ επέκταση αυτό ισχύει για όλα τα πιθανά n, οπότε για n=3 θα έχουμε το SA:

Σειρά	Δείκτης	επίθεμα
1	*	000…
2	*	001…
3	*	010…
4	*	011…
5	*	100…
6	*	101…
7	6	110…
8	5	111…

Πίνακας 6-6: Πίνακας Επιθεμάτων με όλες τις 3-άδες μίας De Bruijn περιόδου $2^3 = 8$.

Για να ερμηνεύσουμε την τελευταία παρατήρησή μας, θα δημιουργήσουμε τον SA της δυαδικής ακολουθίας De Bruijn για n=3 με περιτύλιξη (wrapping) 00010111(00), που είναι ακριβώς η ίδια με την ακολουθία για n=3, όπως δείξαμε πιο πάνω, με τη μόνη διαφορά να τοποθετήσουμε τα (n-

1=3-1=2) 2 πρώτα bit της ακολουθίας αυτής στο τέλος της ακολουθίας, ώστε να πάρουμε κάθε πιθανή n-άδα, στην προκειμένη περίπτωση, κάθε πιθανή 3-άδα. Οπότε θα έχουμε τον παρακάτω SA:

Σειρά	Δείκτης	επίθεμα
1	9	0
2	8	00
3	0	00010111
4	1	0010111
5	2	010111
6	4	0111
7	7	100
8	3	10111
9	6	1100
10	5	11100

Πίνακας 6-7: Πίνακας Επιθεμάτων της δυαδικής De Bruijn ακολουθίας 00010111 για n=3 με «wrapping», οπότε 00010111(00).

Παρατηρούμε ότι αλλάζει η δομή του SA, που είναι και απολύτως λογικό, ωστόσο οι διαφορές δεν είναι τεράστιες, αφού οι Πίνακες 7-2 και 7-6 διαφέρουν στα εξής σημεία:

- Στη θέση 1 και 2 του Πίνακα 6-6 τοποθετούνται τα επιθέματα με δείκτες 9 και 8 αντίστοιχα, δηλαδή τα (n-1) bits που προσθέσαμε στο τέλος της ακολουθίας, όπως ήταν και αναμενόμενο και τα οποία αγνοούμε παντελώς.
- Στις θέσεις 3 έως 10 του Πίνακα 6-6, υπάρχει μία απόλυτη αντιστοίχιση με τις θέσεις 1 έως 8 του Πίνακα 6-2, σχετικά με τις 3-άδες με τις οποίες αρχίζουν τα επιθέματα, με την μόνη διαφορά ότι στις θέσεις 7, 9 και 10 του Πίνακα 6-6 που αντιστοιχούν στις θέσεις 5, 7 και 8 του Πίνακα 6-2, προσθέτουμε στο τέλος (n-1=3-1=2) τα δύο πρώτα bits της αρχικής ακολουθίας De Bruijn, λόγω της περιτύλιξης.

Αυτό ουσιαστικά είναι πολύ χρήσιμο συμπέρασμα, διότι μας επιτρέπει να αγνοήσουμε την ακολουθία, που σχηματίζεται με περιτύλιξη. Εναλλακτικά χρησιμοποιούμε την ακολουθία χωρίς περιτύλιξη, όπου τοποθετούμε αυθαίρετα τα μηδενικά που λείπουν, ώστε να δημιουργήσουμε τις 3-άδες που θέλουμε, για να έχουμε όλες τις δυνατές 3-άδες της ακολουθίας De Bruijn, που όπως θα δείξουμε παρακάτω ίσως μας βοηθήσουμε να ολοκληρώσουμε κάποιες σκέψεις μας.

<u>Μία δεύτερη ιδιότητα – παρατήρηση είναι:</u>

- Τα επιθέματα τελειώνουν εναλλάξ σε «0» και «1», κοιτάζοντας πάντα τα πρώτα n bits αυτών (αν δηλαδή κοιτάμε μόνο το πρόθεμα των πρώτων n ψηφίων κάθε επιθέματος). Συγκεκριμένα, αν το k είναι περιττός αριθμός (1,3,5,…), τα πρώτα n ψηφία του επιθέματος στη θέση k του SA τελειώνουν σε «0», ενώ αντίστοιχα αν το k είναι άρτιος (0,2,4,6,…), τα πρώτα n ψηφία του επιθέματος στη θέση k του SA τελειώνουν σε «1». Με άλλα λόγια, για k περιττό, αν $SA[k] = m$, τότε η ακολουθία μας στο σημείο $m + n - 1$ αρχίζει με την τιμή «0», ενώ για k άρτιο, η ακολουθία στο σημείο $m + n - 1$ αρχίζει με την τιμή «1». Να αναφέρουμε ότι λόγω της περιτύλιξης ή αλλιώς «wrapping», η ποσότητα $m + n - 1$ εκλαμβάνεται πάντα modulo (2^{n-1}). Άρα, μία πρώτη ιδιότητα που έχουμε είναι:
- Για οποιοδήποτε k, αν $SA[k] = m$, τότε,

  Για $k = 1,3,5, \ldots$ (μονό): το SA θα πάρει την τιμή $(m + n - 1)$ $(mod\ 2^{n-1})$ σε κάποιο k' μικρότερο του 2^{n-1}, δηλαδή αν $SA[k'] = m + n - 1$, τότε $k' < 2^{n-1}$.

  Για $k = 0, 2,4,6, \ldots$ (ζυγό): το SA θα πάρει την τιμή $(m + n - 1)$ $(mod\ 2^{n-1})$ σε κάποιο k' μεγαλύτερο ή ίσο του 2^{n-1}, δηλαδή αν $SA[k'] = m + n - 1$, τότε $k' \geq 2^{n-1}$.

Η παραπάνω ιδιότητα επιβεβαιώνεται και πειραματικά.

Για παράδειγμα, έστω ότι έχουμε την De Bruijn ακολουθία για n=3 (00010111) και τη δομή του SA όπως δείχνει ο Πίνακας 6-2, οπότε,

στη θέση SA[1] έχουμε το επίθεμα που αρχίζει με 000, συνεπώς αν SA[1]=k, τότε η ακολουθία μας στη θέση k+2 έχει την τιμή «0» (το 2 εδώ είναι το (n-1) του γενικού τύπου της ιδιότητας).

στη θέση SA[2] έχουμε το επίθεμα που αρχίζει με 001, συνεπώς αν SA[2]=m, τότε η ακολουθία μας στη θέση m+2 έχει την τιμή «1».

Γενικότερα λοιπόν, στις μονές θέσεις του SA (1,3,5,...) έχουμε επιθέματα που αρχίζουν με **0, και στις ζυγές θέσεις του SA έχουμε επιθέματα που αρχίζουν από **1. Ας σκεφτούμε τη σειρά των επιθεμάτων που είναι: 000, 001, 010, 011, 100, 101, 110, 111. Τότε πάντα το τελευταίο bit αλλάζει.

Επανερχόμαστε πάλι στα παραπάνω: Αφού η ακολουθία μας στη θέση (k+2) έχει την τιμή «0», σε ποια θέση του SA θα βάλουμε το επίθεμα που αρχίζει από την τιμή k+2; Αναγκαστικά σε

κάποια που βρίσκεται στο πρώτο μισό του SA, γιατί εκεί βρίσκονται όλα τα επιθέματα που αρχίζουν με «0».

Αντίστοιχα, η τιμή (m+2) στο SA θα τεθεί σε θέση του στο δεύτερο μισό, γιατί εκεί αντιστοιχούν όλα τα επιθέματα που αρχίζουν με «1». Αυτό το απλό μας λέει η ιδιότητα αυτή.

Οπότε μία γενική παρατήρηση είναι:

- Οι υπακολουθίες-επιθέματα κάθε δυαδικής ακολουθίας De Bruijn μήκους 2^n (περίοδος) είναι ταξινομημένες στο SA. Κατά συνέπεια, στις πρώτες μισές θέσεις του SA (2^{n-1}) αντιστοιχούν οι υπακολουθίες που αρχίζουν με μηδέν και ακολουθούνε αυτές που αρχίζουν με άσσο κ.ο.κ..

Μία τρίτη βασική ιδιότητα – παρατήρηση είναι:

- Η υπακολουθία-επίθεμα που αρχίζει με συνεχόμενη n-άδα από μηδενικά και έχει δείκτη k είναι πάντα στη πρώτη θέση του SA, δηλαδή είναι στην SA[1] και ακολουθείται στην επόμενη θέση του SA, δηλαδή στην SA[2], πάντα από το μοναδικό επίθεμα που αρχίζει με συνεχόμενα (n-1) μηδενικά και με δείκτη επιθέματος πάντα το k+1. Συνοπτικά αν $SA[1] = k, \tau\acute{o}\tau\varepsilon\ SA[2] = k + 1$.

Συμπέρασμα: Αν $n > 2$, οι δύο πρώτες θέσεις του SA, δηλαδή οι SA[1], SA[2], που δομείται από οποιαδήποτε δυαδική ακολουθία De Bruijn της μορφής (s=0****111), είναι δεσμευμένες με τρόπο που περιγράφηκε από τα παραπάνω συμπεράσματα.

Επόμενες παρατηρήσεις – ιδιότητες είναι:

- Αν $n > 2$, οι υπακολουθίες-επιθέματα που αντιστοιχούν στις θέσεις από το SA[1] μέχρι και το $SA[2^{n-1}]$ (δηλαδή τη μεσαία θέση του SA) είναι οι υπόλοιπες που αρχίζουν με μηδέν και ταξινομημένες σε αύξουσα σειρά.
- Αν $n > 2$, οι υπακολουθίες-επιθέματα που αντιστοιχούν στις θέσεις από το $SA[2^{n-1} + 1]$ (δηλαδή από την επόμενη της μεσαίας θέσης) μέχρι και το $SA[2^n]$ είναι οι υπόλοιπες που αρχίζουν με άσσο και ταξινομημένες σε αύξουσα σειρά.
- Σε κάθε SA που δομείται από μία δυαδική ακολουθία De Bruijn στο $SA[2^n]$, δηλαδή στην τελευταία θέση του πίνακα αντιστοιχεί η υπακολουθία-επίθεμα που ξεκινά με n άσσους.
- Σε κάθε SA που παράγεται από μία δυαδική ακολουθία De Bruijn στο $SA[2^n - 1]$, δηλαδή στην προτελευταία θέση του πίνακα αντιστοιχεί η υπακολουθία που ξεκινά με (n-1) άσσους.

