



Πολυτεχνική Σχολή
Τμήμα Μηχανικών Η/Υ & Πληροφορικής

Διπλωματική Εργασία

«Σχεδιασμός και Υλοποίηση
Lightweight Authenticated
Συστήματος Κρυπτογράφησης,
για το Διαδίκτυο των
Πραγμάτων»

Γεώργιος Ντάκος
A.M. 1059569

Επιβλέπων
Νικόλαος Σκλάβος

Γεώργιος Ντάκος, «Σχεδιασμός και Υλοποίηση Lightweight Authenticated Συστήματος
Κρυπτογράφησης, για το Διαδίκτυο των Πραγμάτων»

Μέλη Επιτροπής Αξιολόγησης
Βλάχος Κυριάκος, Καθηγητής
Νικολετσέας Σωτήριος, Καθηγητής
Σκλάβος Νικόλαος, Καθηγητής

Πάτρα, Οκτώβριος 2024

© Copyright συγγραφής Γεώργιος Ντάκος, 2024

© Copyright θέματος SCYTALE Group, Τμήμα Μηχανικών Η/Υ και Πληροφορικής,
Πανεπιστήμιο Πατρών

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών & Πληροφορικής του Πανεπιστημίου Πατρών δεν υποδηλώνει απαραιτήτως και αποδοχή των απόψεων του συγγραφέα εκ μέρους του Τμήματος.

Περίληψη

Η ραγδαία ανάπτυξη του Internet of Things (IoT) έχει δημιουργήσει την ανάγκη για νέες εφαρμογές που ικανοποιούν τις απαιτήσεις χαμηλής καθυστέρησης, υψηλής ταχύτητας, χαμηλής κατανάλωσης ενέργειας και ενισχυμένης ασφάλειας. Καθώς οι IoT συσκευές διεισδύουν ολοένα και περισσότερο στην καθημερινότητα, η ζήτηση για πιο αποδοτικά πρωτόκολλα ασφάλειας και βελτιωμένες επιδόσεις αυξάνεται ραγδαία. Για την αντιμετώπιση αυτών των προκλήσεων, έχει προταθεί ο υβριδικός, ultra-lightweight κρυπταλγόριθμος Hummingbird-2, ο οποίος επιτυγχάνει μια ισορροπία μεταξύ αποδοτικότητας και ασφάλειας, καθιστώντας τον ιδιαίτερα κατάλληλο για εφαρμογές σε ασύρματους αισθητήρες και συσκευές RFID. Σκοπός της παρούσας εργασίας είναι η υλοποίηση μιας ολοκληρωμένης μονάδας Hummingbird-2, η οποία περιλαμβάνει διάφορες βελτιστοποιήσεις στην αρχιτεκτονική του για τη βελτίωση της απόδοσης και τη μείωση της επιφάνειας που καταλαμβάνει. Το σύστημα υλοποιείται σε επιτόπιο προγραμματιζόμενο πίνακα πυλών (Field Programmable Gate Array - FPGA) της σειράς Artix-7, μοντέλο xc7a200tffg1156-3, και λειτουργεί σε μέγιστη συχνότητα 244 MHz.

Λέξεις-κλειδιά: Internet of Things, Κρυπταλγόριθμος, Hummingbird-2, RFID, Ασφάλεια, Χαμηλή Κατανάλωση, FPGA, Ασύρματοι Αισθητήρες, Βελτιστοποίηση Απόδοσης

Abstract

The rapid development of the Internet of Things (IoT) has created the need for new applications that meet the demands for low latency, high speed, low power consumption, and enhanced security. As IoT devices increasingly integrate into everyday life, the demand for more efficient security protocols and improved performance is growing significantly. To address these challenges, the hybrid, ultra-lightweight cryptographic algorithm Hummingbird-2 has been proposed. It strikes a balance between efficiency and security, making it highly suitable for wireless sensors and RFID devices. The aim of this thesis is the implementation of a complete Hummingbird-2 module, including various architectural optimizations to improve performance and reduce the area it occupies. The system is implemented on a Field Programmable Gate Array (FPGA) of the Artix-7 family, model xc7a200tffg1156-3, and operates at a maximum frequency of 244 MHz.

Keywords: Internet of Things, Cryptographic Algorithm, Hummingbird-2, RFID, Security, Low Power Consumption, FPGA, Wireless Sensors, Performance Optimization

Ευχαριστίες

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου, Αναπληρωτή Καθηγητή Νικόλαο Σκλάβο, για την εμπιστοσύνη που μου έδειξε αναθέτοντάς μου το θέμα της παρούσας εργασίας, τις ανεκτίμητες συμβουλές του, την καθοδήγηση του, καθώς και την συνεχή αδιάλειπτη υποστήριξή του κατά τη διάρκεια της συγγραφής της. Επίσης, θα ήθελα να εχαριστήσω την Υποψήφια Διδάκτρωρ Εύα Κωνσταντοπούλου, για τις πολύτιμες γνώσεις, συμβουλές και τον χρόνο που αφιέρωσε, οι οποίες με βοήθησαν να προχωρήσω και να ξεπεράσω τις δυσκολίες μέχρι την ολοκλήρωση της εργασίας.

Συνεχίζοντας, θα ήθελα να ευχαριστήσω θερμά τους γονείς μου και την αδερφή μου, για τις θυσίες που έκαναν καθ' όλη τη διάρκεια των σπουδών μου, αλλά και σε όλη μου τη ζωή, προκειμένου να φέρω εις πέρας την προσπαθειά μου.

Τέλος, θα ήθελα ευχαριστήσω τους φίλους μου, που με κατανόησαν και στάθηκαν δίπλα τόσο στις εύκολες όσο και τις δύσκολες στιγμές της ζωής.

Περιεχόμενα

1.....	
Εισαγωγή	1
1.1 Η Εξέλιξη του Internet of Things και οι Σύγχρονες Προκλήσεις Ασφάλειας	1
1.2 Στόχοι της Διπλωματικής Εργασίας	2
1.3 Διάρθρωση της Διπλωματικής Εργασίας	3
2.....	
Κρυπτογραφία και Εφαρμογές στο Internet of Things	5
2.1 Ιστορική Αναδρομή Κρυπτογραφίας	5
2.2 Βασικές Έννοιες της Κρυπτογραφίας	6
2.3 Είδη Κρυπτογραφίας	8
2.3.1 Συμμετρική Κρυπτογραφία	8
2.3.2 Ασύμμετρη Κρυπτογραφία	9
2.4 Στόχοι της Κρυπτογραφίας στο IoT	10
3.....	
Βασικά Κρυπτογραφικά Δομικά Στοιχεία	14
3.1 Εισαγωγή στα Βασικά Κρυπτογραφικά Δομικά Στοιχεία: Κατηγορίες, Λειτουργίες και Χαρακτηριστικά	14
3.2 Αλγόριθμοι Τμήματος	15
3.2.1 Substitution-Permutation Network Δομή	16
3.2.2 Feistel Δομή	21
3.2.3 Add-Rotate-XOR Δομή	25
3.2.4 Τρόποι Λειτουργίας των Block Ciphers	26

3.3	Αλγόριθμοι Ροής	34
3.3.1	Linear Feedback Shift Registers	35
3.3.2	Non-linear Feedback Shift Registers	36
4.....		
	Message Authentication Codes	38
4.1	Εισαγωγή	38
4.2	Cipher-based Message Authentication Code (CMAC)	38
4.3	Encrypt-then-MAC	39
4.4	Galois/Counter Mode (GCM)	40
5.....		
	FPGAs & ASICs	43
5.1	Εισαγωγή στα FPGAs	43
5.2	Εισαγωγή στα ASICs	44
5.3	Εισαγωγή στις Γλώσσες Υλικού (HDLs)	45
6.....		
	Οικογένεια Hummingbird	48
6.1	Εισαγωγή στην Οικογένεια των Hummingbird Αλγορίθμων	48
6.2	Hummingbird-1	49
6.2.1	Ευπάθειες Hummingbird-1	54
6.3	Hummingbird-2	55
6.3.1	Εισαγώγη στον Hummingbird-2	55
6.3.2	Γενική Αρχιτεκτονική Σχεδιασμού	55
7.....		
	Βελτιστοποίησις του Σχεδιασμού του Hummingbird-2	70
7.1	Προτεινόμενος Σχεδιασμός	70
7.2	Βελτιστοποίηση Σχεδιασμού στο Κόστος Υλικού	71

7.3	Βελτιστοποίηση Σχεδιασμού στην Συχνότητα Ρολογιού	72
8.		
	Προσομοίωση και Υλοποίηση Προτεινόμενου Σχεδιασμού	76
8.1	Εξομοίωση Λειτουργίας Σχεδιασμού	76
8.2	Υλοποίηση Σχεδιασμού	80
8.3	Σύγκριση Σχεδιασμού Hummingbird-2 με Αντίστοιχους Αλγορίθμους	82
8.4	Σύγκριση Αποτελεσμάτων με Άλλες Υλοποιήσεις της Οικογένειας Hummingbird	
	84	
9.		
	Συμπεράσματα και Μελλοντικές Βελτιστοποιήσεις	85
9.1	Σύνοψη	85
9.2	Ευπάθειες του Hummingbird-2	86
9.3	Μελλοντική Εργασία	86

Λίστα Εικόνων

2.1	Βασικό κρυπτογραφικό μοντέλο.	7
2.2	Συμμετρική Κρυπτογραφία.	8
2.3	Ασύμμετρη Κρυπτογραφία.	9
2.4	Σημαντικά ζητήματα ασφάλειας στο IoT.	13
3.1	Λειτουργία Κρυπταλγόριθμου Τμήματος.	16
3.2	Βασική αρχιτεκτονική SPN.	17
3.3	Ένας γύρος ενός SPN [28].	18
3.4	AES Γύρος Κρυπτογράφησης [22].	20
3.5	Κρυπτογράφηση και αποκρυπτογράφηση Feistel (16 γύροι) [22].	22
3.6	Ο αλγόριθμος DES.	23
3.7	Electronic Codebook λειτουργία.	27
3.8	Cipher Block Chaining λειτουργία.	29
3.9	Cipher Feedback λειτουργία των s-bit.	31
3.10	Output Feedback λειτουργία.	32
3.11	Counter λειτουργία.	34
3.12	Λειτουργία Κρυπταλγόριθμου Ροής.	35
3.13	4-bit LFSR.	36
3.14	Η δομή ενός n-bit NLFSR [34].	37
4.1	Αυθεντικοποίηση μηνύματος μέσω MAC.	39
4.2	Κώδικας αυθεντικοποίησης μηνύματος CMAC.	40
4.3	Εφαρμογή MAC σε κρυπτογραφημένο μήνυμα.	41
4.4	Τρόπος λειτουργίας GCM [22].	42
5.1	Η εσωτερική δομή ενός FPGA [36].	44
5.2	Βήματα ροής ψηφιακού σχεδιασμού [37].	47
6.1	Περιγραφή Top-Level του κρυπτογραφικού αλγορίθμου Hummingbird [38].	49
6.2	Τα 4 S-boxes και η δομή του μπλοκ κρυπτογράφησης στον κρυπτογραφικό αλγόριθμο Hummingbird [38].	51
6.3	To Hummingbird Πρωτόκολλο Ιδιωτικής Ταυτοποίησης [39].	52
6.4	Το πρωτόκολλο αμοιβαίου ελέγχου ταυτότητας Hummingbird [39].	53
6.5	Ο αναγνώστης εκδίδει μια εντολή στην ετικέτα [39].	53
6.6	Αρχιτεκτονική της μη γραμμικής συνάρτησης ανάμιξης $f(x)$	58
6.7	Αρχιτεκτονική της συνάρτησης $WD16$	64
6.8	Αρχιτεκτονική Μονάδας Αρχικοποίησης [44].	65
6.9	Αρχιτεκτονική Μονάδας μπλοκ 16 bit Κρυπτογράφησης [44].	66
6.10	Αρχιτεκτονική της $f^{-1}(x)$	67
6.11	Αρχιτεκτονική της $WD16^{-1}$	68

6.12	Αρχιτεκτονική Μονάδας μπλοκ των 16 bit Αποκρυπτογράφησης	69
7.1	Διάγραμμα μηχανής πεπερασμένων καταστάσεων.	71
7.2	Αρχιτεκτονική $WD16^{-1}$ με pipe-lining.	72
7.3	Αρχιτεκτονική $f(x)$ με pipe-lining.	73
7.4	Αρχιτεκτονική $f^{-1}(x)$ με pipe-lining.	74
7.5	Αρχιτεκτονική pipe-lining τεσσάρων σταδίων.	74
7.6	Αρχιτεκτονική pipe-lining δύο σταδίων.	75
8.1	Αποτελέσματα αρχικοποίησης Hummingbird-2.	77
8.2	Αποτελέσματα κρυπτογράφησης Hummingbird-2.	77
8.3	Αποτελέσματα αυθεντικοποίησης Hummingbird-2.	78
8.4	Αποτελέσματα αποκρυπτογράφησης Hummingbird-2.	78
8.5	Αποτελέσματα ελέγχου αυθεντικοποίησης του μηνύματος Hummingbird-2.	79
8.6	Αποτελέσματα αρχικοποίησης Hummingbird-2 για το δεύτερο test vector.	79
8.7	Αποτελέσματα κρυπτογράφησης Hummingbird-2 για το δεύτερο test vector.	79
8.8	Αποτελέσματα αυθεντικοποίησης Hummingbird-2 για το δεύτερο test vector.	80
8.9	Αποτελέσματα αποκρυπτογράφησης Hummingbird-2 για το δεύτερο test vector.	80
8.10	Αποτελέσματα ελέγχου αυθεντικοποίησης του μηνύματος Hummingbird-2 για το δεύτερο test vector.	81

Κατάλογος Πινάκων

5.1	Διαφορές μεταξύ των FPGAs και ASICs.	45
6.1	Τα S-boxes που χρησιμοποιούνται στον Hummingbird-2 [42]	56
6.2	Τα S-boxes που χρησιμοποιούνται στην αποκρυπτογράφηση στον Hummingbird-2.	62
8.1	Τα test vectors που χρησιμοποιήθηκαν [42].	77
8.2	Τα αποτελέσματα της υλοποίησης του Hummingbird-2.	81
8.3	Σύγκριση αποτελεσμάτων στο κόστος υλικού της υλοποίησης του Hummingbird-2 (HB-2) της εργασίας με άλλες υλοποίησεις κρυπταλγόριθμων.	83
8.4	Σύγκριση αποτελεσμάτων στην συχνότητα/throughput/efficiency της υλοποίησης του Hummingbird-2 (HB-2) της εργασίας με άλλες υλοποίησεις κρυπταλγόριθμων.	83
8.5	Σύγκριση αποτελεσμάτων της υλοποίησης του Hummingbird-2 (HB-2) της εργασίας με άλλες υλοποίησεις της οικογένειας Hummingbird.	84

Συντομογραφίες

AEAD	Authenticated Encryption with Associated Data
AES	Advanced Encryption Standard
ARX	Addition Rotation XOR
ASIC	Application Specific Integrated Circuit
DDoS	Distributed Denial of Service
DES	Data Encryption Standard
FPGA	Field Programmable Gate Array
HDL	Hardware Description Language
IBM	International Business Machines
IEEE	Institute of Electrical and Electronics Engineers
IPsec	Internet Protocol Security
IoT	Internet of Things
ISO	International Organization for Standardization
IV	Initialization Vector
MAC	Message Authentication Code
NIST	National Institute of Standards and Technology
RFID	Radio Frequency Identification
SoCs	System on Chips
SPN	Substitution Permutation Network
TLS	Transport Layer Security

Γλωσσάρι ή Απόδοση Όρων

Access Control	Έλεγχος Πρόσβασης
Add-Rotate-XOR	Πρόσθεση-Περιστροφή-XOR
ASIC	Ολοκληρωμένα Κυκλώματα Ειδικής Εφαρμογής
Asymmetric cryptography	Ασύμμετρη Κρυπτογραφία
Authentication	Αυθεντικοποίηση
Authorization	Εξουσιοδότηση
Availability	Διαθεσιμότητα
Bit	Δυαδικό Ψηφίο
Block cipher	Κρυπταλγόριθμος Μπλοκ
Botnet	Δίκτυο από Μολυσμένες Συσκευές
Brute force	Επίθεση μέσω Δοκιμών
Cipher	Κρυπταλγόριθμος
Ciphertext	Κρυπτοκείμενο
Clock	Ρολόι
Confidentiality	Εμπιστευτικότητα
Confusion	Σύγχυση
Cryptosystem	Κρυπτοσύστημα
Data stream	Ροή Δεδομένων
DDoS	Κατανεμημένη Άρνηση Υπηρεσίας
Diffusion	Διάχυση

