ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Εργαστήριο Μικροεπεξεργαστών και υλικού

Διπλωματική Εργασία

«Υλοποίηση σε αναδιατασσόμενη λογική των πινάκων ουράνιου τόξου για την αποκρυπτογράφηση του αλγόριθμου A5/1 (GSM δίκτυα)»

Καλεντέρη Μαρία

Επιτροπή:
Παπαευσταθίου Ιωάννης, Επίκουρος Καθηγητής (Επιβλέπων)
Δόλλας Απόστολος, Καθηγητής
Πνευματικάτος Διονύσιος, Καθηγητής

Πρόλογος

Η παρούσα διπλωματική εργασία εκπονήθηκε στο εργαστήριο Μικροεπεξεργαστών και Υλικού στο Πολυτεχνείο Κρήτης.
Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα Επίκουρο Καθηγητή του τμήματος Ηλεκτρονικών Μηχανικών και Μηχανικών Υπολογιστών του Πολυτεχνείου Κρήτης Παπαευσταθίου Ιωάννη για την καθοδήγησή του σε όλα τα στάδια της υλοποίησης, καθώς και τον Επίκουρο Καθηγητή του τμήματος Εφαρμοσμένης Πληροφορικής και Πολυμέσων του ΤΕΙ Κρήτης Χαράλαμπο Μανιφάβα για τις οδηγίες και τις συμβουλές του.
Επίσης να ευχαριστήσω όλα τα παιδιά στο εργαστήριο στους οποίους ανέτρεξα πολλές φορές για συμβουλές καθώς και τον κύριο Μάρκο Κιμιωνή.
Ένα μεγάλο ευχαριστώ σε όλους τους φίλους μου στα Χανιά, ειδικά στις συγκάτοικούς μου Δώρα και Juliana, και στους φίλους μου στο χωριό για την υποστήριξή τους και ειδικά στον Τάκη. Πάνω από όλους ευχαριστώ την οικογένειά μου που με στήριξε με όλους τους τρόπους.
Τέλος αυτή η διπλωματική εργασία αφιερώνεται στον αδερφό μου Στάθη.

Περίληψη

Η εκτενέστατη χρήση της κινητής τηλεφωνίας απαιτεί την ύπαρξη μέτρων ασφαλείας που διασφαλίζουν την εμπιστευτικότητα της επικοινωνίας των χρηστών.

Ο A5 αλγόριθμος είναι ο βασικός κρυπτογραφικός αλγόριθμος που κρυπτογραφεί τις συνομιλίες των χρηστών τόσο στις ζεύξεις ανόδου (uplink-κινητός σταθμός προς σταθμό βάσης) όσο και στις ζεύξεις καθόδου (downlink-σταθμός βάσης προς κινητό σταθμό).

Ο A5 μετασχηματίζει το αρχικό κείμενο (plaintext) σε ένα κρυπτογραφημένο κείμενο (ciphertext) και αντίστροφα, με τη χρήση ενός κρυπτογραφικού κλειδιού Kc.
Στην οικογένεια των A5 αλγορίθμων ανήκουν οι:

- A5/0 αποτελεί υλοποίηση ως ευφημισμός καθώς δεν χρησιμοποιεί κρυπτογράφηση
- A5/1 είναι η επικρατέστερη υλοποίηση την παρούσα στιγμή
- A5/2 είναι η ασθενέστερη εκδοχή του A5/1
- A5/3 είναι η ισχυρότερη εκδοχή

Έχει αποδειχτεί ότι ο βαθμός ασφαλείας που παρέχει ο A5/2 είναι μηδαμινός.
Όσον αφορά τον A5/1 ένας τεράστιος αριθμός επιθέσεων έχει δημοσιευτεί εναντίον του και το συμπέρασμα είναι ότι και αυτός τη δεδομένη στιγμή παρέχει ελάχιστη ασφάλεια.

Η τελευταία επίθεση είναι η επίθεση A5/1 Security Project από τον Karsten Nohl και μια ομάδα κρυπτογράφων, οι οποίοι υπολόγισαν και δημοσίευσαν ανοιχτά τους πίνακες ουράνιου τόξου για το «σπάσιμο» του αλγορίθμου. Οι πίνακες αποτελούνται από αντιστοιχήσεις 64-bit keystream με την 64-bit εσωτερική κατάσταση του αλγορίθμου.

Η παρούσα διπλωματική εργασία υλοποιεί τους πίνακες ουράνιου τόξου σε αναδιατασσόμενη λογική και συγκεκριμένα στην FPGA Virtex5 XC5VLX330T [1].
Η δημιουργία των πινάκων σε hardware είναι σε πολύ μεγάλο βαθμό ταχύτερη από την αντίστοιχη δημιουργία τους σε software. Για τον υπολογισμό 345 παράλληλων αλυσίδων απαιτούνται 830 ms στο hardware έναντι 39,1 λεπτών στο software.

Λόγω της ανεπαρκούς παροχής ασφαλείας από τον A5/1 η GSMA έχει εκδώσει γενική οδηγία στους τηλεπικοινωνιακούς παρόχους για την μεταβίβαση στον A5/3, ο οποίος αν και είναι υλοποιημένος σε περίπου 40% των τηλεφωνικών συσκευών[19] δεν υποστηρίζεται από τα δίκτυα. Η μεταβίβαση αυτή καθυστερεί αδικαιολόγητα.

Περιεχόμενα

Κατάλογος εικόνων

Κατάλογος πινάκων

Κατάλογος γραφημάτων

Κεφάλαιο 1: Ιστορία του GSM και αρχιτεκτονική του GSM

1.1 Ιστορία του GSM

Το Global System for Mobile communication είναι το μεγαλύτερο παγκοσμίως κυψελωτό ψηφιακό σύστημα κινητής τηλεφωνίας δεύτερης γενιάς (2G).

Η δεκαετία του 1980 εισήγαγε την κινητή τηλεφωνία πρώτης γενιάς (1G) στην Ευρώπη και συγκεκριμένα όταν το 1981 το NMT450 (Nordic Mobile Telephone System στη ζώνη συχνοτήτων των 450 MHz) άρχισε να εκπέμπει αναλογικά στη Δανία, Σουηδία, Φινλανδία και Νορβηγία. Ακολούθησαν η Μεγάλη Βρετανία (Total Access Communications System-TACS στα 900 MHz) και αργότερα η Γερμανία (C-Netz), η Γαλλία (Radiocom 2000) και η Ιταλία (RTMI/RTMS)[2], ενώ μερικά από αυτά τα δίκτυα χρησιμοποιούνταν από παραπάνω από μια χώρες, για παράδειγμα η Αυστρία, το Βέλγιο, η Ολλανδία και η Ισπανία εκμεταλλεύονταν το NMT450. Ένα αναλογικό κυψελωτό σύστημα επικοινωνίας χρησιμοποιούσε αναλογική διαμόρφωση συχνότητας (FM) για να εκπέμψει το σήμα φωνής. Όλα αυτά τα δίκτυα ήταν τεχνικά και λειτουργικά ασύμβατα μεταξύ τους γεγονός που οφειλόταν στη χρήση διαφορετικών ζωνών συχνοτήτων, στη χρήση ανακόλουθων σημάτων για τον έλεγχο της αρχής και τέλους της σύνδεσης και στη σχεδόν αδύνατη φορητότητα μεταξύ διαφορετικών χωρών (handoff ή handover)[3].

Η ασυμβατότητα, και εν μέρει οι οικονομικές συνέπειες της, οδήγησαν την Ευρωπαϊκή Επιτροπή Τηλεπικοινωνιών και Ταχυδρομείων (CEPT- Conférence Européenne des administrations des Postes et des Télécommunications [5]) το 1982 να ιδρύσει την Groupe Spécial Mobile με στόχο τη δημιουργία ενός τηλεπικοινωνιακού συστήματος με κοινές προδιαγραφές προς χρήση από όλη την Ευρώπη. Τα αρχικά της αποτελούν την προέλευση του αρκτικόλεξου GSM.

Ύστερα από συσκέψεις της CEPT και δοκιμές διαφόρων πλάνων ψηφιακής μετάδοσης οριστικοποιήθηκαν οι βασικές παράμετροι του συστήματος GSM και το Συμβούλιο της Ευρώπης αποφασίζει να εκδώσει μια επίσημη οδηγία για την προκράτηση της δέσμης συχνοτήτων των 900 MHz. Το 1987 στην Κοπεγχάγη υπογράφεται το Μνημόνιο Συνεργασίας (Memorandum of Understanding-MOU) από 14 παρόχους 13 Ευρωπαϊκών χωρών για την υποστήριξη της ανάπτυξης του GSM και της εφαρμογής του. Το 1989 η ευθύνη του GSM ανατέθηκε στο Ευρωπαϊκό Τηλεπικοινωνιακό Ινστιτούτο Προτύπων (ETSI) και η ομάδα των ειδικών της CEPT μετονομάζεται σε ETSI Technical Committee GSM. Το 1991 γίνεται διαθέσιμο το πλήρες σύνολο των προδιαγραφών της πρώτης φάσης (phase-1) και γίνεται επίδειξη

των πειραματικών GSM δικτύων στην Telecom ‘91 της Διεθνής Ένωσης Τηλεπικοινωνιών (ITU) στη Γενεύη.

Το 1992 το σύστημα μετονομάζεται σε Global System for Mobile Communications και τα πρώτα εμπορικά GSM δίκτυα τίθενται σε λειτουργία. Το 1993 οι Αυστραλοί πάροχοι συνιστούν τους πρώτους μη Ευρωπαίους που αποφασίζουν να εφαρμόσουν την τεχνολογία GSM και υπογράφουν το Μνημόνιο Συνεργασίας ενώ το πρώτο προσωπικό δίκτυο επικοινωνίας DCS 1800 (τώρα GSM 1800) άρχισε να λειτουργεί στο Ηνωμένο Βασίλειο από το Mercury One-2-One. Το 1994 λανσάρεται η δυνατότητα μεταφοράς δεδομένων και το 1995 οι υπηρεσίες fax και sms. Οι χρήστες του GSM έχουν φτάσει τα 10 εκατομμύρια σε 100 δίκτυα σε 60 χώρες σε όλο τον κόσμο, ολοκληρώνονται οι προδιαγραφές για τη δεύτερη φάση και το πρώτο δίκτυο PCS 1900 (τώρα GSM 1900) μπαίνει σε λειτουργία από την American Personal Communications[6]. Στην έκδοση 96 περιλαμβάνεται το HSCSD ενώ το 1997 ολοκληρώνεται η έκδοση 97 που περιλαμβάνει και το GPRS. Το HSCSD (High Speed Circuit Switched Data) είναι η βελτιωμένη έκδοση του CSD(circuit switched data) του μηχανισμού μεταφοράς των δεδομένων. Η βασική ιδέα έγκειται στο γεγονός ότι η αυξανόμενη υπολογιστική δύναμη των ψηφιακών επεξεργαστών σήματος στα τερματικά επέτρεπε τη χρήση παραπάνω από μιας χρονοθυρίδας (time slot) με τον ίδιο εξοπλισμό. Πολλοί πάροχοι εισήγαγαν την υπηρεσία του HSCSD με bit rate έως 42 Kbit/s[6]. Το GPRS (General packet radio service) είναι μια υπηρεσία κινητής τηλεφωνίας που χρησιμοποιεί μεταγωγή πακέτων(διαθέσιμο και στα 3G). Συχνά χαρακτηρίζεται ως 2.5G σαν μια ενδιάμεση κατάσταση ανάμεσα στην 2^{η} και 3^{η} γενιά κινητής τηλεφωνίας και χρησιμοποιεί ανεκμετάλλευτα κανάλια TDMA (Time division multiple access - αρχιτεκτονική διαίρεσης χρόνου πολλαπλής πρόσβασης) των δικτύων GSM με μέγιστους ρυθμούς μετάδοσης της τάξης των 115 Kbps και άνω.

Ως το 1998 οι χρήστες παγκοσμίως έχουν φτάσει τα 100 εκατομμύρια και είναι η χρονιά που αποφασίζονται οι βασικές ιδέες του UMTS ενώ η εποψία του περνάει στη νεοσύστατη Third Generation Partnership Project (3GPP)[8]. Η 3GPP είναι ένας οργανισμός συνεργασίας μεταξύ τηλεπικοινωνιακών φορέων (ETSI-Ευρώπη, ARIB/TTC-Ιαπωνία, CCSA-Κίνα, ATIS-Βόρεια Αμερική, TTA-Νότια Κορέα) που αναπτύσσει τις προδιαγραφές για παγκόσμια εφαρμογή του 3G υπό την αρμοδιότητα του ITU. Το 2000 ολοκληρώνεται, από την 3GPP, την ETSI Technical Committee SMG και την ANSI T1P1, η GSM/UMTS έκδοση 99 που αποτελεί τη βάση για την εξέλιξη του GSM και την έναρξη του UMTS το 2002 και η 3GPP αναλαμβάνει πια τη μελλοντική δουλειά πάνω στο GSM. Το 2003 άρχισε να εφαρμόζεται το EDGE (Enhanced Data Rates for GSM Evolution),αρχικά από την Cingular στις ΗΠΑ, το οποίο είναι μια τεχνολογία που παρέχει βελτιωμένους ρυθμούς μετάδοσης δεδομένων σε σχέση με το GPRS προσθέτοντας στο GSM κανάλι την τεχνική διαμόρφωσης φάσης 8PSK και διαμόρφωση GMSK (Gaussian Minimum Shift Keying) μεταξύ άλλων. Χαρακτηρίζεται και ως 2.75G. Το 2.5G EDGE “compact” χρησιμοποιεί κανάλια 60 KHz και το 3G EDGE “classic” κανάλια 200 KHz και η ταχύτητα

μετάδοσης δεδομένων φτάνει τα 384 kilobits/sec αλλά δεδομένων των συνθηκών του δικτύου επιτυγχάνονται 60-80 kilobits/sec[9]. Οι εφαρμογές μεταφοράς δεδομένων σε υψηλές ταχύτητες όπως διάφορες υπηρεσίες πολυμέσων επωφελούνται από την αυξημένη χωρητικότητα δεδομένων.

Το Μάρτιο του 2003 ο τηλεπικοινωνιακός πάροχος 3 άρχισε να παρέχει την 3G υπηρεσία στο Ηνωμένο Βασίλειο.

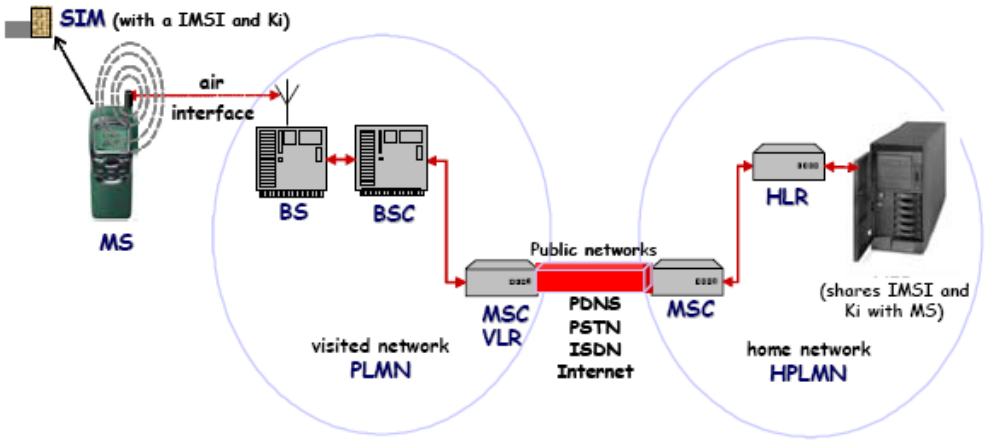
Στον πίνακα 1 παρουσιάζονται τα φάσματα των παρόχων της κινητής τηλεφωνίας στην Ελλάδα σύμφωνα με την ΕΕΤΤ[14].

ΕΤΑΙΡΙΑ ΚΙΝΗΤΗΣ ΤΗΛΕΦΩΝΙΑΣ	**ΥΠΗΡΕΣΙΑ**	**ΑΠΟΝΕΜΗΘΕΝ ΦΑΣΜΑ (MHz)**
COSMOTE	GSM 900	885 – 890 930 – 935
COSMOTE	DCS 1800	1760 – 1785 1855 – 1880
COSMOTE	UMTS	1950.3 – 1965.3 2140.3 – 2155.3 1905.1 – 1910.1
WIND	GSM 900	890 – 900 935 - 945
WIND	DCS 1800	1730 –1745 1825 – 1840
WIND	UMTS	1940.3–1950.3 2130.3–2140.3 1910.1 – 1915.1
VODAFONE	GSM 900	900 – 915 945 - 960
VODAFONE	DCS 1800	1745 – 1760 1840 – 1855
VODAFONE	UMTS	1920.3–1940.3 2110.3–2130.3 915.1 – 1920.1

Πίνακας 1 : Τα δίκτυα κινητής τηλεφωνίας στην Ελλάδα

1.2 Αρχιτεκτονική στο GSM

Η βασική σχεδίαση συνίσταται από τρία μείζονα υποσυστήματα: το ασύρματο υποσύστημα(radio subsystem), το υποσύστημα δικτύου(network subsystem) και το υποσύστημα υποστήριξης λειτουργίας(operation support subsystem).[4]



Εικόνα 1: Επισκόπηση του συστήματος GSM[10]

1.2.1 Το ασύρματο υποσύστημα-Radio subsystem

Περιλαμβάνει τον εξοπλισμό και τις λειτουργίες που σχετίζονται με τη διαχείριση της ραδιοεπαφής συμπεριλαμβανομένου της διαχείρισης των handovers, της μεταβίβασης της λειτουργίας σε διαφορετικές περιοχές. Αποτελείται από τον MS, τον BTS και τον BSC.

Κινητός σταθμός - Mobile Station (MS)
Αποτελείται από τον εξοπλισμό και το λογισμικό που χρειάζεται για να εδραιωθεί η επικοινωνία με το δίκτυο, δηλαδή την εκάστοτε φορητή συσκευή και την κάρτα SIM (Subscriber Identity Module). Ο κινητός σταθμός ανήκει μέσω συμβολαίου στο οικείο δίκτυο (HPLMN-Home Public Land Mobile Network) αλλά όταν βρίσκεται εκτός της περιοχής κάλυψης μπορεί να συνδεθεί στο επισκεπτόμενο VPLMN(Visited PLMN).
Παραδοσιακά ο MS συγκαταλέγεται στο ασύρματο υποσύστημα αν και επικοινωνεί με το υποσύστημα δικτύου όσον αφορά τη φορητότητά του (κάποιες φορές αναφέρεται σαν αυτούσιο υποσύστημα)[4].

Σταθμός βάσης πομποδέκτη - Base Transceiver Station(BTS)
Ο BTS είναι ο διαδραστικός εξοπλισμός που διαχειρίζεται την επικοινωνία μεταξύ του χρήστη και του δικτύου. Σε κάθε κελί (cell) υπάρχει ένας BTS με αρκετούς πομποδέκτες και κωδικοποιεί, κρυπτογραφεί, πολυπλέκει και τροφοδοτεί τα τηλεφωνικά σήματα στην κεραία του[7]. Πολλοί BTS επιβλέπονται από έναν BSC.

Σταθμός βάσης ελεγκτή - Base Station Controller (BSC)
Ο BSC είναι όπως υποδηλώνει και το όνομά του ο ελεγκτικός μηχανισμός των λειτουργιών των BTS και των MS. Μερικές από τις λειτουργίες που επιβλέπει είναι το πρωτόκολλο του handover και ο έλεγχος ισχύος[4]. Επίσης διαχειρίζεται τους πόρους του δικτύου, μέσω της ανακατανομής των συχνοτήτων ανάμεσα στα κελιά και της εκχώρησης και αποδέσμευσης των συχνοτήτων και των χρονοθυρίδων (timeslots) για τους MS, αναλαμβάνει τα σήματα συγχρονισμού χρόνου και συχνοτήτων στους BTS, τη μέτρηση της χρονοκαθυστέρησης και τη γνωστοποίηση ενός κινητού σταθμού στον BTS[7]).

Ο BTS και ο BSC συχνά αναφέρονται και σαν υποσύστημα σταθμού βάσης.

1.2.2 Το υποσύστημα δικτύου-Network subsystem

Το υποσύστημα δικτύου σχετίζεται με τη διαχείριση της επικοινωνίας ανάμεσα στους χρήστες και μέσω των λειτουργιών που εκτελούνται στα τμήματα που το απαρτίζουν εξασφαλίζει τη φορητότητα, την αποθήκευση των στοιχείων των πελατών, την επικοινωνία με χρήστες PSTN-ISDN καθώς και την ασφάλεια των κλήσεων.

Αποτελείται από το MSC, τον Home Location Register (HLR), τον Visitor Location Register (VLR), το Authentication Center (AUC) και τον Equipment ldentity Register (EIR).

Κέντρο Μεταγωγής Κινητής Τηλεφωνίας - Mobile Switching Centre(MSC)
Το κέντρο μεταγωγής είναι υπεύθυνο για την έναρξη και όλα τα στάδια της δρομολόγησης των κλήσεων ανάμεσα στους BSC που ελέγχει καθώς και σε άλλα κέντρα μεταγωγής άλλων περιοχών. Επίσης επιβλέπει τη διεπαφή με το PSTN, το ISDN και το internet.

Μητρώο θέσης οικείων συνδρομητών - Home Location Register(HLR)
Η βάση δεδομένων που περιέχει τις πληροφορίες για τον χρήστη και τη θέση του στο δίκτυο συμπεριλαμβανομένων και των στοιχείων της κάθε SIM (μέσω του IMSI - International Mobile Subscriber Identity).

Μητρώο θέσης επισκεπτών - Visitor Location Register(VLR)
Η προσωρινή βάση δεδομένων που αποθηκεύει τα στοιχεία των MS που εντοπίζονται στη δικαιοδοσία ενός MSC. Ο VLR αναθέτει στα κινητά ένα TIMSI (Temporary Mobile Subscriber Identity) για να περιοριστεί η συνεχής χρήση του IMSI προς αποφυγήν κινδύνων κλοπής των στοιχείων του χρήστη.

Κέντρο Αυθεντικότητας - Authentication Centre(AUC)
Η βάση δεδομένων AUC περιέχει τα κλειδιά των κρυπτογραφικών αλγορίθμων για τη διασφάλιση του απόρρητου των κλήσεων και τις παραμέτρους για την αναγνώριση της αυθεντικότητας του χρήστη.

Μητρώο ταυτότητας εξοπλισμού - Equipment Identity Register(EIR)
Η βάση δεδομένων που αποθηκεύει τα IMEI(International Mobile Equipment Identity) για όλες τις εγγεγραμμένες συσκευές κινητής τηλεφωνίας. ούτως ώστε να αναγνωρίζονται οι παράνομες συσκευές που αναγνωρίζονται στον MSC. Τα IMEI κατηγοριοποιούνται στην άσπρη λίστα(έγκυρα κινητά),στην γκρι λίστα(ελαττωματικά και «ύποπτα» κινητά) και στη μαύρη λίστα(κλεμμένα κινητά)[11].

1.2.3 Το υποσύστημα υποστήριξης λειτουργίας-operation support subsystem

Το υποσύστημα υποστήριξης λειτουργίας αποτελεί τη λειτουργική οντότητα που επιβλέπει και ελέγχει σε σύνολο το σύστημα. Μέσω του OMC-Operation Maintenance Centre διαχειρίζεται τη λειτουργικότητα και αποδοτικότητα του δικτύου, ελέγχει την ποιότητα του σήματος, εκτελεί τις χρεωστικές διαδικασίες και παρέχει τη υποστήριξη και συντήρηση του εξοπλισμού του GSM. Συνδέεται με το υποσύστημα δικτύου και τον σταθμό βάσης ελεγκτή.[4]

Κεφάλαιο 2: Ασφάλεια στο GSM

2.1 Περιγραφή των παραμέτρων της ασφάλειας στο GSM

Η εκτενέστατη χρήση των κινητών τηλεφώνων απαιτεί τη ύπαρξη μέτρων ασφαλείας ως προς τη διασφάλιση της εμπιστευτικότητας της επικοινωνίας των χρηστών και ως προς αποφυγήν των παράνομων υποκλοπών. Ο βασικός λόγος είναι οτι η κινητή τηλεφωνία χρησιμοποιεί την εναέρια διεπαφή (air interface) και συνεπώς είναι πιο ευάλωτη σε επιθέσεις απ'ότι το επίγειο δίκτυο.

Η ασφάλεια στο GSM πρέπει να εγγυάται τη διασφάλιση των παρακάτω:

- Πιστοποίηση και προστασία της ταυτότητας των χρηστών
- Εμπιστευτικότητα και προστασία των μεταφερόμενων πληροφοριών μέσω της κωδικοποίησης και της κρυπτογράφησης

Το πρώτο μέτρο ασφάλειας του GSM έγκειται στην ίδια του τη φύση: είναι ένα ψηφιακό σύστημα που χρησιμοποιεί αλγόριθμο κωδικοποίησης φωνής, ψηφιακή διαμόρφωση GMSK (Gaussian Minimum Shift Keying), την τεχνική του slow frequency hopping και αρχιτεκτονική διαίρεσης χρόνου πολλαπλής πρόσβασης (TDMA). Για την υποκλοπή και την ανακατασκευή τέτοιου σήματος χρειάζεται πιο πολύπλοκος εξοπλισμός από ότι για ένα αναλογικό σήμα όπου αρκεί ένας σαρωτής. Εμφανώς τα παραπάνω δεν αρκούν οπότε οι κυριότερες υπηρεσίες ασφαλείας είναι οι:

1.Αυθεντικότητα

2.Κρυπτογράφηση της επικοινωνίας των χρηστών

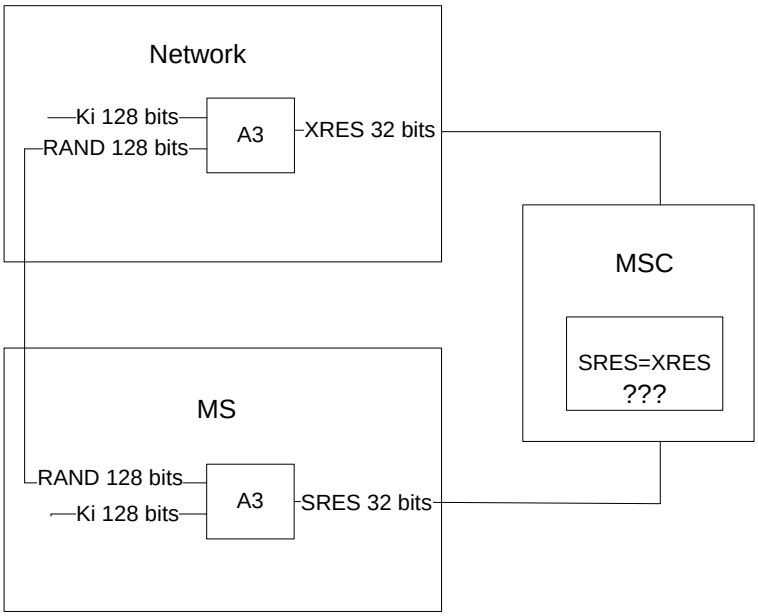
2.1.1 Αυθεντικότητα – A3 αλγόριθμος

Η εγκυρότητα της ταυτότητας ενός χρήστη πιστοποιείται μέσω της κάρτας SIM. Η κάρτα SIM (Subscriber Identity Module) είναι συσκευή με μικροεπεξεργαστή, μοναδική στο δίκτυο και απαραίτητη για τη λειτουργία μιας τηλεφωνικής συσκευής. Περιέχει τον IMSI (International Mobile Subscriber Identity), ο οποίος είναι μόνιμος και αποκλειστικός για κάθε συνδρομητή σε όλο τον κόσμο και το μυστικό κλειδί πιστοποίησης Ki επίσης μοναδικό. Στον μικροεπεξεργαστή της εκτελούνται οι αλγόριθμοι πιστοποίησης και παραγωγής κλειδιών.