Μία τελευταία βασική παρατήρηση – ιδιότητα είναι:

Αν $SA[1] = k$ και $SA[2^n] = m$, τότε για να είναι η ακολουθία De Bruijn πρέπει υποχρεωτικά να ισχύει: $m - k > n - 1$ (και αυτό γιατί στη θέση k της ακολουθίας ξεκινάει υποσυμβολοσειρά με n μηδενικά, ενώ στη θέση m της ακολουθίας ξεκινά υποσυμβολοσειρά με n άσσους).

Η παραπάνω ιδιότητα μπορεί προφανώς να γενικευτεί όχι μόνο για $SA[2^n] = m$ αλλά για οποιοδήποτε $SA[\ j\ ]$, εφόσον $j \geq 2^n - 1$.

6.4 Κατασκευή De Bruijn ακολουθίες από Πίνακες Επιθεμάτων

Για κάθε SA μίας δυαδικής ακολουθίας De Bruijn περιόδου 2^n, ξέρουμε με ακρίβεια τα πρώτα n bits κάθε επιθέματος στο SA. Με αυτό το δεδομένο θα προσπαθήσουμε να κατασκευάσουμε ένα SA που να περιγράφει De Bruijn ακολουθία.

6.4.1 Κατασκευή De Bruijn ακολουθιών για n=3

Σε πρώτο επίπεδο, θα επιχειρήσουμε να εξετάσουμε την περίπτωση μικρής ακολουθία Be Bruijn, για n=3.

Για ακολουθία περιόδου $2^3 = 8$, όποια και αν είναι η De Bruijn ακολουθία, στο SA θα τοποθετηθούν κατά σειρά τα επιθέματα που αρχίζουν από τα 000, 001, 010, 011, 100, 101, 110, 111. Θα τοποθετηθούνε σε έναν πίνακα, ώστε να έχουμε μια συνοπτική εικόνα που θα μας βοηθήσει στον συλλογισμό μας παρακάτω.

Με δεδομένο αυτό θα προσπαθήσουμε να κάνουμε μία αυτόνομη προσέγγιση, που να βασίζεται στις ιδιότητες που αναφέραμε πιο πάνω, με σκοπό την κατασκευή De Bruijn ακολουθιών από ην δομή των SA.

Χωρίς βλάβη της γενικότητας, θεωρούμε ότι η τριπλέτα 111 εμφανίζεται στο τέλος της περιόδου της ακολουθίας s, δηλαδή s=0****111 (όπου με * συμβολίζουμε το άγνωστο bit – σημειώνεται ότι υποχρεωτικά το πρώτο bit της ακολουθίας s θα ισούται με 0).

Επειδή η τριπλέτα 111 είναι στο τέλος της ακολουθίας και βάσει των ιδιοτήτων που περιγράψαμε ανωτέρω, βρίσκεται στην τελευταία θέση του SA[8] (δηλαδή SA[8]=5), είμαστε βέβαιοι ότι η 3-άδα 110 με τους δύο (n-1) άσσους , δηλαδή τους δύο άσσους, θα βρίσκεται στην προτελευταία θέση του SA[7] (δηλαδή SA[7]=6).

Επίσης θα πρέπει στη θέση 4 του SA, δηλαδή στο SA[4] να βάλουμε την τιμή 4, αφού το 6 και 5 είναι στο τέλος αυτού, δηλαδή στις θέσεις SA[7] και SA[8] αντίστοιχα, διότι αυτό μεταφράζεται ότι η τριάδα 111 είναι στο τέλος, οπότε αναγκαστικά θα πρέπει να προηγείται το 0, δηλαδή θα πρέπει η ακολουθία να πάρει τη μορφή s=0***0111, αλλά αυτό θα το δούμε αναλυτικά πιο κάτω και θα το τεκμηριώσουμε. Θα δούμε και την εναλλακτική, αν δηλαδή η s ήταν s=0***1111, τι θα γινόταν, που είναι οφθαλμοφανές από τώρα ότι η s=0***1111 δεν αποτελεί De Bruijn ακολουθία.

Το SA αυτής θα ικανοποιεί τα εξής:

Σειρά	Δείκτης	Υπακολουθία-Επίθεμα
1	*	000…
2	*	001…
3	*	010…
4	4	011…
5	*	100…
6	*	101…
7	6	110…
8	5	111…

Πίνακας 6-8: Πίνακας Επιθεμάτων με όλες τις 3-άδες μίας De Bruijn περιόδου $2^3 = 8$ και τους «τοποθετημένους» δείκτες προς σχηματισμό De Bruijn ακολουθίας.

Πλέον, αρχίζουμε να κατασκευάζουμε το SA.

Από τις παραπάνω ιδιότητες, έχουμε ότι αν SA[1]=k, τότε SA[2]=k+1.

Επειδή SA[1]=k και SA[2]=k+1, θα μπορούσαμε να εξετάσουμε τις παραδοχές για k=0 έως k=2, διότι αν $SA[1] = k \; και \; SA[8] = 5$, τότε θα πρέπει $5 - k > 3 - 1 \Rightarrow k < 3$.

Συνεπώς, το k μπορεί, κατ' αρχάς, να λάβει τιμές από 0 έως 2.

<u>Περίπτωση 1: k=0</u>

Από τις ιδιότητες μπορούμε να συμπεράνουμε ότι αν η ακολουθία αρχίζει από την 3-άδα με τα μηδενικά (k=0), τότε το SA και η αντίστοιχη ακολουθία δομούνται ως εξής:

Σειρά	Δείκτης	Επίθεμα
1	0	000…
2	1	001…
3	*	010…
4	4	011…
5	*	100…
6	*	101…
7	6	110…
8	5	111…

Πίνακας 6-9α: Πίνακας Επιθεμάτων με όλες τις 3-άδες μίας De Bruijn περιόδου $2^3 = 8$ και τους επιπρόσθετους «τοποθετημένους» δείκτες προς σχηματισμό De Bruijn ακολουθίας για k=0.

και s=0001*111.

Αυτό σημαίνει ότι SA[5]=7, συνεπώς έχουμε:

Σειρά	Δείκτης	Επίθεμα
1	0	000…
2	1	001…
3	*	010…
4	4	011…
5	7	100…
6	*	101…
7	6	110…
8	5	111…

Πίνακας 6-9α_1: Πίνακας Επιθεμάτων με όλες τις 3-άδες μίας De Bruijn περιόδου $2^3 = 8$ και τους επιπρόσθετους «τοποθετημένους» δείκτες προς σχηματισμό De Bruijn ακολουθίας για k=0.

Με βάση το ανωτέρω SA, η μόνη πιθανή τιμή για την τιμή του δείκτη 2 είναι το SA[3].

Οπότε η τιμή του SA[3] είναι 2 (αντιστοιχεί στην υπακολουθία-επίθεμα που αρχίζει από 010). Σε αυτήν την περίπτωση, έχουμε:

Σειρά	Δείκτης	Επίθεμα
1	0	000…
2	1	001…
3	2	010…
4	4	011…

5	7	100…
6	*	101…
7	6	110…
8	5	111…

Πίνακας 6-9β: Πίνακας Επιθεμάτων (Suffix Array) με όλες τις 3-άδες μίας De Bruijn περιόδου $2^3 = 8$ και τους επιπρόσθετους «τοποθετημένους» δείκτες προς σχηματισμό De Bruijn ακολουθίας για k=0.

και **s=00010111** και ο τελικός πίνακας θα είναι:

Σειρά	Δείκτης	Επίθεμα
1	0	000…
2	1	001…
3	2	010…
4	4	011…
5	7	100…
6	3	101…
7	6	110…
8	5	111

Πίνακας 6-9γ: Πίνακας Επιθεμάτων με όλες τις 3-άδες μίας De Bruijn περιόδου $2^3 = 8$ και «τοποθετημένους» όλους τους δείκτες με σχηματισμένη De Bruijn ακολουθία για k=0.

Η s είναι πράγματι De Bruijn, όπως μπορεί να επιβεβαιωθεί και από τη συμπλήρωση του SA που ακολουθεί (αφού κάθε επίθεμα αναγνωρίζεται εντός αυτής):

Σειρά	Δείκτης	Επίθεμα
1	0	00010111
2	1	0010111
3	2	010111
4	4	0111
5	7	100
6	3	10111
7	6	110
8	5	111

Πίνακας 6-9δ: Τελικός πίνακας Επιθεμάτων με όλες τις 3-άδες της 00010111 De Bruijn ακολουθίας περιόδου $2^3 = 8$.

Περίπτωση 2: k=1

Δεδομένου ότι έχουμε την ακολουθία s=0***0111 τότε, το SA και η αντίστοιχη ακολουθία, δομούνται με βάση τις ιδιότητες, ως εξής:

Σειρά	Δείκτης	Επίθεμα
1	1	000…
2	2	001…
3	*	010…
4	4	011…
5	*	100…
6	*	101…
7	6	110…
8	5	111…

Πίνακας 6-10α: Πίνακας Επιθεμάτων με όλες τις 3-άδες μίας De Bruijn περιόδου $2^3 = 8$ και τους «τοποθετημένους» δείκτες προς σχηματισμό De Bruijn ακολουθίας για k=1.

και s=00001111. Προφανώς η ακολουθία δεν είναι De Bruijn, κάτι που μπορεί να επιβεβαιωθεί και από το γεγονός ότι δεν μπορεί να ολοκληρωθεί ο σχηματισμός του SA:

Σειρά	Δείκτης	Επίθεμα
1	1	000…
2	2	001…
3	---	010…
4	4	011…
5	7	100…
6	---	101…
7	6	110…
8	5	111…

Πίνακας 6-10β: Πίνακας Επιθεμάτων με όλες τις 3-άδες μίας De Bruijn περιόδου $2^3 = 8$ και τους επιπρόσθετα «τοποθετημένους» δείκτες προς σχηματισμό De Bruijn ακολουθίας για k=1.

<u>Περίπτωση 3: k=2</u>

Τότε, το SA και η αντίστοιχη ακολουθία δομούνται ως εξής:

Σειρά	Δείκτης	Επίθεμα
1	2	000…
2	3	001…
3	*	010…
4	4	011…
5	*	100…
6	*	101…
7	6	110…
8	5	111…

Πίνακας 6-11α: Πίνακας Επιθεμάτων με όλες τις 3-άδες μίας De Bruijn περιόδου $2^3 = 8$ και τους «τοποθετημένους» δείκτες προς σχηματισμό De Bruijn ακολουθίας για k=2.