Encrypt	Κρυπτογράφηση
FPGA	Επιτόπια Προγραμματιζόμενο Πίνακα Πυλών
Flip-flop	Δισταθής Πολυδονητής
Gate-level	Επίπεδο Πυλών
Hybrid	Υβριδικό
Identification	Ταυτοποίηση
Integrity	Ακεραιότητα
Internet of Things	Διαδίκτυο των Πραγμάτων
IV	Διάνυσμα Αρχικοποίησης
Keystream	Ροή Κλειδιού
Key	Κλειδί
Mobile Security	Ασφάλεια Κινητών Συσκευών
MAC	Κωδικός Αυθεντικοποίησης Μηνυμάτων
Nonce	Μοναδικός Αριθμός
Non-repudiation	Μη Αποποίηση Ευθύνης
Padding	Συμπλήρωση
Plaintext	Απλό Κείμενο
Policy Enforcement	Εφαρμογή Πολιτικής
Private key	Ιδιωτικό Κλειδί
Public key	Δημόσιο Κλειδί
Public key cryptography	Κρυπτογραφία Δημόσιου Κλειδιού
RFID	Ταυτοποίηση μέσω Ραδιοσυγνοήτων
Related-key cryptanalysis	Κρυπτοανάλυση Σχετικών Κλειδιών

Reliability	Αξιοπιστία
Reset	Επαναφορά
Responsibility	Ευθύνη
Safety	Ασφάλεια
Secure Middleware	Ενδιάμεσο Λογισμικό Ασφάλειας
Self-Immunity	Αυτο-ανοσία
Stream cipher	Κρυπταλγόριθμος Ροής
Symmetric cryptography	Συμμετρική Κρυπτογραφία
Trade-off	Ανταλλαγή Οφέλους-Κόστους
Trust	Εμπιστοσύνη
VHDL	Γλώσσα Περιγραφής Υλικού Ολοκληρωμένων Πολύ Υψηλής Ταχύτητας

1

Εισαγωγή

1.1 Η Εξέλιξη του Internet of Things και οι Σύγχρονες Προκλήσεις Ασφάλειας

Το διαδίκτυο, τα μέσα κοινωνικής δικτύωσης και γενικά η διαδικτυακή πληροφόρηση αποτελούν πλέον ένα μεγάλο κομμάτι της καθημερινότητας της σύγχρονης ανθρωπότητας. Σε καθημερινή βάση, ανταλλάσσονται μεταξύ χρηστών αμέτρητες πληροφορίες, ενώ διαρκώς ανακαλύπτονται και αναπτύσσονται μέθοδοι για τη διευκόλυνση αυτής της διαδικασίας. Χάρη σε αυτή την εξέλιξη του διαδικτύου, η μέχρι πρότινος περιορισμένη ανταλλαγή πληροφοριών μέσω ηλεκτρονικού ταχυδρομείου έχει επεκταθεί σε άμεση και σε πραγματικό χρόνο επικοινωνία, χρησιμοποιώντας σύγχρονα και ταχύτατα διαδικτυακά κανάλια επικοινωνίας.

Κάθε τέτοιο σύγχρονο κανάλι έχει τα δικά του χαρακτηριστικά και δυνατότητες που προσφέρει στον χρήστη. Το σύνολο όλων αυτών των καναλιών είναι προσβάσιμα από μια κεντρική μονάδα ανεξάρτητης τοποθεσίας, η οποία είναι συνδεδεμένη στο δίκτυο. Αυτή η ευκολία στην πρόσβαση πληροφοριών υπήρξε η χρυσή ευκαιρία για την ανάπτυξη του Internet of Things (IoT), το οποίο έχει ως σκοπό την επικοινωνία μεταξύ συσκευών, επιτρέποντας την ανταλλαγή πληροφοριών και επηρεάζοντας την λειτουργικότητά τους [1].

Έως και τη δεκαετία του '80 όλα τα παραπάνω θα φάνταζαν σαν ένα ακόμα σενάριο επιστημονικής φαντασίας. Ωστόσο, το 1991, ο Mark Weiser, με την εργασία του για τη διάχυτη πληροφορική, έθεσε το όραμα για το σημερινό IoT [2,3]. Στη συνέχεια, το 1999, ο Kevin Ashton επινόησε τον όρο «Internet of Things» και το περιέγραψε ως ένα σύστημα διασύνδεσης του φυσικού κόσμου με το διαδίκτυο, μέσω της χρήσης τεχνολογιών RFID (Radio Frequency Identification) και διάχυτων συσκευών αισθητήρων που παρατηρούν και αναγνωρίζουν τον περιβάλλοντα χώρο [3–5].

Από τα παραπάνω συμπεραίνουμε ότι το IoT αναφέρεται σε ένα δίκτυο διασυνδεδεμένων συσκευών που μπορούν να επικοινωνούν και να αλληλεπιδρούν μεταξύ τους και με το περιβάλλον τους. Η ανάπτυξη του IoT υποστηρίζεται από την προόδο της ασύρματης επικοινωνίας, την αύξηση της υπολογιστικής ισχύος και τη μείωση κόστους των συσκευών. Σήμερα, το IoT αποτελεί έναν από τους βασικούς πυλώνες της ψηφιακής εποχής, επηρεάζοντας τομείς όπως η υγεία, η βιομηχανία, οι έξυπνες πόλεις, τα έξυπνα σπίτια αλλά και άλλους πολύ σημαντικούς

τομείς εφαρμογών [6].

Η αρχιτεκτονική του IoT περιλαμβάνει διάφορα επίπεδα που συνεργάζονται για να επιτύχουν την ανταλλαγή και επεξεργασία δεδομένων:

- Επίπεδο Αισθητήρων:** Περιλαμβάνει τις συσκευές που συλλέγουν δεδομένα από το περιβάλλον τους, όπως αισθητήρες θερμοκρασίας, υγρασίας και κίνησης.
- Επίπεδο Δικτύου:** Ασχολείται με τη μεταφορά των δεδομένων από τους αισθητήρες στους διακομιστές και τα κέντρα δεδομένων.
- Επίπεδο Επεξεργασίας Δεδομένων:** Περιλαμβάνει τις υποδομές και το λογισμικό που επεξεργάζονται και αποθηκεύουν τα δεδομένα, όπως τα συστήματα υπολογιστικού νέφους (cloud).
- Επίπεδο Εφαρμογών:** Αποτελεί επίπεδο στο οποίο οι τελικοί χρήστες αλληλεπιδρούν με τις εφαρμογές IoT, όπως οι εφαρμογές έξυπνων σπιτιών και οι υπηρεσίες υγείας.

Η ανάπτυξη του IoT επηρεάζεται σημαντικά από την πρόοδο της τεχνολογίας αισθητήρων, της ασύρματης επικοινωνίας και των ενσωματωμένων συστημάτων. Η χρήση αισθητήρων RFID, για παράδειγμα, έχει επιτρέψει τη συλλογή και ανταλλαγή δεδομένων σε πραγματικό χρόνο, καθιστώντας δυνατή την παρακολούθηση και διαχείριση διαφόρων διαδικασίων και λειτουργιών.

Ωστόσο, η αύξηση του αριθμού των διασυνδεδεμένων συσκευών δημιουργεί νέες προκλήσεις ασφάλειας, καθώς οι ευπάθειες και οι απειλές για τα συστήματα IoT αυξάνονται. Η ασφάλεια και η προστασία των δεδομένων γίνονται κρίσιμα ζητήματα που πρέπει να αντιμετωπιστούν για να διασφαλιστεί η αξιοπιστία των συστημάτων IoT [6]. Πολλές συσκευές IoT δεν διαθέτουν ισχυρούς μηχανισμούς αυθεντικοποίησης, καθιστώντας τες ευάλωτες σε επιθέσεις από κακόβουλους χρήστες [7]. Η έλλειψη τακτικών ενημερώσεων λογισμικού κάνει τις συσκευές επιρρεπείς σε γνωστές ευπάθειες, ενώ η απουσία κρυπτογράφησης στα δεδομένα καθιστά τις πληροφορίες ευαίσθητες σε μη εξουσιοδοτημένη πρόσβαση [8]. Επιπλέον, οι επιθέσεις DDoS (Distributed Denial-of-Service) αποτελούν μια σοβαρή απειλή για τα δίκτυα IoT [6,7,9]. Η προστασία της ιδιωτικότητας των δεδομένων είναι επίσης κρίσιμη για την εμπιστοσύνη των χρηστών [7–10]. Τέλος, οι επιθέσεις botnet αυξάνονται, με τις συμβιβασμένες IoT συσκευές να χρησιμοποιούνται για κακόβουλες ενέργειες [7,10,11]. Η αντιμετώπιση αυτών των προκλήσεων απαιτεί πολυεπίπεδη προσέγγιση με ισχυρούς μηχανισμούς ασφάλειας, εκπαίδευση χρηστών και συνεχή παρακολούθηση των συστημάτων [8].

1.2 Στόχοι της Διπλωματικής Εργασίας

Οι στόχοι της παρούσας διπλωματικής εργασίας εστιάζονται στην υλοποίηση σε υλικό και βελτιστοποίηση του κρυπτογραφικού αλγορίθμου Hummingbird-2, ο οποίος έχει σχεδιαστεί για συστήματα με περιορισμένους πόρους, όπως τα RFID tags και οι ασύρματοι αισθητήρες. Ο κύριος στόχος της εργασίας είναι η βελτίωση του αλγορίθμου, τόσο σε επίπεδο υλικού όσο και σε επίπεδο χρονισμού, διασφαλίζοντας παράλληλα τη συμμόρφωση με τις απαιτούμενες προδιαγραφές ασφαλείας.

Ο αλγόριθμος περιγράφτηκε σε γλώσσα VHDL, ενώ έγινε προσομοίωση μέσω του εργαλείου ModelSim για την αναλυτική αξιολόγηση της λειτουργικότητάς του πριν την υλοποίηση.

Στη συνέχεια, ο αλγόριθμος υλοποιήθηκε σε ένα FPGA της πλατφόρμας Xilinx Artix-7, χρησιμοποιώντας το εργαλείο Vivado για τη σύνθεση. Μέσω αυτής της διαδικασίας, επιδιώχθηκε η βελτιστοποίηση του αλγορίθμου, με έμφαση στη μείωση του κόστους υλικού και την αύξηση της συχνότητας λειτουργίας, εξασφαλίζοντας ταυτόχρονα την αποτελεσματικότητά του σε συσκευές περιορισμένων πόρων.

Επιπρόσθετα, η παρούσα εργασία συγκρίνει την απόδοση του Hummingbird-2 με άλλους συναφείς κρυπτογραφικούς αλγόριθμους και αξιολογεί τις βελτιώσεις του σε σχέση με τον πρόκατοχό του, Hummingbird-1. Η συγκριτική ανάλυση επικεντρώνεται στην ασφάλεια, την απόδοση και την ανθεκτικότητα του αλγορίθμου απέναντι σε γνωστές επιθέσεις κρυπτανάλυσης, με σκοπό να προταθούν περαιτέρω βελτιώσεις για μελλοντική υλοποίηση σε συστήματα περιορισμένων πόρων.

1.3 Διάρθρωση της Διπλωματικής Εργασίας

Στο πρώτο κεφάλαιο της Διπλωματικής Εργασίας, κάνουμε μια ιστορική ανασκόπηση της ανάπτυξης του IoT και παρουσιάζονται οι σημαντικότερες προκλήσεις ασφάλειας που αντιμετωπίζουν οι διασυνδεδεμένες συσκευές σήμερα, όπως η προστασία δεδομένων, η ιδιωτικότητα και οι ευπάθειες δικτύων.

Στο δεύτερο κεφάλαιο παρουσιάζεται μια συνοπτική ιστορική αναδρομή της κρυπτογραφίας και αναλύονται οι βασικές έννοιες και τα κύρια είδη της, όπως η συμμετρική και ασύμμετρη κρυπτογραφία. Στη συνέχεια, εξετάζονται οι στόχοι και οι εφαρμογές της κρυπτογραφίας στο πλαίσιο του IoT, τονίζοντας τη σημασία της για την ασφάλεια των δικτύων και των συσκευών.

Συνεχίζοντας, το τρίτο κεφάλαιο, επικεντρώνεται στα βασικά κρυπτογραφικά δομικά στοιχεία, παρουσιάζοντας τις κατηγορίες, τις λειτουργίες και τα χαρακτηριστικά τους. Αναλύονται οι σημαντικότεροι αλγόριθμοι κρυπτογράφησης μπλοκ, οι δομές Substitution-Permutation Network (SPN), Feistel και Add-Rotate-XOR που τους σχηματίζουν, καθώς και οι τρόποι λειτουργίας τους. Επιπλέον, εξετάζονται οι αλγόριθμοι ροής και τα βασικά τους στοιχεία, όπως οι γραμμικοί και μη γραμμικοί καταχωρητές.

Στο τέταρτο κεφάλαιο παρουσιάζονται τα Message Authentication Codes (MACs) και η σημασία τους για την εξασφάλιση της ακεραιότητας και αυθεντικότητας των μηνυμάτων. Εξετάζονται διάφορες μέθοδοι, όπως ο Cipher-based Message Authentication Code (CMAC), ο Encrypt-then-MAC, και ο Galois/Counter Mode (GCM), περιγράφοντας τη λειτουργία και τα πλεονεκτήματα καθεμίας.

Στο πέμπτο κεφάλαιο γίνεται εισαγωγή στις τεχνολογίες FPGA (Field Programmable Gate Arrays) και ASIC (Application-Specific Integrated Circuits), αναλύοντας τα χαρακτηριστικά, τις διαφορές και τις εφαρμογές τους. Επιπλέον, παρουσιάζονται οι βασικές γλώσσες υλικού (HDLs) που χρησιμοποιούνται για τον προγραμματισμό και τη σχεδίαση αυτών των κυκλωμάτων.

Το έκτο κεφάλαιο εξετάζει την οικογένεια των Hummingbird αλγορίθμων, οι οποίοι σχεδιάστηκαν για εφαρμογές χαμηλής κατανάλωσης και πόρων, όπως στο IoT. Παρουσιάζεται λεπτομερώς ο αλγόριθμος Hummingbird-1, μαζί με τις γνωστές ευπάθειές του, ενώ ακολουθεί ανάλυση του Hummingbird-2, με έμφαση στη γενική αρχιτεκτονική σχεδιασμού και τις επιθέσεις κρυπτανάλυσης σχετικού κλειδιού που έχουν αναγνωριστεί.

Το έβδομο κεφάλαιο εστιάζει στις βελτιστοποιήσεις του αλγόριθμου Hummingbird-2, προτείνοντας έναν νέο σχεδιασμό που στοχεύει στη βελτίωση της απόδοσης. Αναλύονται οι προσεγγίσεις για τη βελτιστοποίηση του κόστους υλικού, καθώς και οι μέθοδοι βελτίωσης της απόδοσης σε σχέση με τη συχνότητα ρολογιού, εξασφαλίζοντας έτσι αποτελεσματικότερη λειτουργία του αλγορίθμου σε συσκευές με περιορισμένους πόρους.

Στο όγδοο κεφάλαιο περιγράφεται η προσομοίωση και υλοποίηση του προτεινόμενου σχεδιασμού του Hummingbird-2, αξιολογώντας την απόδοση του μέσω εξομοιώσεων. Αναλύονται τα αποτελέσματα της υλοποίησης του σχεδιασμού σε πραγματικό υλικό και συγκρίνεται με άλλους παρόμοιους αλγόριθμους. Τέλος, γίνεται σύγκριση των αποτελεσμάτων με προηγούμενες υλοποιήσεις της οικογένειας Hummingbird, αξιολογώντας τις βελτιώσεις και την αποτελεσματικότητα του νέου σχεδιασμού.

Στο ένατο και τελευταίο κεφάλαιο συνοψίζονται τα κύρια συμπεράσματα της διπλωματικής εργασίας, αναδεικνύοντας τις επιτυχίες και τις προκλήσεις του προτεινόμενου σχεδιασμού του Hummingbird-2. Επιπλέον, παρουσιάζονται προτάσεις για μελλοντικές βελτιστοποιήσεις και ερευνητικές κατευθύνσεις, με στόχο τη βελτίωση της απόδοσης και της ασφάλειας του αλγορίθμου σε πραγματικές εφαρμογές.