Το Ki είναι ένας τυχαίος 128-bit αριθμός, γνωστός μόνο στη SIM και στο AUC που αποδεικνύει την αυθεντικότητα του κινητού σταθμού και τροφοδοτεί την παραγωγή όλων των κλειδιών και των προκλήσεων που χρησιμοποιούνται στο GSM σύστημα.

Η χρήση του TMSI αντί του IMSI για την αναγνώριση ενός κινητού σταθμού σε κάθε αλλαγή της ευρισκόμενης περιοχής εγγυάται την προστασία της αναγνώρισης του χρήστη που βρίσκεται σε κάθε περιοχή.[10]

Η διαδικασία της πιστοποίησης συνίσταται από την απόδειξη της SIM ότι γνωρίζει το Ki. Σε αυτό το σημείο χρησιμοποιείται ο αλγόριθμος A3. Ο AUC εκπέμπει μια τυχαία πρόκληση, έναν 128-bit τυχαίο αριθμό (RAND) στον MS. Στη SIM είναι υλοποιημένος ο αλγόριθμος πιστοποίησης A3, ο οποίος χρησιμοποιώντας το κλειδί Ki κρυπτογραφεί τη RAND και παράγει την SRES μια 32-bit απόκριση. Ταυτόχρονα ο A3 εκτελείται και στον AUC και παράγεται η XRES (32-bit). Οι SRES και XRES στέλνονται στο δίκτυο όπου συγκρίνονται για να ολοκληρωθεί η διαδικασία της πιστοποίησης. Την επόμενη φορά που θα χρειαστεί να ξαναξεκινήσει θα εκδοθεί μια καινούρια RAND έτσι ώστε να καλύπτεται και η περίπτωση κάποιος να έχει υποκλέψει την SRES [10]. Σημειώνεται ότι η απόδειξη της αυθεντικότητας είναι μονόδρομη συνάρτηση, δηλαδή το δίκτυο δεν αποδεικνύει την εγκυρότητά του παρότι είναι πολύ εύκολο να υλοποιηθεί και η αντίστροφη λειτουργία της πιστοποίησης, γεγονός που καθιστά δυνατές ορισμένου είδους επιθέσεις που προέρχονται από αυτή την αδυναμία.



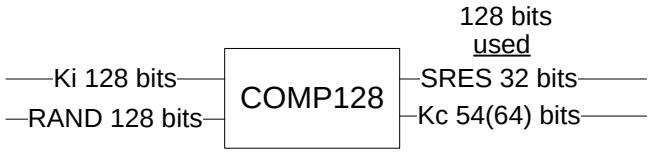
Εικόνα 2 : Πιστοποίηση μέσω του αλγορίθμου A3

2.1.2 Κρυπτογράφηση της επικοινωνίας των χρηστών

Ο κρυπτογραφικός αλγόριθμος χρησιμοποιείται για να διασφαλίσει οτι τα δεδομένα των χρηστών (όπως ομιλία και sms) θα προστατεύονται από αθέμιτες ενέργειες, όπως το «κρυφάκουσμα» στην εναέρια διεπαφή.
Ο βασικός κρυπτογραφικός αλγόριθμος είναι ο A5, ο οποίος έχει ως προαπαιτούμενο το κλειδί που εξάγεται από την εφαρμογή του αλγορίθμου A8.

2.1.2.1 A8 αλγόριθμος

Ο A8 είναι υλοποιημένος και στον κινητό σταθμό(στη SIM) και στο δίκτυο (στον AUC). Είναι συνάρτηση μιας κατεύθυνσης που σκοπός της είναι η εξαγωγή του βασικού κρυπτογραφικού κλειδιού Kc. Εκτελείται ταυτόχρονα με τον υπολογισμό των αποκρίσεων SRES και XRES, με εισόδους το Ki και τη RAND και έξοδο το 64-bits Kc το οποίο αποστέλλεται στον BS για να ξεκινήσει η διαδικασία της κρυπτογράφησης (και της αποκρυπτογράφησης) μέσω του A5 αλγορίθμου.
Ο A3 και ο A8 μπορούν να έχουν διαφορετική σχεδίαση σύμφωνα με τον κάθε διαχειριστή δικτύου αλλά συνήθως συνδυάζονται σε έναν αλγόριθμο A3/A8. Η πιο συνήθης υλοποίηση για το συνδυασμό A3/A8 είναι ο COMP128 που δίνεται στο μνημόνιο συνεργασίας MoU.
Στην παρακάτω εικόνα φαίνονται οι είσοδοι και έξοδοι του COMP128. Το μήκος του κλειδιού Kc είναι 54 bits αντί 64 bits που περιμένει ως είσοδο ο A5. Ο COMP128 προσαρτεί δέκα μηδενικά bits στο Kc, γεγονός που μειώνει εσκεμμένα σημαντικά το εύρος του κλειδιού.



Εικόνα 3 : COMP128 (Αλγόριθμος A3/A8)

2.1.2.2 A5 αλγόριθμος

Ο A5 αλγόριθμος είναι ο βασικός κρυπτογραφικός αλγόριθμος ο οποίος μετασχηματίζει το αρχικό κείμενο (plaintext) σε ένα κρυπτογραφημένο κείμενο (ciphertext) και αντίστροφα, με τη χρήση του κρυπτογραφικού κλειδιού Kc που έχει εξαχθεί από τον A8. Εκτελείται στην κινητή συσκευή, και όχι στη SIM για λόγους ταχύτητας, και στον σταθμό βάσης.

Οι υλοποιήσεις του A5 που παρέχουν διάφορα επίπεδα κρυπτογράφησης είναι οι εξής:

- A5/0 αποτελεί υλοποίηση ως ευφημισμός καθώς δεν χρησιμοποιεί κρυπτογράφηση (χώρες που έγκεινται στους περιορισμούς του ITAR- International Traffic in Arms Regulations). Η Ινδία είναι μια από τις πολλές χώρες που δεν χρησιμοποιούν καθόλου κρυπτογράφηση
- A5/1 είναι ο αρχικός αλγόριθμος που πρωτοχρησιμοποιήθηκε στην Ευρώπη και ο επικρατέστερος την παρούσα στιγμή
- A5/2 είναι η ασθενέστερη εκδοχή του A5/1 που δημιουργήθηκε για να εξαχθεί στις ΗΠΑ και τελικά χρησιμοποιείται ανάμεσα σε άλλες και σε χώρες της Μέσης Ανατολής
- A5/3 είναι η ισχυρότερη εκδοχή που αναπτύχτηκε στα πλαίσια της 3GPP[12]

Σε επίπεδο αρχιτεκτονικής δικτύου οι διαδικασίες που συμβαίνουν για την έναρξη της λειτουργίας κρυπτογράφησης και αποκρυπτογράφησης είναι οι εξής:

Ο BS, μόλις έχει στη διάθεσή του το κλειδί Kc στέλνει στον MS ένα μήνυμα ΄start cipher΄, που περιέχει και πληροφορίες για το ποια έκδοση του A5 διαθέτει και ταυτόχρονα αρχίζει να αποκρυπτογραφεί. Όταν ο MS λάβει αυτό το μήνυμα αρχίζει να κρυπτογραφεί και να αποκρυπτογραφεί ενώ στο άκρο του BS η κρυπτογράφηση αρχίζει μόλις αποκρυπτογραφήσει σωστά ένα ληφθέν πλαίσιο από τον MS[13]. Ένα καινούριο Kc δημιουργείται για κάθε κλήση.
Όταν συμβεί ένα handover κατά τη διάρκεια μιας κλήσης οι απαραίτητες πληροφορίες μεταφέρονται στο νέο σταθμό βάσης και η κρυπτογράφηση συνεχίζεται με το ίδιο Kc.[10]

i) Περιγραφή βασικής λειτουργίας του A5

Ο A5 ακολουθεί τις βασικές αρχές των αλγορίθμων ροής. Η γεννήτρια ροής κλειδιού παράγει το keystream σε συνάρτηση του κλειδιού κρυπτογράφησης Kc και του μετρητή πλαισίου (frame counter). Εν συνεχεία το keystream περνάει από μια πράξη αποκλειστικού-ή (XOR) με το αρχικό κείμενο και δίνει το κρυπτογραφημένο κείμενο. Η αντίστροφη διαδικασία εκτελείται για την αποκρυπτογράφηση.

Encryption	**►**	**Keystream XOR Plaintext = Ciphertext**
Decryption	**►**	**Keystream XOR Ciphertext=Plaintext**

ii) Τεχνικές προδιαγραφές

Είσοδοι
Οι είσοδοι του A5 είναι το Kc(64 bits) και ο Frame COUNT(22 bits).

Kc: Το κρυπτογραφικό κλειδί που εξάγεται από τον αλγόριθμο A8 όπως περιγράφηκε παραπάνω με μήκος 64 bits (από τον A8-COMP128 τα 10 λιγότερο σημαντικά bits είναι 0). Υπενθυμίζουμε ότι ένα καινούριο Kc δημιουργείται για κάθε κλήση.

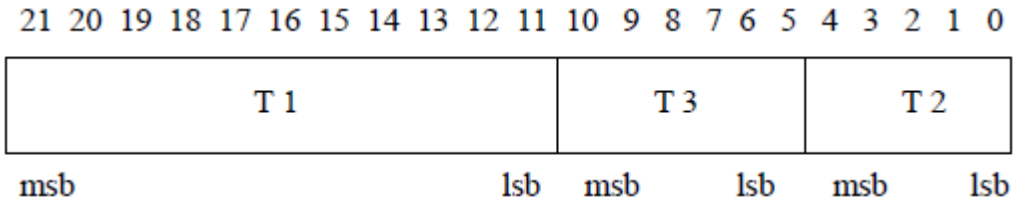
Frame COUNT: Για την επεξήγηση της λειτουργίας του COUNT χρειάζονται κάποιες πληροφορίες σχετικά με την αρχιτεκτονική διαίρεσης χρόνου πολλαπλής πρόσβασης (TDMA).

Στο TDMA ο άξονας του χρόνου διαχωρίζεται επιτρέποντας σε διάφορους χρήστες να μοιράζονται το ίδιο κανάλι συχνοτήτων για συγκεκριμένο χρονικό διάστημα (timeslot). Χρησιμοποίηση ενός καναλιού σημαίνει εκπομπή και λήψη ριπών όπου ριπή (burst) ονομάζεται το φυσικό περιεχόμενο της χρονοθυρίδας, π.χ. πληροφορίες φωνής, δεδομένων ή ελέγχου.

Κάθε κανονική ριπή(normal burst) έχει μήκος 156,25 bits (148 bits πληροφορία και 8.25 guard bits) και διαρκεί 0.577 ms. Ένα πλαίσιο αποτελείται από 8 ριπές.

Επομένως μια συνδιάλεξη GSM αποστέλλεται ως μια αλληλουχία από πλαίσια κάθε 4,615 ms που διαρκεί το καθένα. Κάθε TDMA πλαίσιο συνδέεται με έναν αριθμό πλαισίου(frame number), ο οποίος ορίζεται για όλες τις χρονοθυρίδες του πλαισίου και αυξάνεται κατά 1 πριν αρχίσει το επόμενο πλαίσιο.

Ο COUNT(frame counter) είναι μια 22-bit τιμή η οποία εξάγεται από τον αριθμό πλαισίου όπως περιγράφεται στην παρακάτω εικόνα,



Εικόνα 4 : Τα μέρη του frame counter[20]

όπου T1 είναι το πηλίκο της διαίρεσης του αριθμού πλαισίου με τον αριθμό 51*26=1326, T2 το υπόλοιπο του αριθμού πλαισίου διαιρεμένο με 26 και T3 το υπόλοιπο διαιρεμένο με 51 [20].

Επειδή ο αριθμός πλαισίου αλλάζει κάθε 4.615 ms ο κύκλος επανάληψης του COUNT είναι 5.4 ώρες.

Έξοδοι

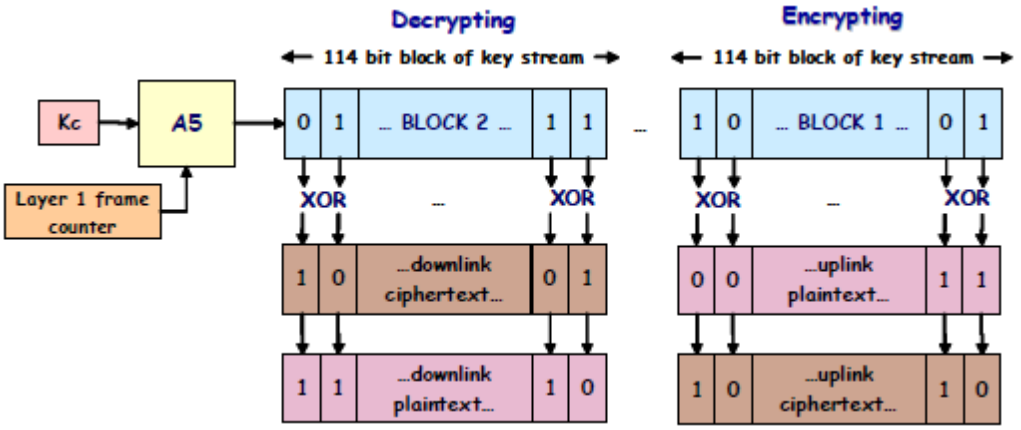
Η εκτέλεση του A5 παράγει το keystream σε δύο τμήματα των 114 bits. Τα πρώτα 114 bits αποτελούν το BLOCK1 και τα επόμενα 114 bits το BLOCK2.

Στον κινητό σταθμό το BLOCK1 χρησιμοποιείται για να κρυπτογραφήσει το αρχικό μήνυμα δηλαδή γίνεται XOR με το plaintext των ζεύξεων ανόδου (uplink-κινητός σταθμός προς σταθμό βάσης) και το BLOCK2 για να αποκρυπτογραφήσει το

κρυπτογραφημένο μήνυμα δηλαδή γίνεται XOR με το ciphertext των ζεύξεων καθόδου (downlink-σταθμός βάσης προς κινητό σταθμό).

Στον σταθμό βάσης το BLOCK1 χρησιμοποιείται για να αποκρυπτογραφήσει το κρυπτογραφημένο μήνυμα των ζεύξεων ανόδου και το BLOCK2 για να κρυπτογραφήσει το κείμενο προς μετάδοση των ζεύξεων καθόδου.

Η περιγραφείσα διαδικασία από την πλευρά του κινητού σταθμού φαίνεται στις εικόνες που ακολουθούν σχηματικά [10,6].



Εικόνα 5 : Λειτουργία του A5/1 στον κινητό σταθμό[10],[6]

Αρχικό κείμενο(Plaintext) και κρυπτογραφημένο κείμενο(Ciphertext)

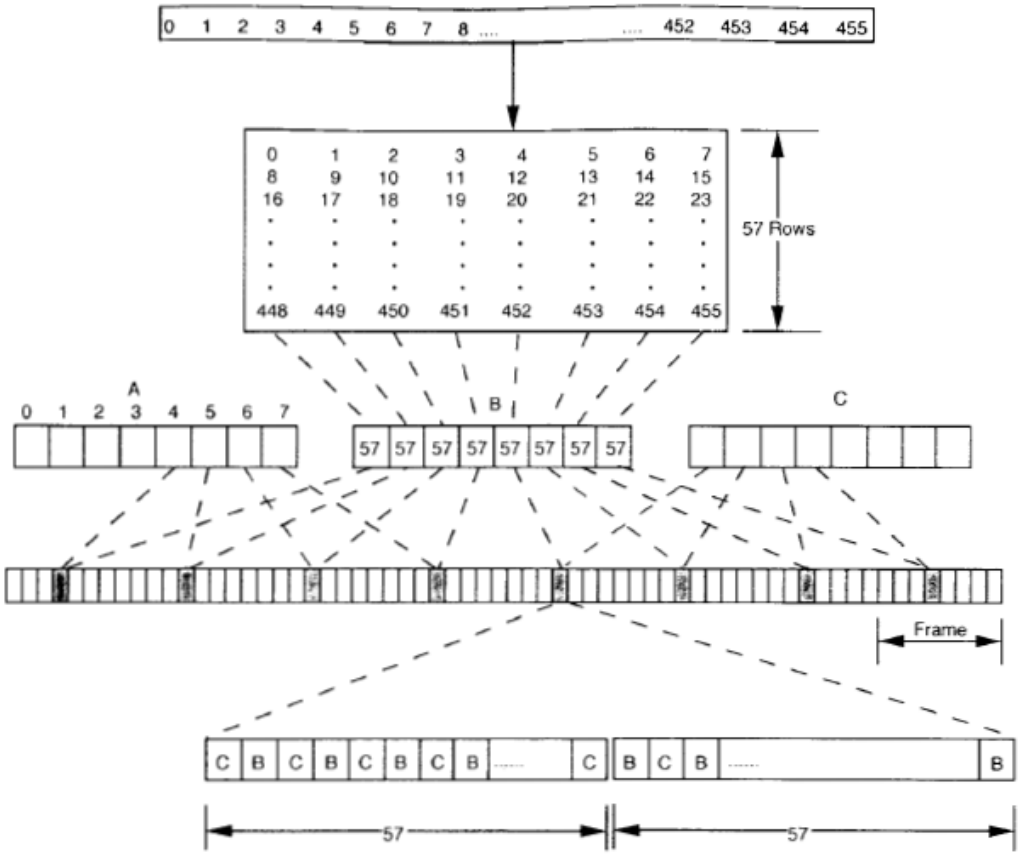
Λόγω της τεχνικής του TDMA το αρχικό κείμενο είναι οργανωμένο σε τμήματα των 114 bits. Κάθε τμήμα ενσωματώνεται σε μια κανονική ριπή και μεταδίδεται κατά τη διάρκεια μιας χρονοθυρίδας, περιέχοντας είτε το ψηφιοποιημένο κείμενο επικοινωνίας από τον κινητό σταθμό στο σταθμό βάσης είτε αντιστρόφως από το σταθμό βάσης στον κινητό σταθμό.

Ο λόγος που το αρχικό κείμενο είναι μήκους 114 bits έγκειται στη διαδικασία δημιουργίας bits από την ομιλία. Μετά την κωδικοποίηση της φωνής, λαμβάνουμε ένα ψηφιακό σήμα μήκους 260 bits και διάρκειας 20 ms, το οποίο μετά την κωδικοποίηση καναλιού είναι 456 bits**.** Μέσω της τεχνικής του interleaving τα 456 bits υποδιαιρούνται σε 8 υποτμήματα των 57 bits με τον ακόλουθο τρόπο: το πρώτο υποτμήμα των 57 bits περιέχει τα bits (0, 8, 16, ...,448), το δεύτερο τα (1, 9, 17, ...,449) και ούτω καθεξής με το τελευταίο να έχει τα bits (7, 15, 23, ...,455). Έτσι τα γειτονικά bits βρίσκονται σε διαφορετικά υποτμήματα. Τα πρώτα τέσσερα υποτμήματα τοποθετούνται στα άρτια αριθμημένα bits τεσσάρων διαδοχικών ριπών, και τα επόμενα τέσσερα υποτμήματα τοποθετούνται στα περιττά bits των 4 επόμενων

διαδοχικών ριπών. Αφού κάθε ριπή περιέχει 114 bits πληροφορίας στην πραγματικότητα είναι μοιρασμένη από 2 υποτμήματα από διαφορετικά ψηφιακά σήματα 20 ms. Κάθε ριπή θα αποτελείται από ένα υποτμήμα από το προηγούμενο σήμα 20 ms και ένα υποτμήμα από το αμέσως επόμενο 20 ms.

Σκοπός της διαδικασίας του interleaving είναι να διασφαλίσει την ευκολότερη διόρθωση σφαλμάτων από τους αντίστοιχους μηχανισμούς, μετά τη διαδικασία του de-interleaving στον αποδέκτη. Λόγω του ότι τα bits κάθε ψηφιοποιημένου μηνύματος διαμοιράζονται σε 8 ριπές η περίπτωση λανθασμένης λήψης μιας ριπής δεν επηρεάζει τη ολική ποιότητα της μετάδοσης καθώς διορθώνεται εύκολα από τις τεχνικές διόρθωσης σφαλμάτων.

Η διαδικασία φαίνεται στην παρακάτω εικόνα.



Εικόνα 6 : Η διαδικασία του interleaving από [4]

2.2 Υλοποιήσεις του A5 αλγορίθμου

Οι αλγόριθμοι συμμετρικού κλειδιού, δηλαδή οι αλγόριθμοι που χρησιμοποιούν το ίδιο κλειδί για την κρυπτογράφηση και την αποκρυπτογράφηση, χωρίζονται σε δύο κατηγορίες:

- Κρυπταλγόριθμους τμήματος (block ciphers)
- Κρυπταλγόριθμους ροής (stream ciphers)

Κρυπταλγόριθμοι τμήματος -Block ciphers

Οι κρυπταλγόριθμοι τμήματος λαμβάνουν ένα τμήμα (block) του αρχικού κειμένου (plaintext) και σε συνδυασμό με το κλειδί κρυπτογράφησης (cipher key) εξάγουν ένα κρυπτογραφημένο κείμενο (ciphertext). Το μήκος του block είναι το ίδιο με το μήκος του ciphertext και συνήθως και με του κλειδιού. Στην κατηγορία αυτή ανήκουν ο DES,3DES,AES, RC2,RC5,RC6, MISTY1.

Κρυπταλγόριθμοι ροής-Stream ciphers

Οι κρυπταλγόριθμοι ροής τυπικά λειτουργούν επί μικρότερων τμημάτων του plaintext (1 bit ή 1 byte). Μια γεννήτρια ροής κλειδιού (keystream generator) παραγάγει μια ακολουθία από ψευδο-τυχαία bits (keystream), είτε συνηθέστερα ανεξάρτητη του αρχικού κειμένου plaintext και του κρυπτογραφημένου κειμένου ciphertext (synchronous stream cipher), είτε σε συνάρτηση των προαναφερθέντων (self-synchronizing stream cipher). Το κρυπτογραφημένο κείμενο (ή η ανακατασκευή του αρχικού) πραγματοποιείται με το συνδυασμό του keystream με το αρχικό (ή το κρυπτογραφημένο αντίστοιχα) συνήθως μέσω μιας πράξης αποκλειστικού-ή (XOR). Οι αλγόριθμοι ροής είναι μια προσέγγιση της λειτουργίας του one-time-pad και εκτελούνται πολύ ταχύτερα από ότι οι αλγόριθμοι τμήματος. Στην κατηγορία αυτή ανήκουν ο RC4 και οι αλγόριθμοι ενδιαφέροντος A5/1 και A5/2.

2.2.1 Υλοποίηση του A5/1

Ο αλγόριθμος A5/1 είναι ένας αλγόριθμος ροής. Η γεννήτρια ροής κλειδιού του A5/1 βασίζεται σε 3 LFSR (Linear Feedback Shift Register) οι οποίες χρονίζονται σύμφωνα με έναν κανόνα παραγόμενο από συγκεκριμένα bits της κάθε LFSR. Η έξοδος ισούται με την πράξη XOR του περισσότερο σημαντικού bit των εξόδων των LFSR.

Αναλυτικά οι 3 LFSRS είναι μεγίστου μήκους (maximum-length) με πρωταρχικά πολυώνυμα ανάδρασης (primitive polynomials). Ονομάζοντας τις LFSRs R1,R2 και R3 τα χαρακτηριστικά τους είναι τα παρακάτω:

- Η R1 έχει μήκος 19 bits, περίοδο 2^{19}-1 και πρωταρχικό πολυώνυμο $\mathbf{x^{19} + x^5 + x^2 + x + 1}$, δηλαδή τα bits που συμμετέχουν στην ανάδραση(tap bits) είναι τα 13,16,17,18 (ξεκινώντας από το 0).

- Η R2 έχει μήκος 22 bits, περίοδο 2^{22}-1 και πρωταρχικό πολυώνυμο $\mathbf{x^{22} + x + 1}$, δηλαδή τα bits που συμμετέχουν στην ανάδραση είναι τα 20,21.

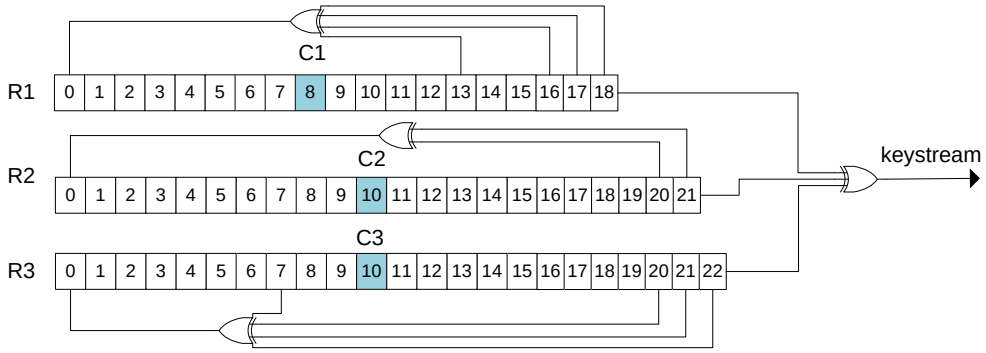
- Η R3 έχει μήκος 23 bits, περίοδο 2^{23}-1 και πρωταρχικό πολυώνυμο $\mathbf{x^{23} + x^{15} + x^2 + x + 1}$, δηλαδή τα bits που συμμετέχουν στην ανάδραση είναι τα 7,20,21,22.

Register Number	Length In bits	Primitive Polynomial	Clock-controlling bit (LSB is 0)	Bits that are XORed
1	19	$\mathbf{x^{19} + x^5 + x^2 + x + 1}$	8	18,17,16,13
2	22	$\mathbf{x^{22} + x + 1}$	10	21,20
3	23	$\mathbf{x^{23} + x^{15} + x^2 + x + 1}$	10	22,21,20,7

Πίνακας 2 : Οι παράμετροι του A5/1[15]

Κάθε ένας από τους R1,R2,R3 ανανεώνεται ακολουθώντας το δικό του πολυώνυμο και δεν εξαρτάται από τους άλλους δυο. Η λειτουργία του χρονισμού όμως είναι πιο πολύπλοκη και βασίζεται στον μηχανισμό του ρολογιού. Ο μηχανισμός του ρολογιού λειτουργεί ως εξής: κάθε καταχωρητής έχει ένα bit ρολογιού που συμβολίζεται με C1,C2 και C3 αντίστοιχα. Το C1 είναι το bit 8 του R1, το C2 είναι το bit 10 του R2 και το C3 είναι το bit 10 του R3. Ονομάζουμε bit πλειοψηφίας την τιμή του bit που έχουν στα bits ρολογιού οι περισσότεροι καταχωρητές. Ένας καταχωρητής χρονίζεται μόνο όταν το bit ρολογιού του «συμφωνεί» με την τιμή του bit πλειοψηφίας. Με αυτόν τον τρόπο τουλάχιστον οι 2 καταχωρητές χρονίζονται σε κάθε κύκλο ρολογιού δίνοντας πιθανότητα 3/4 να προχωρήσει μια lfsr και 1/4 να σταματήσει.

Στην εικόνα 7 αναπαρίσταται η δομή των καταχωρητών, ο μηχανισμός του ρολογιού και η έξοδος του A5/1.



Εικόνα 7 : Οι καταχωρητές R1,R2 και R3

Οι διαδικασίες που διενεργούνται στην εσωτερική κατάσταση του A5/1 είναι οι εξής[29]:

Βημα 1^{ο}: Οι καταχωρητές μηδενίζονται και το 64-bit Kc (από το lsb στο msb) γίνεται παράλληλα XOR με το lsb του καθενός από τους 3 καταχωρητές. Η διαδικασία αυτή διαρκεί 64 κύκλους κατά τους οποίους οι LFSRs χρονίζονται κανονικά και όχι σύμφωνα με το μηχανισμό του ρολογιού που περιγράφηκε παραπάνω. Ακολουθεί ο 22-bit COUNT(από το lsb στο msb) ο οποίος γίνεται παράλληλα XOR στο lsb των καταχωρητών, όπως το Kc, μέσα σε 22 κύκλους κανονικού χρονισμού. Στο τέλος αυτού του βήματος τα περιεχόμενα των 3 καταχωρητών ονομάζονται η αρχική κατάσταση του αλγορίθμου(initial state).