και s=0*000111. Μένει λοιπόν να βρούμε τα SA[3], SA[5], SA[6], – οι πιθανές τιμές που μπορούν να λάβουν είναι 0,1,7 και ανάλογα θα συμπληρωθούν και οι τιμές SA[6] και SA[7], αφού είναι απόλυτα εξαρτώμενες από την τιμή που θα λάβει το SA[3].

Υποπερίπτωση α:

Θέτοντας SA[3]=0, τότε θα έχουμε τον SA:

Σειρά	Δείκτης	Επίθεμα
1	2	000…
2	3	001…
3	0	010…
4	4	011…
5	*	100…
6	*	101…
7	6	110…
8	5	111…

Πίνακας 6-11δ: Πίνακας Επιθεμάτων με όλες τις 3-άδες μίας De Bruijn περιόδου $2^3 = 8$ και τους «τοποθετημένους» δείκτες προς σχηματισμό De Bruijn ακολουθίας για k=2 με το SA[3]=0.

Οπότε προκύπτει αναγκαστικά η ακολουθία **s=01000111**, με (τελικό) SA:

Σειρά	Δείκτης	Επίθεμα
1	2	000…
2	3	001…
3	0	010…
4	4	011…
5	1	100…
6	7	101…
7	6	110…
8	5	111…

Πίνακας 6-11ε: Πίνακας Επιθεμάτων με όλες τις 3-άδες μίας De Bruijn περιόδου $2^3 = 8$ και τους «τοποθετημένους» δείκτες προς σχηματισμό De Bruijn ακολουθίας για k=2 με το SA[3]=0 και τον SA ολοκληρωμένο.

που είναι προφανής De Bruijn και ο τελικός SA της ακολουθίας θα είναι:

Σειρά	Δείκτης	Επίθεμα
1	2	000111
2	3	00111
3	0	01000111
4	4	0111
5	7	1
6	1	1000111
7	6	11
8	5	111

Πίνακας 6-11ζ: Τελικός Πίνακας Επιθεμάτων με της De Bruijn ακολουθίας 01000111.

και ο πίνακας με περιτύλιξη θα γίνει:

Σειρά	Δείκτης	Επίθεμα
1	2	000111
2	3	00111
3	0	01000111
4	4	0111
5	1	1000111
6	7	101
7	6	110
8	5	111

Πίνακας 6-11κ: Τελικός Πίνακας Επιθεμάτων με της De Bruijn ακολουθίας 01000111 με «wrapping».

<u>Υποπερίπτωση β:</u>

Η τιμή SA[3]=1 δεν μπορεί να τεθεί (αφού το επίθεμα της υποακολουθίας που θα ξεκινούσε από τη θέση 2 της S θα ήταν 100 και όχι το 000, όπως υπάρχει στο SA).

Ας το δούμε στον SA:

Σειρά	Δείκτης	Επίθεμα
1	---	000…
2	3	001…
3	1	010…
4	4	011…
5	3	100…
6	---	101…
7	6	110…
8	5	111…

Πίνακας 6-12α: Πίνακας Επιθεμάτων με όλες τις 3-άδες μίας De Bruijn περιόδου $2^3 = 8$ και τους «τοποθετημένους» δείκτες προς σχηματισμό De Bruijn ακολουθίας για k=2 με το SA[3]=1, αλλά καταλήγουμε σε ακολουθία που δεν είναι De Bruijn.

Οπότε προκύπτει αναγκαστικά η αλλαγή της ακολουθίας σε s=00100111, η οποία δεν είναι De Bruijn, διότι φαίνεται καθαρά από τον SA, ότι δεν υπάρχουν οι πιθανές 3-άδες 000 και 101 σε κανένα επιθέματα.

<u>Υποπερίπτωση γ:</u>

Η τιμή SA[3]=7 επίσης δεν μπορεί να τεθεί (αφού το επίθεμα της υποσυμβολοσειράς που θα ξεκινούσε από τη θέση 7 της S θα ήταν 010 και όχι 1**).

Ας το δούμε στον SA:

Σειρά	Δείκτης	Επίθεμα
1	2	000…
2	3	001…
3	7	010…
4	4	011…
5	0	100…
6	6	101…
7	5	110…
8	---	111…

Πίνακας 6-12β: Πίνακας Επιθεμάτων με όλες τις 3-άδες μίας De Bruijn περιόδου $2^3 = 8$ και τους «τοποθετημένους» δείκτες προς σχηματισμό De Bruijn ακολουθίας για k=2 με το SA[3]=7, αλλά καταλήγουμε σε ακολουθία που δεν είναι De Bruijn.

Οπότε προκύπτει αναγκαστικά η ακολουθία s=10000110, η οποία δεν είναι De Bruijn, διότι φαίνεται καθαρά από τον SA, ότι δεν υπάρχει η 3-άδα 111 σε κανένα επίθεμα.

<u>Περίπτωση 4: k=3</u>

Τότε, το SA και η αντίστοιχη ακολουθία δομούνται ως εξής:

Σειρά	Δείκτης	Επίθεμα
1	3	000…
2	4	001…
3	*	010…

4	*	011…
5	*	100…
6	*	101…
7	6	110…
8	5	111…

Πίνακας 6-13: Πίνακας Επιθεμάτων με όλες τις 3-άδες μίας De Bruijn περιόδου $2^3 = 8$ και τους «τοποθετημένους» δείκτες προς σχηματισμό De Bruijn ακολουθίας για k=3, , αλλά καταλήγουμε σε ακολουθία που δεν είναι De Bruijn, γιατί SA[8]-SA[1]<5-3=2<3.

Από την ιδιότητα $SA[1] = k$ και $SA[2^n] = m$, τότε $m - k > n - 1$, προκύπτει ότι η διαφορά είναι: $SA[1] = 3$ και $SA[8] = 5$, τότε: $5 - 3 = 3 - 1 = 2$.

Είναι προφανές, από την ιδιότητα, ότι είναι αδύνατο, διότι η διαφορά τιμών των δεικτών του SA[1] με το SA[8] δεν μπορεί να είναι μικρότερη ή ίση του (n-1), εν προκειμένω, μικρότερη ή ίση του της τιμής 2, που στην προκειμένη περίπτωση είναι η απορριπτέα τιμή 2.

Κατά συνέπεια, βρήκαμε με αυτόν τον τρόπο δύο ακολουθίες De Bruijn:

- 00010111
- 01000111

Αν επανέλθουμε στον τύπο που μας λέει πόσες είναι οι πιθανές ακολουθίες De Bruijn που μπορούν να παραχθούνε για μεταβλητό αριθμό n και θέσουμε για n=3, τότε θα πάρουμε: $2^{2^{n-1}-n} = 2^{2^{3-1}-3} = 2^{2^2-3} = 2^{4-3} = 2$.

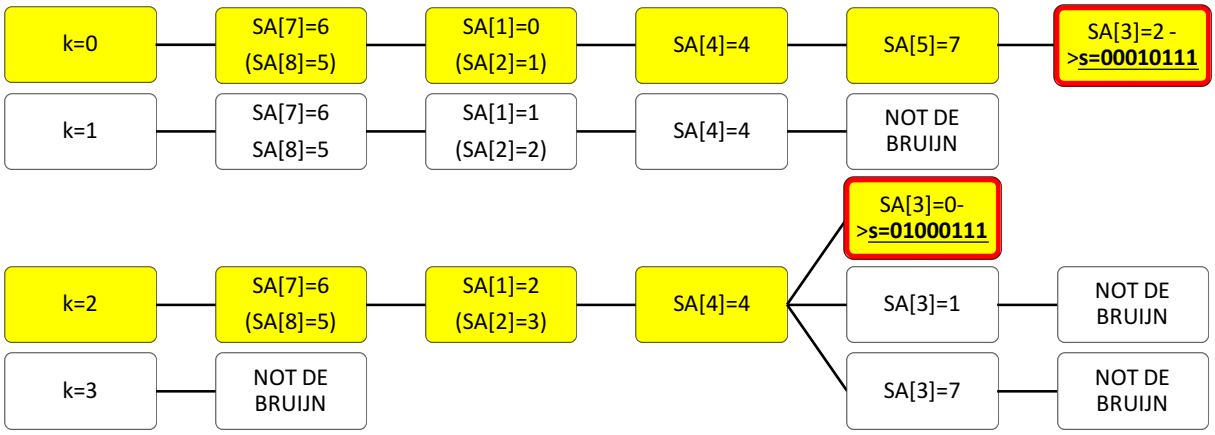
Οπότε είμαστε βέβαιοι ότι έχουμε βρει όλες τις πιθανές δυαδικές De Bruijn ακολουθίας για n=3.

Παρακάτω περιγράφουμε τη διαδικασία και με ένα διάγραμμα-δέντρο - όπου κάθε φύλλο αποτελεί μία επιλογή ("choice") και το φύλλο-παιδί αυτού είναι μία άλλη επιλογή ("choice") δεδομένου της επιλογής του φύλλου-πατέρα.

Οι επιλογές δεν είναι τίποτα άλλο παρά οι τιμές που θέτουμε στις θέσεις του SA (δηλαδή οι διάφορες περιπτώσεις που εξετάσαμε).

Με αυτόν τον τρόπο, θα δείξουμε πώς θα μπορούσε να δομηθεί ένας γενικός αλγόριθμος δυναμικού προγραμματισμού - (τεχνική οπισθοδρόμησης - «backtracking dynamic programming»).

Με το κίτρινο μονοπάτι είναι οι επιλογές που οδηγούν σε De Bruijn ακολουθίες, όπου το τελευταίο φύλλο που δημιουργείται οριστικά η κάθε De Bruijn ακολουθία είναι με κόκκινο περίγραμμα.



Εικόνα 6-1: Διάγραμμα – Δέντρο που απεικονίζει όλες τις επιλογές δημιουργίας De Bruijn ακολουθίας για n=3 από τη δομή των SA.

6.4.2 Κατασκευή De Bruijn ακολουθιών για n=4

Για ακολουθία περιόδου $2^4 = 16$, όποια και αν είναι η De Bruijn ακολουθία στο SA θα τοποθετηθούν κατά σειρά οι υπακολουθίες που αρχίζουν από τα 0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111. Θα τοποθετηθούνε σε έναν πίνακα, ώστε να έχουμε μια συνοπτική εικόνα που θα μας βοηθήσει στον συλλογισμό μας παρακάτω.

Με δεδομένο αυτό θα προσπαθήσουμε να κάνουμε μία αυτόνομη προσέγγιση, που να βασίζεται στις ιδιότητες που αναφέραμε πιο πάνω, με σκοπό την κατασκευή De Bruijn ακολουθιών από την δομή των SA.