2

Κρυπτογραφία και Εφαρμογές στο Internet of Things

2.1 Ιστορική Αναδρομή Κρυπτογραφίας

Η κρυπτογραφία δεν αποτελεί ένα νέο πεδίο που ανακαλύφθηκε και αναπτύχθηκε τον 20ό αιώνα, αλλά έχει τις ρίζες της χιλιάδες χρόνια πριν. Η πρώτη καταγεγραμμένη εφαρμογή κρυπτογραφίας, αν και όχι με τη μορφή που τη γνωρίζουμε σήμερα, εμφανίστηκε στην Αίγυπτο, σε μια επιγραφή σκαλισμένη γύρω στο 1900 π.Χ. στον κύριο θάλαμο του τάφου του ευγενούς Khnumhotep II. Η επιγραφή αυτή περιλαμβανε ασυνήθιστα ιερογλυφικά σύμβολα, όχι για να κρύψει το μήνυμα, αλλά για να αλλάξει τη μορφή του, καθιστώντας το πιο εντυπωσιακό και αξιοπρεπές. Ωστόσο, η πρώτη χρήση κρυπτογραφίας με σκοπό την απόκρυψη σημαντικών πληροφοριών καταγράφηκε στη Μεσοποταμία περίπου το 1500 π.Χ., όπου μια σφηνοειδής επιγραφή περιέγραφε μια μέθοδο κατασκευής σμάλτων για αγγειοπλαστική.

Η κρυπτογραφία συνέχισε να εξελίσσεται μέσα στους αιώνες, με αξιοσημείωτες αναφορές σε αρχαίους πολιτισμούς, όπως η αρχαία Σπάρτη με τη λεγόμενη Σκυτάλη [12], και η αρχαία Ρώμη με τον "Κώδικα του Καίσαρα," ο οποίος αποτελεί τη βάση της σύγχρονης κρυπτογραφίας [13]. Σταθμοί στην εξέλιξη της κρυπτογραφίας περιλαμβάνουν την επιτυχημένη κρυπτανάλυση της γερμανικής συσκευής Enigma από τον Alan Turing και την ομάδα του κατά τη διάρκεια του Β' Παγκοσμίου Πολέμου, καθώς και την καθοριστική εργασία του Claude Shannon το 1949 με τίτλο "Communication Theory of Secrecy Systems," η οποία καθόρισε για πάντα το τρόπο σκέψης γύρω από την κρυπτογραφία, εισάγοντας αυστηρούς μαθηματικούς ορισμούς και εννοιολογικές ιδιότητες όπως η τέλεια μυστικότητα [12].

Στις μέρες μας, η κρυπτογραφία δεν είναι πλέον ένα αποκλειστικό προνόμιο στρατιωτικών εφαρμογών και κρατικών μυστικών, αλλά έχει διαποτίσει κάθε πτυχή της ανθρώπινης ζώης, από στρατιωτικές εφαρμογές και συστήματα αεροπορίας μέχρι τα προσωπικά μας κινητά τηλέφωνα και τις οικιακές συσκευές. Έχει περιγραφεί ως ένας διαρκής αγώνας μεταξύ αυτών που

επιδιώκουν να αποκρύψουν πληροφορίες για το κοινό καλό και εκείνων που προσπαθούν να υποκλέψουν αυτές τις πληροφορίες για ίδιον όφελος. Καθώς αυτός ο ανταγωνισμός συνεχίζεται, η κρυπτογραφία αναμένεται να εξελίσσεται διαρκώς, με την ανάπτυξη νέων τεχνικών που προσφέρουν βελτιωμένα αποτελέσματα.

Το συμπέρασμα αυτής της ραγδαίας εξέλιξης τόσο της κρυπτογραφίας όσο και της IoT βιομηχανίας, η οποία εκτιμάται ότι θα φτάσει σε έσοδα 4 τρισεκατομμυρίων δολαρίων έως το 2025 [14], είναι η ανάγκη για σαφείς και αυστηρούς νόμους και πρωτόκολλα. Η ιδρυση οργανισμών όπως το IoT Security Foundation (IoTSF) και οι πρωτοβουλίες από κυβερνήσεις, όπως η έκθεση "Secure by Design" του Ηνωμένου Βασιλείου, αποτελούν σημαντικά πρώτα βήματα προς αυτήν την κατεύθυνση. Ωστόσο, η έλλειψη υποχρεωτικών κανονισμών και η ανεπαρκής τεχνική καθοδήγηση υποδεικνύουν ότι απαιτείται περαιτέρω δράση για να διασφαλιστεί ότι οι αλγόριθμοι κρυπτογραφίας και οι IoT συσκευές θα συμμορφώνονται με τα υψηλότερα πρότυπα ασφάλειας και προστασίας προσωπικών δεδομένων [15].

2.2 Βασικές Έννοιες της Κρυπτογραφίας

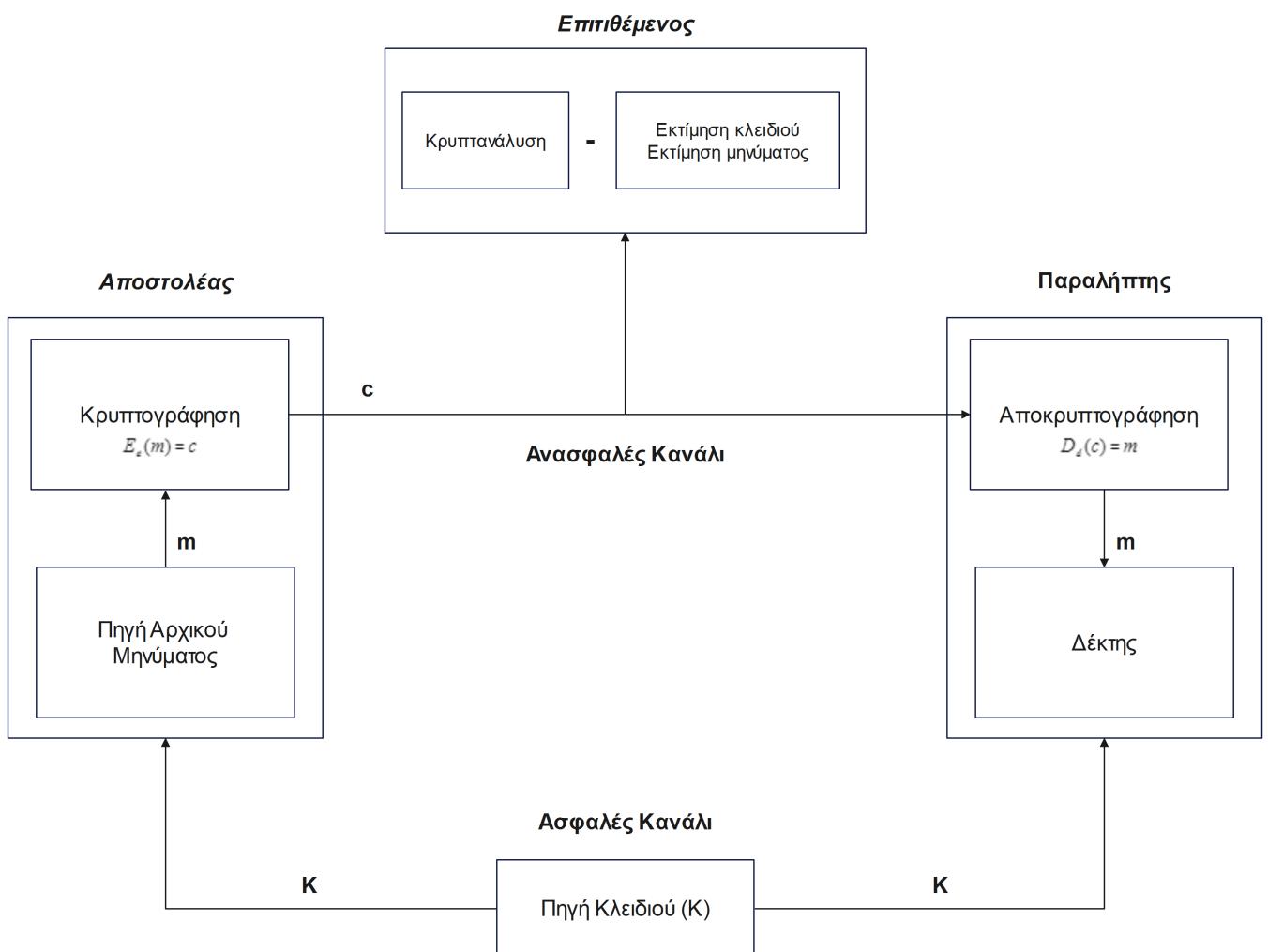
Το κρυπτογραφικό μοντέλο αποτελεί ένα θεμελιώδες πλαίσιο για την εξασφάλιση της ασφάλειας στην επικοινωνία και την προστασία δεδομένων. Βασίζεται σε αυστηρές μαθηματικές αρχές και αλγορίθμους, οι οποίοι διασφαλίζουν την εμπιστευτικότητα, την ακεραιότητα και τη γνησιότητα των πληροφοριών, προφυλάσσοντάς τις από μη εξουσιοδοτημένη πρόσβαση ή αλλοίωση. Ένα βασικό κρυπτογραφικό μοντέλο για την επικοινωνία μεταξύ δύο οντοτήτων μέσω κρυπτοσυστήματος (cryptosystem) φαίνεται στην παρακάτω εικόνα (εικόνα 2.1).

Όπως φαίνεται, το σύστημα περιλαμβάνει τρεις βασικούς ρόλους: τον αποστολέα, τον παραλήπτη και τον επιτιθέμενο. Ο αποστολέας παράγει ένα μήνυμα (απλό κείμενο ή plaintext), το οποίο μπορεί να διαβαστεί από οποιονδήποτε σε αυτή την αρχική του μορφή. Για να εξασφαλίσει την προστασία του μηνύματος, το απλό κείμενο περνά από μια διαδικασία κρυπτογράφησης, με αποτέλεσμα να παράγεται το κρυπτογράφημα (ciphertext), το οποίο είναι ακατανόητο χωρίς το σωστό κλειδί αποκρυπτογράφησης.

Αφού παραχθεί το κρυπτογράφημα, αποστέλλεται μέσω ενός μη ασφαλούς καναλιού στον παραλήπτη. Ο στόχος του παραλήπτη είναι να ανακτήσει το αρχικό απλό κείμενο, το οποίο επιτυγχάνεται μέσω της αντίστροφης διαδικασίας, δηλαδή της αποκρυπτογράφησης. Η διαδικασία της κρυπτογράφησης και της αποκρυπτογράφησης απαιτεί τη χρήση ενός σημαντικού στοιχείου, του κλειδιού (key), το οποίο είναι μια ακολουθία ψηφίων που εισάγεται ως είσοδος στους αντίστοιχους αλγορίθμους.

Το κρυπτογράφημα, ωστόσο, διακινείται μέσω ενός μη ασφαλούς καναλιού, το οποίο είναι ευάλωτο σε επιθέσεις από τρίτους. Ο επιτιθέμενος μπορεί να υποκλέψει το κρυπτογράφημα και, μέσω της διαδικασίας της κρυπτανάλυσης, να προσπαθήσει να αναλύσει τον κρυπταλγόριθμο και να ανακτήσει το αρχικό μήνυμα. Ο στόχος του κρυπτοσυστήματος είναι να διασφαλίσει ότι το κρυπτογράφημα είναι τόσο ισχυρό ώστε να αντιστέκεται σε τέτοιες επιθέσεις.

Σημειώνεται ότι στα πληροφοριακά συστήματα, όλες οι πληροφορίες αναπαρίστανται ως αληλουχίες δυαδικών ψηφίων (bits). Συνεπώς, το «απλό κείμενο» μπορεί να είναι όχι μόνο ένα κείμενο, αλλά επίσης μια εικόνα, ένα ηχητικό μήνυμα ή οποιοσδήποτε άλλος τύπος δεδομένων [16].



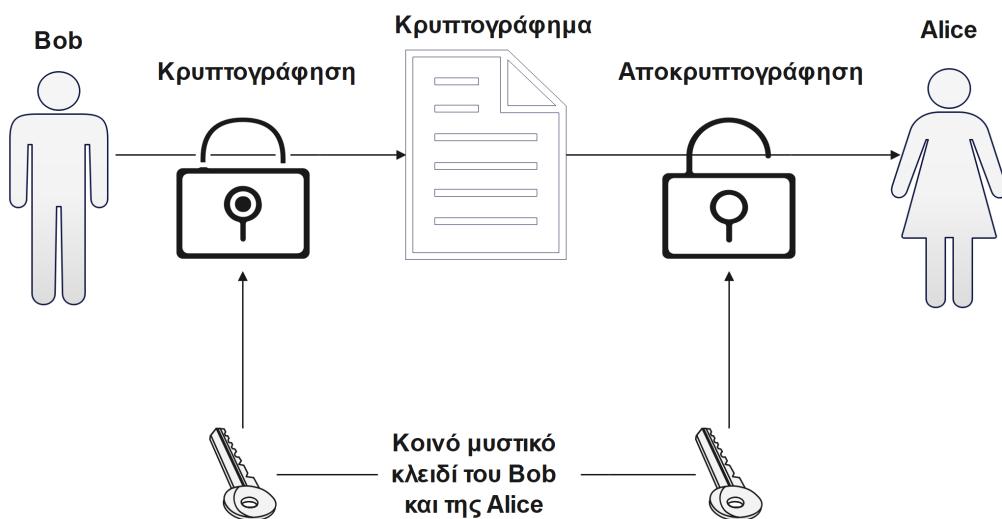
Σχήμα 2.1: Βασικό κρυπτογραφικό μοντέλο.

2.3 Είδη Κρυπτογραφίας

Η κρυπτογραφία διακρίνεται σε διάφορα είδη, τα οποία καθορίζονται από τη μέθοδο κρυπτογράφησης και τη χρήση των κλειδιών. Τα βασικά είδη κρυπτογραφίας περιλαμβάνουν την συμμετρική και την ασύμμετρη κρυπτογραφία, όπου η πρώτη χρησιμοποιεί το ίδιο κλειδί για την κρυπτογράφηση και την αποκρυπτογράφηση, ενώ η δεύτερη βασίζεται σε ζεύγος κλειδιών, δημόσιου και ιδιωτικού. Επιπλέον, υπάρχουν υβριδικές τεχνικές που συνδυάζουν τα πλεονεκτήματα των δύο μεθόδων για βέλτιστη ασφάλεια [17].

2.3.1 Συμμετρική Κρυπτογραφία

Η συμμετρική κρυπτογραφία, ή αλλιώς κρυπτογράφηση συμμετρικού κλειδιού (symmetric cryptography), βασίζεται στη χρήση ενός κοινού μυστικού κλειδιού, το οποίο χρησιμοποιείται τόσο για την κρυπτογράφηση όσο και για την αποκρυπτογράφηση του απλού κειμένου. Το κοινό αυτό κλειδί είναι γνωστό μόνο στα μέρη που συμμετέχουν στην ανταλλαγή της πληροφορίας. Στην παρακάτω εικόνα (εικόνα 2.2) απεικονίζεται η διαδικασία αυτή.



Σχήμα 2.2: Συμμετρική Κρυπτογραφία.

Το κύριο πρόβλημα που αντιμετωπίζουν οι αλγόριθμοι συμμετρικής κρυπτογράφησης είναι η ασφαλής ανταλλαγή κλειδιού. Στη σύγχρονη ψηφιακή εποχή, όπου ο αποστολέας και ο παραλήπτης συχνά δεν γνωρίζονται προσωπικά, απαιτείται η ύπαρξη ενός ασφαλούς καναλιού επικοινωνίας για τη μεταφορά του μυστικού κλειδιού. Συνεπώς, αν ένας επιτιθέμενος καταφέρει να αποκτήσει το κλειδί, μπορεί εύκολα να αποκρυπτογραφήσει οποιοδήποτε μήνυμα έχει κρυπτογραφηθεί με αυτό το κλειδί.

Στη συμμετρική κρυπτογραφία χρησιμοποιούνται οι εξής δύο βασικές εξισώσεις:

$$E_{k_1}(m) = c$$

η οποία αναπαριστά τη διαδικασία κρυπτογράφησης, και

$$D_{k_2}(c) = m$$

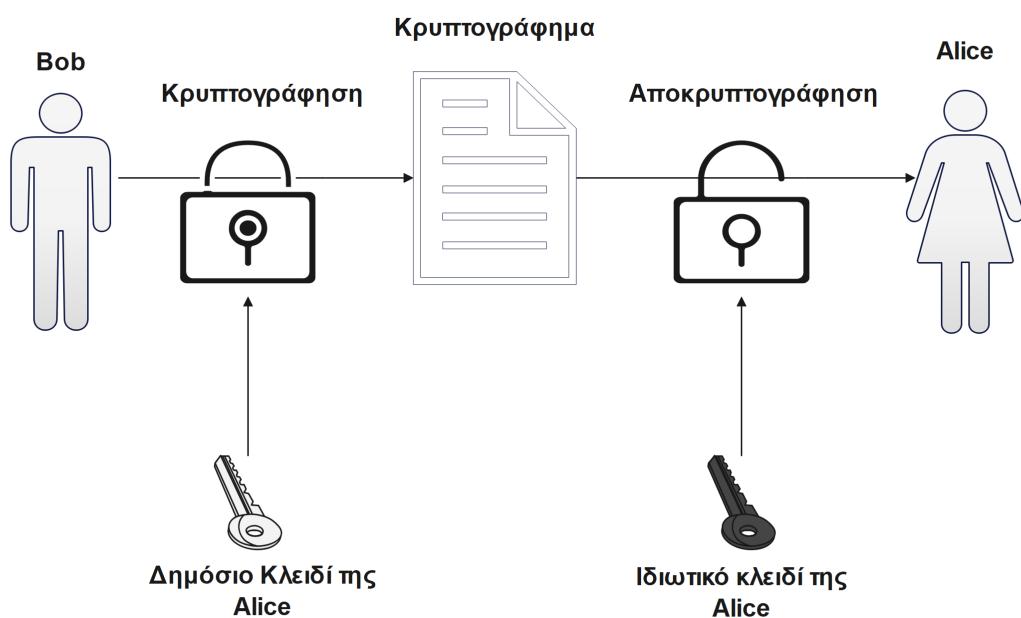
η οποία αναπαριστά τη διαδικασία αποκρυπτογράφησης, όπου το κλειδί k παραμένει το ίδιο για την κρυπτογράφηση και την αποκρυπτογράφηση (δηλαδή $k_1 = k_2$).