Βήμα 2^{ο}: Εν συνεχεία όλοι οι καταχωρητές χρονίζονται για 100 κύκλους χρησιμοποιώντας το μηχανισμό του ρολογιού(τον κανόνα της πλειοψηφίας) χωρίς να παράγουν έξοδο.

Βήμα 3^{ο}: Όλοι οι καταχωρητές χρονίζονται για 228 κύκλους σύμφωνα με το μηχανισμό του ρολογιού παράγοντας 228 bits εξόδου(keystream output). Σε κάθε κύκλο παράγεται ένα bit εξόδου μέσω της πράξης XOR στα msb των τριών καταχωρητών.

Τα παραπάνω βήματα φαίνονται σχηματικά στην εικόνα 8.

1. Set R1 = R2 = R3 = 0.
   For i = 0 to 63 do
   a) R1[0] ←R1[0] ⊕ Kc[i];
   b) R2[0] ←R2[0] ⊕ Kc[i];
   c) R3[0] ←R3[0] ⊕ Kc[i];
   d) Clock all three registers.
   For i = 0 to 21 do
   a) R1[0] ←R1[0] ⊕ COUNT[i];
   b) R2[0] ←R2[0] ⊕ COUNT[i];
   c) R3[0] ←R3[0] ⊕ COUNT[i];
   d) Clock all three registers.

2. For i = 0 to 99 do
   Clock with majority rule and discard the output.

3. For i = 0 to 227 do
   Clock with majority rule and produce two 114-bits outputs.

Εικόνα 8 : Οι διεργασίες της εσωτερικής λειτουργίας του A5/1

2.2.2 Υλοποίηση του A5/2

Ο αλγόριθμος A5/2 είναι η πιο «αδύναμη» έκδοση του A5/1. Δημιουργήθηκε για να εξαχθεί σε χώρες εκτός Ευρώπης καθώς ο A5/1 υπόκειτο σε περιορισμούς εξαγωγής. Η βασική λειτουργία του είναι ίδια με του A5/1, δηλαδή είναι ένας αλγόριθμος ροής που παίρνει ως είσοδο το Kc(64 bits) και το COUNT(22 bits) και μετασχηματίζει μέσω XOR το αρχικό κείμενο σε ένα κρυπτογραφημένο και αντιστρόφως.

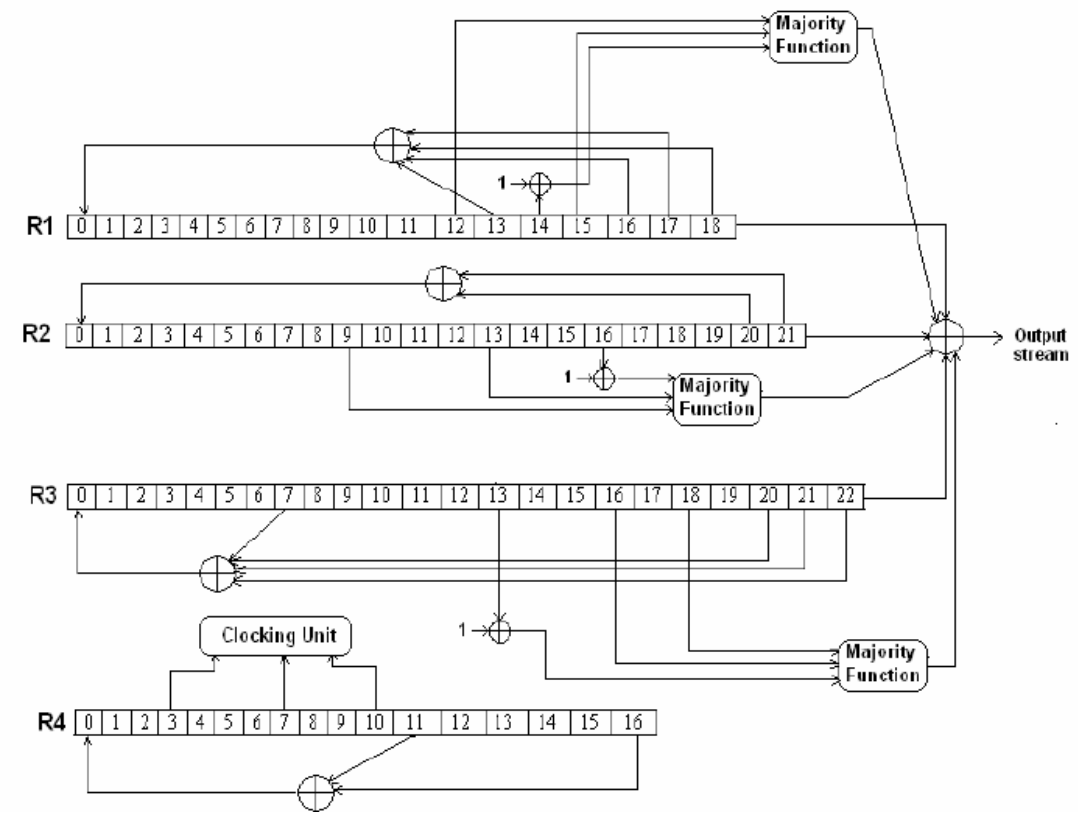
Η γεννήτρια ροής κλειδιού χρησιμοποιεί 4 μεγίστου μήκους LFSRs, με σύμβολα R1,R2,R3,R4 και μήκους 19,22,23 και 17 bits αντιστοίχως. Τα πρωταρχικά πολυώνυμα των τριών πρώτων είναι ίδια με τα R1,R2,R3 του A5/1 ενώ ο R4 έχει πολυώνυμο $\mathbf{x^{17} + x^{5} + 1}$, δηλαδή τα bits που συμμετέχουν στην ανάδραση είναι τα 16,11.

Register Number	Length In bits	Primitive Polynomial	Bits that are XORed
1	19	$\mathbf{x^{19} + x^{5} + x^{2} + x + 1}$	18,17,16,13
2	22	$\mathbf{x^{22} + x + 1}$	21,20
3	23	$\mathbf{x^{23} + x^{15} + x^{2} + x + 1}$	22,21,20,7
4	17	$\mathbf{x^{17} + x^{5} + 1}$	16,11

Πίνακας 3 : Οι παράμετροι του A5/2

Ο χρονισμός των R1,R2,R3 ρυθμίζεται από τον R4 ενώ ο R4 χρονίζεται κανονικά σε κάθε κύκλο. Ο μηχανισμός του ρολογιού είναι ως εξής: τα bits 3,7,10 του R4 αντιστοιχούν στα bits ρολογιού καθενός καταχωρητή, δηλαδή C1=R4(10),C2= R4(3), C3=R4(7). Κατόπιν υπολογίζεται το bit πλειοψηφίας από τα C1,C2,C3 και ανάλογα με το αν το κάθε bit ρολογιού είναι ίδιο με το bit πλειοψηφίας ο αντίστοιχος καταχωρητής χρονίζεται ή όχι. Μετά από το χρονισμό των R1,R2 και R3 (ή 2 εξ αυτών) χρονίζεται και ο R4.

Η έξοδος δημιουργείται ως εξής: σε κάθε καταχωρητή υπολογίζεται η πλειοψηφία από 2 bits και του συμπληρώματος ενός τρίτου bit και κατόπιν τα αποτελέσματα από όλες τις πλειοψηφίες και τα msb bits καθενός γίνονται XOR για να παραχθεί το bit εξόδου.



Εικόνα 9 : Η δομή των καταχωρητών, ο μηχανισμός του ρολογιού και η έξοδος του A5/2 από [16]

Οι διαδικασίες που διενεργούνται στην εσωτερική κατάσταση του A5/2 είναι οι εξής[20]:

Βήμα 1ο: Όπως και στον A5/1 οι καταχωρητές μηδενίζονται και το 64-bit Kc και ο 22-bit COUNT «φορτώνονται» στους R1,R2,R3,R4. Επιπλέον τα bits R1[15], R2[16], R3[18], and R4[10] τίθενται 1.

Βήμα 2ο: Εν συνεχεία οι καταχωρητές χρονίζονται για 99 κύκλους χρησιμοποιώντας το μηχανισμό του ρολογιού χωρίς να παράγουν έξοδο.

Βήμα 3^{ο}: Οι καταχωρητές χρονίζονται για 228 κύκλους σύμφωνα με το μηχανισμό του ρολογιού παράγοντας 228 bits εξόδου.

Τα παραπάνω βήματα φαίνονται σχηματικά στην εικόνα 10 από [20].

1. Set R1 = R2 = R3 = R4 = 0.
   For i = 0 to 63 do
   a) Clock all four registers.
   b) R1[0] ←R1[0] ⊕ Kc[i];
   c) R2[0] ←R2[0] ⊕ Kc[i];
   d) R3[0] ←R3[0] ⊕ Kc[i];
   e) R4[0] ←R4[0] ⊕ Kc[i];
   For i = 0 to 21 do
   a) Clock all four registers.
   b) R1[0] ←R1[0] ⊕ COUNT[i];
   c) R2[0] ←R2[0] ⊕ COUNT[i];
   d) R3[0] ←R3[0] ⊕ COUNT[i];
   e) R4[0] ←R4[0] ⊕ COUNT[i];
   Set the bits R1[15] ←1, R2[16] ←1, R3[18] ←1, R4[10] ←1.
2. For i = 0 to 98 do
   Clock with majority rule and discard the output.

3. For i = 0 to 227 do
   Clock with majority rule and produce two 114-bits outputs.

Εικόνα 10 : Οι διεργασίες της εσωτερικής λειτουργίας του A5/2[20]

2.2.3 Υλοποίηση του A5/3(και GEA3)

Ο A5/3 προστέθηκε το 2002 στο σύνολο των κρυπτογραφικών αλγορίθμων του GSM και οι προδιαγραφές του δημοσιοποιήθηκαν το 2003, σε αντίθεση με τους A5/1 και A5/2 που δε δημοσιοποιήθηκαν επίσημα. Είναι ήδη υλοποιημένος σε περίπου 40% των τηλεφωνικών συσκευών αλλά πολύ λίγοι τηλεπικοινωνιακοί πάροχοι έχουν αρχίσει να τον χρησιμοποιούν[19].

Ο A5/3 αποτελεί μια παραλλαγή του κρυπταλγόριθμου τμήματος MISTY [21], ονομαζόμενη αλγόριθμος KASUMI που ορίστηκε από την 3GPP. Οι διαφορές που υπάρχουν ανάμεσα στον MISTY και τον KASUMI αφορούν την ταχύτητα και την ευκολότερη υλοποίηση σε επίπεδο υλικού. Ο KASUMI ως κρυπτογραφικός αλγόριθμος χρησιμοποιείται στον A5/3 στο GSM και ECSD (Enhanced Circuit Switched Data), και στον GEA3 στο GPRS[17].

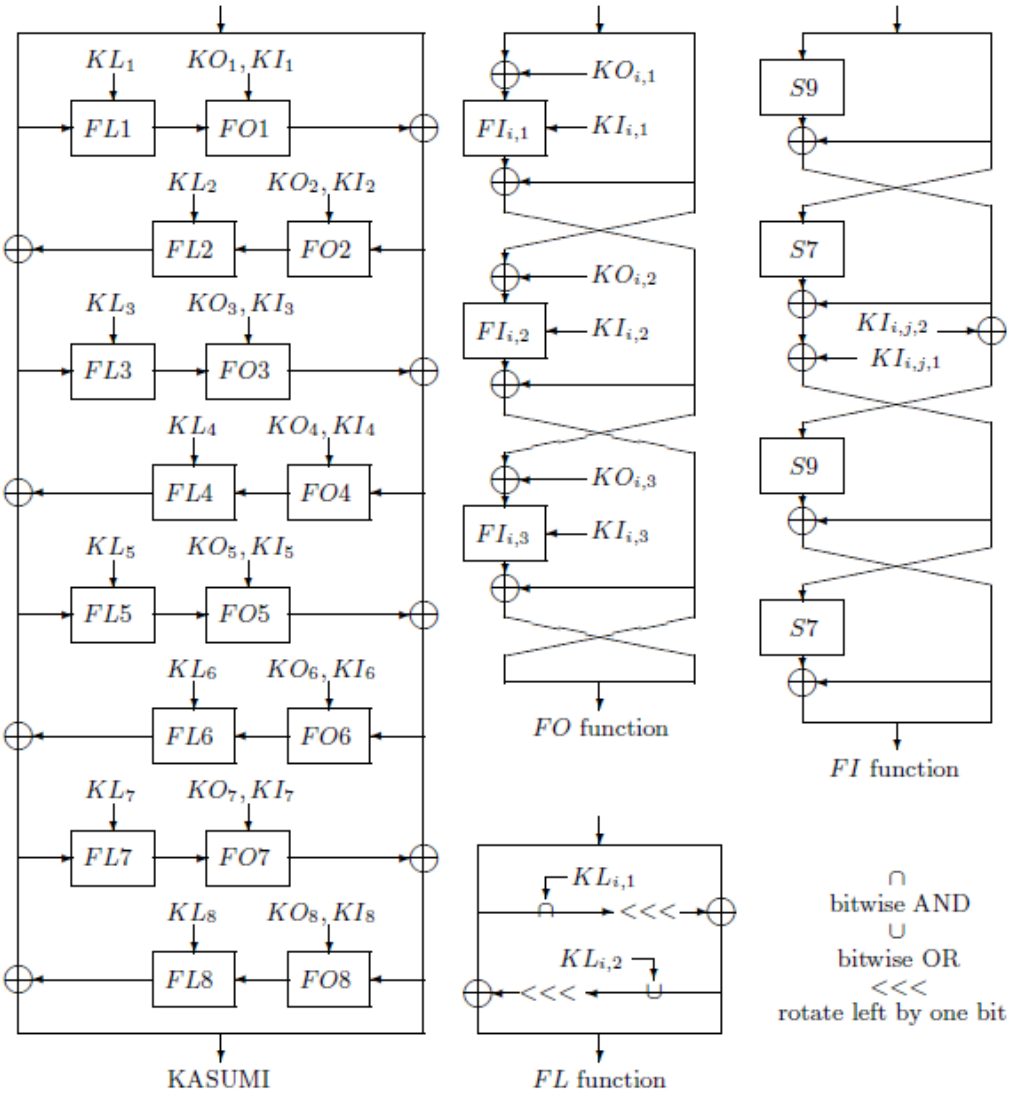
Η βασική του υλοποίηση αποτελείται από μια αναδρομική δομή Feistel με 8 γύρους καθένας από τους οποίους συγκροτείται από 2 συναρτήσεις: τη συνάρτηση FO, η οποία είναι αυτού καθεαυτού μια τριών γύρων Feistel δομή, και από τη συνάρτηση FL η οποία «αναμιγνύει» με γραμμικό τρόπο ένα 32-bit υποκλειδί με τα δεδομένα. Η σειρά των 2 συναρτήσεων εξαρτάται από τον αριθμό του γύρου με την παρακάτω σχέση: στους άρτιους γύρους εφαρμόζεται πρώτα η FO και στους περιττούς πρώτα η FL.

Οι παράμετροι του αλγορίθμου είναι το Kc (64 έως 128 bits), ο αριθμός πλαισίου COUNT (22 bits και 32 bits για το GPRS) και τέλος οι έξοδοι BLOCK1 και BLOCK2 (114 bits, ενώ για το EDGE είναι 348 bits και για το GPRS 64 bits).

Ο αλγόριθμος χρησιμοποιεί μια κεντρική συνάρτηση παραγωγής ροής κλειδιού την KGCORE η οποία βασίζεται στον αλγόριθμο τμήματος KASUMI. Η κρυπτογραφική λειτουργία ορίζεται από την αντιστοίχηση των παραμέτρων εισόδου του A5/3 στις εισόδους της KGCORE και των bits εξόδου της KGCORE στις εξόδους του A5/3. Εσωτερικά το κλειδί Kc (CK ως παράμετρος της KGCORE) «επεκτείνεται» στα 128 bits.

Αναλυτικά οι τεχνικές προδιαγραφές του KASUMI παρουσιάζονται στο [18].

Στην εικόνα 19 φαίνεται το σχεδιάγραμμα του KASUMI.



Εικόνα 11 : Ο αλγόριθμος KASUMI[19]

Κεφάλαιο 3: Κρυπτανάλυση και επιθέσεις

3.1 Εισαγωγή στην κρυπτογραφία

3.1.1 Ορισμοί

Η κρυπτολογία (cryptology) είναι η επιστήμη της ασφάλειας των πληροφοριών και της προστασίας των προσωπικών δεδομένων και χρησιμοποιεί μαθηματικές τεχνικές για να παρέχει πιστοποίηση (authenticity), εμπιστευτικότητα, ακεραιότητα και άλλες υπηρεσίες ασφαλείας στην πληροφορία που αναμεταδίδεται, αποθηκεύεται ή βρίσκεται υπό επεξεργασία σε ένα σύστημα πληροφοριών.

Το σχεδιαστικό μέρος της επιστήμης της κρυπτολογίας ονομάζεται κρυπτογραφία(cryptography) ενώ η διερεύνηση της ασφάλειας και η ανάλυση ονομάζεται κρυπτανάλυση (cryptanalysis). Δηλαδή οι κρυπτογράφοι δημιουργούν κρυπτογραφικές σχεδιάσεις τις οποίες οι κρυπταναλυτές προσπαθούν να «σπάσουν».

Ένα κρυπτογραφικό σύστημα είναι ένα σύνολο κρυπτογραφικών συναρτήσεων παραμετροποιημένο με το κρυπτογραφικό κλειδί.

Η λειτουργία ενός κρυπτογραφικού συστήματος είναι να κρυπτογραφήσει ένα μήνυμα (plaintext) με βάση ένα κλειδί με αποτέλεσμα ένα κρυπτογραφημένο μήνυμα (ciphertext). Η ισχύς ενός τέτοιου συστήματος έγκειται στο μέγεθος του κλειδιού και στην τυχαιότητά του.

Ένα κρυπτογραφικό σύστημα θεωρείται «σπασμένο» εάν υπάρχει μια αποτελεσματική μέθοδος μέσω της οποίας το κλειδί είναι δυνατόν να εξάγεται συστηματικά, με μεγάλη πιθανότητα, από πρακτικά διαθέσιμες πληροφορίες.

3.1.2 Μοντέρνα προσέγγιση στην κρυπτογραφία

Το πρώτο ορόσημο στη μοντέρνα κρυπτογραφία ήταν το 1949 με τη δημοσίευση του Claude Shannon *Communication Theory of Secrecy Systems* [22] η οποία εδραίωσε τη θεωρητική βάση της κρυπτογραφίας με την περιγραφή των απόρρητων συστημάτων(secrecy systems). Το δεύτερο σημαντικό ορόσημο ήταν η δημοσίευση των Diffie και Hellman *New Directions in Cryptography* [23] το 1976 και ακολούθησε η δημοσίευση του DES το 1977 ο οποίος από τότε αποτέλεσε μια ανεξάντλητη πηγή υλικού κρυπτολογικής έρευνας.

3.1.3 Παράμετροι καθορισμού της ασφάλειας

Σύμφωνα με τον Shannon[22] ένας αλγόριθμος είναι τέλειος αν διασφαλίζει την «απόλυτη μυστικότητα» (perfect secrecy), δηλαδή αν είναι ασφαλής ακόμα και αν ο εισβολέας διαθέτει απεριόριστους υπολογιστικούς πόρους. Πρακτικά αυτό είναι αδύνατο οπότε οι όροι που έχουν αναπτυχθεί για την περιγραφή της ασφάλειας ενός κρυπταλγόριθμου είναι οι παρακάτω[24]:

- Υπολογιστική ασφάλεια (computational security)

Ένας αλγόριθμος θεωρείται υπολογιστικά ασφαλής εάν ο βέλτιστος αλγόριθμος για το «σπάσιμό» του απαιτεί έναν επαρκές μεγάλο αριθμό λειτουργιών-εργασιών(για αλγόριθμους σταθερού μεγέθους) ή ταχύτατα αυξανόμενους υπολογιστικούς πόρους(για αλγόριθμους μεταβλητού μεγέθους).

- Πρακτική ασφάλεια (practical security)

Ένας αλγόριθμος θεωρείται πρακτικά ασφαλής εάν παρέχει υπολογιστική ασφάλεια για όλες τις ήδη γνωστές επιθέσεις εναντίον του (όλους τους ήδη γνωστούς αλγορίθμους για το «σπάσιμό» του).

- Αποδείξιμη-ευαπόδεικτη ασφάλεια (provable security)

Ένας αλγόριθμος θεωρείται ασφαλής εάν υπάρχουν αποδείξεις για την ασφάλεια του σύμφωνα υπό ορισμένες παραδοχές, πχ για μια υποκατηγορία επιθέσεων.

3.1.4 Ταξινόμηση των επιθέσεων

Ο πρώτος παράγοντας που καθορίζει την απόδοση ενός κρυπταλγορίθμου είναι το μέγεθος του κλειδιού. Ένας αλγόριθμος είναι θεωρητικά «σπασμένος» αν η χρονική πολυπλοκότητα (time complexity) μιας επίθεσης είναι μικρότερη από τη χρονική πολυπλοκότητα μιας εξαντλητικής αναζήτησης όλων των δυνατών κλειδιών, δηλαδή αν είναι μικρότερη από 2^k, όπου k το μήκος του κλειδιού σε bits. Σε πρακτική βάση, ταξινομούμε τις επιθέσεις σύμφωνα με τα δεδομένα που βρίσκονται στην κατοχή ενός εισβολέα, και σύμφωνα με το χρόνο και τη μνήμη που απαιτούνται για την επίθεση[24].

Διαθέσιμα δεδομένα

- Known plaintext attack

Ο εισβολέας έχει στην κατοχή του ζευγάρια από αρχικό και κρυπτογραφημένο κείμενο. Για τον A5/1 (και τους περισσότερους αλγόριθμους ροής) αυτό σημαίνει ότι έχει πρόσβαση στο keystream αφού η λειτουργία κρυπτογράφησης είναι μια πράξη bit προς bit αποκλειστικού-ή, δηλαδή εάν P το αρχικό κείμενο, C το κρυπτογραφημένο και K το keystream, τότε C=PxorK και ισοδύναμα K=PxorC.

- Ciphertext-only attack

Ο εισβολέας έχει στην κατοχή του μόνο το κρυπτογραφημένο κείμενο. Παρόλο που δεν έχει πρόσβαση σε αρχικό κείμενο έχει κάποιες πληροφορίες για αυτό. Για τον

A5/1 οι πληροφορίες αυτές αφορούν το ψηφιοποιημένο σήμα φωνής και την επέκτασή του με τη χρήση κώδικα διόρθωσης λαθών.

- Chosen plaintext attack

Ο εισβολέας μπορεί να διαλέξει κάποιο αρχικό κείμενο και να αποκτήσει το αντίστοιχο κρυπτογραφημένο κείμενο.

Πολυπλοκότητα Χρόνου
Ο μεγαλύτερος χρόνος που απαιτείται για το «σπάσιμο» ενός αλγορίθμου είναι 2^k, όπου k είναι το μήκος του κλειδιού σε bits. Για τον A5/1 αυτός ο χρόνος είναι 2^{64}.

Ο χρόνος που χρειάζεται για να ολοκληρωθεί μια επίθεση χωρίζεται σε χρόνο προϋπολογισμού (precomputation time) και χρόνο επίθεσης (attack time). Ο χρόνος προϋπολογισμού γίνεται μια φορά και για τις επιθέσεις στις οποίες αυτός χρησιμοποιείται, ο χρόνος επίθεσης συνήθως είναι πολύ μικρός (περισσότερα στο κεφάλαιο 4).

Πολυπλοκότητα Μνήμης
Έχει άμεση σχέση με το χρόνο προϋπολογισμού, καθώς ο τελευταίος συνεπάγεται μεγαλύτερες απαιτήσεις μνήμης.

3.2 Επιθέσεις (Attacks)

Η κρυπτανάλυση είναι ένα αναπόσπαστο κομμάτι της δύναμης ενός κρυπτογραφικού αλγορίθμου, καθώς όσο το δυνατόν περισσότερες επιθέσεις αναλυθούν για έναν αλγόριθμο τόσο περισσότερο αποδεικνύεται η ασφάλειά του και καθίσταται εφικτή η προσαρμογή του ώστε να διασφαλίζεται η «ανθεκτικότητά» του.

Η κρυπτανάλυση των αλγορίθμων ροής βασίζεται στην υπόθεση ότι έχουμε στην κατοχή μας επαρκές γνωστό κείμενο (known plaintext).

Οι κυριότερες τάξεις κρυπταναλυτικών επιθέσεων στον A5/1 είναι οι επιθέσεις guess-and-determine και οι επιθέσεις time-memory-data tradeoff. Στις guess-and-determine επιθέσεις, όπως υποδηλώνει και το όνομα, ο κρυπταναλυτής μαντεύει όσο το δυνατόν λιγότερες μεταβλητές εσωτερικής κατάστασης, και υπολογίζει τις εναπομείναντες μεταβλητές κατάστασης από την πραγματική έξοδο (known plaintext attack ή known keystream). Η μεγάλη υπολογιστική πολυπλοκότητά τους αντισταθμίζεται από το γεγονός ότι συνήθως έχουν χαμηλές απαιτήσεις σε known plaintext. Οι time-memory tradeoff επιθέσεις (βλ Κεφάλαιο 4) αποτελούνται από δύο στάδια: το στάδιο προϋπολογισμού και το στάδιο πραγματικού χρόνου επίθεσης. Στο πρώτο στάδιο υπολογίζονται οι ακολουθίες εξόδου για κάποιες εσωτερικές καταστάσεις, δηλαδή από κάποιο μικρό ή μεγάλο υποσύνολο των πιθανών εσωτερικών καταστάσεων, και αποθηκεύονται, συνήθως σε ένα πίνακα. Στο δεύτερο στάδιο ο πίνακας εξερευνείται ώστε να βρεθεί η αντιστοίχηση με την γνωστή έξοδο και με αυτόν τον τρόπο

ανακτάται η εσωτερική κατάσταση. Όπως υποδηλώνει το όνομά τους οι time-memory-data tradeoff attacks στηρίζονται στην ανταλλαγή χρόνου επίθεσης - αποθηκευτικής μνήμης(π.χ. αριθμός ζευγαριών) - δεδομένων(απαιτήσεις σε known plaintext). Για παράδειγμα όσο περισσότερη μνήμη διαθέτει το στάδιο προϋπολογισμού (αριθμός ζευγαριών) τόσο μικρότερος είναι ο χρόνος της πραγματικής επίθεσης.

3.2.1 Επιθέσεις στον αλγόριθμο A5/1

Η σχεδίαση του A5/1 κρατήθηκε μυστική ώσπου κάποιες πληροφορίες για τον αλγόριθμο άρχισαν να δημοσιεύονται το 1994 [25],[26]. Το πρώτο βήμα της κρυπτανάλυσής του έγινε το 1997 από τον Golic[27] ως ένα πρώτο σχεδιάγραμμα (περιγραφή) του αλγορίθμου. Το 1999 ο αλγόριθμος (και ο A5/2) έγινε reverse engineered από τους Briceno, Goldberg, Wagner[28] χρησιμοποιώντας ένα κανονικό GSM τηλέφωνο. Οι πρώτες επιθέσεις στον A5/1 δεδομένου του [28] ήταν από τους Eli Biham,Orr Dunkelman το 2000 [15], από τους Biryukov,Shamir,Wagner το 2000 [29] (με preliminary draft από Biryukov,Shamir το 1999**)** και από τους Keller,Seitz το 2001[30]. Έκτοτε έχουν δημοσιευτεί πολλές επιθέσεις διαφόρων τεχνικών στον A5/1 όπως [31],[32],[33],[20],[38]. Η πιο πρόσφατη επίθεση πάνω στην οποία έχει στηριχτεί αυτή η διπλωματική εργασία είναι του Karsten Nohl και μιας ομάδας κρυπτογράφων οι οποίοι δημιούργησαν έναν πίνακα ουράνιου τόξου 2 Terabyte για την εύρεση του κλειδιού μιας συνομιλίας [36] (βλ. Κεφάλαιο 5).