Το SA αυτής θα ικανοποιεί τα εξής:

Σειρά	Δείκτης	Υπακολουθία - Επίθεμα
1	*	0000…
2	*	0001…

3	*	0010…
4	*	0011…
5	*	0100…
6	*	0101…
7	*	0110…
8	*	0111…
9	*	1000…
10	*	1001…
11	*	1010…
12	*	1011…
13	*	1100…
14	*	1101…
15	13	1110…
16	12	1111…

Από τις ιδιότητες που αντλήσαμε αρχίζουμε με παρόμοιο τρόπο, όπως κάναμε και για τη δημιουργία De Bruijn για n=3.

Έτσι υποχρεωτικά στις θέσεις SA[15] και SA[16], τοποθετούμε τα επιθέματα που έχουν δείκτη 13 και 12 αντίστοιχα. Με βάση την ιδιότητα αυτή, θα πρέπει να έχουμε κάνει το πρώτο βήμα, για να «χτίσουμε» μία δυαδική ακολουθία De Bruijn η οποία θα έχει τη μορφή:

Δείκτης	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s=	0	*	*	*	*	*	*	*	*	*	*	*	1	1	1	1

Επειδή υπάρχει το γνωστό «wrapping» το επίθεμα που αρχίζει από τον δείκτη 13, θα μας «υποχρεώσει» να αρχίσουμε την ακολουθία μας από το «0», άρα:

s=0**********1111

Επειδή SA[1]=k και SA[2]=k+1, θα μπορούσαμε να εξετάσουμε τις παραδοχές για k=0 έως k=8, διότι αν $SA[1] = k \; και \; SA[16] = 12$, τότε θα πρέπει $12 - k > 4 - 1 \Rightarrow k < 9$.

Σίγουρα υπάρχει μεγάλη πολυπλοκότητα στην εξάντληση όλων των πιθανών τιμών για k, και «χειροκίνητα» είναι μία επώδυνη διαδικασία. Για το λόγο αυτό θα επικεντρωθούμε για την τιμή του k=0.

Οπότε αν k=0, τότε βάσει των ιδιοτήτων ο παραπάνω SA, θα πάρει την μορφή:

Σειρά	Δείκτης	Επίθεμα
1	0	0000…
2	1	0001…

3	*	0010…
4	*	0011…
5	*	0100…
6	*	0101…
7	*	0110…
8	*	0111…
9	*	1000…
10	*	1001…
11	*	1010…
12	*	1011…
13	*	1100…
14	*	1101…
15	13	1110…
16	12	1111…

και η ακολουθία, θα σχηματιστεί ως εξής:

Δείκτης	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s=	0	0	0	0	1	*	*	*	*	*	*	*	1	1	1	1

Επειδή στο τέλος της ακολουθίας υπάρχουνε τέσσερις άσσοι, δεν μπορεί στην θέση πριν την τετράδα με άσσους, δηλαδή το επίθεμα από τον δείκτη 11 να ξεκινάει με «1», αλλά υποχρεωτικά με «0», διαφορετικά θα είχαμε διπλή εμφάνιση της τετράδας «1111» στην De Bruijn ακολουθία, που έρχεται σε αντίθεση με τον γνωστό κανόνα της, ο οποίος λέει, ότι σε μία περίοδο μίας De Bruijn ακολουθίας, εμφανίζεται ακριβώς μία φορά κάθε πιθανή τετράδα και όχι δύο φορές.

Επομένως η ακολουθία θα πάρει τη μορφή:

Δείκτης	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s=	0	0	0	0	1	*	*	*	*	*	*	0	1	1	1	1

και ο SA θα σχηματιστεί ως εξής:

Σειρά	Δείκτης	Επίθεμα
1	0	0000…
2	1	0001…
3	*	0010…
4	*	0011…
5	*	0100…
6	*	0101…
7	*	0110…
8	11	0111…
9	*	1000…

10	*	1001…
11	*	1010…
12	*	1011…
13	*	1100…
14	*	1101…
15	13	1110…
16	12	1111…

Πλέον υπάρχουμε δύο περιπτώσεις που μπορούμε να ακολουθήσουμε. Η μία είναι το SA[3] να είναι το 2 και η άλλη είναι το SA[4] να είναι το 2.

- 1η Περίπτωση είναι το SA[3]=2
- 2η Περίπτωση είναι το SA[4]=2

1η Περίπτωση SA[3]=2, οπότε ο SA θα γίνει:

Σειρά	Δείκτης	Επίθεμα
1	0	0000…
2	1	0001…
3	2	0010…
4	*	0011…
5	*	0100…
6	*	0101…
7	*	0110…
8	11	0111…
9	*	1000…
10	*	1001…
11	*	1010…
12	*	1011…
13	*	1100…
14	*	1101…
15	13	1110…
16	12	1111…

και η ακολουθία θα πάρει τη μορφή:

Δείκτης	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s=	0	0	0	0	1	0	*	*	*	*	*	0	1	1	1	1

Η περίπτωση αυτή, θα έχει και άλλες υποπεριπτώσεις στη συνέχεια:

1η Υπό-περίπτωση είναι το SA[5]=3

Σειρά	Δείκτης	Επίθεμα
1	0	0000…

2	1	0001…
3	2	0010…
4	*	0011…
5	3	0100…
6	*	0101…
7	*	0110…
8	11	0111…
9	*	1000…
10	*	1001…
11	*	1010…
12	*	1011…
13	*	1100…
14	*	1101…
15	13	1110…
16	12	1111…

Δείκτης	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s=	0	0	0	0	1	0	0	*	*	*	*	0	1	1	1	1

1α) Υπό-περίπτωση είναι το SA[9]=4

Σειρά	Δείκτης	Επίθεμα
1	0	0000…
2	1	0001…
3	2	0010…
4	*	0011…
5	3	0100…
6	*	0101…
7	*	0110…
8	11	0111…
9	4	1000…
10	*	1001…
11	*	1010…
12	*	1011…
13	*	1100…
14	*	1101…
15	13	1110…
16	12	1111…

Δείκτης	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s=	0	0	0	0	1	0	0	0	*	*	*	0	1	1	1	1

Εδώ οδηγούμαστε σε ακολουθία που δε θα είναι De Bruijn, γιατί το θα πρέπει SA[0]=5 ή SA[1]=5, πράγμα που δε μπορεί να γίνει.

1β) Υπό-περίπτωση είναι το SA[10]=4

Σειρά	Δείκτης	Επίθεμα
1	0	0000…
2	1	0001…
3	2	0010…
4	*	0011…
5	3	0100…
6	*	0101…
7	*	0110…
8	11	0111…
9	*	1000…
10	4	1001…
11	*	1010…
12	*	1011…
13	*	1100…
14	*	1101…
15	13	1110…
16	12	1111…

Δείκτης	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s=	0	0	0	0	1	0	0	1	*	*	*	0	1	1	1	1

Υποχρεωτικά θα πρέπει ως επόμενο βήμα να έχουμε SA[4]=5 και άρα θα σχηματιστεί ο SA και η ακολουθία ως εξής:

Σειρά	Δείκτης	Επίθεμα
1	0	0000…
2	1	0001…
3	2	0010…
4	5	0011…
5	3	0100…
6	*	0101…
7	*	0110…
8	11	0111…
9	*	1000…
10	4	1001…
11	*	1010…
12	*	1011…
13	*	1100…
14	*	1101…
15	13	1110…
16	12	1111…

Δείκτης	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s=	0	0	0	0	1	0	0	1	1	*	*	0	1	1	1	1

Υποχρεωτικά θα πρέπει ως επόμενο βήμα να έχουμε SA[7]=6 και άρα θα σχηματιστεί ο SA και η ακολουθία ως εξής:

Σειρά	Δείκτης	Επίθεμα
1	0	0000…
2	1	0001…
3	2	0010…
4	5	0011…
5	3	0100…
6	*	0101…
7	6	0110…
8	11	0111…
9	*	1000…
10	4	1001…
11	*	1010…
12	*	1011…
13	*	1100…
14	*	1101…
15	13	1110…
16	12	1111…

Δείκτης	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s=	0	0	0	0	1	0	0	1	1	0	*	0	1	1	1	1

Θα πρέπει ως επόμενο βήμα να έχουμε SA[13]=7 ή SA[14]=7:

$1\beta_1$) Υπό-περίπτωση είναι το SA[13]=7:

Σειρά	Δείκτης	Επίθεμα
1	0	0000…
2	1	0001…
3	2	0010…
4	5	0011…
5	3	0100…
6	*	0101…
7	6	0110…
8	11	0111…
9	*	1000…
10	4	1001…
11	*	1010…
12	*	1011…
13	7	1100…
14	*	1101…
15	13	1110…
16	12	1111…

Δείκτης	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s=	0	0	0	0	1	0	0	1	1	0	0	0	1	1	1	1

και άρα δεν έχουμε De Bruijn ακολουθία.

$1\beta_2$) Υπό-περίπτωση είναι το SA[14]=7:

Σειρά	Δείκτης	Επίθεμα
1	0	0000…
2	1	0001…
3	2	0010…
4	5	0011…
5	3	0100…
6	9	0101…
7	6	0110…
8	11	0111…
9	15	1000…
10	4	1001…
11	8	1010…
12	10	1011…
13	14	1100…
14	7	1101…
15	13	1110…
16	12	1111…

Δείκτης	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s=	0	0	0	0	1	0	0	1	1	0	1	0	1	1	1	1

και άρα έχουμε De Bruijn ακολουθία, την s=0000100110101111.