Το βασικό πλεονέκτημα των αλγορίθμων συμμετρικής κρυπτογράφησης είναι η υψηλή ταχύτητα και η χαμηλή υπολογιστική ισχύς που απαιτούν, γεγονός που τους καθιστά ιδανικούς για χρήση σε συσκευές με περιορισμένους πόρους, όπως αυτές του Διαδικτύου των Πραγμάτων (IoT).

Μερικοί από τους πιο γνωστούς αλγόριθμους συμμετρικής κρυπτογράφησης είναι οι DES, 3DES, IDEA, RC2, RC4 και AES.

2.3.2 Ασύμμετρη Κρυπτογραφία

Η κρυπτογράφηση δημόσιου κλειδιού (public key cryptography) ή αλλιώς ασύμμετρη κρυπτογραφία (asymmetric cryptography) αναπτύχθηκε στα τέλη της δεκαετίας του 1970 από τους Diffie και Hellman [12]. Η βασική ιδέα τους ήταν ότι ο αποστολέας και ο παραλήπτης δεν χρειάζεται να μοιράζονται ένα κοινό μυστικό κλειδί, όπως συμβαίνει στη συμμετρική κρυπτογράφηση. Αντίθετα, κάθε χρήστης διαθέτει ένα ζεύγος κλειδιών για διαφορετικές λειτουργίες: ένα ιδιωτικό κλειδί (private key) και ένα δημόσιο κλειδί (public key).



Σχήμα 2.3: Ασύμμετρη Κρυπτογραφία.

Το ιδιωτικό κλειδί παραμένει κρυφό και γνωστό μόνο στον κάτοχό του, ενώ το δημόσιο κλειδί είναι διαθέσιμο σε όλους τους χρήστες. Η σχέση ανάμεσα σε αυτά τα δύο κλειδιά είναι

συμπληρωματική: ό,τι κρυπτογραφείται με το δημόσιο κλειδί μπορεί να αποκρυπτογραφηθεί μόνο με το αντίστοιχο ιδιωτικό κλειδί και αντίστροφα.

Για να κατανοήσουμε καλύτερα αυτή τη διαδικασία, ας δούμε ένα παράδειγμα με δύο οντότητες, τον Bob και την Alice. Και οι δύο διαθέτουν ένα ιδιωτικό και ένα δημόσιο κλειδί. Αν ο Bob θέλει να στείλει ένα μήνυμα στην Alice, θα το κρυπτογραφήσει χρησιμοποιώντας το δημόσιο κλειδί της Alice, το οποίο είναι διαθέσιμο σε όλους. Μόνο η Alice, με τη χρήση του ιδιωτικού της κλειδιού, θα μπορεί να αποκρυπτογραφήσει και να διαβάσει το μήνυμα (εικόνα 2.3). Αντίστοιχα, αν η Alice θέλει να στείλει ένα μήνυμα στον Bob, θα χρησιμοποιήσει το δημόσιο κλειδί του Bob για την αποκρυπτογράφηση. Με αυτόν τον τρόπο, εξασφαλίζεται ότι οποιοσδήποτε μπορεί να επιβεβαιώσει την αυθεντικότητα της οντότητας που κρυπτογράφησε το μήνυμα, καθώς μόνο ο κάτοχος του ιδιωτικού κλειδιού μπορεί να το αποκρυπτογραφήσει [16].

2.4 Στόχοι της Κρυπτογραφίας στο IoT

Η κρυπτογραφία είναι ένας από τους κεντρικούς πυλώνες της ασφάλειας στο IoT. Η αυξημένη διασυνδεσιμότητα των συσκευών IoT, σε συνδυασμό με την ανάγκη για ασφαλή επικοινωνία και προστασία δεδομένων, δημιουργεί αυξημένες απαιτήσεις για την εφαρμογή αξιόπιστων κρυπτογραφικών μηχανισμών. Οι στόχοι της κρυπτογραφίας στο πλαίσιο των IoT δικτύων επικεντρώνονται στη διασφάλιση της ακεραιότητας της εμπιστευτικότητας και της αυθεντικότητας των δεδομένων και των κόμβων που επικοινωνούν στο δίκτυο. Οι βασικοί στόχοι που καθορίζουν την αποτελεσματικότητα της κρυπτογραφίας στα IoT περιλαμβάνουν :

- **Αυθεντικοποίηση (Authentication) και Εξουσιοδότηση (Authorization):** Η αυθεντικοποίηση και η εξουσιοδότηση αποτελούν δύο διακριτές αλλά αλληλένδετες διαδικασίες που είναι θεμελιώδεις για την ασφάλεια των IoT συστημάτων. Η αυθεντικοποίηση διασφαλίζει ότι μόνο νόμιμοι χρήστες και συσκευές μπορούν να εισέλθουν στο σύστημα, ενώ η εξουσιοδότηση καθορίζει τα δικαιώματα και τα επίπεδα πρόσβασης αυτών των οντοτήτων στους διαθέσιμους πόρους και υπηρεσίες του δικτύου. Ο συνδυασμός αυτών των δύο διαδικασιών αποτρέπει τη μη εξουσιοδοτημένη πρόσβαση και ενισχύει την αξιοπιστία των αλληλεπιδράσεων εντός του IoT περιβάλλοντος [18,19].
- **Πρόσβαση και Έλεγχος (Access Control) και Ταυτοποίηση (Identification):** Η ταυτοποίηση των συσκευών και χρηστών στο IoT αποτελεί κρίσιμο βήμα για την αποτελεσματική εφαρμογή των μηχανισμών ελέγχου πρόσβασης. Ο έλεγχος πρόσβασης βασίζεται στην ικανότητα του συστήματος να αναγνωρίζει και να επιβεβαιώνει την ταυτότητα των οντοτήτων που επιθυμούν να αλληλεπιδράσουν με το δίκτυο. Μέσω της ταυτοποίησης, διασφαλίζεται η ορθή κατανομή των δικαιωμάτων πρόσβασης και η αξιοπιστία των επικοινωνιών, ιδίως σε περιβάλλοντα με μεγάλο αριθμό διασυνδεδεμένων συσκευών [19–21].
- **Ιδιωτικότητα (Privacy) και Εμπιστευτικότητα (Confidentiality):** Η ιδιωτικότητα και η εμπιστευτικότητα αποτελούν δύο βασικές διαστάσεις για την προστασία των δεδομένων σε IoT περιβάλλοντα. Η εμπιστευτικότητα εξασφαλίζει ότι τα δεδομένα είναι προσβάσιμα μόνο από εξουσιοδοτημένα μέρη, μέσω της χρήσης κρυπτογραφικών τεχνικών. Παράλληλα, η διασφάλιση της ιδιωτικότητας των χρηστών και συσκευών, μέσω της εφαρμογής

ανώνυμων ή ψευδωνυμικών μοντέλων, είναι απαραίτητη για την προστασία από κακόβουλες επιθέσεις και τη διατήρηση της ακεραιότητας των προσωπικών δεδομένων [18–20].

- **Επιβολή Πολιτικών (Policy Enforcement):** Η επιβολή πολιτικών ασφάλειας και ιδιωτικότητας σε IoT περιβάλλοντα είναι απαραίτητη για την εξασφάλιση της συμμόρφωσης με τις κανονιστικές απαιτήσεις και τη διατήρηση της ασφάλειας. Η αναζήτηση ισορροπίας μεταξύ της εφαρμογής πολιτικών και των υπολογιστικών πόρων που απαιτούνται για αυτήν είναι μια σημαντική πρόκληση [19,20].
- **Ακεραιότητα (Integrity):** Η προστασία της ακεραιότητας των δεδομένων διασφαλίζει ότι οι πληροφορίες που μεταφέρονται στο δίκτυο δεν έχουν τροποποιηθεί ή παραπομπεί, εξασφαλίζοντας έτσι την αξιοπιστία των επικοινωνιών [18].
- **Μη Αποποίηση Ευθύνης (Non-repudiation):** Με την εφαρμογή κρυπτογραφικών μεθόδων, εξασφαλίζεται η δυνατότητα απόδειξης της εκτέλεσης ενεργειών από συγκεκριμένους χρήστες, αποτρέποντας την άρνηση συμμετοχής σε κακόβουλες δραστηριότητες [18].
- **Κινητή Ασφάλεια (Mobile Security):** Οι κινητοί κόμβοι στα IoT δίκτυα κινούνται συχνά από το ένα σύμπλεγμα στο άλλο, δημιουργώντας ανάγκες για γρήγορη ταυτοποίηση και προστασία ιδιωτικότητας. Πρωτόκολλα όπως αυτά που διαχειρίζονται την ασφάλεια κατά τη μετακίνηση κόμβων είναι κρίσιμα για την αποτροπή επιθέσεων παρακολούθησης ή παραβίασης της ιδιωτικότητας [20].
- **Ενδιάμεσο Λογισμικό Ασφάλειας (Secure Middleware):** Η ενσωμάτωση συσκευών IoT μέσω ενδιάμεσων λογισμικών (middleware) αποτελεί κρίσιμο στοιχείο για τη διαχείριση και την ασφάλεια των δεδομένων που ανταλλάσσονται στο δίκτυο. Τα middleware παρέχουν προστασία κατά τη μεταφορά δεδομένων και την επικοινωνία μεταξύ των διασυνδεδεμένων συσκευών, ωστόσο η ανάγκη για πλήρη ασφάλεια και διαλειτουργικότητα αποτελεί ακόμα σημαντική πρόκληση [20].
- **Επικαιρότητα Μηνυμάτων (Freshness):** Η διασφάλιση ότι τα μηνύματα που διακινούνται στο δίκτυο είναι πρόσφατα αποτρέπει επιθέσεις επαναμετάδοσης δεδομένων (replay attacks) [18].
- **Εμπιστοσύνη (Trust):** Η διαχείριση της εμπιστοσύνης μεταξύ των συσκευών σε ένα IoT δίκτυο είναι ζωτικής σημασίας για την αξιολόγηση της αξιοπιστίας των κόμβων και την αποτροπή κακόβουλων ενεργειών [19].
- **Προστασία από Προηγούμενες και Μελλοντικές Απειλές (Forward & Backward Secrecy):** Η προστασία αυτή αποτρέπει τη διαρροή ευαίσθητων πληροφοριών σε νέες ή αποχωρούσες συσκευές, διασφαλίζοντας την ασφάλεια των δεδομένων κατά τη μετάβαση από ή προς το δίκτυο [18].
- **Αξιοπιστία (Reliability) και Διαθεσιμότητα (Availability):** Η αξιοπιστία και η διαθεσιμότητα αποτελούν δύο σημαντικές απαιτήσεις για την αδιάλειπτη λειτουργία των IoT συστημάτων. Η αξιοπιστία σχετίζεται με την ικανότητα του συστήματος να λειτουργεί χωρίς σφάλματα και με συνέπεια, ενώ η διαθεσιμότητα διασφαλίζει ότι οι υπηρεσίες

του δικτύου είναι πάντα προσβάσιμες από εξουσιοδοτημένους χρήστες, ακόμη και υπό συνθήκες επίθεσης ή τεχνικών προβλημάτων. Η συνδυασμένη επίτευξη αξιοπιστίας και διαθεσιμότητας είναι θεμελιώδης για την ομαλή και ασφαλή λειτουργία των IoT δικτύων [18,21].

- **Ασφάλεια (Safety):** Η ασφάλεια των αυτόνομων συστημάτων είναι κρίσιμη καθώς η τυχόν μη ορθή λειτουργία τους μπορεί να προκαλέσει σοβαρές επιπτώσεις τόσο στο ίδιο το σύστημα όσο και στο φυσικό περιβάλλον. Ο έλεγχος των απρόβλεπτων συμπεριφορών αυτών των συστημάτων πρέπει να διασφαλίζεται ώστε να αποφεύγονται καταστροφικές συνέπειες, ειδικά σε συστήματα που βασίζονται σε αισθητήρες και δεδομένα από το περιβάλλον [21].
- **Ευθύνη (Responsibility):** Η ευθύνη σχέτιζεται άμεσα με την κατανομή των δικαιωμάτων πρόσβασης και των εξουσιοδοτήσεων. Η εξουσιοδότηση επιτρέπει τη διασφάλιση ότι μόνο οι αρμόδιοι χρήστες ή συσκεύες μπορούν έχουν πρόσβαση σε δεδομένα ή υπηρεσίες, και να αναλαμβάνουν την ευθύνη για τις ενέργειες τους στο δίκτυο [21].
- **Αυτο-Ανοσία (Self-Immunity):** Τα IoT συστήματα που λειτουργούν σε απομακρυσμένες ή επικίνδυνες περιοχές εκτίθενται σε φυσικές απειλές λόγω περιορισμών όπως μη αξιόπιστες επικοινωνιακές συνδέσεις, περιορισμένη φυσική προστασία και ανεπαρκή συστήματα διαχείρισης εμπιστοσύνης. Η αυτο-ανοσία είναι η ικανότητα των κόμβων να προστατεύονται από τέτοιες φυσικές επιθέσεις και να συνεχίζουν να λειτουργούν ορθά παρά τις αντιξότητες [21].

Οι παραπάνω στόχοι θέτουν τα θεμέλια για την ενίσχυση της ασφάλειας στα IoT δίκτυα, υποδεικνύοντας την ανάγκη για συνεχή έρευνα και ανάπτυξη νέων μεθόδων που θα αντιμετωπίζουν τις προκλήσεις της ασφάλειας, ιδιωτικότητας και διαλειτουργικότητας που αναδύονται στο ψηφιακό οικοσύστημα του IoT.



Σχήμα 2.4: Σημαντικά ζητήματα ασφάλειας στο IoT.

3

Βασικά Κρυπτογραφικά Δομικά Στοιχεία

3.1 Εισαγωγή στα Βασικά Κρυπτογραφικά Δομικά Στοιχεία: Κατηγορίες, Λειτουργίες και Χαρακτηριστικά

Η κρυπτογραφία διαδραματίζει κρίσιμο ρόλο στην ασφάλεια των πληροφοριακών συστημάτων, με τις δύο κύριες κατηγορίες κρυπτογραφικών αλγορίθμων να είναι οι αλγόριθμοι τμήματος (block ciphers) και οι αλγόριθμοι ροής (stream ciphers). Επίσης υπάρχουν και κατηγορίες κρυπτογραφικών αλγορίθμων που κληρονομούν χαρακτηριστικά τόσο από τους block ciphers όσο και από τους stream ciphers, οι οποίοι αποκαλούνται υβριδικοί (hybrid). Σε αυτή την κατηγορία ανήκει και ο αλγόριθμος Hummingbird-2 που αναπτύχθηκε στα πλαίσια της εργασίας. Η κάθε κατηγορία προσφέρει διαφορετικά χαρακτηριστικά και πλεονεκτήματα, καθιστώντας την κατάλληλη για διάφορες εφαρμογές. Στην παρούσα ενότητα γίνεται μία περιεκτική παρουσίαση των βασικών λειτουργιών και χαρακτηριστικών τους, ενώ σε επόμενες ενότητες θα ακολουθήσει μια λεπτομερέστερη ανάλυση.

Οι αλγόριθμοι ροής κρυπτογραφούν τα δεδομένα διαδοχικά, επεξεργαζόμενοι τα bits ένα τη φορά. Ένα από τα χαρακτηριστικά παραδείγματα αυτής την κατηγορίας είναι ο RC4, ο οποίος χρησιμοποιεί ψευδοτυχαία παραγωγή αριθμών για την παραγωγή ενός κρυπτογραφικού ρεύματος κλειδιών (keystream), το οποίο συνδυάζεται με το κείμενο μέσω της πράξης XOR για την παραγωγή του κρυπτοκειμένου [22]. Αυτού του τύπου οι αλγόριθμοι είναι ιδιαίτερα αποδοτικοί για την κρυπτογράφηση ροών δεδομένων σε πραγματικό χρόνο, όπως σε περιπτώσεις επικοινωνίας μέσω δικτύου ή streaming πολυμέσων. Παρακάτω αναφέρονται μερικά από τα πλεονεκτήματα και μειονεκτήματα τους:

- + Οι αλγόριθμοι ροής είναι ταχύτεροι σε σύγκριση με τους αλγορίθμους τμήματος, καθώς επεξεργάζονται τα δεδομένα αμέσως και χρησιμοποιούν λιγότερους υπολογιστικούς πόρους [22].