Παρακάτω θα αναφερθούν με περισσότερες πληροφορίες κάποιες από τις σημαντικότερες επιθέσεις.

Η πρώτη επίθεση στον A5/1 που πρέπει να αναφερθεί είναι η επίθεση ωμής βίας-brute force attack που περιλαμβάνει την εξαντλητική δοκιμή όλων των δυνατών κλειδιών (πολυπλοκότητα 2^{64}). Η πολυπλοκότητα αυτής της επίθεσης είναι 2^{54} (τα τελευταία 10 bits του κλειδιού είναι 0) γεγονός που καθιστά ανέφικτη την αποκρυπτογράφηση σε real time.

i) Golic [27]

Η επίθεση του Golic κατηγοριοποιείται στις guess-and-determine επιθέσεις, χρησιμοποιεί known plaintext (σύμφωνα με τον Golic χρειάζονται μόνο περίπου 64 bits διαδοχικού keystream) και εφαρμόστηκε σε ένα εικαζόμενο σχεδιάγραμμα του A5/1 (alleged). Η βασική ιδέα είναι ο εισβολέας να μαντέψει τα «χαμηλότερα» bits του κάθε καταχωρητή R1,R2,R3, και συνεπώς τα bits ρολογιού, και στη συνέχεια να χρονίσει τους καταχωρητές επαληθεύοντας σε κάθε χρονισμό ότι ισχύουν οι περιορισμοί που εισάγονται από το known keystream. Σε κάθε βήμα από κάθε bit του keystream λαμβάνεται μια γραμμική εξίσωση με μεταβλητές κάθε bit της εσωτερικής κατάστασης έως ότου να έχουν ληφθεί τουλάχιστον 64 γραμμικά ανεξάρτητες

εξισώσεις, αρκετές για την επίλυση του συστήματος. Σύμφωνα με τον Golic η πολυπλοκότητα της επίθεσης είναι $2^{40.16}$ αλλά πρέπει να ληφθεί υπόψιν και η πολυπλοκότητα υπολογισμού της λύσης των γραμμικών εξισώσεων η οποία δεν είναι αμελητέα.

Ο Golic εισήγαγε επίσης και άλλα 2 σημαντικά σημεία στη δημοσίευσή του:

1.Ανακατασκευή της εσωτερικής κατάστασης του αλγορίθμου στο σημείο μετά τη «φόρτωση» του κλειδιού S(0), δεδομένου του S(101) του σημείου που ο κρυπταλγόριθμος παράγει το πρώτο bit εξόδου. Αυτό επιτυγχάνεται χρησιμοποιώντας γραμμική αναδρομή προς τα πίσω (backward linear recursion) και επαλήθευση κάθε πιθανής περίπτωσης με χρονισμό προς τα μπροστά των καταχωρητών. Με αυτόν τον τρόπο διαπιστώνεται αν παράγεται το S(101) από όλες τις πιθανές λύσεις S(0).

2.Μια επίθεση με ανταλλαγή χρόνου-μνήμης, βασισμένη στο παράδοξο των γενεθλίων (birthday paradox) που αποφέρει την S(0) εάν $T \cdot M \geq 2^{63.32}$ όπου T είναι ο απαιτούμενος χρόνος υπολογισμού και M η απαιτούμενη μνήμη (σε 128 bits words). Αυτή η επίθεση είναι αναποτελεσματική καθώς χρειάζεται περισσότερο από 3 ώρες τηλεφωνικής συνδιάλεξης και οι απαιτήσεις σε χρόνο και μνήμη είναι πολύ μεγάλες. Η τεχνική της ανταλλαγής χρόνου-μνήμης στους αλγόριθμους ροής δημοσιεύθηκε ανεξάρτητα από τον Babbage[37].

ii) Biryukov,Shamir,Wagner [29]

Σε αυτή τη δημοσίευση οι συγγραφείς δηλώνουν ότι ο οργανισμός GSM επιβεβαίωσε την ορθότητα του αλγορίθμου A5/1 (A pedagogical implementation) που έγινε reverse engineered από τους Briceno, Goldberg, Wagner[28].

Οι Biryukov,Shamir,Wagner εκτελούν 2 επιθέσεις στον A5/1, οι οποίες ανήκουν στην κατηγορία των time-memory-data tradeoff επιθέσεων και βασίζονται στην ιδέα του Golic. Και οι δυο αποτελούνται από ένα στάδιο προϋπολογισμού με πολυπλοκότητα 2^{48} όπου αποθηκεύονται ζευγάρια εσωτερικών καταστάσεων και συστοιχίες από bits εξόδου (τα πρώτα bits), δηλαδή σε ζεύγη (prefix,state). Στην εκτέλεση της επίθεσης μόλις αναγνωριστεί η συγκεκριμένη συστοιχία είναι δυνατή η ανάκτηση των πιθανών εσωτερικών καταστάσεων χρησιμοποιώντας τον πίνακα προϋπολογισμού και τα υπόλοιπα bits εξόδου.

Η πρώτη επίθεση, η οποία ονομάζεται the biased birthday attack, έχει απαιτήσεις σε known plaintext 2 λεπτά μιας συνομιλίας και χρόνο εκτέλεσης (χωρίς χρόνο προϋπολογισμού) περίπου ένα δευτερόλεπτο. Η δεύτερη επίθεση, ονομαζόμενη the random subgraph attack, απαιτεί 2 περίπου δευτερόλεπτα συνομιλίας και έχει χρόνο εκτέλεσης (χωρίς χρόνο προϋπολογισμού) μερικά λεπτά. Κάποιες πιθανές tradeoff παράμετροι με τα αποτελέσματά τους σε χρόνο παρουσιάζονται από τους συγγραφείς στον παρακάτω πίνακα.

Attack Type	Preprocessing steps	Available data	Number of 73GB disks	Attack time
Biased Birthday attack (1)	2^{42}	2 minutes	4	1 second
Biased Birthday attack (2)	2^{48}	2 minutes	2	1 second
Random Subgraph attack	2^{48}	2 seconds	4	minutes

Πίνακας 4 : Παράμετροι ανταλλαγής [29]

iii) Biham, Dunkelman [15]

Η επίθεση των Biham,Dunkelman κατηγοριοποιείται στις guess-and-determine επιθέσεις, χρησιμοποιεί known plaintext ($2^{20.8}$ bits δεδομένων ισοδύναμα με 2.36 λεπτά συνομιλίας) και σκοπός είναι να ληφθεί η S(101), η εσωτερική κατάσταση του κρυπταλγόριθμου στο σημείο που παράγεται το πρώτο bit εξόδου. Η βασική ιδέα είναι ο εισβολέας να μαντέψει κάποια bits των καταχωρητών τη χρονική στιγμή που συμβαίνει ένα γεγονός που μπορεί να αποφέρει σημαντικές πληροφορίες για την εσωτερική κατάσταση των καταχωρητών, π.χ. όταν ένας εξ αυτών δε χρονίζεται για μεγάλο χρονικό διάστημα. Η επίθεση αυτή απαιτεί $2^{39.91}$ A5/1 clockings. Επίσης απαιτεί 64 GB αποθηκευμένα δεδομένα μετά από ένα στάδιο προεπεξεργασίας με χρόνο υπολογισμού 2^{37} A5/1 clockings.

Attack	Precomputation Workload	Complexity of Analysis	Time Unit	Data Complexity (bits)	Memory Requirements
[7] - Basic Attack	0	$2^{40.16}$	Linear eq. set solving	64	0
[7] - TM Tradeoff	$2^{35.65}$	$2^{27.67}$	Linear eq. set solving	$2^{28.8}$	862GB
[3] - Baised Birthday Attack	2^{48}	1 second	A5/1 Clocking	$2^{20.5}$	146 GB
[3] - Baised Birthday Attack	2^{42}	1 second	A5/1 Clocking	$2^{20.5}$	292 GB
[3] - Random Subgraph Attack	2^{48}	minutes	A5/1 Clocking	$2^{14.7}$	146 GB
Our Results	2^{38}	$2^{39.91}$	A5/1 Clocking	$2^{20.8}$	64 GB
Our Results	$2^{33.6}$	$2^{40.97}$	A5/1 Clocking	$2^{20.8}$	4 GB

Εικόνα 12 : Πολυπλοκότητα επιθέσεων στον A5/1[15]

iv) Keller, Seitz [30]

Η επίθεση των Keller,Seitz κατηγοριοποιείται στις guess-and-determine επιθέσεις, χρησιμοποιεί known plaintext (λίγα frames-συνήθως μόνο ένα) και στηρίζεται στην πρόταση του Anderson [26]. Η βασική ιδέα είναι ο εισβολέας να μαντέψει τους καταχωρητές R1,R2 και σύμφωνα με αυτούς να προσδιορίσει τον R3 μειώνοντας τον αριθμό των υποψηφίων καταστάσεων με την αναγνώριση τυχόν αντιφάσεων που προκαλούνται από την τιμή του bit ρολογιού της R3 (σύμφωνα με το αν τελικά ο R3 χρονίζεται ή όχι). Η επίθεση χωρίζεται σε 2 φάσεις: τη φάση προσδιορισμού των

καταχωρητών και τη φάση επαλήθευσης όπου η υποψήφια κατάσταση ελέγχεται ως προς την ορθότητά της. Ένα σημαντικό μέρος της επίθεσης υλοποιείται σε FPGA (Xilinx XC4062), ενώ επίσης εκτιμήθηκε ότι σε 1000 Alcatel ASICs ο χρόνος αποκρυπτογράφησης θα είναι λιγότερος από ένα λεπτό.

v) Barkan,Biham[33]

Η επίθεση χρησιμοποιεί known plaintext (περίπου 4.9–9.2 δευτερόλεπτα) και ολοκληρώνεται σε χρόνο που κυμαίνεται ανάμεσα σε δέκατα του δευτερολέπτου και σε λίγα λεπτά σε ένα PC, με ποσοστό επιτυχίας περίπου 91%.

vi)Gendrullis, Novotny, Rupp[38]

Η επίθεση δημοσιοποιήθηκε το 2008, κατηγοριοποιείται στις guess-and-determine επιθέσεις, χρησιμοποιεί known plaintext (64 διαδοχικά bits) και έχει υλοποιηθεί πλήρως σε hardware και συγκεκριμένα σε μια ειδική μηχανή που ονομάζεται COPACOBANA.

Η μηχανή COPACOBANA (Cost-Optimized Parallel COde Breaker)[39][40] είναι ένα «συγκρότημα» από FPGAs το οποίο είναι βελτιστοποιημένο για να τρέχει κρυπταναλυτικούς αλγόριθμους. Τα χαρακτηριστικά των κρυπταναλυτικών αλγορίθμων που αξιοποιούνται στην COPACOBANA είναι η μεγάλη απαίτηση για παράλληλους υπολογισμούς και η αντίστοιχη μικρή απαίτηση σε μνήμη καθώς και η χαμηλή ανάγκη για είσοδο νέων δεδομένων ή για ανταλλαγή δεδομένων ανάμεσα στα στιγμιότυπα. Η νεότερη έκδοση αποτελείται από 128 Virtex4 SX35 FPGAs ενώ η παλιότερη έκδοση πάνω στην οποία έχει στηριχτεί η δουλειά των Gendrullis, Novotny, Rupp αποτελούταν από 120 Spartan3-XC3S1000 FPGAs.

Η βασική ιδέα είναι των Keller,Seitz[30] με κάποιες διαφοροποιήσεις όσον αφορά την άμεση απόρριψη των υποψηφίων R3[10] που θα οδηγούσαν σε αντιφάσεις ως προς τη συστοιχία εξόδου, και την εξέταση όλων των πιθανοτήτων για τα υποψήφια R3[10] που δεν οδηγούν σε αντιφάσεις. Το αποτέλεσμα συγκριτικά με τους Keller,Seitz είναι να βρεθεί σε κάθε περίπτωση η σωστή εσωτερική κατάσταση αλλά ταυτόχρονα να αυξηθεί ο αριθμός των κύκλων ρολογιού για τον προσδιορισμό μιας υποψήφιας κατάστασης από 14 σε 17 $\frac{2}{3}$ κύκλους. Η πολυπλοκότητα της επίθεσης είναι $2^{54.02}$ και η εσωτερική κατάσταση ανακτάται σε 7 ώρες κατά μέσο όρο (έως 14 ώρες στη χειρότερη περίπτωση).

vii) Barkan, Biham, Keller[20]

Η επίθεση κατηγοριοποιείται στις time-memory tradeoff επιθέσεις, χρησιμοποιεί μόνο ciphertext (ciphertext only attack) και χρησιμοποιεί το time-memory-data tradeoff από τους Biryukov,Shamir[34]. Η βασική ιδέα είναι ο εισβολέας να εκμεταλλευτεί το γεγονός ότι οι κώδικες διόρθωσης σφαλμάτων (error correction codes) εφαρμόζονται πριν την κρυπτογράφηση και συνεπώς αποκτάται known

plaintext χωρίς να χρειάζεται να το «μαντέψει». Σύμφωνα με τους συγγραφείς μπορούν να καθοριστούν διάφορες παράμετροι της ανταλλαγής όπως 10 λεπτά ciphertext θα χρειαστούν 930 PC να προϋπολογίσουν 50 TB σε ένα χρόνο, με τελική πραγματική επίθεση σε ένα PC διάρκειας 1.53 λεπτών. Επίσης 64 δευτερόλεπτα ciphertext θα χρειαστούν 2800 PC να προϋπολογίσουν 50 TB σε ένα χρόνο, με τελική πραγματική επίθεση σε ένα PC διάρκειας 13.33 λεπτών.

3.2.2 Επιθέσεις στον αλγόριθμο A5/2

Ο αλγόριθμος A5/2 έγινε reverse engineered μαζί με τον A5/1 από τους Briceno, Goldberg,Wagner[28]. Στη συνεδρία της Crypto99 οι Goldberg,Wagner ανακοίνωσαν μια επίθεση στον A5/2 που ολοκληρώνεται σε μόνο $O(2^{16})$ βήματα[29].
Οι [20],[35] περιλαμβάνουν μερικές από τις μετέπειτα επιθέσεις στον A5/2. Στην [20] ένα παράδειγμα είναι ότι απαιτούνται περίπου 8.5 ώρες προεπεξεργασίας, 780 MB μνήμης και η επίθεση ολοκληρώνεται σε λιγότερο από ένα δευτερόλεπτο.

Λόγω του γεγονότος ότι ο A5/2 έχει αποδειχθεί επισφαλής η ETSI και η 3GPP έχουν αρχίσει την σταδιακή κατάργησή του.

3.2.3 Επιθέσεις στον αλγόριθμο A5/3

Η σχεδίαση του A5/3 δημοσιεύθηκε το 2003 και συνεπώς τέθηκε στη διαθεσιμότητα των κρυπταναλυτών. Παρόλο που είναι πολύ ισχυρότερος από τον A5/1 αρκετές επιθέσεις έχουν δημοσιευτεί εναντίον του KASUMI όπως οι [41],[42],[43]. Η σημαντικότερη είναι των Dunkelman, Keller, Shamir [44] οι οποίοι χρησιμοποιούν ένα τύπο επίθεσης στον KASUMI, την επίθεση sandwich, και βρίσκουν το κλειδί με 2^{26} δεδομένα, 2^{30} bytes μνήμης και χρόνο 2^{32}. Η επίθεση χρειάζεται λιγότερο από 2 ώρες για να ολοκληρωθεί σε ένα PC αλλά επειδή χρησιμοποιεί συσχετισμένα κλειδιά και επιλεγμένα μηνύματα είναι πιθανό να μην είναι πλήρως εφαρμόσιμη στον A5/3. Οι συγγραφείς παρατηρούν στη δημοσίευσή τους ότι η παραλλαγή του KASUMI είναι πολύ πιο εύκολο να «σπάσει» από ότι ο αρχικός MISTY αλγόριθμος ο οποίος χρειάζεται εξαντλητική αναζήτηση κλειδιού με πολυπλοκότητα 2^{128}. Αυτό το γεγονός αποδεικνύει ότι οι αλλαγές που έγιναν στον MISTY οδήγησαν σε πιο αδύναμο κρυπτοσύστημα [44].

Κεφάλαιο 4: TMTO – Rainbow tables

4.1 Ανταλλαγή χρόνου- μνήμης (TMTO)

Η επίθεση με ανταλλαγή χρόνου-μνήμης (time-memory tradeoff attack) προτάθηκε το 1980 από τον Hellman[45] και έκτοτε βελτιώθηκε σε πολλά σημεία με κυριότερες βελτιώσεις του Rivest[46] και Oechslin[47].

Η επίθεση στοχεύει στην ανάκτηση της εσωτερικής κατάστασης ενός αλγορίθμου δεδομένου κάποιου γνωστού plaintext (known plaintext). Ένα προσχέδιο της επίθεσης είναι η υλοποίηση μιας εξαντλητικής αναζήτησης για κάθε κλειδί και το ciphertext που του αντιστοιχεί και η αποθήκευσή των ζευγαριών αυτών σε ένα πίνακα. Ωστόσο αυτός ο τρόπος δεν είναι πρακτικά δυνατός γιατί θα απαιτούσε ένα τεράστιο μέγεθος χώρου και χρόνου. Σε αυτό το σημείο εισάγεται η σημασία της ανταλλαγής χρόνου και μνήμης όπου γίνεται ένας συμβιβασμός ανάμεσα στον χώρο δεδομένων του προϋπολογισμού και στον τελικό χρόνο επίθεσης, όπου τελικός χρόνος επίθεσης είναι ο χρόνος εύρεσης της αντιστοιχίας στα αποθηκευμένα δεδομένα με το ζητούμενο προς αποκρυπτογράφηση keystream και η εύρεση της εσωτερικής του κατάστασης. Δηλαδή αν υποθέσουμε ότι υπάρχει μια καμπύλη ανταλλαγής (tradeoff curve), τα ζητούμενα σημεία θα καθορίζονται ως εξής: όσο μεγαλύτερος χρόνος, τόσο μικρότερη μνήμη και το αντίστροφο. Οι παράμετροι που καθορίζουν τα σημεία αυτής της καμπύλης εξαρτώνται από τη συνάρτηση του αλγορίθμου και τους διαθέσιμους πόρους για τον προϋπολογισμό και για την τελική επίθεση.

Η επίθεση με ανταλλαγή χρόνου-μνήμης αποτελείται από 2 στάδια όπως έχει προαναφερθεί: το στάδιο προϋπολογισμού και το στάδιο πραγματικού χρόνου επίθεσης. Το στάδιο προϋπολογισμού είναι πολύ χρονοβόρο και το αποτέλεσμά του είναι ένας πίνακας (ή πολλοί) με αποθηκευμένα ζευγάρια εσωτερικής κατάστασης-keystream. Το στάδιο πραγματικού χρόνου ολοκληρώνεται όσο το δυνατόν ταχύτερα και το αποτέλεσμά του είναι η εσωτερική κατάσταση μετά από εξερεύνηση των πινάκων για μια αντιστοιχία του διαθέσιμου keystream.

Οι παράμετροι μιας επίθεσης ανταλλαγής χρόνου μνήμης είναι οι παρακάτω[34]:

- N: το μέγεθος του διαστήματος αναζήτησης(αριθμός εσωτερικών καταστάσεων)
- P: χρόνος προϋπολογισμού
- M: η διαθέσιμη μνήμη(σε hard disks ή DVDs)
- T: πραγματικός χρόνος επίθεσης
- D: τα διαθέσιμα (realtime) δεδομένα

Η καμπύλη ανταλλαγής που περιγράφηκε από τον Hellman για τους αλγόριθμους τμήματος (block ciphers) είναι η $TM^2 = N^2$ (τυχαίες συναρτήσεις-random functions), για $1 \leq T \leq N$, $P = N$, $D = 1$ *και* $TM = N$ (τυχαίες μεταθέσεις-random permutations). Σημειώνεται ότι ο χρόνος προϋπολογισμού δεν πρέπει να είναι πιο πολύπλοκος από την αναζήτηση όλων των πιθανών *N*. Με την επιλογή $T = M$ λαμβάνεται το σημείο $M = T = N^{2/3}$ *ή* $M = T = N^{1/2}$.
Η καμπύλη ανταλλαγής που περιγράφηκε από τους Babbage[37], Golic[27], είναι η $TM = N$, *για* $1 \leq T \leq D$, $P = M$ για τους αλγόριθμους ροής.
Η καμπύλη ανταλλαγής που περιγράφηκε από τους Biryukov,Shamir[34], είναι η $TM^2D^2 = N^2$, *για* $D^2 \leq T \leq N$ για τους αλγόριθμους ροής.

Πρέπει να αναφερθεί ότι η επίθεση στοχεύει κυρίως τους αλγόριθμους ροής (stream ciphers) ενώ στους περισσότερους αλγόριθμους τμήματος (block ciphers) ισοδυναμεί με μια εξαντλητική αναζήτηση του κλειδιού [48].

4.1.1 Διαδικασία πίνακα – αλυσίδες Hellman

Η μέθοδος του Hellman εφαρμόζεται και αντιστρέφει τις one-way συναρτήσεις (με κάποιες εξαιρέσεις), δηλαδή δεδομένου ενός $y= f(x)$ βρίσκει το *x*. Χρησιμοποιεί chosen plaintext αν και μπορεί να χρησιμοποιηθεί και με ciphertext only.
Η βασική δομή του πίνακα (ή των πινάκων) είναι ως εξής: αποτελείται από αλυσίδες(chains) μήκους *t*, και είναι μεγέθους *m* x *t*, όπου *m* είναι ο αριθμός των σημείων εκκίνησης (start points). Σύμφωνα με τον Hellman[45] τα startpoints-SPs (θεωρητικά όλο το keyspace) είναι είσοδοι της συνάρτησης *f* και η έξοδος ξανατροφοδοτείται στη συνάρτηση *f* για *t* φορές, όπου *t* το μήκος της αλυσίδας.

$$For\ 1 \leq i \leq m$$
$$X_{i0} = SP_i$$
$$X_{ij} = f(X_{i,j-1})\ όπου\ 1 \leq j \leq t$$

Το τελικό σημείο(endpoint-EP) της κάθε i αλυσίδας είναι το:

$$EP_i = f^t(SP_i)$$

Τα ενδιάμεσα σημεία απορρίπτονται και αποθηκεύονται στον πίνακα μόνο τα:

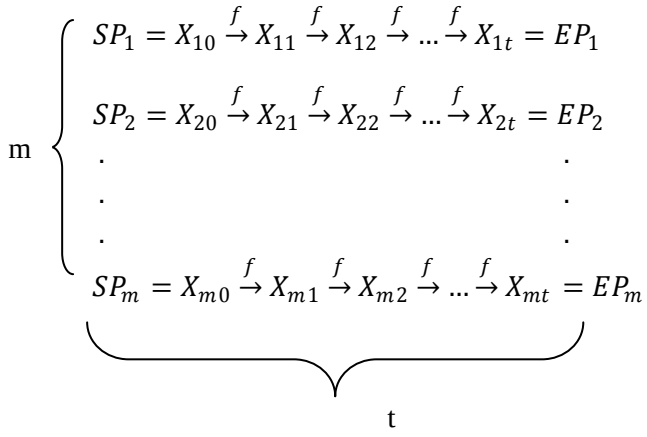
$$\{SP_i, EP_i\}^m{}_{i=1}$$

ταξινομημένα ως προς τα EP.

Η λειτουργία της συνάρτησης f είναι η εξής:

$$F(K) = R[S_K(P_0)]$$

όπου S_K είναι η συνάρτηση του κρυπταλγόριθμου με ένα κλειδί *K*, P_0 ένα fixed plaintext και *R* είναι μια συνάρτηση μετατροπής-αναγωγής (reduction function).



Εικόνα 13: Ένας πίνακας Hellman *mxt* [45]

Στο τελικό στάδιο της επίθεσης δεδομένου ενός plaintext P_0, ή ενός ciphertext C_0 ($C_0 = S_K(P_0)$), υπολογίζεται το $Y_1=R(C_0) = f(K)$ και το $Y1$ ελέγχεται αν υπάρχει σαν τελικό σημείο στον πίνακα (endpoint).

Αν υπάρχει σαν τελικό σημείο τότε υπάρχει περίπτωση το ζητούμενο κλειδί είναι μέσα στην αλυσίδα. Συνεπώς αρχίζοντας από το SP της συγκεκριμένης αλυσίδας γίνεται ο επαναϋπολογισμός της αλυσίδας και ελέγχεται το προτελευταίο σημείο της αλυσίδας εάν είναι το ζητούμενο κλειδί δηλαδή αν η αποκωδικοποίησή του ciphertext παράγει το plaintext.

Εάν δεν υπάρχει σαν τελικό σημείο τότε υπολογίζεται το $Y_2=f(Y_1)$ και ελέγχεται εάν αυτό είναι ένα τελικό σημείο στον πίνακα. Εάν είναι ελέγχεται αν είναι κλειδί το δεύτερο σημείο από το τέλος της αλυσίδας. Εάν ούτε το $Y2$ είναι τελικό σημείο υπολογίζονται με τον παραπάνω τρόπο, και ελέγχονται για τελικά σημεία και τα αντίστοιχα κλειδιά τους, όλα τα $Y_t = f(Y_{t-1})$.

Σημειώνεται ότι η διαδικασία του Hellman στη δημοσίευσή του απευθύνεται στους αλγόριθμους τμήματος (block ciphers) όπου τα πρώτα *k* bits (μέγεθος του κλειδιού) του keystream λαμβάνονται ως η συνάρτηση $f(K)$.

4.1.2 Συγκρούσεις-Συγχωνεύσεις-Ψευδείς ειδοποιήσεις (Collisions- Merges-False alarms)

Ένα μειονέκτημα των πινάκων είναι οι συγκρούσεις (collisions). Σύγκρουση ονομάζεται η παραγωγή του ίδιου τελικού σημείου (endpoint) από δύο ή περισσότερες διαφορετικές αλυσίδες με αποτέλεσμα τη συγχώνευση (merge) των αλυσίδων. Κάθε συγχώνευση μειώνει τον αριθμό των διακριτών κλειδιών που καλύπτονται από έναν πίνακα.

Ψευδή ειδοποίηση (false alarm) ονομάζεται η κατάσταση στην οποία η αλυσίδα που αποτελείται από τα *Yt* έχει συγχωνευτεί σε κάποια θέση με μια ήδη υπάρχουσα αποθηκευμένη αλυσίδα η οποία όμως δεν περιέχει το ζητούμενο κλειδί. Συνεπώς η εύρεση μιας αντιστοίχησης στον πίνακα δε σηματοδοτεί την εύρεση του κλειδιού και είναι ένας από τους λόγους για τους οποίους πρέπει να ελέγχεται κάθε πιθανό υποψήφιο κλειδί που ανακτάται από τον πίνακα για την ορθότητά του. Στο [49] προτείνεται μια μέθοδος βασισμένη σε σημεία ελέγχου (checkpoints) που μειώνει το χρόνο που καταναλώνεται στα false alarms και συνεπώς το χρόνο της επίθεσης αυξάνοντας τη μνήμη (αλλά κερδίζοντας σε κατά πολύ μεγαλύτερο ποσοστό σε χρόνο).

4.1.3 Πιθανότητα επιτυχίας

Η πιθανότητα επιτυχούς εύρεσης ενός κλειδιού σε έναν πίνακα εξαρτάται από τον αριθμό των διακριτών κλειδιών που καλύπτει και συνεπώς από τις συγκρούσεις και τις αλυσίδες που μπαίνουν σε βρόχους (loops).