2η Υπό-περίπτωση είναι το SA[6]=3

Σειρά	Δείκτης	Επίθεμα
1	0	0000…
2	1	0001…
3	2	0010…
4	*	0011…
5	*	0100…
6	3	0101…
7	*	0110…
8	11	0111…
9	*	1000…
10	*	1001…
11	*	1010…
12	*	1011…
13	*	1100…
14	*	1101…
15	13	1110…
16	12	1111…

Δείκτης	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s=	0	0	0	0	1	0	1	*	*	*	*	0	1	1	1	1

Εδώ υπάρχουνε δύο περιπτώσεις:

SA[11]=4 ή SA[12]=4

2α) Υπό-περίπτωση: SA[11]=4

Σειρά	Δείκτης	Επίθεμα
1	0	0000…
2	1	0001…
3	2	0010…
4	*	0011…
5	*	0100…
6	3	0101…
7	*	0110…
8	11	0111…
9	*	1000…
10	*	1001…
11	4	1010…
12	*	1011…
13	*	1100…
14	*	1101…
15	13	1110…
16	12	1111…

Δείκτης	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s=	0	0	0	0	1	0	1	0	*	*	*	0	1	1	1	1

Εδώ έχουμε μία περίπτωση, όπου SA[5]=5, οπότε:

Σειρά	Δείκτης	Επίθεμα
1	0	0000…
2	1	0001…
3	2	0010…
4	*	0011…
5	5	0100…
6	3	0101…
7	*	0110…
8	11	0111…
9	*	1000…
10	*	1001…
11	4	1010…
12	*	1011…
13	*	1100…
14	*	1101…
15	13	1110…

16	12	1111…

Δείκτης	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s=	0	0	0	0	1	0	1	0	0	*	*	0	1	1	1	1

Εδώ υπάρχουνε δύο περιπτώσεις:

SA[9]=6 ή SA[10]=6

$2\alpha_1$) Υπό-περίπτωση: SA[9]=6

Σειρά	Δείκτης	Επίθεμα
1	0	0000…
2	1	0001…
3	2	0010…
4	*	0011…
5	5	0100…
6	3	0101…
7	*	0110…
8	11	0111…
9	6	1000…
10	*	1001…
11	4	1010…
12	*	1011…
13	*	1100…
14	*	1101…
15	13	1110…
16	12	1111…

Δείκτης	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s=	0	0	0	0	1	0	1	0	0	0	*	0	1	1	1	1

Δε μπορεί να οδηγήσει σε De Bruijn ακολουθία.

$2\alpha_2$) Υπό-περίπτωση: SA[10]=6

Σειρά	Δείκτης	Επίθεμα
1	0	0000…
2	1	0001…
3	2	0010…
4	7	0011…
5	5	0100…
6	3	0101…
7	8	0110…
8	11	0111…
9	15	1000…

10	6	1001…
11	4	1010…
12	10	1011…
13	14	1100…
14	9	1101…
15	13	1110…
16	12	1111…

Δείκτης	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s=	0	0	0	0	1	0	1	0	0	1	1	0	1	1	1	1

και άρα έχουμε De Bruijn την s=0000101001101111.

2β) Υπό-περίπτωση: SA[12]=4

Σειρά	Δείκτης	Επίθεμα
1	0	0000…
2	1	0001…
3	2	0010…
4	*	0011…
5	*	0100…
6	3	0101…
7	5	0110…
8	11	0111…
9	*	1000…
10	*	1001…
11	*	1010…
12	4	1011…
13	*	1100…
14	*	1101…
15	13	1110…
16	12	1111…

Δείκτης	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s=	0	0	0	0	1	0	1	1	0	*	*	0	1	1	1	1

Εδώ υπάρχουνε δύο περιπτώσεις:

SA[13]=6 ή SA[14]=6

$2\beta_1$) Υπό-περίπτωση: SA[13]=6

Σειρά	Δείκτης	Επίθεμα
1	0	0000…
2	1	0001…
3	2	0010…

4	*	0011…
5	*	0100…
6	3	0101…
7	5	0110…
8	11	0111…
9	*	1000…
10	*	1001…
11	*	1010…
12	4	1011…
13	6	1100…
14	*	1101…
15	13	1110…
16	12	1111…

Δείκτης	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s=	0	0	0	0	1	0	1	1	0	0	*	0	1	1	1	1

Εδώ υπάρχουνε δύο περιπτώσεις:

SA[9]=7 ή SA[10]=7

$2\beta_{1_\alpha}$) Υπό-περίπτωση: SA[9]=7

Σειρά	Δείκτης	Επίθεμα
1	0	0000…
2	1	0001…
3	2	0010…
4	*	0011…
5	*	0100…
6	3	0101…
7	5	0110…
8	11	0111…
9	7	1000…
10	*	1001…
11	*	1010…
12	4	1011…
13	6	1100…
14	*	1101…
15	13	1110…
16	12	1111…

Δείκτης	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s=	0	0	0	0	1	0	1	1	0	0	0	0	1	1	1	1

δεν οδηγεί σε De Bruijn ακολουθία (δύο «0000» σε μία «περίοδο»)

$2\beta_{1_\beta}$) Υπό-περίπτωση: SA[10]=7

Σειρά	Δείκτης	Επίθεμα
1	0	0000…
2	1	0001…
3	2	0010…
4	8	0011…
5	*	0100…
6	3	0101…
7	5	0110…
8	11	0111…
9	15	1000…
10	7	1001…
11	*	1010…
12	4	1011…
13	6	1100…
14	*	1101…
15	13	1110…
16	12	1111…

Δείκτης	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s=	0	0	0	0	1	0	1	1	0	0	1	0	1	1	1	1

δεν οδηγεί σε De Bruijn (δε συμπληρώνονται οι τετράδες όλες).

$2\beta_2$) Υπό-περίπτωση: SA[14]=6

Σειρά	Δείκτης	Επίθεμα
1	0	0000…
2	1	0001…
3	2	0010…
4	10	0011…
5	8	0100…
6	3	0101…
7	5	0110…
8	11	0111…
9	15	1000…
10	9	1001…
11	7	1010…
12	4	1011…
13	14	1100…
14	6	1101…
15	13	1110…
16	12	1111…

Δείκτης	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s=	0	0	0	0	1	0	1	1	0	1	0	0	1	1	1	1

και άρα θα έχουμε De Bruijn ακολουθία, την s=0000101101001111

2η Περίπτωση SA[4]=2, οπότε ο SA θα γίνει:

Σειρά	Δείκτης	Επίθεμα
1	0	0000…
2	1	0001…
3	*	0010…
4	2	0011…
5	*	0100…
6	*	0101…
7	3	0110…
8	11	0111…
9	*	1000…
10	*	1001…
11	*	1010…
12	*	1011…
13	*	1100…
14	*	1101…
15	13	1110…
16	12	1111…

και η ακολουθία θα πάρει τη μορφή:

Δείκτης	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s=	0	0	0	0	1	1	0	*	*	*	*	0	1	1	1	1

Υπάρχουνε δύο περιπτώσεις στο επόμενο βήμα:

SA[13]=4 ή SA[14]=4

2α) Υπό-περίπτωση SA[13]=4

Σειρά	Δείκτης	Επίθεμα
1	0	0000…
2	1	0001…
3	*	0010…
4	2	0011…
5	*	0100…
6	*	0101…
7	3	0110…
8	11	0111…
9	*	1000…
10	*	1001…
11	*	1010…
12	*	1011…

13	4	1100…
14	*	1101…
15	13	1110…
16	12	1111…

και η ακολουθία θα πάρει τη μορφή:

Δείκτης	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s=	0	0	0	0	1	1	0	0	*	*	*	0	1	1	1	1

Υπάρχουμε δύο περιπτώσεις στο επόμενο βήμα:

SA[9]=5 ή SA[10]=5

$2\alpha_1$) Υπό-περίπτωση SA[9]=5

Σειρά	Δείκτης	Επίθεμα
1	0	0000…
2	1	0001…
3	*	0010…
4	2	0011…
5	*	0100…
6	*	0101…
7	3	0110…
8	11	0111…
9	5	1000…
10	*	1001…
11	*	1010…
12	*	1011…
13	4	1100…
14	*	1101…
15	13	1110…
16	12	1111…

και η ακολουθία θα πάρει τη μορφή:

Δείκτης	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s=	0	0	0	0	1	1	0	0	0	*	*	0	1	1	1	1

δεν οδηγεί σε De Bruijn ακολουθία, αφού τα επιθέματα με δείκτες 0 και 1 υπάρχουν ήδη στην ακολουθία, οπότε δεν υπάρχει επίθεμα πλέον που αρχίζει με 000*, ώστε να τοποθετηθεί στον δείκτη 6 από τον οποίο αρχίζει υποχρεωτικά επίθεμα με 000*.

$2\alpha_2$) Υπό-περίπτωση SA[10]=5

Σειρά	Δείκτης	Επίθεμα
1	0	0000…
2	1	0001…
3	6	0010…
4	2	0011…
5	*	0100…
6	*	0101…
7	3	0110…
8	11	0111…
9	*	1000…
10	5	1001…
11	*	1010…
12	*	1011…
13	4	1100…
14	*	1101…
15	13	1110…
16	12	1111…

και η ακολουθία θα πάρει τη μορφή:

Δείκτης	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s=	0	0	0	0	1	1	0	0	1	0	*	0	1	1	1	1

Υπάρχουνε δύο περιπτώσεις στο επόμενο βήμα:

SA[5]=7 ή SA[6]=7

$2\alpha_{2_\alpha}$) Υπό-περίπτωση SA[5]=7

Σειρά	Δείκτης	Υπακολουθία - Επίθεμα
1	0	0000…
2	1	0001…
3	6	0010…
4	2	0011…
5	7	0100…
6	*	0101…
7	3	0110…
8	11	0111…
9	15	1000…
10	5	1001…
11	*	1010…
12	*	1011…
13	4	1100…
14	*	1101…
15	13	1110…
16	12	1111…

και η ακολουθία θα πάρει τη μορφή:

Δείκτης	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s=	0	0	0	0	1	1	0	0	1	0	0	0	1	1	1	1

Οπότε οι επιλογές αυτές για τον SA δεν οδηγούν σε De Bruijn ακολουθία.

$2\alpha_{2_\beta}$) Υπό-περίπτωση SA[6]=7

Σειρά	Δείκτης	– Επίθεμα
1	0	0000…
2	1	0001…
3	6	0010…
4	2	0011…
5	*	0100…
6	7	0101…
7	3	0110…
8	11	0111…
9	15	1000…
10	5	1001…
11	8	1010…
12	10	1011…
13	4	1100…
14	*	1101…
15	13	1110…
16	12	1111…

και η ακολουθία θα πάρει τη μορφή:

Δείκτης	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s=	0	0	0	0	1	1	0	0	1	0	1	0	1	1	1	1

Οπότε οι επιλογές για τον SA δεν οδηγούν σε De Bruijn ακολουθία.