- + Λόγω της χαμηλής υπολογιστικής πολυπλοκότητας, είναι κατάλληλοι για συστήματα με περιορισμένους πόρους.
- Η χρήση της ίδιας κλειδοροής (keystream) για την κρυπτογράφηση διαφορετικών μηνυμάτων μπορεί να οδηγήσει σε ευπάθειες που διευκολύνουν την κρυπτανάλυση [22,23].
- Η ασφάλεια των αλγορίθμων ροής εξαρτάται άμεσα από την ποιότητα της ψευδοτυχαίας γεννήτριας αριθμών που χρησιμοποιείται για την παραγωγή του keystream.

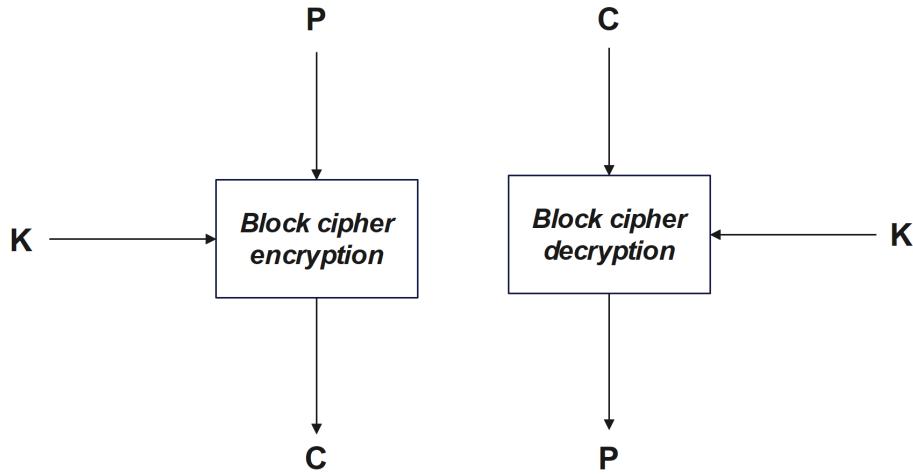
Οι αλγόριθμοι τμήματος διαφέρουν σημαντικά από τους αλγορίθμους ροής, καθώς επεξεργάζονται τα δεδομένα σε σταθερού μεγέθους μπλοκ, συνήθως 64 ή 128 bits. Ένας από του πιο διαδεδομένους αλγόριθμους αυτής της κατηγορίας είναι ο Advanced Encryption Standard (AES), ο οποίος έχει καθιερωθεί ως το παγκόσμιο πρότυπο για την κρυπτογράφηση δεδομένων [24]. Οι αλγόριθμοι τμήματος είναι ιδανικοί για εφαρμογές όπου απαιτείται η κρυπτογράφηση μεγάλων όγκων δεδομένων, όπως στην κρυπτογράφηση αρχείων ή τη διαβίβαση μεγάλων μηνυμάτων. Παρακάτω αναφέρονται μερικά από τα πλεονεκτήματα και μειονεκτήματα τους:

- + Παρέχουν υψηλό επίπεδο ασφάλειας λόγω της πολυπλοκότητας των μετασχηματισμών που εφαρμόζουν στα μπλοκ δεδομένων [24].
- + Είναι καταλληλότεροι για εφαρμογές που απαιτούν τη διασφάλιση της ακεραιότητας και της εμπιστευτικότητας σε μεγάλους όγκους δεδομένων, όπως στην κρυπτογράφηση βάσεων δεδομένων.
- Η κρυπτογράφηση μπλοκ απαιτεί μεγαλύτερους υπολογιστικούς πόρους, καθιστώντας τους αλγόριθμους τμήματος λιγότερο αποδοτικούς σε εφαρμογές που απαιτούν επεξεργασία δεδομένων σε πραγματικό χρόνο [22].
- Η ταχύτητα επεξεργασίας είναι χαμηλότερη σε σύγκριση με τους αλγόριθμους ροής, γεγονός που μπορεί να προκαλέσει καθυστερήσεις σε συγκεκριμένες εφαρμογές.

Συνολικά, οι αλγόριθμοι ροής υπερέχουν σε εφαρμογές που απαιτούν γρήγορη κρυπτογράφηση και χαμηλή πολυπλοκότητα, όπως η κρυπτογράφηση δικτυακών ροών δεδομένων. Αντίθετα, οι αλγόριθμοι τμήματος είναι καταλληλότεροι για περιπτώσεις όπου απαιτείται αυξημένη ασφάλεια και επεξεργασία μεγάλων δεδομένων, όπως σε κρυπτογράφηση αρχείων ή βάσεων δεδομένων. Παρά τα πλεονεκτήματα των αλγορίθμων ροής, οι αλγόριθμοι τμήματος χρησιμοποιούνται συχνότερα σε εφαρμογές υψηλής ασφάλειας, καθώς προσφέρουν μεγαλύτερη ευελιξία στη διαχείριση της εμπιστευτικότητας και της ακεραιότητας [22].

3.2 Αλγόριθμοι Τμήματος

Οι αλγόριθμοι τμήματος αποτελούν τη βάση για πολλές σύγχρονες κρυπτογραφικές εφαρμογές, καθώς προσφέρουν υψηλά επίπεδα ασφάλειας μέσω της επεξεργασίας των δεδομένων σε σταθερού μεγέθους μπλοκ. Η λειτουργία τους βασίζεται σε συγκεκριμένες αρχιτεκτονικές, οι οποίες καθορίζουν τον τρόπο με τον οποίο τα δεδομένα υποβάλλονται σε κρυπτογραφικούς μετασχηματισμούς. Από τις πιο σημαντικές αρχιτεκτονικές είναι η Feistel δομή, η οποία χρησιμοποιείται στον αλγόριθμο DES, και το Substitution-Permutation Network (SPN), που αποτελεί τη βάση για τον αλγόριθμο AES [25]. Ακόμα μία αρχιτεκτονική είναι και η Add-Rotate-XOR (ARX) [25]. Αυτές οι αρχιτεκτονικές παρέχουν διαφορετικούς μηχανισμούς επεξεργασίας των δεδομένων και εξασφαλίζουν διαφορετικά επίπεδα ανθεκτικότητας σε επιθέσεις.

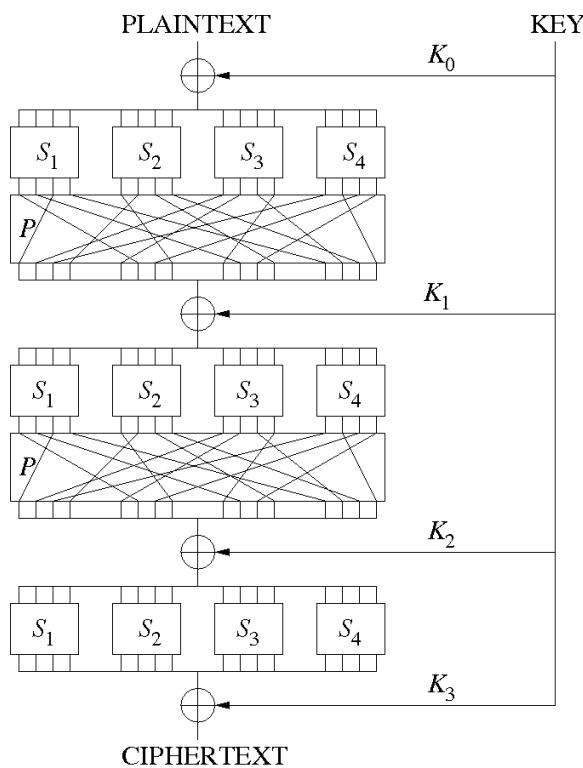


Σχήμα 3.1: Λειτουργία Κρυπταλγόριθμου Τμήματος.

3.2.1 Substitution-Permutation Network Δομή

Η Substitution-Permutation Network (SPN), γνωστή και ως Δίκτυο Υποκατάστασης – Αντιμετάθεσης, αποτελεί ένα από τα πιο σημαντικά θεμέλια στον τομέα της σύγχρονης συμμετρικής κρυπτογραφίας. Αυτή η δομή εισήχθη ως απάντηση στην ανάγκη για ένα σύστημα που θα συνδύαζε αποτελεσματικά τις έννοιες της σύγχυσης (confusion) και της διάχυσης (diffusion) που είχε περιγράψει ο Claude Shannon το 1945 [26]. Η σύγχυση επιτυγχάνεται μέσω της μη γραμμικής υποκατάστασης, ενώ η διάχυση μέσω της γραμμικής αντιμετάθεσης [27]. Η δομή SPN συνδυάζει αυτές τις ιδιότητες, κάνοντας χρήση ειδικών λειτουργιών υποκατάστασης και αντιμετάθεσης σε διαδοχικούς γύρους για να μετασχηματίσει δεδομένα και να προσφέρει υψηλό επίπεδο ασφάλειας.

Τα SPN καθιερώθηκαν ως το πιο αποτελεσματικό μέσο για την εξασφάλιση της ασφάλειας των κρυπτογραφικών αλγορίθμων, και έχει χρησιμοποιηθεί σε πολλούς γνωστούς αλγορίθμους, όπως ο AES, ο PRESENT, και ο NOEKEON [25]. Η διάδοση της χρήσης αυτής της δομής οδήγησε σε σημαντική ανάπτυξη της κρυπτογραφικής επιστήμης, ιδίως όσον αφορά τις συσκευές με περιορισμένους πόρους, όπως οι έξυπνες κάρτες και οι αισθητήρες.

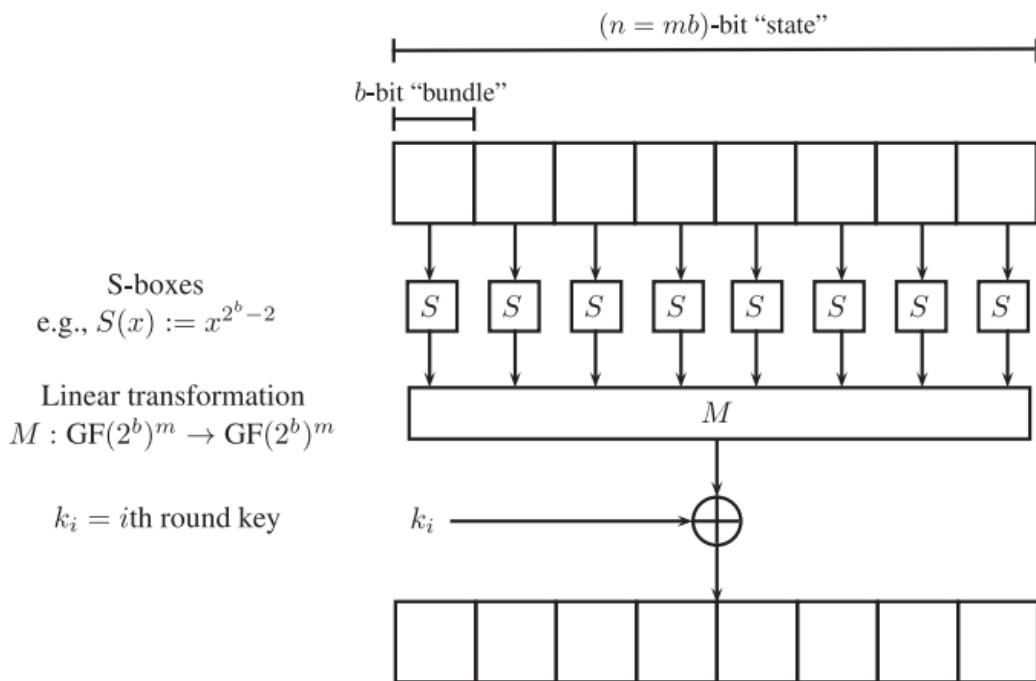


Σχήμα 3.2: Βασική αρχιτεκτονική SPN.

Δομή και Στάδια SPN

Ένα τυπικό SPN αποτελείται από 1 γύρους, και κάθε γύρος περιλαμβάνει τρία βασικά στάδια:

- Υποκατάσταση (Substitution) μέσω S-Boxes:** Κάθε μπλοκ του δεδομένου χωρίζεται σε μικρότερα μπλοκ (συνήθως 4, 8 ή 16) τα οποία υποβάλλονται σε μη γραμμική υποκατάσταση. Αυτή η υποκατάσταση πραγματοποιείται μέσω των λεγόμενων S-boxes (Substitution Boxes), οι οποίες εφαρμόζουν μια μη γραμμική συνάρτηση υποκατάστασης (π.χ., $S(x) = x^{2^b - 2}$). Τα S-boxes εξασφαλίζουν τη σύγχυση (confusion) μεταξύ της εισόδου και της εξόδου, δημιουργώντας πολύπλοκες σχέσεις μεταξύ τους [28].
- Αντιμετάθεση (Permutation) μέσω P-boxes:** Μετά την υποκατάσταση, τα αποτελέσματα των S-boxes περνούν από τα P-boxes (Permutation Boxes), τα οποία πραγματοποιούν γραμμικές αντιμεταθέσεις (bitwise permutations). Τα P-boxes φροντίζουν να μετακινούν τα bits έτσι ώστε κάθε bit εξόδου από ένα S-box να εισέρχεται σε ένα διαφορετικό S-box στον επόμενο γύρο. Αυτό βοηθά στη διάχυση της πληροφορίας σε όλο το μπλοκ δεδομένων, κάτι που καθιστά τον αλγόριθμο ανθεκτικό σε επιθέσεις ανάλυσης [27].
- Προσθήκη Κλειδιού (Key Mixing) μέσω XOR:** Σε κάθε γύρο, ένα κλειδί (round key) παράγεται από το κύριο κλειδί (master key) μέσω μιας διαδικασίας γνωστής ως «Key schedule». Το round key προστίθεται στο μπλοκ δεδομένων μέσω της λειτουργίας XOR, προσθέτοντας ένα επιπλέον επίπεδο σύγχυσης [27].



Σχήμα 3.3: Ένας γύρος ενός SPN [28].

Το αποτέλεσμα αυτού του συνδυασμού υποκατάστασης, αντιμετάθεσης και προσθήκης κλειδιού είναι η δημιουργία ενός ισχυρού κρυπτογραφικού συστήματος που εξασφαλίζει ότι ακόμη και η παραμικρή αλλαγή στην είσοδο ή στο κλειδί θα επιφέρει μεγάλες αλλαγές στην έξοδο, εξασφαλίζοντας την ασφάλεια του συστήματος.

S-boxes

Τα S-boxes είναι ένας πίνακας υποκατάστασης που λαμβάνει ως είσοδο ένα μικρό μπλοκ από bits (π.χ. 4, 8 ή 16 bits) και το αντικαθιστά με άλλο μπλοκ από bits ίσης διάστασης. Η υποκατάσταση αυτή δεν είναι απλώς μια ανακατάταξη των bits αλλά μια μη γραμμική λειτουργία που εξασφαλίζει ότι καμία απλή γραμμική συνάρτηση δεν μπορεί να περιγράψει τη σχέση μεταξύ των bits εισόδου και εξόδου [25].

Για παράδειγμα, στον AES, το S-box χρησιμοποιεί τη συνάρτηση $S(x) = x^{2^8 - 2}$, η οποία εφαρμόζεται σε κάθε byte της εισόδου για να δημιουργήσει το αντίστοιχο byte εξόδου. Η μη γραμμική αυτή λειτουργία προσθέτει σημαντική ασφάλεια στον αλγόριθμο, καθώς κάθε αλλαγή στην είσοδο επηρεάζει σχεδόν όλες τις εξόδους του S-box με τρόπο που είναι δύσκολο να προβλεφθεί [26]. Τα καλά σχεδιασμένα S-boxes πρέπει να πληρούν ορισμένες προδιαγραφές για να εξασφαλίζουν την ασφάλεια:

- **Μη γραμμικότητα:** Πρέπει να είναι μη γραμμικά, ώστε να παρέχουν σύγχυση και να μην επιτρέπουν στο επιτιθέμενο να εκμεταλλευτεί γραμμικές σχέσεις μεταξύ εισόδου και εξόδου.