Θεωρητικά αν όλα τα κλειδιά σε όλες τις αλυσίδες είναι διαφορετικά και εάν το *K* έχει επιλεχθεί ομοιόμορφα από όλες τις πιθανές τιμές, η πιθανότητα επιτυχίας θα είναι *mt/N* [45].
Η πιθανότητα επιτυχίας μέσω του «προβλήματος κατοχής»(occupancy problem) στο [50] είναι:

$$\mathrm{P}(success) = 1 - e^{-mtr/2^k} \qquad \textit{εξίσωση 1}$$

Στη δημοσίευση [51] παρουσιάζεται μια πιο λεπτομερής ανάλυση με πιθανότητα επιτυχίας

$$\Pr(success) \geq 1 - e^{-g(u)\frac{rmt}{2^k}} \qquad \textit{εξίσωση 2}$$

όπου $g(u) = \frac{1}{u}\int_0^u \frac{1-e^{-x}}{x}dx$ *και* $u = \frac{mt^2}{2^k}$

Θεωρώντας ότι έχουμε $r = 2^{k/3}$ πίνακες και ότι $m = t = 2^{k/3}$ όπου *k* είναι το μήκος του κλειδιού με πολυπλοκότητα προϋπολογισμού 2^k, η πιθανότητα επιτυχίας, όπως αναφέρεται και στο [52], είναι περίπου 0,63[45] για την πρώτη εξίσωση και 0,55 για τη δεύτερη.

mtr	P(success)
0	0
2^{k-5}	0.03
2^{k-4}	0.06
2^{k-3}	0.12
2^{k-2}	0.22
2^{k-1}	0.39
2^{k}	0.63
2^{k+1}	0.86
2^{k+2}	0.98
2^{k+3}	0.99
∞	1

Πίνακας 5 : Πιθανότητες επιτυχίας για την εξίσωση 1 για διάφορες επιλογές του *mtr* [52]

4.1.4 Διακριτά σημεία

Το 1982 ο Rivest πρότεινε μια βελτιστοποίηση που στηρίζεται στα διακριτά σημεία (distinguished points-DP), η οποία μειώνει σημαντικά τις πράξεις αναζήτησης(look-ups) που απαιτούνται για τον εντοπισμό της αντιστοίχησης μιας καταχώρησης στον πίνακα. Τα διακριτά σημεία ικανοποιούν μια συγκεκριμένη συνθήκη όπως για παράδειγμα τα τελευταία *ν* bits να είναι μηδενικά. Συνεπώς ο πίνακας αποτελείται από καταληκτικά (ενίοτε και ενδιάμεσα) σημεία που είναι διακριτά και κατά τη διάρκεια της επίθεσης πραγματικού χρόνου χρειάζεται η πρόσβαση στο δίσκο μόνο όταν συναντήσουμε κάποιο διακριτό σημείο στη διαδικασία που εφαρμόζεται πάνω στο δεδομένο keystream. Δηλαδή υπολογίζονται οι αλυσίδες έως ότου οδηγηθούν σε διακριτό σημείο σε *t* ή λιγότερο από *t* επαναλήψεις ή έως ότου το μέγεθος της αλυσίδας γίνει *t+1*. Αποθηκεύονται μόνο αυτές που οδηγούν σε διακριτό σημείο και αν υπάρχουν περισσότερες από μια που οδηγούν στο ίδιο σημείο αποθηκεύεται μόνο η μεγαλύτερη.
Αυτή η βελτιστοποίηση μειώνει τον αριθμό των αναζητήσεων σημαντικά αλλά αυξάνει την πιθανότητα να συγχωνευτούν κάποιες αλυσίδες (merge).

4.2 Πίνακες ουράνιου τόξου-Rainbow tables

Το 2003 ο Philippe Oechslin [47] πρότεινε μια βελτιστοποίηση στην ανταλλαγή χρόνου-μνήμης του Hellman την οποία ονόμασε πίνακες ουράνιου τόξου (rainbow tables).
Η βασική ιδέα του Oechslin είναι η χρήση μιας ακολουθίας από διαφορετικές συναρτήσεις μετατροπής-αναγωγής *R* (reduction functions) στην κατασκευή μιας

αλυσίδας. Δεδομένου ότι σε κάθε βήμα αντιστοιχεί μια *Ri*, όπου $1 \leq i \leq t-1$ συνεπάγεται ότι αντιστοιχεί και διαφορετική συνάρτηση *fi* με $1 \leq i \leq t-1$. Με αυτόν τον τρόπο μειώνεται η πιθανότητα των συγχωνεύσεων των αλυσίδων (πιθανότητα merge *1/t* σε μια σύγκρουση) αφού για να συγχωνευτούν πρέπει να έχουν το ίδιο κλειδί στην ίδια θέση, ενώ οι υπάρχουσες συγχωνεύσεις εντοπίζονται εύκολα αφού καταλήγουν στο ίδιο τελικό σημείο. Λόγω της μείωσης αυτής το μέγεθος των πινάκων είναι μεγαλύτερο και ο αριθμός των πινάκων μικρότερος. Στο [47] ο Oechslin υπολογίζει ότι η πιθανότητα επιτυχίας *t* πινάκων Hellman με μέγεθος *m* x *t* είναι περίπου ίδια με την πιθανότητα επιτυχίας ενός πίνακα ουράνιου τόξου με μέγεθος *mt* x *t*. Επίσης οι πίνακες ουράνιου τόξου μειώνουν τις αναζητήσεις στη μνήμη κατά συντελεστή *t*. Επιπλέον πρέπει να σημειωθεί ότι στους rainbow tables δεν υπάρχουν loops. Όπως δηλώνει ο Oechslin οι πίνακες ουράνιου τόξου έχουν τα προτερήματα των πινάκων που χρησιμοποιούν διακριτά σημεία αλλά χωρίς τους περιορισμούς τους. Για παράδειγμα οι αλυσίδες έχουν σταθερό μέγεθος σε αντίθεση με τις αλυσίδες με διακριτά σημεία, γεγονός που μειώνει τον αριθμό των ψευδών ειδοποιήσεων (false alarms). Τέλος οι υπολογισμοί που απαιτούνται για την εύρεση ενός κλειδιού μειώνονται κατά συντελεστή 2 σε σχέση με τους πίνακες Hellman και κατά συντελεστή 12 σε σχέση με τους πίνακες με διακριτά σημεία[47].
Στο στάδιο της πραγματικής επίθεσης ακολουθείται περίπου η ίδια διαδικασία που έχει περιγραφεί για τους πίνακες Hellman, δηλαδή υπολογίζεται το Y_1 σύμφωνα με την τελευταία συνάρτηση αναγωγής (reduction function) και συγκρίνεται με τα τελικά σημεία του πίνακα $Y_1 = R_{t-1}(C)$, στη συνέχεια υπολογίζεται το Y_2 σύμφωνα με την προτελευταία συνάρτηση αναγωγής και την τελευταία συνάρτηση f, όπου έχουμε $Y_2 = f_{t-1}(R_{t-2}(C))$ και ελέγχεται για τελικό σημείο και ούτω καθεξής προς τα πίσω στις αλυσίδες.

Η πολυπλοκότητα κατασκευής τέλειων πινάκων ουράνιου τόξου είναι πολύ μεγάλη αφού είναι *Nt*. Το όνομα του πίνακα ουράνιου τόξου προέρχεται από το γεγονός ότι αν σε κάθε συνάρτηση R αντιστοιχηθεί ένα διαφορετικό χρώμα τότε ο πίνακας μοιάζει με ουράνιο τόξο. Στην παρακάτω εικόνα αναπαριστούνται οι αλυσίδες ενός πίνακα ουράνιου τόξου.

$$\text{m} \left\{ \begin{array}{l} SP_1 = X_{10} \xrightarrow{f1} X_{11} \xrightarrow{f2} X_{12} \xrightarrow{f3} \ldots \xrightarrow{ft} X_{1t} = EP_1 \\ SP_2 = X_{20} \xrightarrow{f1} X_{21} \xrightarrow{f2} X_{22} \xrightarrow{f3} \ldots \xrightarrow{ft} X_{2t} = EP_2 \\ \cdot \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \cdot \\ \cdot \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \cdot \\ \cdot \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \cdot \\ SP_m = X_{m0} \xrightarrow{f1} X_{m1} \xrightarrow{f2} X_{m2} \xrightarrow{f3} \ldots \xrightarrow{ft} X_{mt} = EP_m \end{array} \right.$$

$$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}_{\text{t}}$$

Εικόνα 14 : Ένας πίνακας ουράνιου τόξου

Κεφάλαιο 5: A5/1 Security Project

Το Δεκέμβριο του 2009 ανακοινώθηκε το project των Karsten Nohl, Chris Paget, Sascha Krissler και μιας ομάδας κρυπτογράφων, επονομαζόμενο A5/1 security project[36]. Παρουσιάστηκε στο συνέδριο 26C3 στο Βερολίνο και βασίζεται στο project από την ομάδα THC στις αρχές του 2008, το οποίο περιλαμβάνει την κατασκευή πινάκων ουράνιου τόξου για τον A5/1 που δε δημοσιεύθηκαν όμως ποτέ.

Ο κύριος στόχος του project ήταν η υλοποίηση και κατασκευή ενός ολοκληρωμένου συνόλου των πινάκων ουράνιου τόξου που απαιτούνται για το «σπάσιμο» μιας οποιασδήποτε συνομιλίας. Η κατασκευή των πινάκων υλοποιήθηκε σε εξειδικευμένους επεξεργαστές, όπως είναι οι GPUs και το cell του PS3, και το συνολικό μέγεθός τους είναι περίπου 2 Terabyte.

Ενδεικτικά ο ρυθμός δημιουργίας αλυσίδων είναι: 500 αλυσίδες ανά δευτερόλεπτο για την Nvidia GTX280 (CUDA-enabled) και την ATI HD5870, και 120 αλυσίδες ανά δευτερόλεπτο για το cell του PS3.
Το σύνολο του κώδικα βρίσκεται μέσω subversion στο [53].

5.1 Τεχνικά στοιχεία

Η μέθοδος της ανταλλαγής χρόνου-μνήμης που χρησιμοποιείται είναι ένας πίνακας ουράνιου τόξου που συνδυάζεται με τη μέθοδο των διακριτών σημείων, με τρόπο ότι η R αλλάζει όταν βρεθεί ένα διακριτό σημείο καθώς και το τελικό σημείο είναι επίσης διακριτό.

Πιο συγκεκριμένα στο project χρησιμοποιήθηκαν 3 διαφορετικές σχεδιάσεις: η πρώτη είναι η σχεδίαση dp15k32 με διακριτά σημεία 15 bits και 32 διαφορετικές R, η δεύτερη είναι η σχεδίαση dp15k8 με διακριτά σημεία 15 bits και 8 διαφορετικές R και η τρίτη είναι η dp12k8e100 με διακριτά σημεία 12 bits και 8 διαφορετικές R. Επίσης οι 2 πρώτες χρησιμοποιούν την 64-bit έξοδο της μηχανής A5/1 ως είσοδο στο επόμενο βήμα ενώ η τρίτη χρησιμοποιεί την έξοδο αφού χρονιστεί για 100 κύκλους. Κατά τη διάρκεια της υλοποίησης αυτής της διπλωματικής εργασίας η χρησιμοποιούμενη σχεδίαση ήταν η πρώτη.

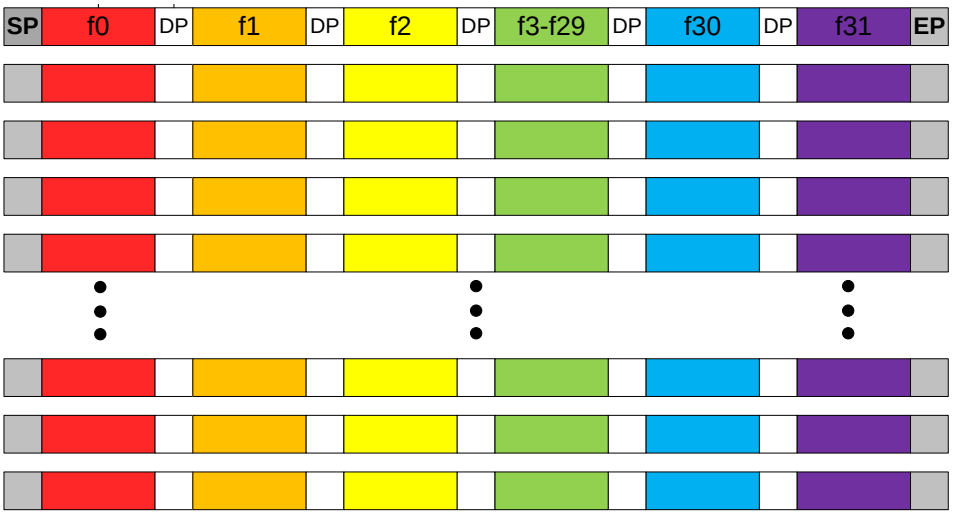
Πρέπει να σημειωθεί ότι οι συναρτήσεις R δεν υπολογίζονται κατά τη διάρκεια της κατασκευής των αλυσίδων αλλά δημιουργούνται από τη CPU και αντιγράφονται στην DRAM της GPU. Επίσης οι R είναι ουσιαστικά συναρτήσεις γύρου (round functions) και όχι συναρτήσεις αναγωγής (reduction functions) αφού το μήκος της εξόδου της μηχανής A5/1 και το μήκος της εσωτερικής κατάστασης του αλγορίθμου είναι ίδια και ίσα με 64 bits, οπότε δε χρειάζεται η συνάρτηση αλλαγής του μεγέθους.

Η κατασκευή των πινάκων γίνεται ως εξής:

Μια γεννήτρια (work generator) παράγει τα αρχικά σημεία των αλυσίδων (θέτει και το τελικό σημείο της αλυσίδας την ίδια τιμή ως ένας τρόπος για να φορτώνει τις ολοκληρωμένες/μισο-ολοκληρωμένες αλυσίδες).

Η διαδικασία υπολογισμού των αλυσίδων ξεκινάει με τη μηχανή A5/1 και τις R.
Η μηχανή A5/1 είναι ουσιαστικά ο χρονισμός των LFSRs 64 φορές ενώ οι R συναρτήσεις είναι 64-bit τιμές.
Το αρχικό σημείο πρώτα τροφοδοτείται στη μηχανή A5/1 και η 64-bit έξοδος της μηχανής εν συνεχεία γίνεται XOR με την πρώτη τιμή της R. Εάν το αποτέλεσμα της πράξης XOR είναι διακριτό σημείο τότε αυτό αποτελεί το πρώτο μετά το αρχικό σημείο της αλυσίδας. Αν δεν είναι διακριτό σημείο το ίδιο αποτέλεσμα μετά την πράξη XOR επανατροφοδοτείται στη μηχανή A5/1 και η ίδια διαδικασία επαναλαμβάνεται έως ότου να βρεθεί ένα διακριτό σημείο.
Η αλυσίδα ολοκληρώνεται μόλις βρεθεί ένα διακριτό σημείο, το τελικό σημείο, μετά την τελευταία συνάρτηση R ή όταν η συνάρτηση R πρόκειται να αλλάξει από την $32^{η}$ στην $33^{η}$.
Τα ενδιάμεσα διακριτά σημεία απομακρύνονται και ένας work consumer(stxxl) απομακρύνει τις συγχωνεύσεις και ταξινομεί τις αλυσίδες ως προς τα τελικά σημεία.
Το τελευταίο βήμα είναι η αποθήκευση στον πίνακα μόνο των αρχικών και των τελικών σημείων των αλυσίδων.

Στο παρακάτω σχήμα παρουσιάζεται σχηματικά η δομή των αλυσίδων.



Εικόνα 15 : Η δομή του πίνακα

Το μήκος της αλυσίδας είναι 2^{20} κατά μέσο όρο (2^{15} * 32), κάθε πίνακας έχει $2^{28.5}$ αλυσίδες και ο αριθμός των πινάκων είναι $2^{8.5}$. Συνολικά 2^{57} ($2^{20+28.5+8.5}$) καταχωρήσεις σε όλους τους πίνακες.

Πρακτικά χρειάζεται ο υπολογισμός παραπάνω από 2^{64}/N τιμών λόγω του ότι μπορεί να υπάρξει μεγάλος αριθμός συγκρούσεων.

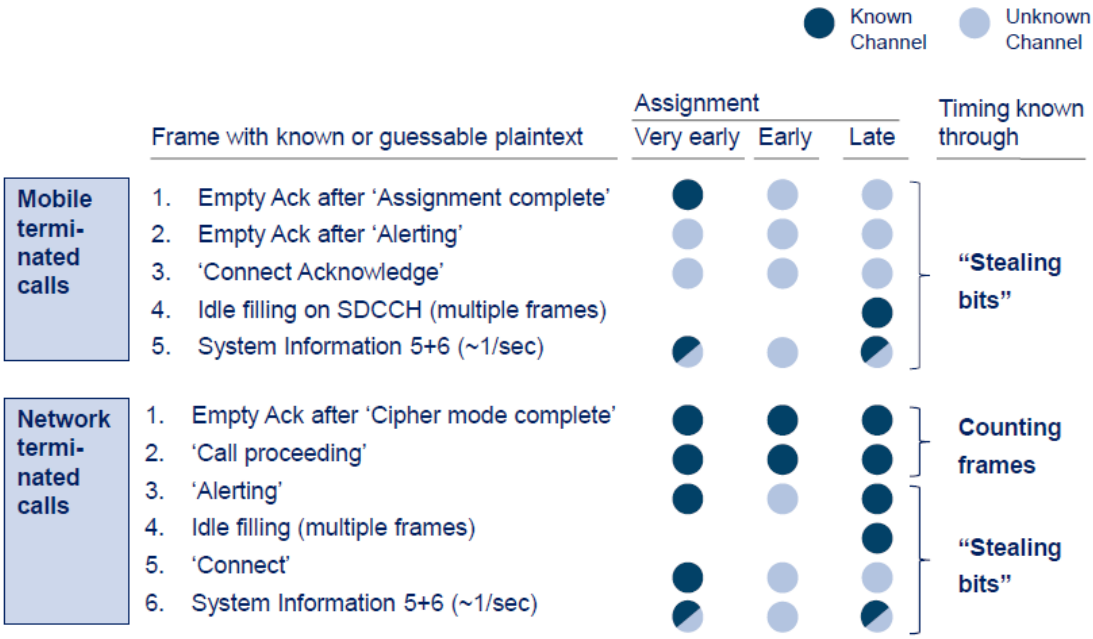
Λαμβάνοντας υπόψιν τη σχεδίαση dp12k8e100 το μήκος της αλυσίδας είναι κατά μέσο όρο 2^{15} και συνολικά υπολογίζονται 2^{53} καταχωρήσεις ενώ αποθηκεύονται μόνο οι 2^{38} καταχωρήσεις (αφού κάθε αλυσίδα έχει μήκος 2^{15} κατά μέσο όρο αν και ο πίνακας περιέχει 2^{53} καταχωρήσεις αποθηκεύονται οι 2^{38}).
Ο αριθμός των υπολογιζόμενων καταχωρήσεων 2^{53} θα εξηγηθεί στην ενότητα 5.3.1.

Η κατασκευή των πινάκων έγινε από χρήστες του internet σε όλο τον κόσμο για να αποφευχθεί η συγκέντρωσή τους σε ένα μόνο σημείο καθώς και για τη διασφάλιση ότι δεν υπάρχει νομικό πρόβλημα για τους κρυπταναλυτές. Ολοκληρώθηκε σε περίπου 2 μήνες και έχουν γίνει upload στο internet.

5.2 Γενικές απαιτήσεις για την επίθεση-εύρεση δεδομένων

Το known plaintext που απαιτείται για την επίθεση μπορεί να αποκτηθεί ακέραιο ή να μαντευτεί από πληθώρα μηνυμάτων που ανταλλάσσονται ανάμεσα στο χρήστη και το δίκτυο .

Κάποιες από αυτές αναπαριστώνται στην παρακάτω εικόνα[36].



Εικόνα 16 : «Χρήσιμα» μηνύματα [36]

Η σημαντικότερη πηγή known plaintext είναι το πρώτο κρυπτογραφημένο μήνυμα που στέλνεται από τον κινητό σταθμό στο σταθμό βάσης και σηματοδοτεί την έναρξη της κρυπτογράφησης μέσω της εντολής CIPHER_MODE_COMPLETE. Αυτό το μήνυμα των ζεύξεων ανόδου (uplink) που είναι layer 3 αποστέλλεται μέσα σε ένα LAPDm frame (layer 2 frame format). Επειδή εσωκλείεται στο LAPDm frame σχεδόν όλα τα υπόλοιπα bits του frame είναι σταθερά και αυτά αποτελούν το known plaintext που απαιτείται για την επίθεση.

Το LAPDm frame είναι 23 bytes ή 184 bits, τα οποία πριν σταλούν κωδικοποιούνται για να υποστηρίζουν τη διόρθωση λαθών (forward error correction), συνεπώς το τελικό μήκος του είναι 456 bits. Τα 456 bits στέλνονται σε 4 ριπές (bursts) των 114-bits.
Κάθε 114 bits παρέχει 51 ακολουθίες keystream των 64-bit. Οι 51 ακολουθίες αποκτούνται χωρίζοντας τα 114 bits σε υποτμήματα των 64-bits με τον εξής τρόπο: το πρώτο είναι τα bits (0…63), το δεύτερο τα bits (1…64), το τρίτο τα bits (2…65) και ούτω καθεξής με το τελευταίο να είναι το (50…113).

Συνεπώς συνολικά αποκτούνται 4 x 51 = 204 ακολουθίες keystream, για καθεμία εκ των οποίων μπορεί να γίνει η αναζήτηση στον πίνακα για την εσωτερική κατάσταση του αλγορίθμου (των καταχωρητών) που έχει σαν αποτέλεσμα τη συγκεκριμένη ακολουθία.

5.3 Η αναζήτηση

Μετά την ολοκλήρωση της κατασκευής των πινάκων ουράνιου τόξου ακολουθεί η εύρεση της 64-bit εσωτερικής κατάστασης που οδηγεί μέσω του αλγορίθμου A5/1 στα πρώτα 64-bit της εξόδου, δηλαδή στην 64-bit δεδομένη ακολουθία keystream. Η αναζήτηση μπορεί να γίνει για μια ακολουθία αλλά και περισσότερες συνεχόμενες τροφοδοτούμενες ακολουθίες που προκύπτουν από τα γραφόμενα της ενότητας 5.2.

5.3.1 Η διαδικασία της αναζήτησης

Η διαδικασία αναζήτησης (lookup process) μιας καταχώρησης στον πίνακα σύμφωνα με τη σχεδίαση που αναλύουμε γίνεται ως εξής: για κάθε 64-bit ακολουθία (θα την ονομάσουμε τελική ακολουθία) υπολογίζονται 32 αλυσίδες έως τα τελικά σημεία τους αρχίζοντας σε κάθε γύρο (χρώμα). Για τα τελικά σημεία αναζητείται η αντιστοίχησή τους στον πίνακα (ή πίνακες) και μόλις βρεθεί, με το ανακτώμενο αρχικό σημείο από το δίσκο υπολογίζεται μια νέα αλυσίδα έως το σημείο που άρχισε η αλυσίδα αναζήτησης που παρείχε το αντίστοιχο τελικό σημείο στο δίσκο.

Το αποτέλεσμα της αναζήτησης (σύμφωνα με το table format που αναλύουμε dp15k32) είναι η εσωτερική κατάσταση πριν από τους 100 χρονισμούς. Ακολουθεί η

επιβεβαίωση του αποτελέσματος χρονίζοντας προς τα μπρος τις LFSRs για 100 κύκλους και παρατηρώντας αν η 64-bit ακολουθία που προκύπτει έπειτα είναι η τελική.

Αφού επιβεβαιωθεί η ορθότητα, η καταχώρηση που ανακτήθηκε χρονίζεται προς τα πίσω 22 φορές για να βγει το frame number και καταλήγουμε στην εσωτερική κατάσταση αμέσως μετά την εισαγωγή του κλειδιού στους καταχωρητές και πριν τον frame number.

Πρέπει να αναφερθούν 2 σημεία στην όλη διαδικασία.

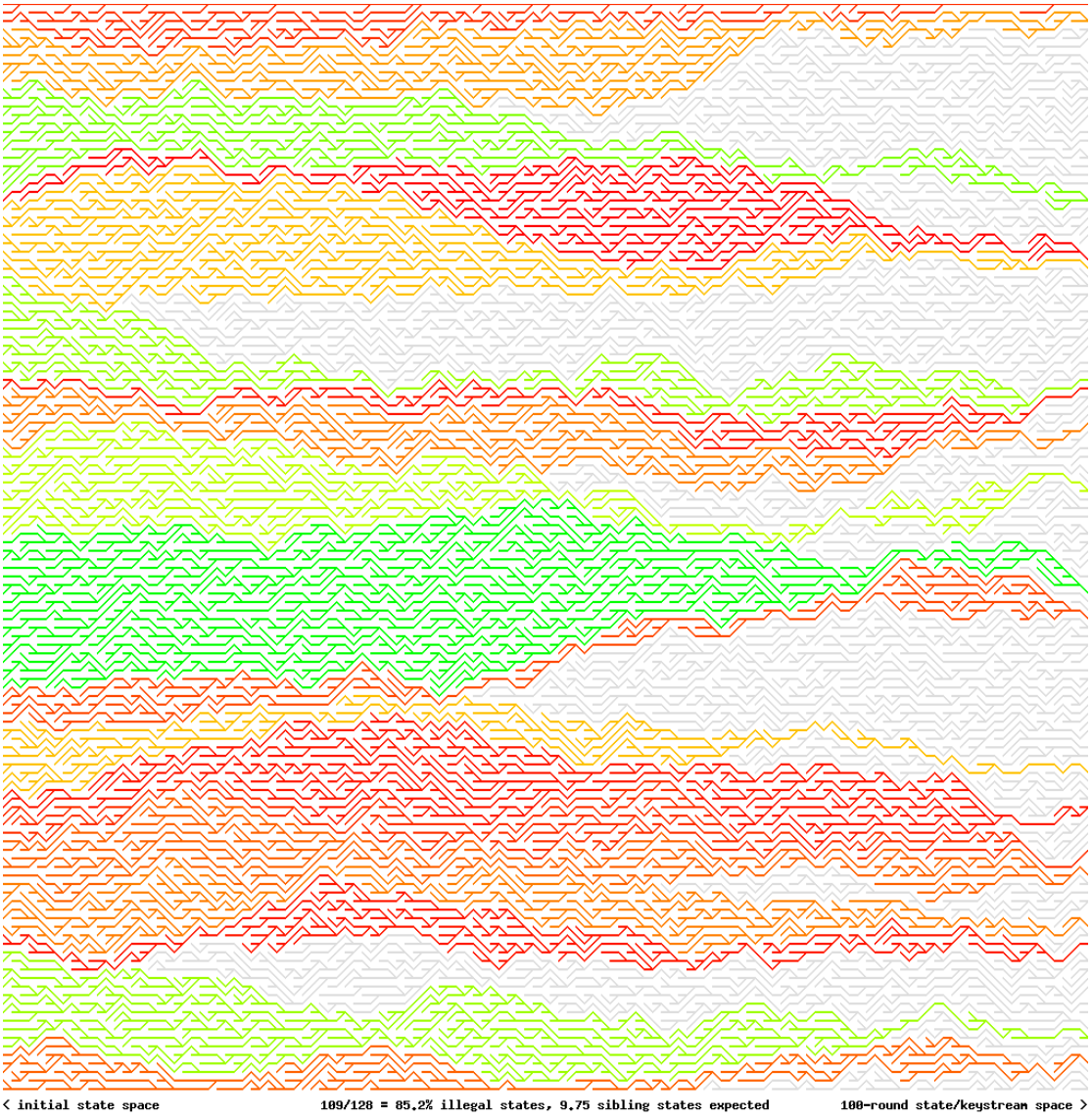
Το πρώτο είναι το γεγονός ότι αν η ανακτώμενη καταχώρηση από τον πίνακα χρονιστεί προς τα εμπρός 100 φορές και μετά προς τα πίσω άλλες 100 φορές, μπορούν να βρεθούν και άλλες καταστάσεις που δημιουργούν την ίδια τελική ακολουθία.