2β) Υπό-περίπτωση SA[14]=4

Σειρά	Δείκτης	Υπακολουθία - Επίθεμα
1	0	0000…
2	1	0001…
3	*	0010…
4	2	0011…
5	*	0100…
6	*	0101…
7	3	0110…
8	11	0111…

9	*	1000…
10	*	1001…
11	*	1010…
12	*	1011…
13	*	1100…
14	4	1101…
15	13	1110…
16	12	1111…

και η ακολουθία θα πάρει τη μορφή:

Δείκτης	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s=	0	0	0	0	1	1	0	1	*	*	*	0	1	1	1	1

Υπάρχουνε δύο περιπτώσεις στο επόμενο βήμα:

SA[11]=5 ή SA[12]=5

$2\beta_1$) Υπό-περίπτωση SA[11]=5

Σειρά	Δείκτης	Επίθεμα
1	0	0000…
2	1	0001…
3	*	0010…
4	2	0011…
5	*	0100…
6	*	0101…
7	3	0110…
8	11	0111…
9	*	1000…
10	*	1001…
11	5	1010…
12	*	1011…
13	*	1100…
14	4	1101…
15	13	1110…
16	12	1111…

και η ακολουθία θα πάρει τη μορφή:

Δείκτης	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s=	0	0	0	0	1	1	0	1	0	*	*	0	1	1	1	1

Υπάρχουνε δύο περιπτώσεις στο επόμενο βήμα:

SA[5]=6 ή SA[6]=6

$2\beta_{1_\alpha}$) Υπό-περίπτωση SA[5]=6

Σειρά	Δείκτης	Επίθεμα
1	0	0000…
2	1	0001…
3	*	0010…
4	2	0011…
5	6	0100…
6	*	0101…
7	3	0110…
8	11	0111…
9	*	1000…
10	*	1001…
11	5	1010…
12	*	1011…
13	*	1100…
14	4	1101…
15	13	1110…
16	12	1111…

και η ακολουθία θα πάρει τη μορφή:

Δείκτης	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s=	0	0	0	0	1	1	0	1	0	0	*	0	1	1	1	1

Υποχρεωτικά θα έχουμε ως επόμενο βήμα:

Σειρά	Δείκτης	Επίθεμα
1	0	0000…
2	1	0001…
3	8	0010…
4	2	0011…
5	6	0100…
6	9	0101…
7	3	0110…
8	11	0111…
9	15	1000…
10	7	1001…
11	5	1010…
12	10	1011…
13	14	1100…
14	4	1101…
15	13	1110…
16	12	1111…

και η ακολουθία θα πάρει τη μορφή:

Δείκτης	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s=	0	0	0	0	1	1	0	1	0	0	1	0	1	1	1	1

και άρα έχουμε De Bruijn ακολουθία, την s=0000110100101111.

$2\beta_{1_\beta}$) Υπό-περίπτωση SA[6]=6

Σειρά	Δείκτης	Επίθεμα
1	0	0000…
2	1	0001…
3	*	0010…
4	2	0011…
5	*	0100…
6	6	0101…
7	3	0110…
8	11	0111…
9	*	1000…
10	*	1001…
11	5	1010…
12	7	1011…
13	14	1100…
14	4	1101…
15	13	1110…
16	12	1111…

και η ακολουθία θα πάρει τη μορφή:

Δείκτης	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s=	0	0	0	0	1	1	0	1	0	1	1	0	1	1	1	1

Άρα οι επιλογές αυτές για τον SA δεν οδηγούν σε De Bruijn ακολουθία.

$2\beta_2$) Υπό-περίπτωση SA[12]=5

Σειρά	Δείκτης	Υπακολουθία - Επίθεμα
1	0	0000…
2	1	0001…
3	*	0010…
4	2	0011…
5	*	0100…
6	*	0101…
7	3	0110…
8	11	0111…
9	*	1000…
10	*	1001…
11	*	1010…

12	5	1011…
13	*	1100…
14	4	1101…
15	13	1110…
16	12	1111…

και η ακολουθία θα πάρει τη μορφή:

Δείκτης	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s=	0	0	0	0	1	1	0	1	1	*	*	0	1	1	1	1

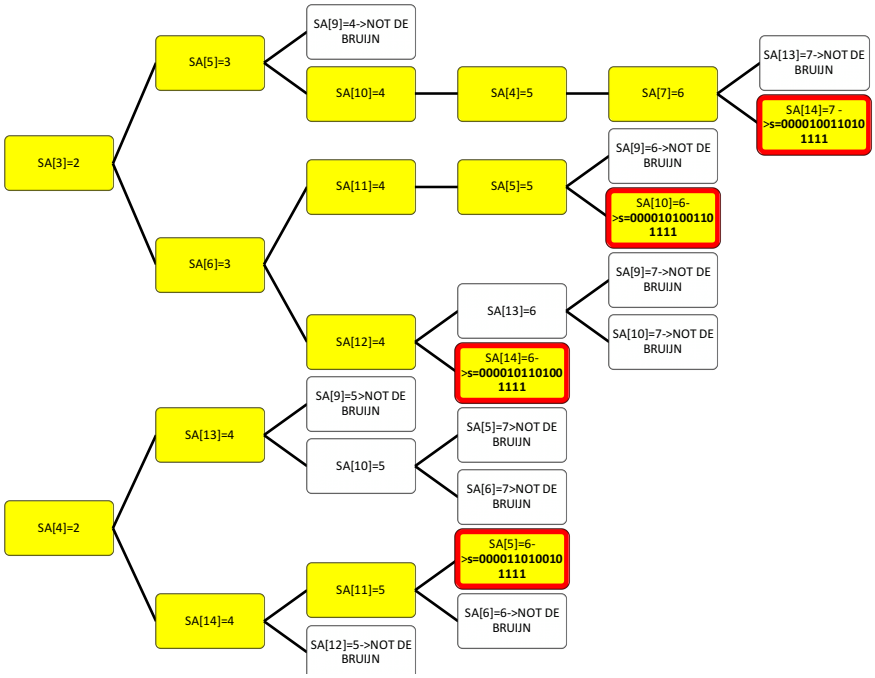
δεν οδηγεί σε De Bruijn.

Οπότε για k=0 βρήκαμε τέσσερις ακολουθίες De Bruijn:

- 0000100110101111
- 0000101001101111
- 0000101101001111
- 0000110100101111

Παρακάτω περιγράφουμε τη διαδικασία και με ένα διάγραμμα - δέντρο – κατ' αναλογία με την περίπτωση n=3.

Με το κίτρινο μονοπάτι είναι οι επιλογές που οδηγούν σε De Bruijn ακολουθίες, όπου το τελευταίο φύλλο που δημιουργείται οριστικά η κάθε De Bruijn ακολουθία είναι με κόκκινο περίγραμμα.



Εικόνα 6-2: Διάγραμμα – Δέντρο που απεικονίζει όλες τις επιλογές δημιουργίας De Bruijn ακολουθίας για n=4 και για μία περίπτωση k=0, από τη δομή των SA.

6.4 Γενικά συμπεράσματα στη νέα κατασκευή De Bruijn ακολουθιών

Δε θα αναφερθούμε ενδελεχώς σε συμπεράσματα σε τούτο το υποκεφάλαιο, αλλά θα συγκρίνουμε τις δύο ακολουθίες De Bruijn που δημιουργήσαμε με τη βοήθεια της δομής των SA, στο προηγούμενο υποκεφάλαιο 6.3 για να εντοπίσουμε ένα «κρίσιμο» σημείο.

Ας δούμε την πρώτη ακολουθία De Bruijn, η οποία δημιουργήθηκε και είναι η 00010111. Ο SA αυτής είναι:

Σειρά	Δείκτης	Επίθεμα
1	0	00010111
2	1	0010111
3	2	010111
4	4	0111
5	7	100
6	3	10111
7	6	110
8	5	111

Πίνακας 6-14: Πίνακας Επιθεμάτων (Suffix Array) της 00010111 De Bruijn ακολουθίας περιόδου $2^3 = 8$.

Και η δεύτερη ακολουθία , η οποία δημιουργήθηκε και είναι η 01000111 με τον SA να είναι:

Σειρά	Δείκτης	Επίθεμα
1	2	000111
2	3	00111
3	0	01000111
4	4	0111
5	7	101*(101 με περιτύλιξη)*
6	1	1000111
7	6	110
8	5	111

Πίνακας 6-15: Πίνακας Επιθεμάτων (Suffix Array) της 01000111 De Bruijn ακολουθίας περιόδου $2^3 = 8$.

Παρατηρούμε αρχικά με «κόκκινα» γράμματα την συμπλήρωση των (n-1) πρώτων bits της αρχικής ακολουθίας, ώστε να «δημιουργηθούνε» οι 3-άδες που υπάρχουνε μέσα στην αρχική De Bruijn ακολουθία, για να επιβεβαιωθούμε διπλά ότι η μελέτη που έγινε, κατέληξε σε ασφαλή συμπεράσματα.

Ο πρώτος SA έχει μεν αναλυθεί και πιο πάνω, αλλά θα αναφερθούμε στην περιτύλιξη (wrapping) που γίνεται σκόπιμα, ώστε να «δημιουργήσουμε» σε αυτόν όλες τις πιθανές 3-άδες που υπάρχουνε στην περίοδο της De Bruijn ακολουθίας του, οπότε και τοποθετούμε τα (n-1) bits της αρχικής ακολουθίας. Βλέπουμε στα SA[5] και SA[7] στον Πίνακα 6-14 ότι νοερά τοποθετούμε τα bits αυτά, ώστε να «αποδείξουμε» ότι υπάρχουνε στο SA όλες οι πιθανές 3-άδες της περιόδου.

Στον δεύτερο SA, παρατηρούμε ότι λείπει το επίθεμα 101... στο SA[6], ενώ κατά τη δημιουργία της ακολουθίας, βασιζόμενοι στη δομή των SA με όλα τα επιθέματα, γνωρίζουμε ότι υποχρεωτικά υπήρχε. Δε μας προκαλεί όμως εντύπωση, εφόσον γνωρίζουμε ότι κάποια επιθέματα που υπάρχουνε στο SA, να μην «φέρουν» όλες τις n-άδες, που υπάρχουνε στην αρχική De Bruijn ακολουθία. Στον πάνω SA το επίθεμα 01000111, δε φέρει την 3-άδα 101, όπως είναι αναμενόμενο, ενώ αυτή «δημιουργείται» νοερά στο SA[5], χωρίς όμως να μεταβάλει τη δομή του SA. Αυτό γίνεται, διότι η 3-άδα είναι «κρυμμένη» στην περιτύλιξη της ακολουθίας και δημιουργείται, όπως δείξαμε πιο πάνω, από την ακολουθία, αν σε αυτήν προσθέσουμε στο τέλος τα (n-1) πρώτα bits της αρχικής ακολουθίας, οπότε θα έχουμε την 01000111(01). Η διαδικασία

αυτή ονομάζεται περιτύλιξη ή αλλιώς «wrapping» και είναι νοερά πάντα δεδομένη (βλ. και σχετική συζήτηση στην Ενότητα 6.3).