- **Αντίσταση σε επιθέσεις διαφορικής κρυπτανάλυσης:** Κάθε μικρή αλλαγή στην είσοδο ενός S-box πρέπει να επηρεάζει σε μεγάλο βαθμό την έξοδο, κάτι που εξασφαλίζει την αντίσταση σε επιθέσεις διαφορικής κρυπτανάλυσης.
- **Αναστρεψιμότητα:** Τα S-boxes πρέπει να είναι αναστρέψιμα για να επιτρέπεται η αποκρυπτογράφηση των δεδομένων.

Σε συνδυασμό με τις λειτουργίες αντιμετάθεσης (P-boxes), τα S-boxes διασφαλίζουν τη σωστή διάχυση της πληροφορίας στο ciphertext και την ανθεκτικότητα του συστήματος σε επιθέσεις, καθιστώντας τα ένα από τα βασικά εργαλεία σε σύγχρονους κρυπτογραφικούς αλγόριθμους [27].

Αλγόριθμος AES

Ένα παράδειγμα κρυπτογραφικού αλγορίθμου που βασίζεται στη δομή SPN όπως ήδη έχουμε αναφέρει είναι ο AES, ο οποίος χρησιμοποίει μπλοκ δεδομένων 128 bits και κλειδιά 128, 192 ή 256 bits. Ο AES εφαρμόζει την κρυπτογράφηση σε διαδοχικούς γύρους, όπου κάθε γύρος περιλαμβάνει τα εξής βήματα:

- **Υποκατάσταση μέσω ενός S-box.**
- **Αντιμετάθεση μέσω των λειτουργιών Μετατόπιση Σειρών (Shift Rows) και Μίξη Στηλών (Mix Columns).**
- **Προσθήκη κλειδιού μέσω της λειτουργίας XOR (Add Round Key).**

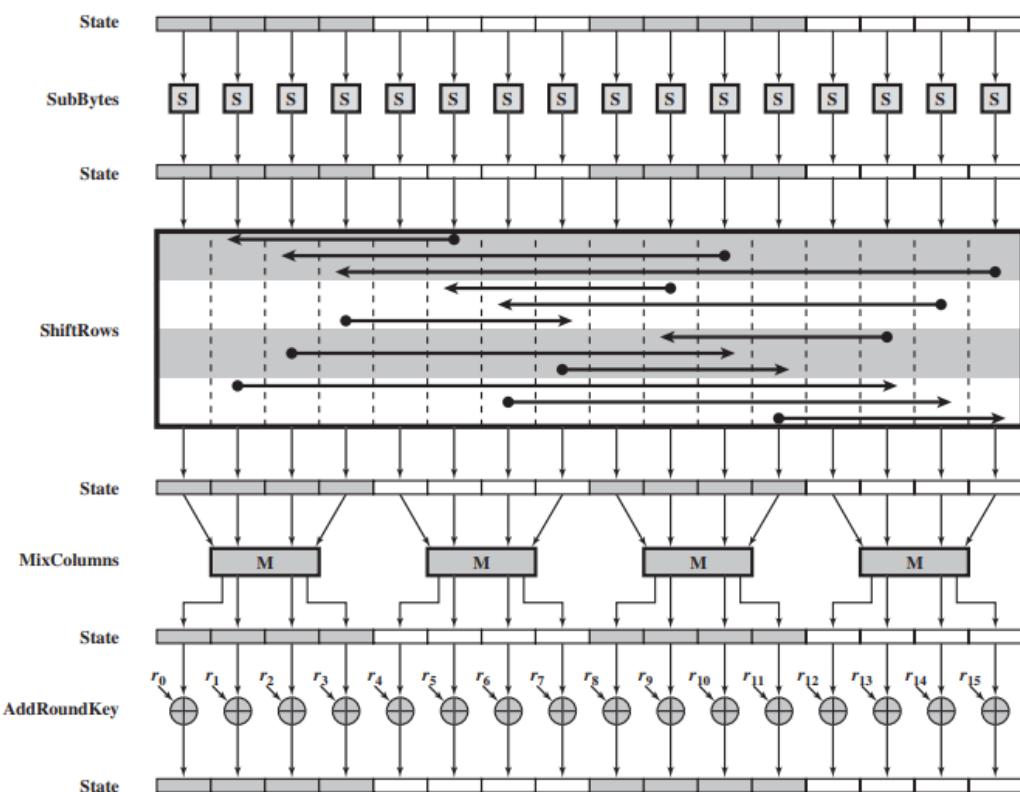
Κάθε S-box στον AES βασίζεται στη μη γραμμική συνάρτηση $S(x) = x^{2^8-2}$ και έχει σχεδιαστεί για να παρέχει υψηλή ασφάλεια απέναντι σε επιθέσεις όπως η διαφορική και γραμμική κρυπτανάλυση [25].

Ανθεκτικότητα του SPN Απέναντι σε Επιθέσεις

Τα SPN είναι ιδιαίτερα ανθεκτικά σε επιθέσεις κρυπτανάλυσης, όπως:

- **Γραμμική κρυπτανάλυση (Linear Cryptanalysis):** Αυτή η μορφή επίθεσης εκμεταλλεύεται πιθανές γραμμικές σχέσεις μεταξύ του plaintext, του ciphertext, και του κλειδιού. Τα SPN αντιμετωπίζουν αυτές τις επιθέσεις χάρη στις μη γραμμικές ιδιότητες των S-boxes, οι οποίες καταρρίπτουν οποιαδήποτε γραμμική συσχέτιση [27].
- **Διαφορική κρυπτανάλυση (Differential Cryptanalysis):** Οι επιθέσεις αυτού του είδους εκμεταλλεύονται τις σχέσεις μεταξύ διαφορών σε ζεύγη εισόδων και εξόδων. Η διαχυτική φύση των P-boxes στα SPN καθιστά δύσκολο για τους επιτιθέμενους να προσδιορίσουν τις διαφορές στο plaintext που προκαλούν συγκεκριμένες αλλαγές στο ciphertext [28].

Συνολικά, η δομή SPN έχει σχεδιαστεί με τρόπο που παρέχει ισχυρή αντοχή σε επιθέσεις, καθιστώντας την ιδανική για τη χρήση σε σύγχρονους αλγορίθμους κρυπτογράφησης.



Σχήμα 3.4: AES Γύρος Κρυπτογράφησης [22].

Συμπέρασμα

Η SPN δομή είναι μία από τις πιο αποτελεσματικές και ανθεκτικές μεθόδους για την κρυπτογράφηση δεδομένων, επιτυγχάνοντας ισχυρή σύγχυση και διάχυση μέσω των S-boxes και P-boxes αντίστοιχα. Ο συνδυασμός αυτών των λειτουργιών σε διαδοχικούς γύρους παρέχει τη βάση για την ανθεκτικότητα πολλών από τους σύγχρονους αλγορίθμους κρυπτογράφησης, όπως ο AES. Παρότι οι επιθέσεις όπως η γραμμική και διαφορική κρυπτανάλυση παραμένουν ισχυρές, τα SPN παραμένουν ιδιαίτερα ανθεκτικά κυρίως λόγω της καλά σχεδιασμένης τους δομής.

3.2.2 Feistel Δομή

Η δομή Feistel είναι μια συμμετρική δομή που χρησιμοποιείται ευρέως στο σχεδιασμό συμμετρικών κρυπτογραφικών αλγορίθμων μπλοκ, και ονομάζεται έτσι από τον Horst Feistel, έναν Γερμανό φυσικό και κρυπτογράφο. Αυτή η δομή αποτελεί τη βάση για πολλούς κρυπτογραφικούς αλγορίθμους, συμπεριλαμβανόμενων των DES (Data Encryption Standard), KASUMI, καθώς και άλλων όπως οι GOST, Blowfish, και TwoFish [29].

Ιστορικές & Σημαντικές Εφαρμογές των Δομών Feistel

Οι δομές Feistel έγιναν ευρέως γνωστές με την εφαρμογή τους στον αλγόριθμο Lucifer της IBM, που αναπτύχθηκε το 1973 από τους Horst Feistel και Don Coppersmith. Αργότερα, αυτές οι δομές κέρδισαν αξιοπιστία με την υιοθέτηση του DES από την Ομοσπονδιακή Κυβέρνηση των ΗΠΑ το 1976, έναν αλγόριθμο βασισμένο στο Lucifer με τροποποιήσεις από την NSA.

Οι δομές Feistel χρησιμοποιούνται σε πολλούς σύγχρονους αλγόριθμους μπλοκ κρυπτογράφησης, όπως ο KASUMI, ο οποίος χρησιμοποιείται σε συστήματα Global System for Mobile Communications (GSM) και Universal Mobile Telecommunications System (UMTS). Ο KASUMI χρησιμοποιεί μια παραλλαγή της δομής Feistel με 8 γύρους και αποτελεί μια σημαντική εφαρμογή της δομής αυτής στην κινητή τηλεφωνία [29]. Οι παραλλαγές των δομών Feistel, όπως οι μη ισορροπημένες δομές Feistel, οπού τα δύο μέρη δεν έχουν το ίδιο μέγεθος, χρησιμοποιούνται επίσης σε ειδικές περιπτώσεις, όπως στον αλγόριθμο Skipjack που αναπτύχθηκε από την Υπηρεσία Εθνικής Ασφάλειας των ΗΠΑ.

Γενική Δομή Feistel

Μια κρυπτογράφηση Feistel χωρίζει το κείμενο σε δύο ίσα μέρη, τα οποία ονομάζονται L_0 (αριστερό μέρος) και R_0 (δεξιό μέρος). Σε κάθε γύρο i , εφαρμόζεται μια λειτουργία γύρου F στο δεξιό μέρος R_i , και το αποτέλεσμα \oplus με το αριστερό μέρος L_i . Στη συνέχεια, τα δύο μέρη ανταλλάσσονται για τον επόμενο. Μαθηματικά, αυτό εκφράζεται ως εξής:

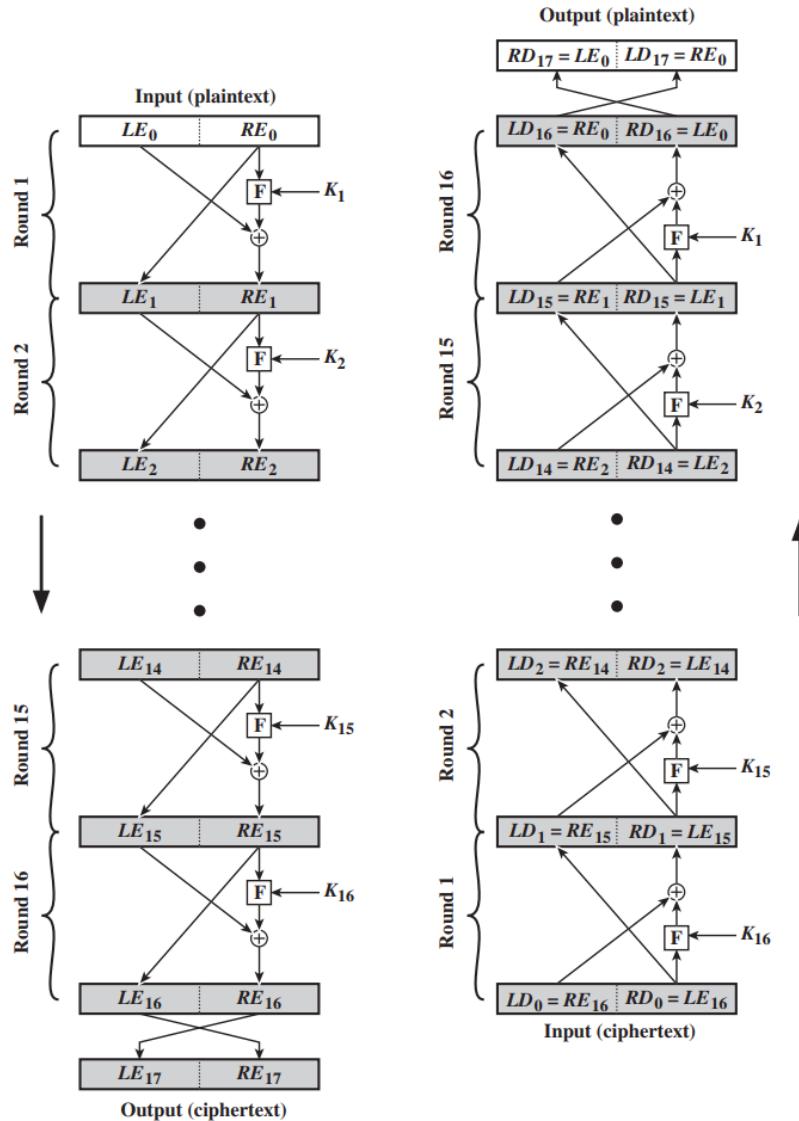
$$L_{i+1} = R_i,$$

$$R_{i+1} = L_i \oplus F(R_i, K_i)$$

οπού K_i είναι το υποκλειδί για τον γύρο i . Αυτή η διαδικασία επαναλαμβάνεται για αρκετούς γύρους ώστε να επιτευχθεί η ασφάλεια [22].

Ένα σημαντικό χαρακτηριστικό της δομής Feistel είναι ότι ο ίδιος αλγόριθμος μπορεί να χρησιμοποιηθεί και για αποκρυπτογράφηση αλλά τα υποκλείδια εφαρμόζονται με αντίστροφη σειρά. Αυτό κάνει τις δομές Feistel αποδοτικές και πρακτικές για εφαρμογή τόσο σε υλικό όσο

και σε λογισμικό. Στην παρακάτω εικόνα (εικόνα 3.5) βλέπουμε τόσο την κρυπτογράφηση όσο και την αποκρυπτογράφηση Feistel των 16 γύρων.



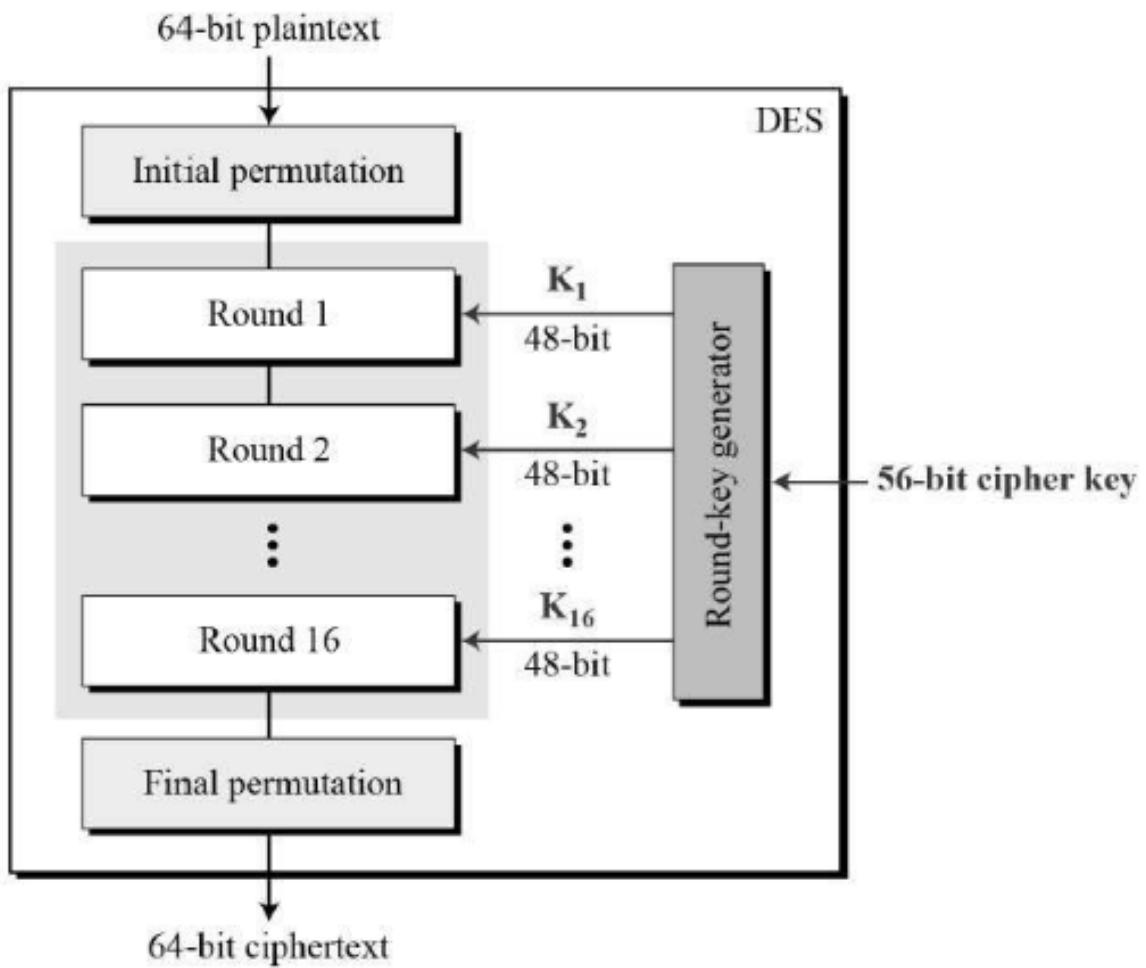
Σχήμα 3.5: Κρυπτογράφηση και αποκρυπτογράφηση Feistel (16 γύροι) [22].