Με αυτόν τον τρόπο μια τυχαία κατάσταση μπορεί να μετατραπεί σε άλλες 1.4 καταστάσεις κατά μέσο όρο που οδηγούν στην ίδια τελική ακολουθία δεδομένου του μειωμένου υποσυνόλου (αφαιρώντας τις καταστάσεις που δεν μπορούν να υπάρξουν μετά τους 100 χρονισμούς βλ.παρακάτω).

Στην εικόνα 17 αναπαριστούνται τα μονοπάτια που δημιουργούν οι καταστάσεις μέσω του χρονισμού μπρος/πίσω. Στην αριστερή πλευρά της εικόνας είναι το αρχικό εύρος των δυνατών καταστάσεων και στη δεξιά πλευρά το εύρος των καταστάσεων μετά τους 100 χρονισμούς.

Όλες οι καταστάσεις εντός ενός πράσινου και κόκκινου μέρους συγκλίνουν στην ίδια κατάσταση μετά τους 100 χρονισμούς δηλαδή στα δεξιά της εικόνας. Τα κόκκινα κομμάτια έχουν λιγότερες δυνατές καταστάσεις που οδηγούν στην ίδια τελική ακολουθία από ότι τα πράσινα κομμάτια, ενώ το χρώμα γκρι δίνει καταστάσεις που δε δύναται να προσπελαστούν με το χρονισμό προς τα εμπρός.

Μετά από μια επιτυχή αναζήτηση στους πίνακες λαμβάνεται μια τιμή που ανήκει στα αριστερά της εικόνας. Με το χρονισμό μπροστά 100 φορές λαμβάνεται η αντίστοιχη τιμή που ανήκει στη στήλη στα δεξιά. Ακολούθως με το χρονισμό προς τα πίσω 100 φορές παράγεται ολόκληρο το πράσινο τμήμα, σε περίπτωση που υπάρχουν επιπλέον τιμές που δίνουν την ίδια τελική ακολουθία. Δηλαδή ανακτώνται επιπλέον δυνατές εσωτερικές καταστάσεις δεδομένης μιας τελικής ακολουθίας.



Εικόνα 17 : Τα μονοπάτια που δημιουργούν οι καταστάσεις μέσω του χρονισμού μπρος/πίσω[36]

Σε αυτό το σημείο μπορεί να εξηγηθεί ο αριθμός των καταχωρήσεων 2^{53} που έχει αναφερθεί στην ενότητα 5.1.

Από τις 2^{64} πιθανές καταστάσεις οι 2^{3} είναι αδύνατες μετά από τους 100 χρονισμούς. Όταν ολοκληρωθεί το βήμα του αλγορίθμου που περιλαμβάνει τους 100 χρονισμούς σύμφωνα με το μηχανισμό της πλειοψηφίας, μόνο το 16% των καταστάσεων μπορούν να υπάρξουν. Επίσης δεδομένου των προαναφερθέντων για τον αριθμό των δυνατών εσωτερικών καταστάσεων που μπορούν να υπάρξουν για μια τελική ακολουθία, ο εναπομείναντας αριθμός 2^{61} διαιρείται δια 1.4. Άρα ο αριθμός φτάνει στην τιμή $2^{60,515}$. Αν αυτός ο αριθμός διαιρεθεί με την τιμή 204, δηλαδή τον αριθμό των ακολουθιών για τις οποίες μπορεί να γίνει αναζήτηση στον πίνακα και να δώσουν μια καταχώρησή του, τότε το καλυπτόμενο εύρος φτάνει στο $2^{52,843}$, συνεπώς περίπου 2^{53}.

Το δεύτερο σημείο που πρέπει να αναφερθεί είναι η διαδικασία χρονισμού προς τα πίσω. Εξαιρώντας το μηχανισμό της πλειοψηφίας (τον κανόνα του ρολογιού) μια LFSR χρονίζεται προς τα πίσω με αντίστροφο τρόπο από τον κανονικό, δηλαδή καθορίζοντας το lsb bit μέσω της XOR στα bits που συμμετέχουν στην ανάδραση (tap bits) και του lsb. Όταν ο χρονισμός προς τα πίσω πρέπει να γίνει σύμφωνα με τον κανόνα του ρολογιού η διαδικασία που ακολουθείται είναι πιο περίπλοκη και χρονοβόρα καθώς πρέπει να ληφθεί υπόψιν κάθε συνδυασμός των bits ρολογιού και κάθε τρόπος με τον οποίον ο συνδυασμός μπορεί να χρονιστεί προς τα πίσω σε κάθε κύκλο. Συνολικά υπάρχουν 2^6 συνδυασμοί εκ των οποίων 24 οδηγούν σε αδύνατο αποτέλεσμα.

5.3.2 Πολυπλοκότητα της αναζήτησης

Όσον αφορά την πολυπλοκότητα της αναζήτησης (lookup complexity) ισχύουν τα ακόλουθα. Όπως έχει αναφερθεί ο συνολικός αριθμός των πινάκων όπως προκύπτει από τους θεωρητικούς υπολογισμούς για τη σχεδίαση dp15k32 είναι $2^{8.5}$. Συνεπώς για τις 4 x 51 = 204 ακολουθίες οι οποίες βρίσκονται στην κατοχή του κρυπταναλυτή απαιτούνται $(1 + 2 + 3 + ... + 32) * 2^{15} * 204 = 528 * 2^{15} * 204 \approx 3,5 * 10^9$ A5/1 σημεία κατά μέσο όρο. Με έναν επεξεργαστή ικανό να διεξάγει 162 εκατομμύρια διεργασίες το δευτερόλεπτο θα χρειαστούν 21 δευτερόλεπτα για να δημιουργηθούν αυτές οι αλυσίδες. Σε αυτό το χρόνο πρέπει να προστεθεί ο χρόνος περαίωσης των αναζητήσεων στο δίσκο ο οποίος είναι 65 δευτερόλεπτα (32 *204 =6528 προσβάσεις, επί 10 ms χρόνο η καθεμία).

Πρακτικά η αναζήτηση μπορεί να γίνει για μια ακολουθία ή περισσότερες παράλληλα. Αν υποθέσουμε ότι έχει γίνει ο υπολογισμός 40 πινάκων και διατίθενται 400 τελικές ακολουθίες τότε ο ελάχιστος χρόνος της αναζήτησης ανακτάται όταν 40* 32 * 204 = 261120 αλυσίδες μπορούν να επεξεργαστούν παράλληλα στο hardware. Σε περίπτωση που το εύρος του hardware είναι μικρότερο ή μεγαλύτερο από 261120 τότε ο χρόνος μεγαλώνει ή μικραίνει αντίστοιχα.

5.3.3 Βαθμός επιτυχίας

Από τη δομή της επίθεσης συμπεραίνεται ότι το ποσοστό επιτυχίας είναι ανάλογο των κατασκευασθέντων αλυσίδων-πινάκων, της υπολογιστικής δυνατότητας, του χρόνου της τελικής επίθεσης καθώς και των διαθέσιμων ακολουθιών προς αποκρυπτογράφηση.

Η πιθανότητα επιτυχίας είναι 50% αν καλύπτονται στους πίνακες $2^{64}/N$ τιμές, όπου N είναι ο αριθμός των διαθέσιμων ακολουθιών προς αποκρυπτογράφηση. Πρακτικά για να καλύπτονται $2^{64}/N$ τιμές χρειάζεται ο υπολογισμός μεγαλύτερου αριθμού τιμών για να ληφθούν υπόψιν οι συγκρούσεις.

Εάν οι αποθηκευμένες τιμές είναι 2^{57} με πιθανότητα επιτυχίας 50%, το ποσοστό επιτυχίας αυξάνεται σε 75% όταν γίνουν 2^{58}, ενώ όταν καλύπτουν το εύρος του κλειδιού (keyspace) 2^{64} τότε το ποσοστό επιτυχίας είναι 100 % .

Τα παραπάνω ποσοστά αφορούν τυχαίες τιμές για τις οποίες γίνεται αναζήτηση στον πίνακα, και όχι πραγματικές ακολουθίες από ένα κινητό τηλέφωνο. Όταν η αναζήτηση γίνεται για τις τελευταίες ακολουθίες τότε το ποσοστό επιτυχίας είναι πάνω από 90%. Αυτό οφείλεται στο γεγονός ότι ο A5/1 εσωτερικά μειώνει το εύρος των δυνατών καταστάσεων, δηλαδή μετά από το βήμα των 100 χρονισμών οι 64-bit πιθανές καταστάσεις δεν είναι 2^{64} αλλά περίπου 2^{61} και οι κατασκευασμένοι πίνακες εμπεριέχουν τιμές που αντιστοιχούν στο ελαττωμένο υποσύνολο. Το Kraken σε έναν υπολογιστή βρίσκει την κατάσταση μετά την είσοδο του κλειδιού σε μερικά δευτερόλεπτα με περίπου 90% πιθανότητα χρησιμοποιώντας 2 κρυπτογραφημένα μηνύματα.

5.4 Kraken το νορβηγικό τέρας

Το Kraken είναι μια μηχανή που χρησιμοποιεί το σύνολο των πινάκων για να «σπάσει» μια 64-bit ακολουθία. Η αναζήτηση γίνεται σε σύνολο 40-48 πινάκων, ο καθένας εκ των οποίων καταλαμβάνει 42 GB στο δίσκο και συνεπώς απαιτείται 1,7 έως 2 TB χωρητικότητα στο δίσκο διαμοιρασμένη σε διαμερίσματα χωρίς το σύστημα αρχείων. Επίσης το Kraken απαιτεί μια πολυπύρηνη συσκευή με 3GB RAM που τρέχει Linux, ενώ υπάρχει και η δυνατότητα υποστήριξης από τη GPU.

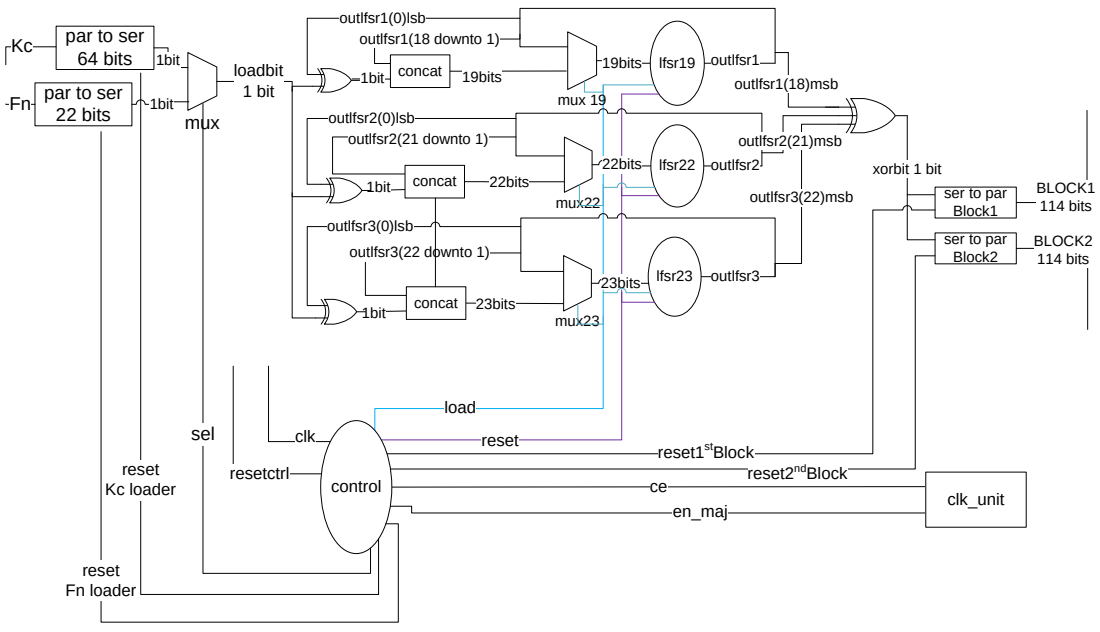
Κεφάλαιο 6: Η υλοποίησή μου

Σε αυτό το κεφάλαιο θα περιγραφεί η υλοποίηση του αλγορίθμου A5/1 σε VHDL καθώς και η υλοποίηση της κατασκευής των πινάκων ουράνιου τόξου. Η κατασκευή των πινάκων στηρίζεται στο θεωρητικό υπόβαθρο που έχει περιγραφεί στο Κεφάλαιο 5 όσον αφορά το A5/1 Security Project του Karsten Nohl [36].

6.1 Η υλοποίηση του A5/1 αλγόριθμου

Η σχεδίαση αποτελείται από 2 βασικές υπομονάδες : τη μονάδα A5/1 module και την μονάδα ελέγχου (control). Η συνολική σχεδίαση αναπαρίσταται στην εικόνα 18.

Στο A5/1 module υλοποιούνται οι lfsrs, οι μονάδες που μετατρέπουν το κλειδί Kc και το μετρητή πλαισίου Fn από παράλληλο σήμα σε σειριακό, οι μονάδες που μετατρέπουν την έξοδο των lfsrs από σειριακό σήμα σε δύο 114-bit σήματα, οι πολυπλέκτες που καθορίζουν τις εισόδους των lfsrs, οι πύλες XOR που απαιτούνται για την φόρτωση του κλειδιού και του μετρητή πλαισίου καθώς και για την έξοδο των lfsrs, και τέλος η μονάδα του ρολογιού (clk_unit).



Εικόνα 18 : Η συνολική σχεδίαση της υλοποίησης του A5/1

6.1.1 Περιγραφή της εσωτερικής λειτουργίας

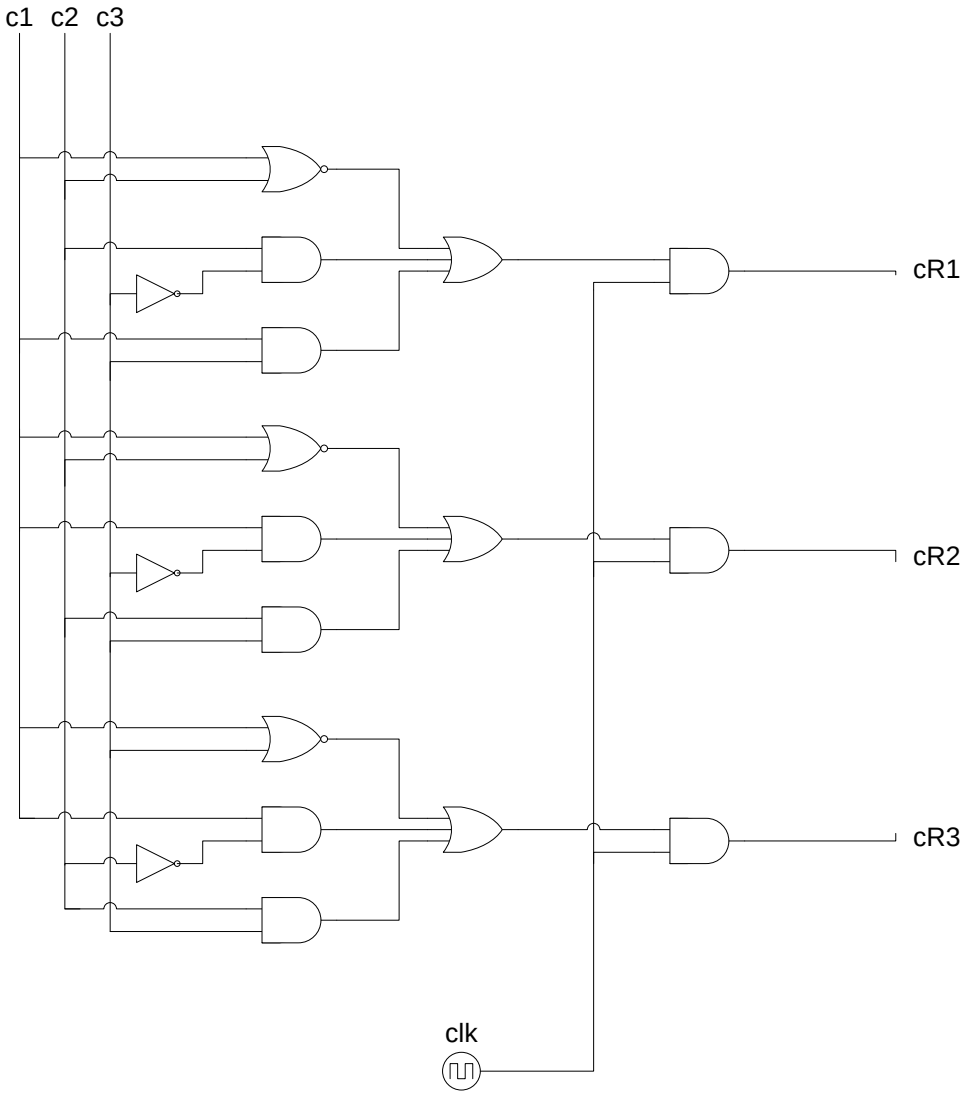
Το κλειδί Kc και ο μετρητής πλαισίου Fn εισάγονται στις μονάδες par to ser 64 bits και par to ser 22 bits, από τις οποίες εξάγονται σειριακά τα bits τους από το lsb στο msb. Ο πολυπλέκτης καθορίζει τη σειρά με την οποία φορτώνονται στις lfsrs, πρώτα το Kc και κατόπιν ο Fn.

Στη συνέχεια το κάθε bit γίνεται xor με το lsb της κάθε lfsr και το αποτέλεσμα συνενώνεται (concatenation) με τα υπόλοιπα bits της lfsr η οποία είναι αρχικοποιημένη στο 0.

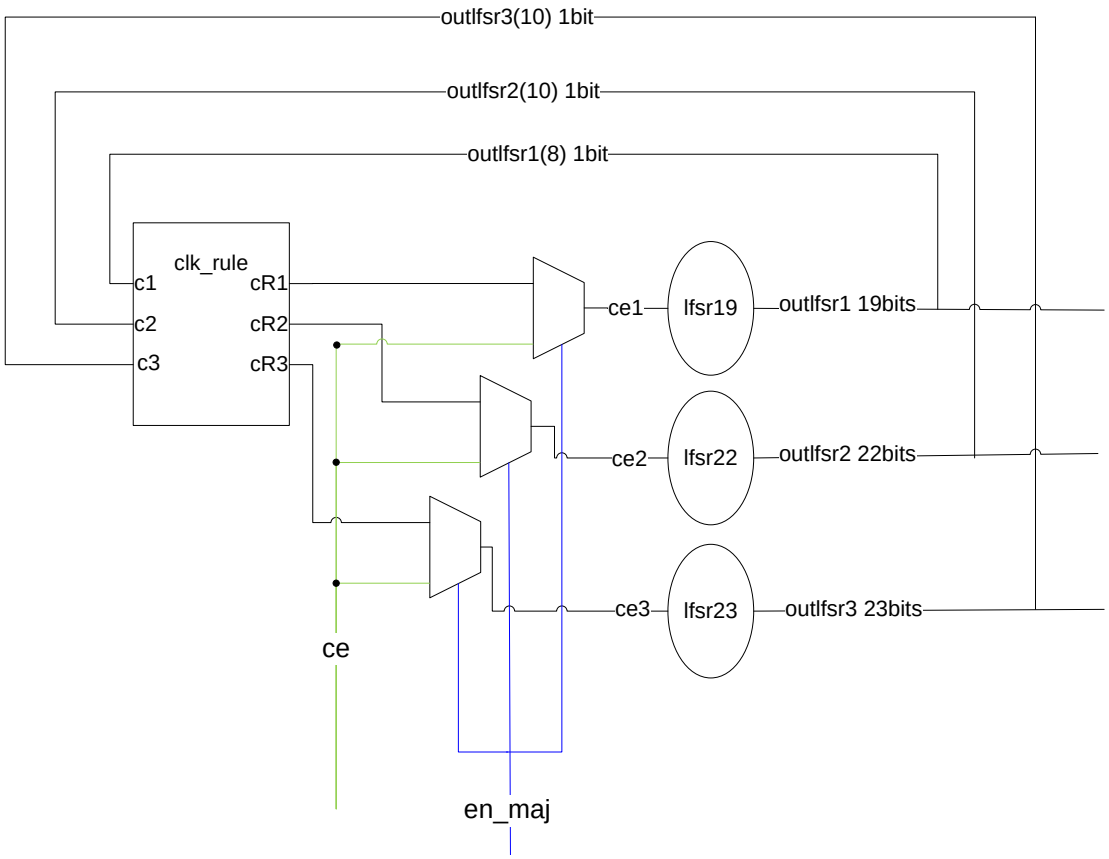
Οι πολυπλέκτες mux19, mux22 και mux23 καθορίζουν την είσοδο των lfsrs με τον εξής τρόπο: όταν το load είναι 0 τότε οι παράλληλες είσοδοι των lfsrs είναι οι αντίστοιχες παράλληλες έξοδοι τους, ενώ όταν το load είναι 1 τότε οι είσοδοι τους είναι οι έξοδοι αφού το lsb bit τους έχει γίνει xor με το εκάστοτε bit των Kc και Fn. Οι lfsrs σε κάθε περίπτωση χρονίζονται σύμφωνα με το μηχανισμό του ρολογιού που περιγράφεται παρακάτω.

Τέλος η σειριακή έξοδος η οποία προκύπτει από το xor των msb των lfsrs εισάγεται διαδοχικά στις μονάδες ser to par Block1 και Block2 έτσι ώστε να παραχθούν οι έξοδοι του αλγορίθμου BLOCK1 (114-bits) και BLOCK2 (114-bits).

Ο μηχανισμός του ρολογιού υλοποιείται στο component clk_unit, το οποίο περιλαμβάνει το clk_rule όπως περιγράφονται στις παρακάτω εικόνες 19 και 20. Οι είσοδοι c1,c2,c3 είναι αντίστοιχα τα bits χρονισμού δηλαδή τα outlfsr1(8), outlfsr2(10) και outlfsr3(10) και οι έξοδοι καθορίζουν τα ce των lfsrs με τον εξής τρόπο: εάν είναι ενεργοποιημένο το en_maj (ο κανόνας του ρολογιού) τότε οι έξοδοι του clk_rule cR1,cR2,cR3 είναι τα ce των lfsrs και εάν το en_maj είναι 0 τότε οι lfsrs προχωράνε κανονικά σύμφωνα με το γενικό ce.



Εικόνα 19 : Η διάταξη με πύλες του clk_rule



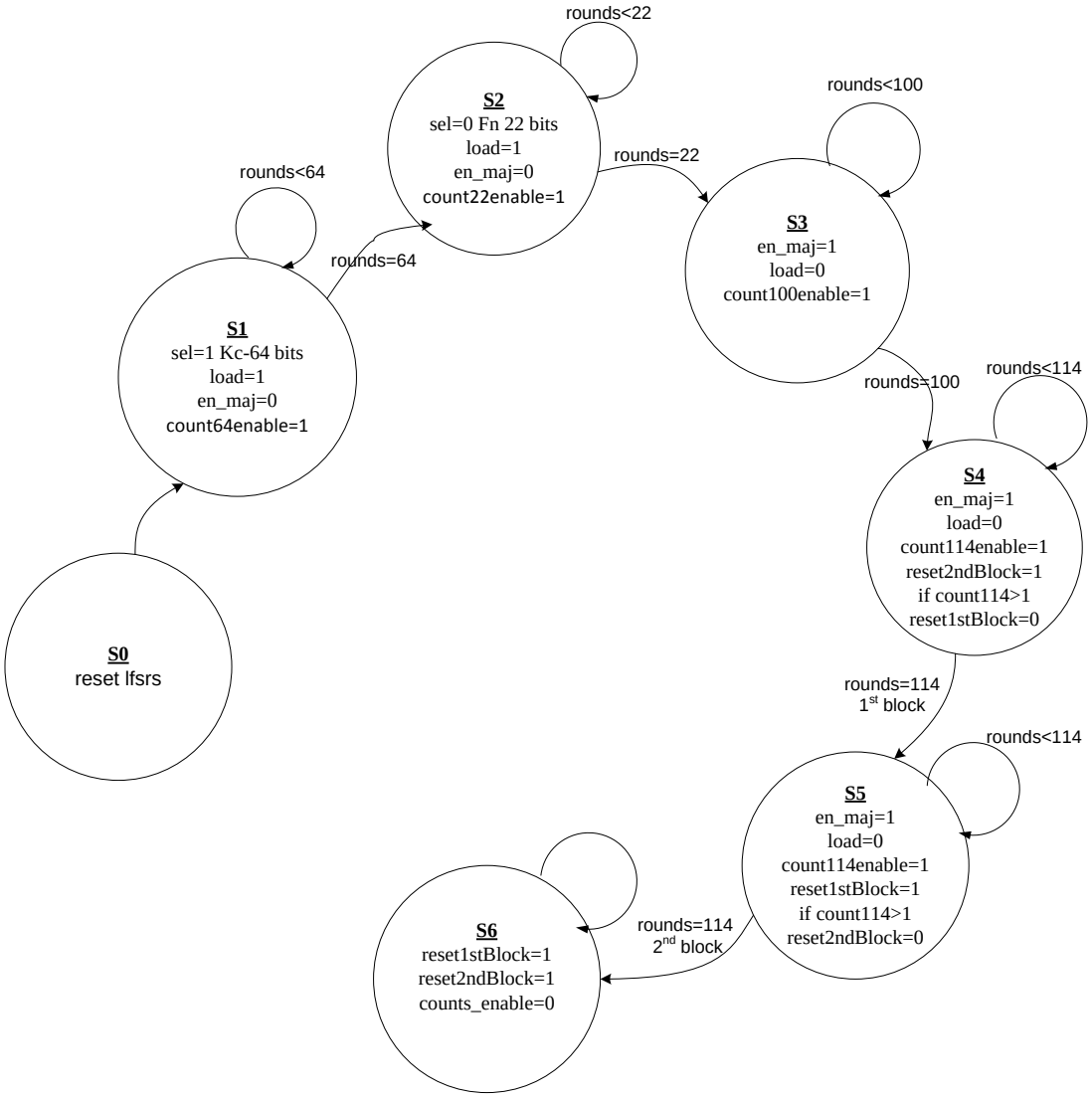
Εικόνα 20 : Ο μηχανισμός του ρολογιού clk_unit

Στη μονάδα control πραγματοποιούνται χρονικά τα βασικά βήματα του αλγορίθμου δηλαδή η εισαγωγή στο lsb bit των lfsrs το κλειδί και το frame number bit προς bit, ο χρονισμός 100 κύκλων με ενεργοποιημένο τον κανόνα της πλειοψηφίας και κατόπιν ο χρονισμός 2x114 κύκλους με εξόδους τα BLOCK1 και BLOCK2. Τα βήματα αυτά περιγράφονται στις διεργασίες της fsm στην παρακάτω εικόνα 21.

Η FPGA που χρησιμοποιήθηκε για το simulation είναι η Virtex-5 XC5VLX110T. Η σχεδίαση λειτουργεί σε συχνότητα ρολογιού 168.265 MHz (ελάχιστη περίοδος ρολογιού 5.943ns). Οι πόροι που καταναλώνει η σχεδίαση φαίνονται στον παρακάτω πίνακα.