Εξετάσαμε και επιβεβαιώσαμε ότι είναι εφικτό, εφόσον αξιοποιηθούν οι ιδιότητες που αντλήσαμε από τη δομή των SA που δομούνται από De Bruijn ακολουθίες, να παράγουμε «χειροκίνητα» δυαδικές ακολουθίες οι οποίες είναι De Bruijn. Η θετική έκβαση του γεγονότος βασίζεται στην «εμφύτευση» των ιδιοτήτων αυτών, ακολουθώντας την ανάποδη λογική. Η λογική είναι απλή και βασίζεται στα επιθέματα του SA, που φέρουν όλες τις πιθανές n-άδες μίας De Bruijn ακολουθίας. Ωστόσο θα πρέπει παράλληλα, όπως αναφέραμε, να υπολογίζεται νοερά η ύπαρξη της κυκλικής συνένωσης, ώστε να καταστεί εφικτή η δημιουργία της De Bruijn ακολουθίας με τη δομή του SA.

Κεφάλαιο 7
Επίλογος

7.1 Εισαγωγή

Σ' αυτό το κεφάλαιο παρατίθενται τα βασικά συμπεράσματα της διατριβής και οι προτάσεις για μελλοντική έρευνα.

7.2 Επισκόπηση - Συμπεράσματα

Αντικείμενο της διατριβής αποτέλεσε η μελέτη τεχνικών για την κατασκευή ψευδοτυχαίων ακολουθιών. Απώτερος σκοπός αυτής ήταν η ανάπτυξη νέων μεθόδων κατασκευής κρυπτογραφικά σημαντικών ακολουθιών – ειδικότερα, ακολουθιών De Bruijn - που θα βασίζονται στη δομή των SA.

Στο πλαίσιο της εργασίας αρχικά αναφερθήκαμε στην ανάγκη ύπαρξης ψευδοτυχαίων ακολουθιών, που δεν είναι τίποτα άλλο από μία αλληλουχία αριθμών (στη συγκεκριμένη περίπτωση, δυαδικών αριθμών), που έχουν «καλά» χαρακτηριστικά τυχαιότητας. Ως εκ τούτου μία «καλή» ψευδοτυχαία ακολουθία χρησιμοποιείται στην κρυπτογραφία ως κλειδοροή.

Οι κλειδοροές παράγονται με ποικίλους τρόπους. Ένας ικανοποιητικός, μη κοστοβόρος και γρήγορος τρόπος παραγωγής ακολουθιών με κάποια καλά, κατ' αρχάς, χαρακτηριστικά είναι οι γραμμικοί καταχωρητές ολίσθησης με ανάδραση (LFSR), οι οποίοι έχουν αποτελέσει για πολλές δεκαετίες αντικείμενο έρευνας. Ως εκ τούτου, οι LFSR μελετήθηκαν ενδελεχώς στην παρούσα διατριβή, όπου πέρα από τα ανωτέρω πλεονεκτήματα επιτρέπουν την εύκολη κατασκευή ακολουθιών με τη μέγιστη δυνατή περίοδο (αφού η μεγάλη περίοδος της κλειδοροής αποτελεί βασικό κρυπτογραφικό της χαρακτηριστικό). Ωστόσο, οι ερευνητές Berlekamp και Massey κατέστησαν τους LFSR μη κατάλληλες πηγές παραγωγής ψευδοτυχαίων ακολουθιών, αφού ανέπτυξαν αλγόριθμο (γνωστό με το όνομα Berlekamp-Massey) που επιτρέπει την πρόβλεψη ολόκληρης της ακολουθίας αν κανείς γνωρίζει ένα μικρό τμήμα της, μεγέθους ίσου με τη λεγόμενη γραμμική πολυπλοκότητα αυτής (η οποία δεν μπορεί να είναι μεγαλύτερη από το μέγεθος του LFSR). Συνεπώς, προέκυψε η ανάγκη παραγωγής ακολουθιών με υψηλή γραμμική πολυπλοκότητα. Μελετήθηκαν το πλήθος των γνωστών τεχνικών που ακολουθούνται προς αυτήν την κατεύθυνση οι οποίες βασίζονται στην εισαγωγή μη γραμμικοτήτων (μη γραμμικά φίλτρα, μη γραμμικοί συνδυαστές κτλ.) σε LFSR.

Πιο πρόσφατες επιθέσεις σε κρυπτογραφικά συστήματα (π.χ. οι αλγεβρικές επιθέσεις) καταδεικνύουν ότι η εγγενής γραμμικότητα του LFSR που χρησιμοποιείται ως δομικό συστατικό για την παραγωγή ψευδοτυχαίων ακολουθιών αποτελεί μειονέκτημα και «ευπαθές» σημείο. Ως εκ τούτου, έμφαση έχει δοθεί τα τελευταία χρόνια στη μελέτη μη γραμμικών καταχωρητών ολίσθησης με ανάδραση (NLFSR). Στην παρούσα διατριβή παρουσιάσαμε τους διάφορους NLFSR, με τις γνωστές μη γραμμικές συναρτήσεις που τους χαρακτηρίζουν και δείξαμε ότι και αυτοί δεν είναι απαραίτητα άτρωτοι. Ωστόσο η μελέτη των NLFSR δεν έχει ολοκληρωθεί ακόμα – για παράδειγμα, σε αντίθεση με τους LFSR, δεν γνωρίζουμε πώς να κατασκευάζουμε NLFSR που να παράγουν ακολουθίες με τη μέγιστη δυνατή περίοδο. Οι ακολουθίες αυτές μεγίστης περιόδου που παράγονται από NLFSR αποτελούν μια σημαντική κατηγορία ακολουθιών, γνωστής ως De Bruijn ακολουθίες, οι οποίες έχουν ιδιαίτερη αξία τόσο στην κρυπτογραφία όσο και σε άλλες εφαρμογές. Συνεπώς, στο πλαίσιο της διατριβής παρουσιάσαμε τις ιδιότητες των De Bruijn ακολουθιών και έπειτα, μέσα από τις σύγχρονες βιβλιογραφικές αναφορές, αναδείξαμε το σημείο στο οποίο βρίσκεται η τρέχουσα έρευνα αναφορικά κυρίως με τις τεχνικές κατασκευής τους – για παράδειγμα, καταδείξαμε ότι βρέθηκαν τρόποι να παράγονται De Bruijn ακολουθίες από NLFSR αλλά όχι αρκετά μεγάλης περιόδου.

Στη συνέχεια, ακολουθήσαμε μία νέα προσέγγιση, σε μία προσπάθεια να αναπτύξουμε μία καινούρια αλγοριθμική διαδικασία για την παραγωγή δυαδικών De Bruijn ακολουθιών. Βασικός

πυλώνας της ερευνητικής αυτής προσπάθειας ήταν η αξιοποίηση της δομής των πινάκων επιθεμάτων (SA), με κύριο σκεπτικό το ότι είναι εφικτή η κατασκευή De Bruijn ακολουθιών μέσω της δομής των SA. Για να γίνει αυτό κατανοητό, περιγράψαμε αναλυτικά τη δομή των δέντρων επιθεμάτων (ST), που οδηγούν σε SA. Στην έρευνά μας, αναλύσαμε τις De Bruijn ακολουθίες με βάση τις ιδιότητες που εμφανίζουν τα SA αυτών. Με αυτόν τον τρόπο «αλιεύσαμε» ιδιότητες που είναι κοινές σε όλους τους SA των De Bruijn ακολουθιών, ανεξάρτητα από την περίοδο της ακολουθίας. Έτσι καταγράψαμε όλες τις παρατηρήσεις μας με περιεκτικό τρόπο και τις εντάξαμε σε μαθηματικό πλαίσιο, ώστε να έχουμε μία βάση αναφοράς με την όποια θα συνεχίσουμε την έρευνά μας. Για το σκοπό αυτό, στο πλαίσιο της διατριβής αναπτύχθηκαν και αξιοποιήθηκαν εφαρμογές λογισμικού, τόσο για την παραγωγή ακολουθιών De Bruijn όσο και για τη δημιουργία των SA αυτών.

Η κύρια ερευνητική συνεισφορά της διατριβής είναι η παραγωγή De Bruijn ακολουθιών βασισμένες στις ιδιότητες των SA που καταγράφηκαν από την ανωτέρω διαδικασία. Καταδείξαμε ότι, τουλάχιστον για μικρές τιμές της περιόδου 2^n των ακολουθιών, είναι εφικτό να «εμφυτεύσουμε» συγκεκριμένες τιμές σε ένα SA, που να πληρούν το σύνολο των επιθυμητών ιδιοτήτων, έτσι ώστε, το παραγόμενο SA να αντιστοιχεί σε ακολουθία De Bruijn. Πετύχαμε δηλαδή να αναδείξουμε μία νέα τεχνική για κατασκευή ακολουθιών De Bruijn, ακολουθώντας μία κατά κάποιο τρόπο «αντίστροφη» λογική, κατασκευάζοντας πρώτα το SA αυτών.

Παρόλο που δεν μπορούμε να ισχυριστούμε ότι παρουσιάζουμε μία ολοκληρωμένη νέα τεχνική, εν τούτοις διαφαίνεται ότι ανοίξαμε την «πόρτα» που οδηγεί στο νέο «μονοπάτι»: το πρώτο βήμα έγινε και η παρούσα διατριβή ενδεχομένως αποτελέσει εφαλτήριο για μελλοντική έρευνα.

7.3 Μελλοντική έρευνα

Όπως ήδη αναφέραμε, η ερευνητική προσπάθεια της παρούσας διατριβής αποτελεί ένα πρώτο βήμα: η αναζήτηση περισσότερων χαρακτηριστικών ιδιοτήτων του SA μίας De Bruijn ακολουθίας μοιάζει να είναι εφικτή, η οποία με τη σειρά της πιθανότατα θα διευκολύνει την ανάπτυξη μιας συστηματικής διαδικασίας για την παραγωγή ακολουθιών De Bruijn μεγαλύτερης περιόδου. Με βάση τη μέχρι τώρα έρευνα, διαφαίνεται ότι αυτό θα μπορούσε να γίνει αξιοποιώντας τις τεχνικές του δυναμικού προγραμματισμού με περιορισμούς (ειδικότερα, με προγραμματισμό τύπου οπισθοδρόμησης - «backtracking») για την ανάπτυξη συστηματικής αλγοριθμικής διαδικασίας, που θα μπορεί πλέον αυτόνομα να κατασκευάζει De Bruijn

ακολουθίες μεγάλων περιόδων. Σε κάθε περίπτωση, θα πρέπει να αποτιμηθεί και η υπολογιστική πολυπλοκότητα της νέας αυτής προσέγγισης, ώστε να μπορεί να αξιολογηθεί και η δυνατότητα εφαρμογής της στην πράξη.