Αλγόριθμος DES

Ο DES είναι ένας συμμετρικός αλγόριθμος κρυπτογράφησης μπλοκ που αναπτύχθηκε τη δεκαετία του '70 και χρησιμοποιήθηκε ευρέως ως το πρότυπο κρυπτογράφησης δεδομένων για πολλές δεκαετίες. Σχεδιάστηκε από την IBM και υιοθετήθηκε ως το επίσημο πρότυπο των Ηνωμένων Πολιτειών το 1977. Η γενική λειτουργία του φαίνεται στην εικόνα 3.6.

Τα κύρια Χαρακτηριστικά του DES είναι τα εξής:

1. Μέγεθος Μπλοκ και Κλειδιού:



Σχήμα 3.6: Ο αλγόριθμος DES.

- Ο DES λειτουργεί με 64-bit μπλοκ δεδομένων, δηλαδή χωρίζει τα δεδομένα που θέλουμε να κρυπτογραφήσουμε σε κομμάτια των 64-bit και τα επεξεργάζεται ξεχωριστά.
 - Χρησιμοποιεί ένα κλειδί 64-bit, εκ των οποίων τα 56-bit χρησιμοποιούνται πραγματικά για την κρυπτογράφηση, ενώ τα υπόλοιπα 8-bit χρησιμοποιούνται για έλεγχο parity [29].
2. **Δομή Feistel:** Ο DES είναι βασισμένος στη δομή Feistel όπως ανάφεραμε, πράγμα που σημαίνει ότι χωρίζει το μπλοκ δεδομένων σε δύο ίσα μέρη: το αριστερό (L) και το δεξί (R). Κατά τη διάρκεια κάθε γύρου, το δεξί μέρος (R) υποβάλλεται σε μια συνάρτηση που συνδυάζεται με ένα υποκλειδί, και το αποτέλεσμα \oplus με το αριστερό μέρος (L). Μετά από αυτή τη διαδικασία, τα δύο μέρη αντιστρέφονται και συνεχίζεται ο επόμενος γύρος [22].
3. **Πλήθος Γύρων:** Ο DES περιλαμβάνει 16 γύρους επεξεργασίας. Κάθε γύρος αυξάνει τη δυσκολία της ανάλυσης του αλγορίθμου, καθώς κάθε γύρος εισάγει σύγχυση και διάχυση, καθιστώντας τον DES πιο ασφαλή [22].
4. **Κρυπτογραφική Συνάρτηση:** Η κύρια συνάρτηση του DES (F) είναι αρκετά πολύπλοκη και περιλαμβάνει:
- **Επέκταση (Expansion) :** Το δεξί μέρος (32-bit) επεκτείνεται σε 48-bit.
 - **Εξόρυξη Υποκλειδιού :** Ένα μοναδικό υποκλειδί χρησιμοποιείται σε κάθε γύρο.
 - **Εφαρμογή S-boxes:** Μετά από την \oplus με το υποκλειδί, τα δεδομένα εισάγονται σε 8 S-boxes, που μειώνουν τα 48-bit σε 32-bit μέσω μη αναστρέψιμων υποκαταστάσεων.
 - **Αναδιάταξη:** Τέλος, γίνεται μια αναδιάταξη των bit, εξασφαλίζοντας τη διάχυση των δεδομένων.
5. **Αρχική και Τελική Αντιστροφή:**
- Πριν από τους 16 γύρους, γίνεται μια αρχική αντιστροφή των bit του μπλοκ δεδομένων.
 - Μετά τους γύρους, εφαρμόζεται μια τελική αντιστροφή για να παραχθεί το τελικό κρυπτογράφημα [22].
6. **Αποκρυπτογράφηση:** Η διαδικασία της αποκρυπτογράφησης είναι σχεδόν ίδια με την κρυπτογράφηση, αλλά τα υποκλειδιά χρησιμοποιούνται σε αντίστροφη σειρά. Αυτό οφείλεται στη δομή Feistel, η οποία επιτρέπει την εύκολη αναστροφή της κρυπτογράφησης για την αποκρυπτογράφηση [22].

Παρότι ο DES ήταν εξαιρετικά σημαντικός στην ιστορία της κρυπτογραφίας, η χρήση ενός 56-bit κλειδιού τον καθιστά ευάλωτο σε επιθέσεις brute force. Σήμερα, έχουν αναπτυχθεί νέοι αλγόριθμοι όπως ο 3DES για να βελτιώσουν την ασφάλεια [29].

Σχεδιαστικές Προσεγγίσεις για τις Δομές Feistel

Η δομή Feistel προσφέρει διάφορα σχεδιαστικά πλεονεκτήματα:

- **Ιδιότητα αναστρεψιμότητας:** : Η δομή Feistel εγγυάται την αναστρεψιμότητα, που σημαίνει ότι η κρυπτογράφηση και η αποκρυπτογράφηση μπορούν να εκτελέσουν παρόμοιες λειτουργίες.
- **Ευελιξία της λειτουργίας γύρου:** Η λειτουργία γύρου F μπορεί να είναι οποιαδήποτε συνάρτηση που ικανοποιεί τις απαιτήσεις ασφάλειας, καθώς η ίδια η αναστρεψιμότητα διασφαλίζεται από τη δομή και όχι από την αναστρεψιμότητα της λειτουργίας F [29].

Η δομή Feistel είναι κεντρική σε πολλούς κρυπτογραφικούς αλγόριθμους, καθιστώντας την καίρια για την ανάπτυξη ασφαλών συστημάτων κρυπτογράφησης. Στηρίζεται στην ιδέα της σύγχυσης και της διάχυσης και συνδυάζει αυτές τις δύο ιδιότητες με τρόπο που διασφαλίζει την κρυπτογραφική ασφάλεια του συστήματος.

Η δομή Feistel έχει αναλυθεί εκτενώς από κρυπτογράφους. Ο Michael Luby και ο Charles Rackoff απέδειξαν ότι αν η λειτουργία είναι κρυπτογραφικά ασφαλής ψευδοτυχαία συνάρτηση, τότε τρεις γύροι είναι αρκετοί για να καταστήσουν την κρυπτογράφηση μια ψευδοτυχαία απεικόνιση. Τέσσερις γύροι είναι αρκετοί για να καταστήσουν την κρυπτογράφηση «ισχυρή» ψευδοτυχαία απεικόνιση, διασφαλίζοντας την ασφάλεια ακόμη και απέναντι σε επιθέσεις αντιστροφής [29].

Οι παραπάνω επιστημονικές θεμελιώσεις κάνουν τις δομές Feistel μια από τις πιο ασφαλείς και αξιόπιστες μεθόδους για την κατασκευή αλγορίθμων κρυπτογράφησης τμήματος.

3.2.3 Add-Rotate-XOR Δομή

Οι Add-Rotate-XOR (ARX) δομές αποτελούν έναν δημοφιλή σχεδιασμό για αλγορίθμους κρυπτογράφησης και κατακερματισμού (hash), οι οποίοι στηρίζονται αποκλειστικά σε τρεις βασικές λειτουργίες:

- **Προσθήκη (Addition):** Η προσθήκη πραγματοποιείται σε λέξεις δεδομένων με όριο 2^n , δηλαδή, οι αριθμοί προστίθενται και αν υπερβούν το ανώτερο όριο, το κρατούμενο απορρίπτεται. Αυτή η πράξη είναι μη γραμμική και εισάγει μεγάλη σύγχυση στα δεδομένα, καθιστώντας δυσκολότερη την πρόβλεψη του αποτελέσματος.
- **Περιστροφή (Rotation):** Η κυκλική μετατόπιση bit εντός μιας λέξης δεδομένων είναι μια πράξη που μετακινεί τα bits, αλλά δεν αλλάζει τα περιεχόμενά τους. Αυτή η λειτουργία προσφέρει διάχυση και είναι πολύ σημαντική για την αύξηση της αλληλεπίδρασης μεταξύ των bits.
- **Λογική πράξη XOR:** Η πράξη XOR εφαρμόζεται μεταξύ δύο λέξεων δεδομένων, παράγοντας εξόδους στις οποίες τα bits είναι 1 μόνο όταν τα αντίστοιχα bits των εισόδων διαφέρουν. Πρόκειται για μια γραμμική λειτουργία που προσφέρει επιπλέον σύγχυση στο κρυπτογραφημένο κείμενο [25].

Αυτές οι τρεις λειτουργίες επιτρέπουν στους αλγορίθμους αυτούς να έχουν ταχύτητα και αποτελεσματικότητα, είτε υλοποιούνται σε hardware είτε σε software, ενώ προσφέρουν ασφάλεια ενάντια σε επιθέσεις χρονομέτρησης (timing attacks). Η ARX δομή είναι ιδανική για ελαφρούς αλγορίθμους κρυπτογράφησης που απαιτούν χαμηλό υπολογιστικό κόστος, γεγονός που την καθιστά δημοφιλή σε εφαρμογές όπως το διαδίκτυο των πραγμάτων [25].

Οι ARX αλγόριθμοι έχουν επίσης το πλεονέκτημα ότι δεν απαιτούν S-boxes, τα οποία χρησιμοποιούνται σε άλλους κρυπτογραφικούς αλγόριθμους για την παροχή μη γραμμικότητας. Αυτή η απλότητα καθιστά τους ARX αλγόριθμους εύκολους στην υλοποίηση και αρκετά αποδοτικούς σε υπολογιστικούς πόρους [30]. Επιπλέον οι αλγόριθμοι ARX δεν εξαρτώνται από το χρόνο εκτέλεσης για την ασφάλεια τους, καθιστώντας τους ανθεκτικούς σε τέτοιες επιθέσεις. Μερικά παραδείγματα χρήσης της δομής ARX είναι τα εξής:

- **ChaCha20 :** Ένας από τους πιο γνωστούς αλγορίθμους ARX, ο ChaCha20, χρησιμοποιείται ευρέως για κρυπτογραφία υψηλής απόδοσης και ασφάλειας. Το κλειδί του αλγόριθμου είναι η ταχύτητα και η αποδοτικότητα, ιδιαίτερα σε συστήματα όπου οι χρονικές καθυστερήσεις μπορεί να αποτελούν κίνδυνο.
- **SPECK :** Ένας άλλος γνωστός ARX αλγόριθμος, ο SPECK, έχει σχεδιαστεί για ελαφρές εφαρμογές, και χρησιμοποιεί περιστροφές και προσθέσεις για να δημιουργήσει ισχυρούς κρυπτογραφικούς μηχανισμούς [30].

Οι αλγόριθμοι ARX είναι ανθεκτικοί σε πολλές παραδοσιακές επιθέσεις κρυπτογράφησης λόγω της απλότητας των λειτουργιών τους. Ωστόσο, υπάρχουν ορισμένες επιθέσεις που έχουν σχεδιαστεί ειδικά για ARX δομές, όπως η περιστροφική κρυπτανάλυση (rotational cryptanalysis), η οποία προσπαθεί να εκμεταλλευτεί την περιοδικότητα της περιστροφής των bit [31]. Μια άλλη γνωστή επίθεση που αποτελεί επέκταση της περιστροφικής κρυπτανάλυσης είναι η γνωστή Rotational-XOR επίθεση ή αλλιώς RX κρυπτανάλυση και χρησιμοποιείται για την ανάλυση ARX αλγορίθμων. Οι επιθέσεις RX βασίζονται στη μελέτη του τρόπου με τον οποίο οι περιστροφές και οι πράξεις XOR αλληλοεπιδρούν με σταθερά bits που εισάγονται σε κάθε γύρο ενός κρυπτογραφικού αλγόριθμου [30].

Οι ARX δομές αποτελούν βασικό συστατικό πολλών σύγχρονων ελαφρών κρυπτογραφικών αλγορίθμων λόγω την απλότητας, της αποδοτικότητας και της ασφάλειας τους. Ωστόσο, πρέπει να λαμβάνονται υπόψη νέες επιθέσεις όπως η RX κρυπτανάλυση, που στοχεύουν στις δομές αυτές.

3.2.4 Τρόποι Λειτουργίας των Block Ciphers

Ένας αλγόριθμος τμήματος δέχεται ως είσοδο ένα μπλοκ δεδομένων σταθερού μήκους b bits και ένα κλειδί k , και παράγει ένα κρυπτογραφημένο μπλοκ μήκους b bits. Ωστόσο, όταν το προς κρυπτογράφηση δεδομένο υπερβαίνει τα b bits, ο αλγόριθμος τμήματος μπορεί να εφαρμοστεί σπάζοντας το δεδομένο σε μπλοκ μεγέθους b bits. Όταν κρυπτογραφούνται πολλαπλά μπλοκ δεδομένων χρησιμοποιώντας το ίδιο κλειδί, προκύπτουν ορισμένα ζητήματα ασφαλείας [22]. Το Εθνικό Ινστιτούτο Προτύπων και Τεχνολογίας (NIST) έχει ορίσει πέντε τρόπους λειτουργίας (modes of operation), προκειμένου να λυθούν τα ζητήματα αυτά [22]. Αυτός ο τρόπος λειτουργίας είναι μια τεχνική που ενισχύει την απόδοση ενός κρυπτογραφικού αλγορίθμου ή προσαρμόζει τον αλγόριθμο σε μια συγκεκριμένη εφαρμογή, όπως η κρυπτογράφηση μιας ακολουθίας μπλοκ δεδομένων ή ενός ρεύματος δεδομένων (data stream) [22]. Οι πέντε βασικοί τρόποι λειτουργίας που έχουν καθοριστεί από το NIST είναι οι παρακάτω :

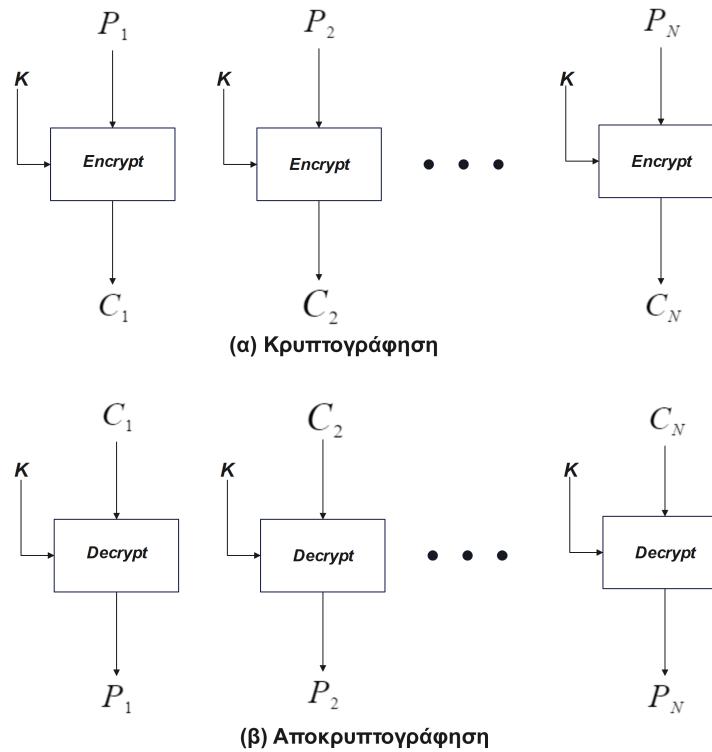
- **Electronic Codebook (ECB)**
- **Cipher Block Chaining (CBC)**

- **Cipher Feedback (CFB)**
- **Output Feedback (OFB)**
- **Counter (CTR)**

Ένας ακόμα τρόπος λειτουργίας που προτάθηκε από την IEEE (IEEE Std 1619-2007) αλλά και από το NIST είναι ο λεγόμενος XTS-AES λόγω ειδικών, σε σχέση με την κρυπτογράφηση, δεδομένων επικοινωνίας, απαιτήσεων που έχουν καταγραφεί για την κρυπτογράφηση αποθηκευμένων δεδομένων σε σκληρό δίσκο [12].

Electronic Codebook

Η τεχνική λειτουργίας ECB (εικόνα 3.7) είναι η απλούστερη των αλγορίθμων τμήματος. Στον τρόπο αυτό, το δεδομένο χωρίζεται σε μπλοκ σταθερού μεγέθους, και κάθε μπλοκ κρυπτογραφείται ξεχωριστά χρησιμοποιώντας το ίδιο κλειδί. Ο όρος «codebook» προέρχεται από την ιδέα ότι για ένα δεδομένο κλειδί, υπάρχει ένα μοναδικό κρυπτοκείμενο για κάθε μπλοκ δεδομένων b bits. Αυτό σημαίνει ότι μπορούμε να φανταστούμε ένα τεράστιο «βιβλίο κωδικών», όπου για κάθε πιθανό συνδυασμό των b bits υπάρχει ένα αντίστοιχο κρυπτοκείμενο [22]. Στη



Σχήμα 3.7: Electronic Codebook λειτουργία.