	Used	Available	Utilization-Percentage
Number of Slice Registers:	416	69120	0%
Number of Slice LUTs:	375	69120	0%
Number of LUT Flip Flop pairs used:	597		
Number of fully used LUT-FF pairs:	194	597	32%
Number of bonded IOBs:	316	640	49%
Number of BUFG/BUFGCTRLs:	1	32	3%

Πίνακας 6 : Χρησιμοποίηση πόρων της FPGA



Εικόνα 21 : Τα βήματα της fsm

6.1.2 Σύγκριση με software

Η σύγκριση του χρόνου υπολογισμού των 114-bit τμημάτων σε hardware και σε software από τα ίδια Kc και Frame number, παρουσιάζεται παρακάτω. Για τον υπολογισμό του χρόνου σε software χρησιμοποιήθηκε μια υλοποίηση σε γλώσσα C (A pedagogical implementation of A5/1[28]) και ο Intel(R) VTune(TM) Performance Analyzer 9.1 σε περιβάλλον Windows XP με επεξεργαστή Pentium 4 3.01 GHz, και ο χρόνος είναι ο καλύτερος ύστερα από πολλαπλά τρεξίματα του VTune.
Η μεγάλη διαφορά στους χρόνους είναι αναμενόμενη καθώς ο αλγόριθμος A5/1 είναι σχεδιασμένος για υλοποιήσεις σε hardware (στον κινητό σταθμό).

Χρόνος software	Χρόνος hardware
14950,166113 μs	**2,995272 μs**

Πίνακας 7 : Σύγκριση χρόνων για την υλοποίηση του A5/1

6.2 Η υλοποίηση των πινάκων ουράνιου τόξου για τον A5/1

Σε αυτή την υποενότητα θα περιγραφεί ο τρόπος υπολογισμού ενός πίνακα ουράνιου τόξου με δύο κύριες υλοποιήσεις. Αρχικά όμως θα ακολουθήσει η διαδικασία υπολογισμού ενός EP από ένα SP σε μια αλυσίδα και κατόπιν ο παράλληλος υπολογισμός πολλών αλυσίδων που απαρτίζουν έναν πίνακα ουράνιου τόξου.

6.2.1 Βασική δομή της υλοποίησης μιας αλυσίδας

Η διαδικασία που ακολουθείται συνοπτικά για μια αλυσίδα είναι η εξής:

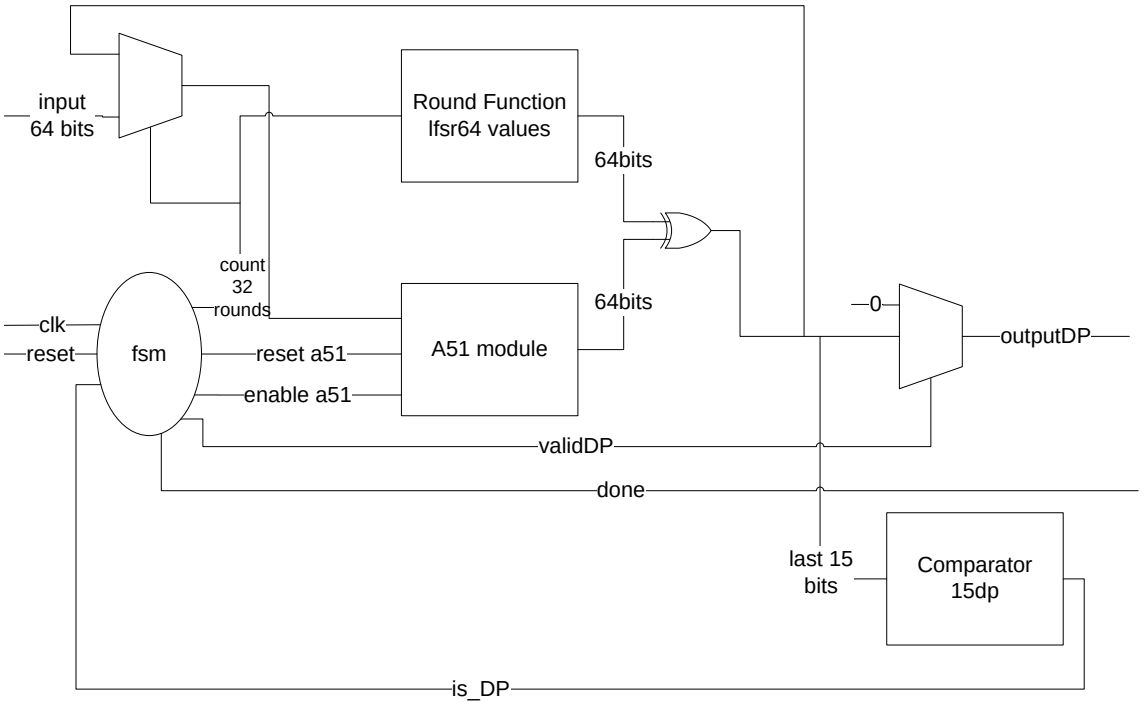
Το σημείο εκκίνησης-SP (εισάγεται στο A5/1 module) διαχωρίζεται σε 3 μέρη τα οποία τροφοδοτούνται στις lfsrs οι οποίες στη συνέχεια χρονίζονται για 64 κύκλους με ενεργοποιημένο το μηχανισμό της πλειοψηφίας. Από την πράξη xor στα msb bits προκύπτει ένα 64-bit σήμα.

Αυτό το σήμα γίνεται xor με την 64-bit τιμή που προέρχεται από τη συνάρτηση γύρου και που αντιστοιχεί στον πρώτο γύρο. Το αποτέλεσμα της πράξης xor ελέγχεται αν είναι διακριτό σημείο, δηλαδή στην προκειμένη περίπτωση αν τα lsb 15 bits είναι 0, μέσω της διαδικασίας ενός συγκριτή.

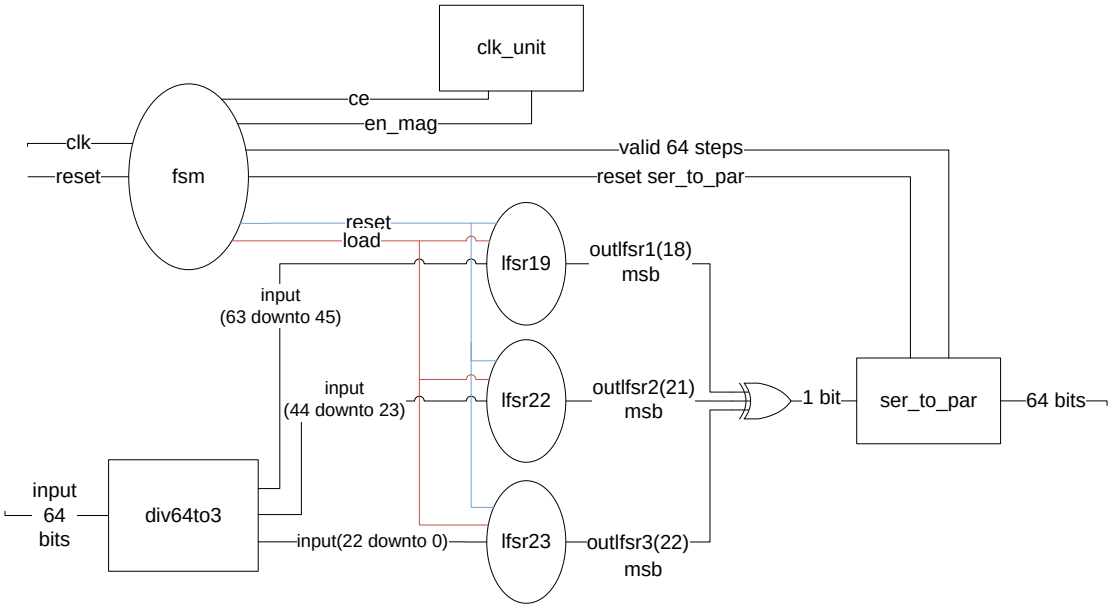
Στην περίπτωση που δεν είναι διακριτό σημείο τότε το αποτέλεσμα αυτό επανατροφοδοτείται σαν «νέο» SP στην αρχή της σχεδίασης και η ίδια διαδικασία επαναλαμβάνεται ώσπου το αποτέλεσμα της XOR με τη συνάρτηση γύρου να είναι διακριτό σημείο.

Μόλις βρεθεί το διακριτό σημείο τότε αυτό αποτελεί τον πρώτο σύνδεσμο της αλυσίδας ο οποίος επανατροφοδοτείται σαν SP αλλάζοντας όμως τη συνάρτηση γύρου να αντιστοιχεί στο δεύτερο χρώμα του ουράνιου τόξου. Η ίδια πορεία ακολουθείται για όλα τα προκύπτοντα αποτελέσματα ώσπου να τελειώσει η $32^{η}$ συνάρτηση γύρου οπότε ενεργοποιείται το σήμα done και το τελευταίο αποτέλεσμα αποτελεί το τελικό σημείο της αλυσίδας και αποθηκεύεται.

Η συνολική περιγραφή της υλοποίησης μιας αλυσίδας φαίνεται στην εικόνα 22 και οι εσωτερικές διεργασίες του A5/1 module στην εικόνα 23.



Εικόνα 22 : Διαδικασία υπολογισμού μιας αλυσίδας(chain)



Εικόνα 23 : Υλοποίηση του A5/1 module (clk_unit από εικόνα 20 ενότητα 6.1.1)

Ακολουθεί η αναλυτικότερη παρουσίαση της διαδικασίας υπολογισμού μιας αλυσίδας.

Η κύρια μονάδα της σχεδίασης είναι η μονάδα A5/1 module (Εικόνα 23). Σε αυτό το τμήμα υλοποιούνται οι lfsrs και οι 64 χρονισμοί τους με ενεργοποιημένη τη συνάρτηση πλειοψηφίας.
Περιλαμβάνει το clk_unit το οποίο υποστηρίζει τη δυνατότητα χρονισμού με το μηχανισμό της πλειοψηφίας και του κανονικού χρονισμού, την υπομονάδα διαχωρισμού του 64-bit σήματος, την υπομονάδα σύνθεσης της σειριακής εξόδου σε 64-bit έξοδο, τη μονάδα ελέγχου και την πύλη XOR.

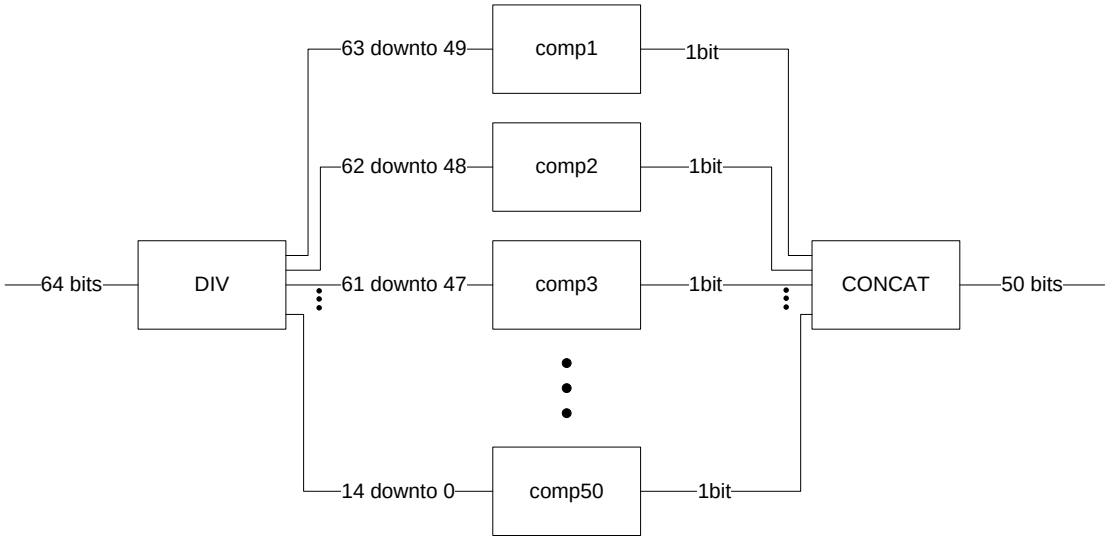
Η είσοδος-SP διαχωρίζεται σε 3 τμήματα μήκους 19,22 και 23 bits, και συνεπώς τα bits 63-45 της εισόδου εισάγονται στην lfsr19, τα bits 44-23 στην lfsr22 και τα bits 22-0 στην lfsr23. Οι R1, R2, R3 είναι Fibonacci lfsrs με πύλη ανάδρασης XOR. Στη συνέχεια η fsm ορίζει το χρονισμό των 64 κύκλων και κατόπιν τα bits που δημιουργούνται από την πράξη XOR στα msb bit των lfsrs συνθέτονται στην 64-bit έξοδο. Το σημείο εκκίνησης-SP αποτελεί την είσοδο του A5/1 module μόνο την πρώτη φορά ενώ όλες τις επόμενες ο πολυπλέκτης επιλέγει την έξοδο της XOR των 64 χρονισμών με τη συνάρτηση γύρου.

Η υπομονάδα Round Function (RF) υλοποιεί τη λειτουργία της συνάρτησης γύρου η οποία στην προκειμένη περίπτωση αποτελείται από την πράξη XOR με ψευδοτυχαίες τιμές που παράγονται από μια lfsr 64 bits. Συγκεκριμένα οι τιμές είναι το αποτέλεσμα του χρονισμού 64*N φορές μιας μεγίστου μήκους Fibonacci lfsr 64 bits με πύλη ανάδρασης XNOR, όπου N είναι ο γύρος στον οποίον βρισκόμαστε κάθε φορά ($1 \leq N \leq 32$). Η lfsr64 είχε αρχικά υλοποιηθεί να προχωράει παράλληλα με τα υπόλοιπα βήματα της σχεδίασης αλλά λόγω του γεγονότος ότι προσέθετε επιπλέον πόρους και χρόνο εκτέλεσης, προτιμήθηκε οι τιμές να προϋπολογιστούν και να αποθηκευτούν σε ένα array από το οποίο αντλείται η επιθυμητή τιμή σύμφωνα με έναν counter που σηματοδοτεί σε ποιο γύρο βρίσκεται η εκτέλεση.

Ο σκοπός της μονάδας σύγκρισης (comparator) είναι να ελέγξει αν κάθε υποψήφιο αποτέλεσμα, δηλαδή η έξοδος των 64 χρονισμών έχοντας γίνει XOR με την τιμή της συνάρτησης γύρου, είναι διακριτό σημείο. Συνεπώς τα τελευταία 15 bits του αποτελέσματος συγκρίνονται με τα μηδενικά bits και στην έξοδο του συγκριτή αποδίδεται 1 αν είναι διακριτό σημείο και 0 αν δεν είναι.

Στη συνέχεια η έξοδος του συγκριτή εισάγεται στην fsm έτσι ώστε να προχωρήσει η συνάρτηση γύρου στην επόμενη τιμή ή να επαναληφθεί η διαδικασία από την αρχή με αυτήν την έξοδο που «ακυρώθηκε» ώσπου να βρεθεί ένα διακριτό σημείο.

Τέλος πρέπει να αναφερθεί ότι στην υλοποίηση παρέχεται η δυνατότητα αλλαγής των βασικών παραμέτρων της λειτουργίας των διακριτών σημείων. Στις παραπάνω σχεδιάσεις διακριτό σημείο ορίζεται ένα σήμα με τα τελευταία 15 lsb bits μηδενικά. Η εικόνα 24 δείχνει τη σχεδίαση της μονάδας του συγκριτή έτσι ώστε διακριτό σημείο να θεωρείται ένα σήμα με 15 διαδοχικά μηδενικά bits σε οποιαδήποτε θέση στο σήμα. Το σήμα «σπάει» σε 50 κομμάτια, το καθένα από τα οποία τροφοδοτείται σε ένα συγκριτή. Οι έξοδοι των συγκριτών συνθέτονται σε ένα σήμα μήκους 50 bits και εάν το σήμα αυτό δεν είναι μηδενικό τότε το αρχικό σήμα αποτελεί διακριτό σημείο.



Εικόνα 24 : Η μονάδα του συγκριτή για οποιαδήποτε 15 διαδοχικά μηδενικά bits

6.2.2 Βασική δομή της υλοποίησης όλων των αλυσίδων

Για τον παράλληλο υπολογισμό πολλών αλυσίδων που απαρτίζουν έναν πίνακα ουράνιου τόξου έχουν υλοποιηθεί δύο διαφορετικές σχεδιάσεις:
1) με σήματα εξόδου
2)με αποθήκευση σε μνήμη

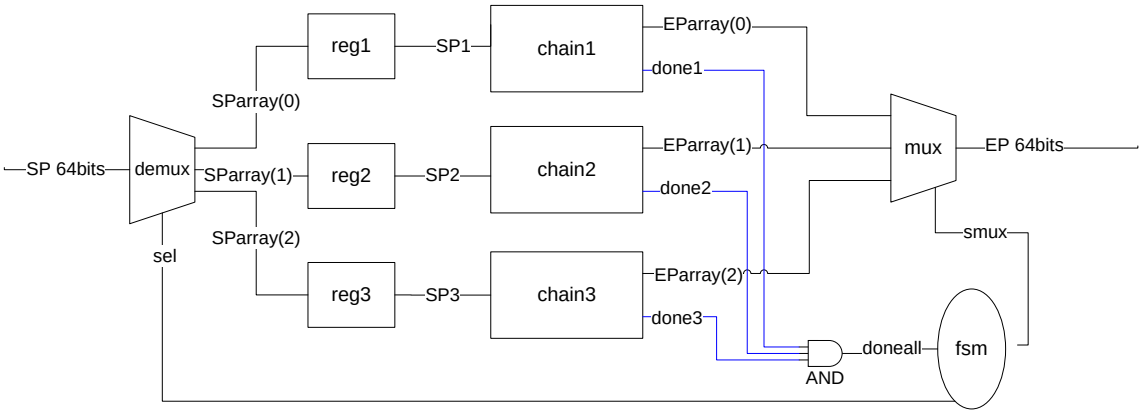
Η FPGA που χρησιμοποιήθηκε για τις 2 υλοποιήσεις είναι η Virtex5 XC5VLX330T.

6.2.2.1 Υλοποίηση με σήματα εισόδου και εξόδου

Σε αντίθεση με τη χρησιμοποίηση μνήμης αυτή η υλοποίηση δημιουργήθηκε για να αντιμετωπίσει τον τεράστιο χρόνο που καταναλωνόταν σε όλα τα στάδια (synthesize, place and route, simulation).

Το σχεδιάγραμμα της συνολικής υλοποίησης όλων των αλυσίδων με την προϋπόθεση εισαγωγής των SP και την εξαγωγή των EP ως σήματα παρουσιάζεται στην παρακάτω εικόνα 25.

Με τη φράση συνολική υλοποίηση των αλυσίδων αντικατοπτρίζεται η παράλληλη επεξεργασία ενός αριθμού αλυσίδων. Ο μέγιστος αριθμός των αλυσίδων καθορίζεται από τους πόρους της εκάστοτε FPGA.



Εικόνα 25 : Πίνακας ουράνιου τόξου με 3 ενδεικτικές αλυσίδες με σήματα εξόδου

Με την πρόσθεση του γεγονότος ότι τις διεργασίες εισόδου και εξόδου αναλαμβάνει ένας από-πολυπλέκτης και ένας πολυπλέκτης αντίστοιχα, αντιμετωπίζεται το πρόβλημα των IOBs.

Η λειτουργία του από-πολυπλέκτη είναι η εξής: η είσοδος του είναι ένα σήμα εισόδου 64 bits κάθε φορά, η έξοδος του ένα array των 64-bit με μέγεθος όσα στιγμιότυπα των αλυσίδων έχουμε ορίσει, και το sel στη δεκαδική αναπαράστασή του αντιστοιχεί στον αριθμό των στιγμιοτύπων. Δεδομένης μιας εισόδου και ενός sel, ο πολυπλέκτης εξάγει στη θέση του array που αντιστοιχεί στη δεκαδική τιμή του sel την 64-bit είσοδο. Για παράδειγμα όταν έχουμε 128 παράλληλες αλυσίδες τότε το sel είναι μήκους 6 downto 0, ξεκινώντας από το 0000000. Για τη μηδενική τιμή του select η πρώτη είσοδος αντιστοιχίζεται στο array(0), η 0000001 αντιστοιχίζεται στο array(1) στον επόμενο κύκλο και ούτω καθεξής έως την τιμή του select 1111111 που βγάζει την αντίστοιχη έξοδο στο array(127). Οι κύκλοι που καταναλώνονται είναι υπερβολικά αμελητέοι αν συγκριθούν με το συνολικό αριθμό κύκλων της εκτέλεσης (βλ ενότητα 6.3.1).

Οι τιμές του array στη συνέχεια τροφοδοτούνται στον αντίστοιχο καταχωρητή οι οποίες εισάγονται ταυτόχρονα στις υπομονάδες chain που έχουν περιγραφεί παραπάνω στην ενότητα 6.2.1. Οι καταχωρητές επίσης εξυπηρετούν την μείωση του

datapath για το synthesize καθώς μειώνεται ο χρόνος που αναλίσκεται για να γίνουν route τα σήματα.

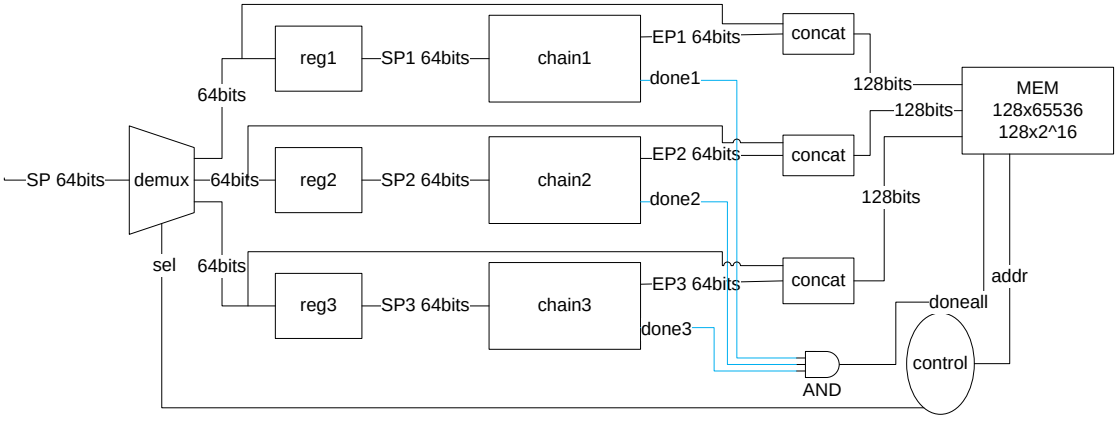
Από την κάθε υπομονάδα chain εξάγεται το σήμα done που ενεργοποιείται όταν βρεθεί το τελικό σημείο της αλυσίδας. Όταν όλα τα σήματα done ενεργοποιηθούν τότε με τη σειρά του ενεργοποιείται το σήμα doneall το οποίο εισάγεται στην fsm και ξεκινάνε αυτόματα οι διεργασίες του πολυπλέκτη.

Ο πολυπλέκτης παίρνει ως είσοδο τα τελικά σημεία των αλυσίδων με μορφή array και εξάγει σε κάθε κύκλο μετά την ενεργοποίηση του doneall όλα τα τελικά σημεία ακολουθώντας τη σειρά των αλυσίδων (και όχι τη σειρά που ολοκληρώθηκε η καθεμιά). Το select του πολυπλέκτη αυξάνεται μέσα στην fsm.

Με αυτή τη μέθοδο τα SP εισάγονται αυξητικά, δηλαδή ορίζοντας ένα αρχικό SP τα υπόλοιπα αποτελούν τιμές μιας αριθμητικής προόδου που έχουμε ορίσει. Στη συγκεκριμένη υλοποίηση έχουμε ορίσει την πρόσθεση +1 στην προηγούμενη τιμή του SP. Ο λόγος είναι ο εξής: όταν κατασκευάζουμε πολλούς διαφορετικούς πίνακες αυτοί πρέπει να καλύπτουν το μεγαλύτερο δυνατό εύρος πιθανών SP, και υποθέτοντας ότι ξεκινάμε τον επόμενο πίνακα μετά το τέλος του προηγούμενου τότε θεωρητικά οι πίνακες θα καλύπτουν όλο το δυνατό εύρος των SP.

6.2.2.2 Υλοποίηση με μνήμη

Αποτελεί την κύρια υλοποίηση των πινάκων ουράνιου τόξου δεδομένου ότι τα τελικά σημεία αποθηκεύονται στη μνήμη ως ζευγάρια αρχικού σημείου-τελικού σημείου (SP-EP).
Το σχεδιάγραμμα της σχεδίασης παρουσιάζεται στην παρακάτω εικόνα 26.



Εικόνα 26 : Πίνακας ουράνιου τόξου με 3 ενδεικτικές αλυσίδες με μνήμη

Η διαδικασία που ακολουθείται είναι η ίδια με αυτή που έχει περιγραφεί στην ενότητα 6.2.2.1 με τη διαφορά ότι τα τελικά σημεία συνενώνονται με το αντίστοιχο αρχικό τους και αποθηκεύονται στη μνήμη. Το we της μνήμης είναι το σήμα doneall.

Η μνήμη που χρησιμοποιήθηκε σε αυτή την υλοποίηση είναι μια Block Memory η οποία παράχθηκε με coregen και συγκεκριμένα με Block Memory Generator v2.8.
Η Block RAM επιλέχθηκε σε σχέση με την Distibuted RAM λόγω του γεγονότος ότι οι Block RAMs είναι καταλληλότερες για μεγάλο αριθμό καταχωρήσεων. Έχει βάθος 2^{16} = 65536 και πλάτος 128 bits, αφού η κάθε καταχώρηση είναι SP(64 bits)-EP(64 bits).

6.2.2.3 Παραλληλισμός για την υλοποίηση με μνήμες

Σύμφωνα με τους πόρους της σχεδίασης με μνήμη που προκύπτουν από τα αποτελέσματα του synthesize και του place and route, είναι δυνατός ο παράλληλος υπολογισμός περισσοτέρων από μιας αλυσίδων. Συνεπώς υλοποιήθηκαν διάφοροι αυξανόμενοι αριθμοί παράλληλων αλυσίδων και συγκεκριμένα 10, 128, 181, 200, 250, 300 και 350.
Στις 350 αλυσίδες ο αριθμός των Slice LUTs είχε ξεπεράσει το 100% οπότε έγινε προσαρμογή για 345 αλυσίδες οι οποίες είναι ο μέγιστος αριθμός που χωράει σε μια FPGA Virtex5 XC5VLX330T.

Η χρησιμοποίηση των πόρων της FPGA για τις σχεδιάσεις με όλους τους υλοποιημένους αριθμούς παράλληλων στιγμιοτύπων παρουσιάζονται στον παρακάτω πίνακα.

# of instances	Slice Registers	Slice LUTs	LUT Flip Flop pairs	Fully used LUT-FF pairs	Bonded IOBs	Block RAM/FIFO	BUFG/BUFGCTRLs
1	488/207360 (0%)	507/207360 (0%)	803	192/803 (23%)	131/960 (13%)	2/324 (0%)	4/32 (12%)
10	4955/207360 (2%)	6791/207360 (3%)	9073	2673/9073 (29%)	136/960 (14%)	228/324 (70%)	16/32 (50%)
128	61566/207360 (29%)	77775/207360 (37%)	106862	32479/106862 (30%)	139/960 (14%)	228/324 (70%)	16/32 (50%)
181	86922/207360 (41%)	109371/207360 (52%)	150513	45780/150513 (30%)	140/960 (14%)	228/324 (70%)	16/32 (50%)
200	96001/207360 (46%)	120065/207360 (57%)	165399	50667/165399 (30%)	140/960 (14%)	228/324 (70%)	16/32 (50%)
250	122802/207360 (59%)	154103/207360 (74%)	212225	64680/212225 (30%)	140/960 (14%)	228/324 (70%)	16/32 (50%)
300	143869/207360 (69%)	180535/207360 (87%)	248621	75783/248621 (30%)	141/960 (14%)	228/324 (70%)	16/32 (50%)
345	165386/207360 (79%)	206911/207360 (99%)	285007	87290/285007 (30%)	141/960 (14%)	228/324 (70%)	16/32 (50%)

Πίνακας 8 : Χρησιμοποίηση πόρων της FPGA για την υλοποίηση με μνήμη

Το synthesize και το place and route πραγματοποιήθηκαν στον server iraklis αφού ένα απλό PC δεν είχε επαρκή μνήμη. Επίσης πρέπει να αναφερθεί ότι ακόμα και το simulation για τον υπολογισμό μιας μόνο αλυσίδας δεν ήταν δυνατό λόγω της μνήμης που απαιτούνταν ούτε στο PC αλλά ούτε και στον server iraklis χρησιμοποιώντας τον ISE simulator. Το πρόβλημα λύθηκε με τη χρήση του Modelsim.