Θα πρέπει τέλος να σημειωθεί ότι η εύρεση ακολουθιών De Bruijn μέσω αλγοριθμικής διαδικασίας, όπως αυτή που περιγράφηκε, παρόλο που αποκτά ιδιαίτερη σημασία και για κρυπτογραφικές εφαρμογές, δεν δίνει απάντηση στο πρόβλημα εύρεσης γενικού κανόνα κατασκευής NLFSR που να παράγει ακολουθίες μεγίστης περιόδου (De Bruijn). Συνεπώς, το ανοιχτό αυτό ερευνητικό πρόβλημα παραμένει.

Βιβλιογραφία

[01] M. I. Abouelhoda, E. Ohlebusch and S. Kurtz, "Optimal Exact String Matching Based on Suffix Arrays", Faculty of Technology, University of Bielefeld, 2002.

[02] C. Blondeau and K. Nyberg , "Finite Fileds and their Applications", "Perfect nonlinear function and cryptography", Aalto University, School of Science, Department of Information and Computer Science, 2014.

[03] M. Burmester, Σ. Γκρίτζαλης, Σ. Κάτσικας, Β. Χρυσικόπουλος, "Σύγχρονη Κρυπτογραφία: Θεωρία και Εφαρμογές, Παπασωτηρίου", 2011.

[04] B. Cazaux, T. Lecroq and E. Rivals, "From Indexing Data Structures to De Bruijn Graphs", Universite de Montpellier, 2014.

[05] E. Dubrova, "A method for Generating Full Cycles by a Composition of NLFSRs", Design, Codes and Cryptography, DOI 10.1007/s10623-014-9947-3, 2014.

[06] A. J. Ferreira, and A. L. Oliveira M. A. T. Figueiredo, "Suffix Arrays – A Competitive Choice fot Fast Lempel-Ziv Compresions", Instituto Superior de Engenharia de Lisboa, Portugal, 2009.

[07] R. Forre, "A fast correlation attack on nonlinearly feedforward filtered shift-register sequences" in Advances in Cryptology-Eurocrypt '89 (Lecture Notes in Computer Science Springer), vol. 334, pp. 586-595, 1990.

[08] G. Gong, "Sequence Analysis", (Lectures Notes for CO739x, Department of Combinatories and Optimization), University of Eaterloo, 1999.

[09] M. Hell and T. Johansson and W. Meier, "Grain – A Stream Cipher for Constrained Environments", Dept. of Information Technology, Lund University, 2007.

[10] P. Ko and S. Aluru, "Space efficient linear time construction of suffix arrays", Department of Electrical and Computer Engineering, Iowa State University, 2004.

[11] M. Li, D. Lin, “De Bruijn Sequences from Nonlinear Feedback Shift Registers”, State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, 2015.

[12] K. Mandal and G. Gong, “Cryptographically strong De Bruijn sequences with large periods”, CACR Technical Report, 2012.

[13] S. Marcus, H. Lee and M. C. Schaltz, Simons Center for Quantitive Biology, Cold Springer Harbor Laboratory, Cold Spring Harbor, NY, Department of Computer Scinece, 2014.

[14] A. Menezes, P. V. Oorschot and S. Vanstone, “Handbook of Applied Cryptography", CRC Press, 1996.

[15] C. Paar, J. Pelzl, “Understanding Cryptography: A Textbook for Studends and Practitioners”, Springer Science & Business Media, 2009.

[16] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, S. Vo, “A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications”, NIST National Institute of Standarts and Technology, 2010.

[17] C. Shannon, “A Mathematical Theory of Communication”, The Bell System Technical Journal, 1948.

[18] Y. She, Supervisor Professor P. J. Giblin, Universal Cycles, “Wirral Grammar School for Girls”, Department of Mathematical Sciences, University Liverpoll, 2011.

[19] T. Siegenthaler, “Cryptanalysts representasion of nonlinearly filtered m-sequences” in Advances in Cryptology-Eurocrypt ’85 (Lecture Notes in Computer Science Springer), vol. 219, pp. 103-110, 1986.

[20] H. C. A. van Tilborg and S. Jajodia (Eds.), “Encyclopedia of Cryptography and Security”, 2011.

[21] Ν. Θεοδωρόπουλος και Α. Δημόπουλος, ”Μελέτη των αλγεβρικών επιθέσεων σε σύγχρονους αλγόριθμους κρυπτογράφησης”, Πτυχιακή εργασία, Τεχνολογικό Εκπαιδευτικό Ίδρυμα Στερεάς Ελλάδας, 2010.

[22] Κ. Λιμνιώτης, “Κρυπτογραφία”, Σημειώσεις μαθήματος Κρυπτογραφίας στο Ανοικτό Πανεπιστήμιο Κύπρου, 2014.

Δικτυακοί Τόποι

[Web01] http://el.wikipedia.org/wiki/Κρυπτογράφηση_Συμμετρικού_Κλειδιού

[Web02] http://el.wikipedia.org/wiki/Κρυπτογράφηση_Δημόσιου_Κλειδιού

[Web03] http://el.wikipedia.org/wiki/Κρυπτογραφικοί_Αλγόριθμοι_Ροής

[Web04] http://datagenetics.com/blog/october22013/index.html

[Web05] http://en.wikipedia.org/wiki/De_Bruijn_sequence

[Web06] http://users.uom.gr/~steph/material/crypto/HAC_Ch06.pdf

[Web07] http://www.homolog.us/Tutorials/index.php?p=2.2&s=1

[Web08] https://en.wikipedia.org/wiki/Substring#Suffix

[Web09] http://www.sanfoundry.com/cpp-program-implement-suffix-array/

[Web10] https://en.wikipedia.org/wiki/Gilbert_Vernam

[Web11] http://www.ecrypt.eu.org/stream/

Ακρωνύμια

bit Binary Digit (Δυαδικό ψηφίο «0» ή «1»)

deg Degree (Βαθμός)

FSR Feedback Shift Register (Καταχωρητής Ολίσθησης με ανάδραση)

GF Galois field

GSM Global System for Mobile Communications (Παγκόσμιο Σύστημα Κινητών Επικοινωνιών)

LFSR Linear Feedback Shift Register (Γραμμικός Καταχωρητής Ολίσθησης με ανάδραση)

m-sequence Maximal Length Sequence

mod Modulo operation (Πράξη υπολοίπων)

NLFSR Non Linear Feedback Shift Register (Μη Γραμμικός Καταχωρητής Ολίσθησης με ανάδραση)

RC4 Rivest Cipher 4

RFID Radio Frequency Identification (Ταυτοποίηση μέσω Ραδιοσυχνοτήτων)

SA Suffix Array (Πίνακας Επιθεμάτων)

SSL Secure Sockets Layer

ST Suffix Tree (Δένρδο Επιθεμάτων)

TLS Transport Layer Security

XOR Exclusive OR

3G Third generation of mobile telecommunications technology (Τρίτης γενιάς στο Σύστημα Κινητών Επικοινωνιών)

Η/Υ Ηλεκτρονικός Υπολογιστής

Παράρτημα Α

Λογισμικά που χρησιμοποιήθηκαν για την έρευνα

Στο Παράρτημα αναφέρονται οι κώδικες που αναπτύχθηκαν για την έρευνα.

A1. **ConstructDeBruijn** - Κώδικας δημιουργίας δυαδικής De Bruijn ακολουθίας για αριθμό n.

A2. **ConstructSuffixArray** - Κώδικας δημιουργίας SA της δυαδικής De Bruijn ακολουθίας που δημιουργήθηκε.

A-1. ConstructDeBruijn - Κώδικας δημιουργίας δυαδικής De Bruijn ακολουθίας για αριθμό n.

```
def de_bruijn(k,n):
 try:
  alphabet = list(map(str, range(k)))
 except (ValueError, TypeError):
  alphabet = k
  k = len(k)
 a = [0] * k * n
 sequence = []
 def db(t, p):
  if t > n:
   if n % p == 0:
    sequence.extend(a[1:p + 1])
  else:
   a[t] = a[t - p]
   db(t + 1, p)
   for j in range(a[t - p] + 1, k):
    a[t] = j
    db(t + 1, t)
```

```
  db(1, 1)

  return "".join(alphabet[i] for i in sequence)

n = int(input("Enter a number: "))

file = open("newfile.txt", "w")

file.write(de_bruijn(2,n))

file.close()

#print(de_bruijn(2,n))
```

[Web05]

A-2. ConstructSuffixArray - Κώδικας δημιουργίας SA της δυαδικής De Bruijn ακολουθίας που δημιουργήθηκε.

```
/*

* C++ Program to Implement Suffix Array
```

```
*/
#include <iostream>
#include <cstdlib>
#include <cstring>
#include <string>
#include <fstream>
using namespace std;


class SuffixArray
{
 private:
  string *text;
  int length;
  int *index;
  string *suffix;
 public:
  SuffixArray(string text)
  {
   this->text = new string[text.length()];
   for (int i = 0; i < text.length(); i++)
   {
    this->text[i] = text.substr(i, 1);
   }
```

```
 this->length = text.length();
 this->index = new int[length];
 for (int i = 0; i < length; i++)
 {
  index[i] = i;
 }
 suffix = new string[length];
}

void createSuffixArray()
{
 for(int index = 0; index < length; index++)
 {
  string text = "";
  for (int text_index = index; text_index < length; text_index++)
  {
   text += this->text[text_index];
  }
  suffix[index] = text;
 }
 int back;
 for (int iteration = 1; iteration < length; iteration++)
 {
```

```
  string key = suffix[iteration];
  int keyindex = index[iteration];
  for (back = iteration - 1; back >= 0; back--)
  {
   if (suffix[back].compare(key) > 0)
   {
    suffix[back + 1] = suffix[back];
    index[back + 1] = index[back];
   }
   else
   {
    break;
   }
  }
  suffix[back + 1] = key;
  index[back + 1 ] = keyindex;
 }
 cout<<"SUFFIX \t INDEX"<<endl;
 for (int iterate = 0; iterate < length; iterate++)
 {
  cout<<suffix[iterate] << "\t" << index[iterate]<<endl;
 }
}
```

```
};


int main()
{
            char buffer[256];


             ifstream examplefile ("newfile.txt");


             if (! examplefile.is_open())
             {
              cout << "Error opening file";
              exit (1);
             }


             while (! examplefile.eof() )
             {
              examplefile.getline (buffer,100);
              cout << buffer << endl;
             }


 string text;
 text = buffer;
 SuffixArray suffixarray(text);
```

```
  suffixarray.createSuffixArray();

}
```

[Web09]