6.3 Αποτελέσματα

Σε αυτή την ενότητα παρουσιάζονται συγκριτικά οι χρόνοι υλοποίησης των αλυσίδων από το simulation και συγκρίνονται με τους χρόνους υλοποίησης σε software, και για τις δύο υλοποιήσεις των πινάκων με σήματα και μνήμη.

6.3.1 Για την υλοποίηση με σήματα εισόδου και εξόδου

Όλα τα αποτελέσματα είναι για την FPGA Virtex5 XC5VLX330T. Ο μέγιστος αριθμός παράλληλων στιγμιοτύπων για την $1^{η}$ υλοποίηση είναι 320 και όχι 345 όπως στη δεύτερη με μνήμη.

	Minimum Period (ns)	**Maximum Frequency(MHz)**
1 instance	5.354	186.776
10 instances	7.474	133.797
128 instances	8.729	114.561
181 instances	9.614	104.015
200 instances	9.563	104.570
256 instances	10.219	97.857
300 instances	10.283	97.248
320 instances	13.021	76.799

Πίνακας 9 : Μέγιστη συχνότητα για την $1^{η}$ υλοποίηση

Οι χρόνοι εκτέλεσης σε software υπολογίστηκαν χρησιμοποιώντας τον Intel(R) VTune(TM) Performance Analyzer 9.1 σε περιβάλλον Windows XP με επεξεργαστή Pentium 4 3.01 GHz. Οι κώδικες είναι στη γλώσσα C++ και είναι προσαρμοσμένοι από το A5/1 project [36]. Δεν διαθέτουν καμία εντολή εκτύπωσης του αποτελέσματος στην οθόνη για να έχουν όσο το δυνατόν μικρότερο χρόνο εκτέλεσης. Οι αναφερόμενοι χρόνοι είναι οι καλύτεροι χρόνοι σε κάθε περίπτωση ύστερα από πολλαπλά τρεξίματα του VTune. Στον πίνακα 10 παρουσιάζονται οι χρόνοι για 10 διαφορετικά SP για την υλοποίηση μιας αλυσίδας.

1 instance για 10 διαφορετικά SP	Συνολικός χρόνος Vtune σε **seconds**	Συνολικός χρόνος FPGA σε **ms**
#0	6,258139	358.878620
#1	9,139535	538.826560
#2	5,919269	340.407320
#3	8,693023	508.576460
#4	6,300997	360.110040
#5	6,177409	353.738780
#6	6,172425	353.471080
#7	8,852492	513.020280
#8	8,597342	503.972020
#9	5,931229	347.099820

Πίνακας 10 : Σύγκριση χρόνων για μια αλυσίδα για την 1η υλοποίηση

Στον πίνακα 11 παρουσιάζονται οι χρόνοι εκτέλεσης για τους αριθμούς στιγμιοτύπων 10,128,181,200,256,300 και 320. Σε αυτές τις σχεδιάσεις, όσον αφορά την πλευρά του hardware, οι κύκλοι έως ότου να ολοκληρωθεί η διαδικασία ισούνται με τους κύκλους της εκτέλεσης της μεγαλύτερης σε μήκος αλυσίδας.

Για τα 10 παράλληλα στιγμιότυπα τροφοδοτήθηκαν τα ίδια τυχαία και διαφορετικά SP με τον πίνακα 9 εκ των οποίων μεγαλύτερο μήκος αλυσίδας έχει το δεύτερο (με μήκος 100640000 κύκλους).

Για τους υπόλοιπους αριθμούς στιγμιοτύπων ξεκινώντας από το #9 προστέθηκαν τα SP+1, δηλαδή #10 = #9+1, #11 = #10+1, #12 = #11+1 και ούτω καθεξής. Ως τα 128 στιγμιότυπα έχουμε συναντήσει τη μεγαλύτερη σε μήκος αλυσίδα η οποία είναι η 122η αλυσίδα και καθότι για τις υπόλοιπες σχεδιάσεις χρησιμοποιούνται οι ίδιοι αριθμοί επαυξημένοι συνεχώς κατά 1, αυτή η αλυσίδα παραμένει η μεγαλύτερη έως τα 320 στιγμιότυπα. Το μήκος αυτής της αλυσίδας είναι 121599540 κύκλοι.
Σημειώνεται ότι η μικρότερη σε μήκος αλυσίδα ολοκληρώνεται σε 45614694 κύκλους.

	Συνολικός χρόνος Vtune σε **seconds**	Συνολικός χρόνος FPGA σε **ms**
10 instances	71,440864	752.183360
128 instances	870,004983	1061,442384
181 instances	1217,800664	1169,057977
200 instances	1353,663787	1162,856401
256 instances	1670,042193	1242,625699
300 instances	2036,733887	1250,408069
320 instances	2132,514618	1583,347610

Πίνακας 11 : Σύγκριση χρόνων για 10,128,181,200,256,300 και 320 παράλληλες αλυσίδες για την 1η υλοποίηση

6.3.2 Για την υλοποίηση με μνήμη

Ισχύουν οι ίδιες συνθήκες που περιγράφηκαν στην ενότητα 6.3.1. Η χρησιμοποιούμενη FPGA είναι η Virtex5 XC5VLX330T. Ο μέγιστος αριθμός παράλληλων στιγμιοτύπων για την υλοποίηση με μνήμη είναι 345.

	Minimum Period (ns)	**Maximum Frequency(MHz)**
1 instance	5.599	178.603
10 instances	6.105	163.800
128 instances	6.721	148.787
181 instances	6.848	146.028
200 instances	6.552	152.625
256 instances	6.681	149.678
300 instances	6.592	151.699
345 instances	6.832	146.370

Πίνακας 12 : Μέγιστη συχνότητα για τη βασική υλοποίηση με μνήμη

Στους πίνακες 13 και 14 παρουσιάζονται οι χρόνοι για όλους τους αριθμούς των παράλληλων στιγμιοτύπων έως 345 και η απόδοση του καθενός στην υλοποίηση στην FPGA σε σύγκριση με το software.
Έχουν τροφοδοτηθεί τα ίδια SP με την ενότητα 6.3.1 και ισχύει ότι η μεγαλύτερη σε μήκος αλυσίδα για τα 345 στιγμιότυπα είναι η $122^{η}$ αλυσίδα με μήκος 121599540 κύκλους.

1 instance για 10 διαφορετικά SP	Συνολικός χρόνος Vtune σε **seconds**	Συνολικός χρόνος FPGA σε **ms**	Απόδοση (x φορές)
#0	6,258139	375,300970	16,67
#1	9,139535	563,483360	16,22
#2	5,919269	355,984420	16,63
#3	8,693023	531,849010	16,34
#4	6,300997	376,588740	16,73
#5	6,177409	369,925930	16,70
#6	6,172425	369,645980	16,70
#7	8,852492	536,496180	16,50
#8	8,597342	527,033870	16,31
#9	5,931229	362,983170	16,34

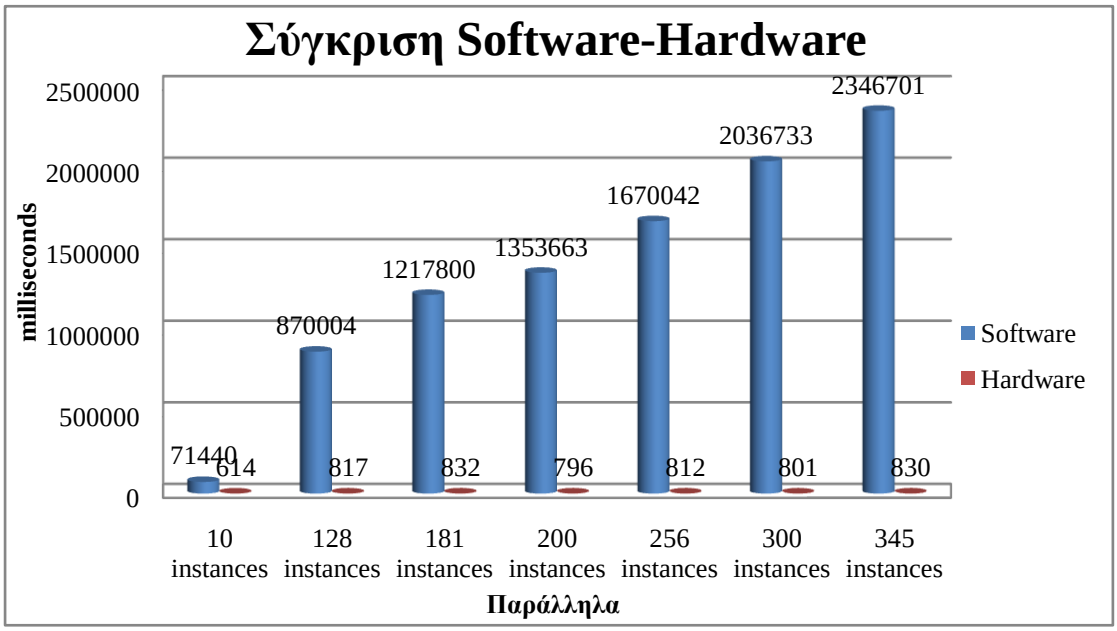
Πίνακας 13 : Σύγκριση χρόνων για 1 αλυσίδα για την υλοποίηση με μνήμη

	Συνολικός χρόνος Vtune σε **seconds**	Συνολικός χρόνος FPGA σε **ms**	Απόδοση (x φορές)
10 instances	71,440864	614,407200	116,28
128 instances	870,004983	817,270508	1064,53
181 instances	1217,800664	832,713649	1462,45
200 instances	1353,663787	796,720186	1699,05
256 instances	1670,042193	812,406526	2055,67
300 instances	2036,733887	801,584167	2540,89
345 instances	2346,701661	830,768057	2824,74

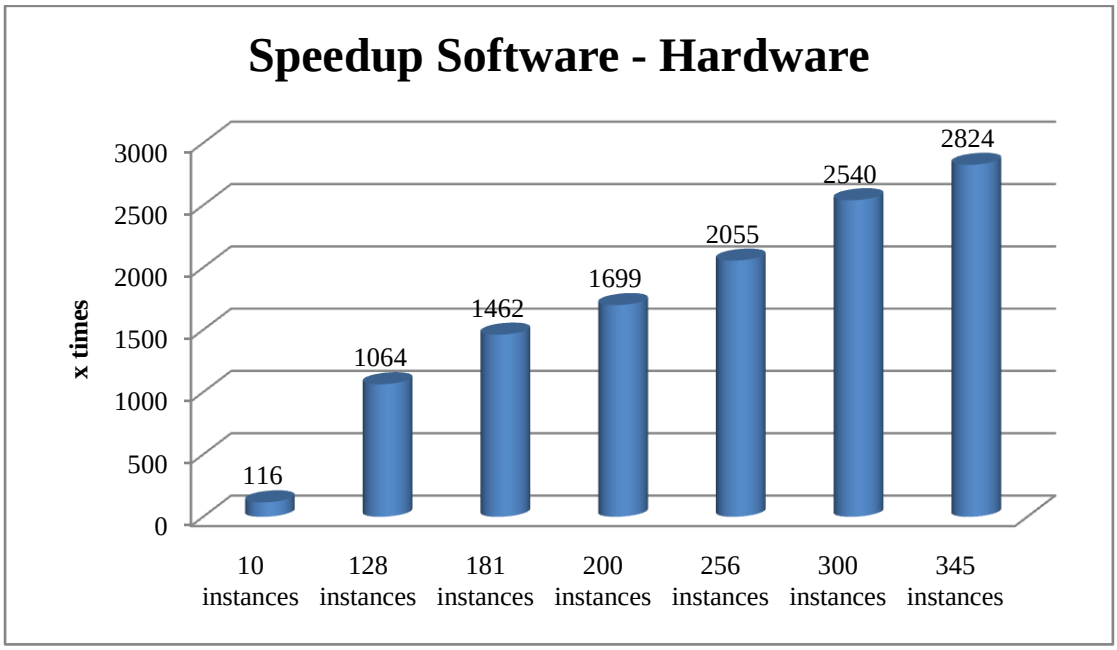
Πίνακας 14 : Σύγκριση χρόνων για 10,128,181,200,256,300 και 345 παράλληλες αλυσίδες για την υλοποίηση με μνήμη

Ο κύκλος ρολογιού σε αυτή την υλοποίηση αλλάζει ελάχιστα όπως παρατηρείται και στον πίνακα 12 και συνεπώς η απόδοση αυξάνεται σε μεγάλο βαθμό όσο αυξάνεται ο αριθμός των παράλληλων στιγμιοτύπων. Στα 345 παράλληλα στιγμιότυπα η υλοποίηση σε FPGA των πινάκων ουράνιου τόξου είναι 2824x φορές γρηγορότερη από την εκτέλεση σε software για τα ίδια SP. Στο γράφημα 2 παρουσιάζεται η επιτάχυνση που επιτυγχάνεται μέσω του hardware.

Στο γράφημα 1 παρουσιάζεται η σύγκριση των παραπάνω χρόνων. Ο χρόνος του hardware φαίνεται ότι είναι πολύ μικρός σε σχέση με το software.
Πράγματι για 10 στιγμιότυπα ο χρόνος ολοκλήρωσης σε hardware είναι 614 ms ενώ σε software 1,19 λεπτά, για τα 200 στιγμιότυπα σε hardware είναι 796 ms σε σύγκριση με τα 22,56 λεπτά και για 345 στιγμιότυπα υπάρχει τεράστια διαφορά ανάμεσα στα 830 ms του hardware έναντι των 39,1 λεπτών του software.



Γράφημα 1: Σύγκριση χρόνων υπολογισμού των πινάκων σε hardware-software



Γράφημα 2: Speedup του υπολογισμού των πινάκων σε hardware σε σχέση με το software

Κεφάλαιο 7: Συμπεράσματα και μελλοντική εργασία

7.1 Συμπεράσματα

Σε αυτή τη διπλωματική εργασία παρουσιάσαμε μια αρχιτεκτονική FPGA για την υλοποίηση πινάκων ουράνιου τόξου για το «σπάσιμο» του αλγορίθμου A5/1. Η επίθεση με την ανταλλαγή χρόνου-μνήμης στοχεύει σε 64 διαδοχικά bits ενός κρυπτογραφημένου μηνύματος και ανακτά την εσωτερική κατάσταση του αλγορίθμου. Δεδομένης της εσωτερικής κατάστασης καθίσταται δυνατή η αποκρυπτογράφηση ολόκληρου του μηνύματος και κατ'επέκταση όλης της συνομιλίας. Παρατηρήσαμε ότι η δημιουργία των πινάκων ουράνιου τόξου στην FPGA είναι σε τεράστιο βαθμό ταχύτερη από τη δημιουργία τους σε PC. Για τον υπολογισμό 345 παράλληλων αλυσίδων απαιτούνται 830 ms στο hardware έναντι 39,1 λεπτών στο software.

7.2 Μελλοντική εργασία

Σαν μελλοντική εργασία στην παρούσα διπλωματική εργασία αφήνεται η εκτέλεση στην FPGA Virtex5 XC5VLX330T.

Επίσης προτείνονται οι παρακάτω βελτιστοποιήσεις:

Για την υλοποίηση όλων των πινάκων που καλύπτουν το συνολικό εύρος των δυνατών εσωτερικών καταστάσεων προτείνεται η μείωση του εύρους σε αυτά τα SP τα οποία είναι δυνατά να προκύψουν μετά το πέρας των 100 χρονισμών, αφού από τις 2^{64} πιθανές καταστάσεις, οι 2^{3} είναι αδύνατες μετά από τους 100 χρονισμούς.

Οι ολοκληρωμένοι πίνακες πρέπει να μεταφερθούν στο δίσκο (ή σε εξωτερική μνήμη) από την FPGA και να ταξινομηθούν έτσι ώστε να είναι δυνατή η αναζήτηση ενός τελικού σημείου γρηγορότερα και η εύρεση του αντίστοιχου αρχικού. Ο χρόνος ταξινόμησης εκτιμάται πάρα πολύ μικρός όπως και η διαδικασία αναζήτησης. Στο συνολικό χρόνο αυτής της επίθεσης με ανταλλαγή χρόνου-μνήμης κυριαρχεί το στάδιο προϋπολογισμού με τον υπολογισμό των πινάκων, ενώ το τελικό στάδιο της αναζήτησης είναι συγκριτικά αμελητέο. Η αναζήτηση μπορεί να υλοποιηθεί σε software ή ακόμα και σε hardware για να ελαττωθεί ακόμα περισσότερο ο απαιτούμενος χρόνος.
Επιπρόσθετα μπορεί να γίνει σύνθεση των μικρών πινάκων σε μεγαλύτερους έτσι ώστε να μειώνεται ο χρόνος των προσβάσεων στο δίσκο.

Βιβλιογραφία

[1] Xilinx Virtex-5 FPGA User Guide, UG190 (v5.3) May 17, 2010
http://www.xilinx.com/support/documentation/user_guides/ug190.pdf

[2] Privateline telecommunications expertise,GSM History
http://www.privateline.com/mt_gsmhistory/02_gsm_history/

[3] Lawrence Harte, Richard Levine, Geoff Livingston.1999. *GSM Superphones*. McGraw-Hill Telecommunications

[4] Asha Mehrotra.1997. *GSM System Engineering* Mobile Communications Series. Artech House Publishers

[5] European Conference of Postal and Telecommunications Administrations
http://www.cept.org/

[6] Friedhelm Hillebrand.2002.*GSM and UMTS: The Creation of Global Mobile Communication*.John Wiley & Sons Ltd

[7] Χαράλαμπος Μανιφάβας.Σημειώσεις χειμερινού εξαμήνου Information Systems Security 2007-2008.Πανεπιστήμιο Κρήτης

[8] The 3rd Generation Partnership Project http://www.3gpp.org/

[9] Annabel Z. Dodd. 2002. *The Essential Guide to Telecommunications* Third Edition. Prentice Hall PTR

[10] Paulo S. Pagliusi.2002.*A Contemporary Foreword on GSM Security*.InfraSec '02 Proceedings of the International Conference on Infrastructure Security, Volume 2437/2002.pages 129-144. Springer-Verlag London, UK

[11] Heikki Kaaranen, Ari Ahtiainen, Lauri Laitinen, Siamäk Naghian, Valtteri Niemi. 2005. *UMTS Networks: Architecture, Mobility and Services,* Second edition. John Wiley & Sons Ltd

[12] http://www.gsm-security.net/

[13] ETSI TS 143 020 V4.0.0 (2000-11) Technical Specification.*Digital cellular telecommunications system (Phase 2+),Security-related network functions.*
http://www.3gpp.org/ftp/specs/html-info/43020.htm

[14] ΕΕΤΤ Εθνική Επιτροπή Τηλεπικοινωνιών και Ταχυδρομείων.
http://www.eett.gr/opencms/opencms/EETT

[15] Eli Biham, Orr Dunkelman.2000.*Cryptanalysis of the A5/1 GSM Stream Cipher*. PROGRESS IN CRYPTOLOGY -INDOCRYPT 2000, Lecture Notes in Computer Science,Volume 1977/2000,43-51

[16] Imran Erguler, Emin Anarim. 2005.*A Modified Stream Generator for the GSM*

Encryption Algorithms A5/1 and A5/2.13th European Signal Processing Conference (EUSIPCO 2005),Sep. 4-8,Antalya, Turkey

[17] 3GPP TS 55.216 V6.2.0 (2003-09) Technical Specification. *Specification of the A5/3 Encryption Algorithms for GSM and ECSD, and the GEA3 Encryption Algorithm for GPRS, Document 1: A5/3 and GEA3 Specifications (Release 6)*

[18] 3GPP TS 35.202 V9.0.0(2009-12) Technical Specification. *Specification of the 3GPP Confidentiality and Integrity Algorithms, Document 2: KASUMI Specification (Release 9)*

[19] Orr Dunkelman, Nathan Keller, Adi Shamir.2010.*A Practical-Time Related-Key Attack on the KASUMI Cryptosystem Used in GSM and 3G Telephony*.Proceedings of CRYPTO 2010, Lecture Notes in Computer Science 6223, pp. 393-410, Springer

[20] Elad Barkan, Eli Biham, Nathan Keller.2006-07.*Instant Ciphertext-Only Cryptanalysis of GSM Encrypted Communication*. JOURNAL OF CRYPTOLOGY Volume 21, Number 3, 392-429

[21] Mitsuru Matsui.1997. *New Block Encryption Algorithm MISTY*. 4th International Workshop, FSE '97, LNCS 1267, pp. 64–74

[22] Claude Shannon. *1949. Communication Theory of Secrecy Systems*. Bell System Technical Journal, vol. 28, page 656–715.

[23] Whitfield Diffie , Martin E Hellman. 1976. *New Directions in Cryptography.* IEEE Transactions on Information Theory, 22(6), pp. 644-654

[24] Valtteri Niemi, Kaisa Nyberg.2003.*UMTS Security*.John Wiley & Sons,Ltd

[25] S. B. Xu, D. K. He, X. M. Wang.1994. *An Implementation of the GSM General Data Encryption Algorithm A5*.CHINACRYPT ’94, Xidian, China, pp. 287-291 (Chinese)

[26] R. Anderson, M. Roe.1994. Subject: A5 (Was: HACKING DIGITAL PHONES). Message at sci.crypt, alt.security, uk.telecom. http://groups.google.com/group/uk.telecom/msg/ba76615fef32ba32

[27]Jovan Dj. Golić. 1997. *Cryptanalysis of Alleged A5 Stream Cipher*. ADVANCES IN CRYPTOLOGY -EUROCRYPT ’97,Lecture Notes in Computer Science, Volume 1233/1997, 239-255. Springer-Verlag

[28] Marc Briceno, Ian Goldberg, David Wagner. 1998-1999. *A pedagogical implementation of the GSM A5/1 and A5/2 “voice privacy” encryption algorithms*. http://www.scard.org/gsm/a51.html

[29] Alex Biryukov, Adi Shamir, David Wagner.2000. *Real Time Cryptanalysis of A5/1 on a PC*. Fast Software Encryption Workshop 2000, New York City.
Alex Biryukov, Adi Shamir. 1999. *Real Time Cryptanalysis of the Alleged A5/1 on a PC (preliminary draft)*

[30] Jörg Keller, Birgit Seitz.2001. *A Hardware-Based Attack on the A5/1 Stream Cipher*

[31] Patrik Ekdahl,Thomas Johansson.2003. Another Attack on A5/1. Information Theory, IEEE Transactions.Volume:49 Issue:1,pages 284 – 289

[32] Alexander Maximov, Thomas Johansson, Steve Babbage.2005. *An Improved Correlation Attack on A5/1*. SELECTED AREAS IN CRYPTOGRAPHY Lecture Notes in Computer Science, Volume 3357/2005, 1-18

[33] Elad Barkan, Eli Biham. 2006. *Conditional Estimators: An Effective Attack on A5/1*. SELECTED AREAS IN CRYPTOGRAPHY Lecture Notes in Computer Science, Volume 3897/2006, 1-19

[34] Alex Biryukov, Adi Shamir.2000. *Cryptanalytic Time/Memory/Data Tradeoffs for Stream Ciphers*. ADVANCES IN CRYPTOLOGY - ASIACRYPT 2000 Lecture Notes in Computer Science, Volume 1976/2000, 1-13

[35] A. Bogdanov, T. Eisenbarth, A. Rupp. 2007. *A Hardware-Assisted Realtime Attack on A5/2 without Precomputations*. In 9th Workshop on Cryptographic Hardware and Embedded Systems - CHES 2007, volume 4727 of Lecture Notes in Computer Science, pages 394– 412. Springer-Verlag

[36] A5/1 Security Project.2009
http://reflextor.com/trac/a51/wiki και http://opensource.srlabs.de/ και http://events.ccc.de/congress/2009/Fahrplan/attachments/1519_26C3.Karsten.Nohl.GSM.pdf
http://events.ccc.de/congress/2009/Fahrplan/events/3654.en.html

[37] S. Babbage.1995. *A Space/Time Trade-off in Exhaustive Search Attacks on Stream Ciphers*. European Convention on Security and Detection, IEE Conference Publication, 408.

[38] Timo Gendrullis, Martin Novotný, Andy Rupp.2008. *A Real-World Attack Breaking A5/1 within Hours*. In 10th Workshop on Cryptographic Hardware and Embedded Systems - CHES 2008, volume 5154 of Lecture Notes in Computer Science, pages 266–282. Springer-Verlag

[39] Sandeep Kumar, Christof Paar, Jan Pelzl, Gerd Pfeiffer, Manfred Schimmler. 2006. *Breaking Ciphers with COPACOBANA –A Cost-Optimized Parallel Code Breaker*. CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS - CHES 2006, Lecture Notes in Computer Science, Volume 4249/2006, 101-118

[40] http://www.sciengines.com/copacobana/index.html

[41] Eli Biham, Orr Dunkelman, Nathan Keller.2005. *A Related-Key Rectangle Attack on the Full KASUMI*. ADVANCES IN CRYPTOLOGY - ASIACRYPT 2005, Lecture Notes in Computer Science, Volume 3788/2005, 443-461

[42] Jongsung Kim, Seokhie Hong, Bart Preneel, Eli Biham, Orr Dunkelman, Nathan Keller.2009. *Related-Key Boomerang and Rectangle Attacks*. IEEE Transactions on Information Theory

[43] Mark Blunden, Adrian Escott.2002. *Related Key Attacks on Reduced Round KASUMI*. FAST SOFTWARE ENCRYPTION, Lecture Notes in Computer Science, Volume 2355/2002, 391-410

[44] Orr Dunkelman, Nathan Keller, Adi Shamir. 2010. *A Practical-Time Attack on the KASUMI Cryptosystem Used in GSM and 3G Telephony*. ADVANCES IN CRYPTOLOGY - CRYPTO 2010, Lecture Notes in Computer Science, Volume 6223/2010, 393-410

[45] M. E. Hellman.1980. *A Cryptanalytic Time-Memory Trade-off*. IEEE Transactions on Information Theory. Vol. 26, pp. 401-406

[46] D.E. Denning.1982.*Cryptography and Data Security*. Addison-Wesley Publishing Company

[47] Philippe Oechslin.2003. *Making a Faster Cryptanalytic Time-Memory Trade-Off*. CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA. Lecture Notes in Computer Science. Volume 2729/2003, 617-630, Springer

[48] Alex Biryukov. 2005.*Some Thoughts on Time-Memory-Data Tradeoffs*. IACR Eprint archive Report 2005/207

[49] Gildas Avoine, Pascal Junod, Philippe Oechslin. 2005. *Time-Memory Trade-Offs: False Alarm Detection Using Checkpoints*. PROGRESS IN CRYPTOLOGY-INDOCRYPT 2005. Lecture Notes in Computer Science, Volume 3797/2005, 183-196

[50] William Feller. 1968 . *An Introduction to Probability Theory and Its Applications*, 3rd Edition, Vol. 1.Wiley

[51] Johan Borst, Bart Preneel, Joos Vandewalle. 1998. *On the Time-Memory Tradeoff Between Exhaustive Key Search and Table Precomputation*. Proceedings of the 19th Symposium in Information Theory in the Benelux, WIC

[52] Mark Stamp. 2003. *Once Upon a Time-Memory Tradeoff*. http://www.cs.sjsu.edu/faculty/stamp/RUA/TMTO.pdf

[53] svn co https://svn.reflextor.com/tmto-svn