διαδικασία κρυπτογράφησης, το μήνυμα χωρίζεται σε b bits μπλοκ, και αν το τελευταίο είναι μικρότερο από b bits, γίνεται προσθήκη padding για να συμπληρωθεί το μέγεθος. Η αποκρυπτογράφηση γίνεται με τον ίδιο τρόπο κρυπτογραφώντας ή αποκρυπτογραφώντας κάθε μπλοκ

ξεχωριστά χρησιμοποιώντας το ίδιο κλειδί [22]. Το ECB ορίζεται ως εξής :

$$C_j = E(K, P_j), \quad j = 1, \dots, N$$

$$P_j = D(K, C_j), \quad j = 1, \dots, N$$

Η εφαρμογή του ECB διευκολύνεται χάρη στην απλότητά του αλλά και στο ότι δεν απαιτεί επιπλέον βήματα συγχρονισμού ή αλγόριθμους για τη διασύνδεση των μπλοκ. Αυτό ανένει αρκετά την απόδοση της κρυπτογράφησης πράγμα που τον κάνει να είναι κατάλληλος για μικρές ποσότητες δεδομένων. Στο αντίοδα βέβαια όταν ένα μπλοκ δεδομένων εμφανιστεί παραπάνω από μία φορά στο μήνυμα, το κρυπτοκείμενο που παράγεται θα είναι το ίδιο κάθε φορά. Αυτό δημιουργεί ευπάθειες, καθώς επιτρέπει σε έναν επιτιθέμενο να αναγνωρίσει μοτίβα στο κρυπτοκείμενο, ειδικά σε μεγάλα και δομημένα μηνύματα [22].

Cipher Block Chaining

Η CBC λειτουργία αναπτύχθηκε ως βελτίωση της ECB λειτουργίας, ξεπερνώντας τις αδυναμίες ασφαλείας που παρουσίαζε η ECB. Στην CBC, κάθε μπλοκ δεδομένων συνδυάζεται μέσω της πράξης XOR με το κρυπτοκείμενο του προηγούμενου μπλοκ πριν κρυπτογραφηθεί με το ίδιο κλειδί, δημιουργώντας έτσι μια αλυσίδα κρυπτογράφησης (εικόνα 3.8). Έτσι εξασφαλίζουμε ότι ακόμα και αν επαναλαμβάνονται τα ίδια μπλοκ δεδομένων, το κρυπτοκείμενο που παράγεται θα είναι διαφορετικό κάθε φορά [12].

Για την κρυπτογράφηση του πρώτου μπλοκ, χρησιμοποιείται ένα διάνυσμα αρχικοποίησης (Initialization Vector – IV), το οποίο συνδυάζεται με το πρώτο μπλοκ μέσω XOR. Η διαδικασία αποκρυπτογράφησης περιλαμβάνει την εφαρμογή της λειτουργίας αποκρυπτογράφησης στο κάθε μπλοκ κρυπτοκειμένου και στη συνέχεια την πράξη XOR με το προηγούμενο μπλοκ κρυπτοκειμένου για την ανάκτηση του αρχικού μπλοκ δεδομένων [22]. Έτσι η τεχνική CBC εκφράζεται ως εξής :

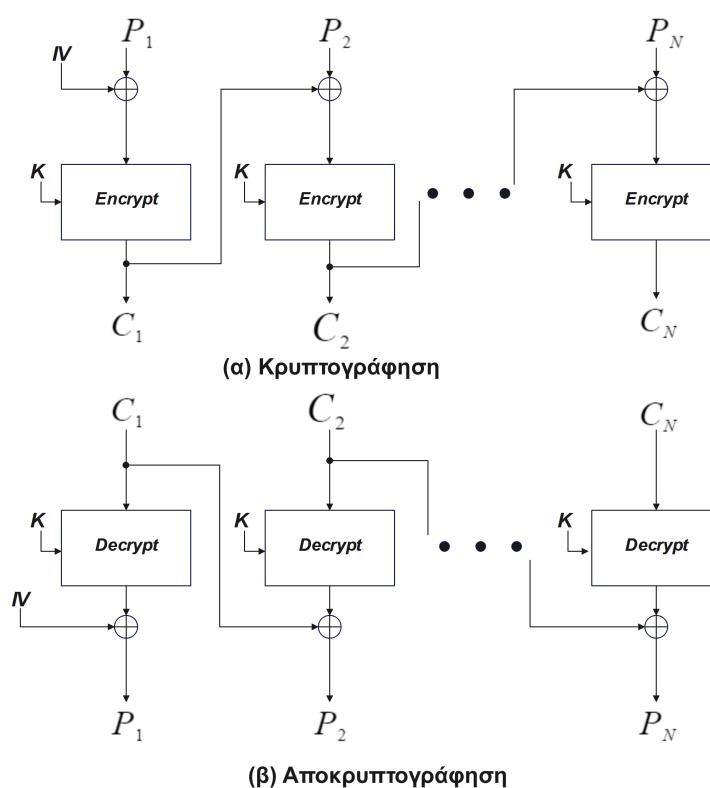
$$\begin{array}{l} C_1 = E(K, [P_1 \oplus IV]) \\ C_j = E(K, [P_j \oplus C_{j-1}]), \quad j = 2, \dots, N \end{array} \quad \left| \quad \begin{array}{l} P_1 = D(K, C_1) \oplus IV \\ P_j = D(K, C_j) \oplus C_{j-1}, \quad j = 2, \dots, N \end{array} \right.$$

Εκτός από την εξάλειψη της επαναληψιμότητας, πετυχαίνουμε καλύτερη ασφάλεια έναντι κρυπταναλυτικών επιθέσεων χάρη στον αλυσιδωτό μηχανισμό. Επιπλέον ο CBC μπορεί να χρησιμοποιηθεί και για αυθεντικοποίηση δεδομένων, προσφέροντας ένα επιπλέον επίπεδο ασφάλειας [22]. Από την άλλη μεριά η ασφάλεια του CBC εξαρτάται από την σωστή επιλογή και χρήση του IV, το οποίο πρέπει να είναι απρόβλεπτο και μοναδικό για κάθε μήνυμα. Επίσης είναι πιο αργή η τεχνική επειδή κάθε μπλοκ εξαρτάται από το προηγούμενο και δεν υπάρχει δυνατότητα παράλληλης επεξεργασίας λόγω των εξαρτήσεων αυτών [22].

Cipher Feedback

Η CFB λειτουργία αποτελεί έναν τρόπο μετατροπής ενός block cipher σε stream cipher, επιτρέποντας την κρυπτογράφηση δεδομένων σε πραγματικό χρόνο χωρίς την ανάγκη για padding του μηνύματος [22].

Η βασική λειτουργία του CFB (εικόνα 3.9) βασίζεται στη χρήση ενός καταχωρητή ολίσθησης μεγέθους ίσου με το μπλοκ κρυπτογράφησης. Η κρυπτογράφηση ξεκινά με ένα IV, το οποίο εισάγεται στον καταχωρητή. Έπειτα εφαρμόζεται η συνάρτηση κρυπτογράφησης στα δεδομένα



Σχήμα 3.8: Cipher Block Chaining λειτουργία.

του καταχωρητή, και τα s πιο σημαντικά bits του αποτελέσματος συνδυάζονται μέσω XOR με το πρώτο τμήμα του plaintext για την παραγωγή του πρώτου ciphertext. Το ciphertext αυτό εισάγεται στον καταχωρητή για την επεξεργασία του επόμενου τμήματος plaintext, και η διαδικασία επαναλαμβάνεται μέχρι να κρυπτογραφηθεί όλο το μήνυμα.

Κατά την αποκρυπτογράφηση, το κρυπτοκείμενο εισάγεται ξανά στον καταχωρητή και η διαδικασία XOR εφαρμόζεται με τον ίδιο τρόπο για ανακτηθεί το αρχικό μήνυμα. Πολύ σημαντικό χαρακτηριστικό του CFB είναι ότι χρησιμοποιεί την συνάρτηση κρυπτογραφίας και για την αποκρυπτογράφηση, αποφεύγοντας τη χρήση διαφορετικών συναρτήσεων για κάθε διαδικασία [22]. Η CFB λειτουργία μπορεί να περιγραφεί ως εξής:

$$\left| \begin{array}{l} I_1 = IV \\ I_j = \text{LSB}_{b-s}(I_{j-1}) \mid C_{j-1}, \quad j = 2, \dots, N \\ O_j = E(K, I_j), \quad j = 1, \dots, N \\ C_j = P_j \oplus \text{MSB}_s(O_j), \quad j = 1, \dots, N \end{array} \right| \quad \left| \begin{array}{l} I_1 = IV \\ I_j = \text{LSB}_{b-s}(I_{j-1}) \mid C_{j-1}, \quad j = 2, \dots, N \\ O_j = E(K, I_j), \quad j = 1, \dots, N \\ P_j = C_j \oplus \text{MSB}_s(O_j), \quad j = 1, \dots, N \end{array} \right.$$

Όπως και στην περίπτωση του CBC έτσι και εδώ λόγω του τρόπου λειτουργίας του CFB, δεν επιτρέπει παράλληλη κρυπτογράφηση, καθώς κάθε τμήμα εξαρτάται από το προηγούμενο. Επιπλέον, αν υπάρξει σφάλμα σε κάποιο τμήμα του ciphertext κατά τη μετάδοση, το σφάλμα αυτό μπορεί να επηρεάσει πολλά τμήματα του plaintext, καθιστώντας το πιο επιρρεπές σε λάθη που προκύπτουν από ζητήματα μετάδοσης δεδομένων [22].

Output Feedback

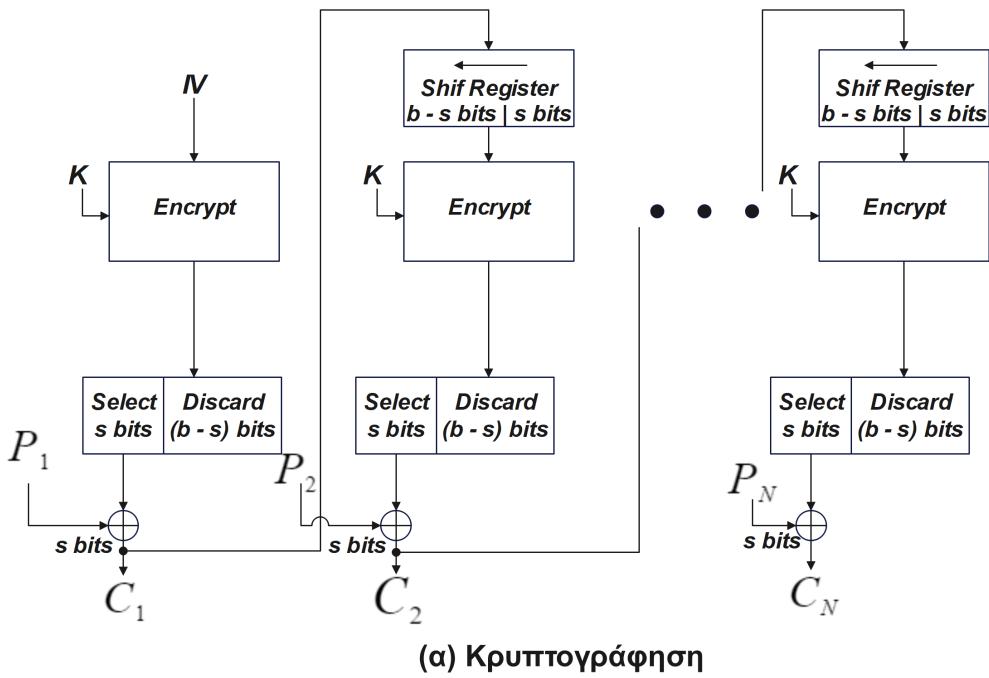
Η OFB λειτουργία (εικόνα 3.10) έχει μια παρόμοια δομή με αυτή της CFB. Στην OFB λειτουργία, η έξοδος της συνάρτησης κρυπτογράφησης ανατροφοδοτείται ώστε να χρησιμοποιηθεί ως είσοδος για την κρυπτογράφηση του επόμενου μπλοκ δεδομένων. Αυτή είναι και η κύρια διαφορά της OFB από την CFB. Επιπλέον στην OFB, κάθε μπλοκ κρυπτογράφησης και αποκρυπτογράφησης βασίζεται αποκλειστικά στο κλειδί και στο IV και όχι στα δεδομένα plaintext ή ciphertext.

Η κρυπτογράφηση στο OFB, εφαρμόζει αρχικά την συνάρτηση κρυπτογράφησης στο IV το οποίο είναι nonce, δηλαδή το IV πρέπει να είναι μοναδικό για κάθε μία εκτέλεση της κρυπτογράφησης [22], και το αποτέλεσμα γίνεται XOR με το P_1 να παράγει το πρώτο μπλοκ C_i . Η ίδια διαδικασία συνεχίζεται για όλα τα μπλοκ του μηνύματος. Κατά την αποκρυπτογράφηση, το εκάστοτε ciphertext μπλοκ γίνεται XOR με την έξοδο της συνάρτησης κρυπτογράφησης και αποκαλύπτει το αντίστοιχο plaintext block. Το OFB ορίζεται ως εξής:

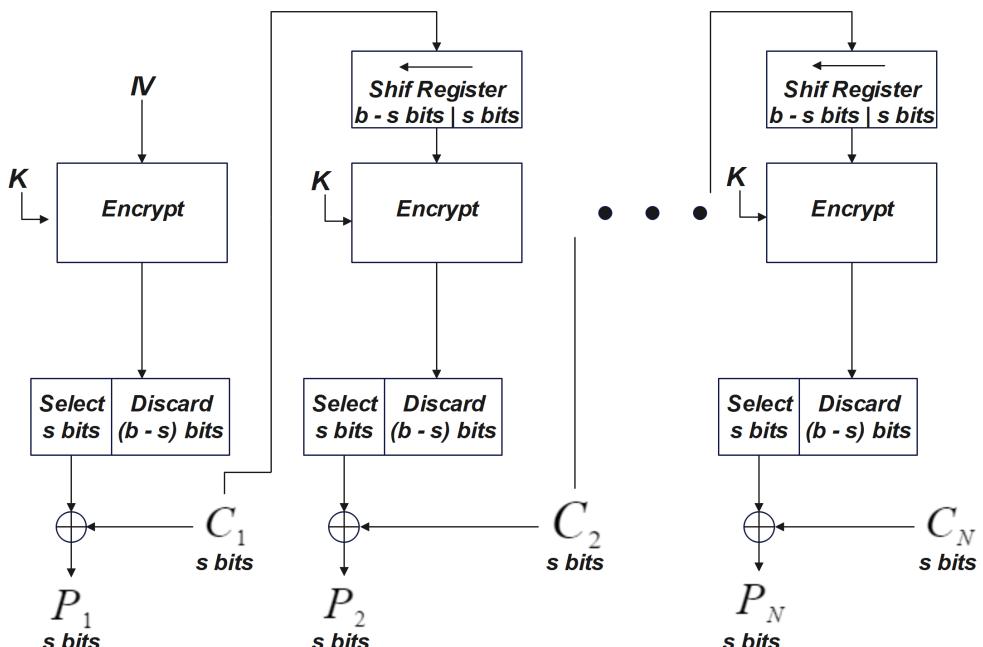
$$\left| \begin{array}{l} I_1 = \text{Nonce} \\ I_j = O_{j-1}, \quad j = 2, \dots, N \\ O_j = E(K, I_j), \quad j = 1, \dots, N \\ C_j = P_j \oplus O_j, \quad j = 1, \dots, N-1 \\ C_N^* = P_N^* \oplus \text{MSB}_u(O_N) \end{array} \right| \quad \left| \begin{array}{l} I_1 = \text{Nonce} \\ I_j = O_{j-1}, \quad j = 2, \dots, N \\ O_j = E(K, I_j), \quad j = 1, \dots, N \\ P_j = C_j \oplus O_j, \quad j = 1, \dots, N-1 \\ P_N^* = C_N^* \oplus \text{MSB}_u(O_N) \end{array} \right.$$

Έστω το μέγεθος ενός μπλοκ είναι b bits. Εάν το τελευταίο μπλοκ απλού κειμένου περιέχει u bits (υποδεικνύεται με *), με $u < b$, τα πιο σημαντικά u bits του τελευταίου μπλοκ εξόδου O_N χρησιμοποιούνται για τη λειτουργία XOR, τα υπόλοιπα $b-u$ bits του τελευταίου μπλοκ εξόδου απορρίπτονται [22].

Η χρήση της OFB λειτουργίας ως stream cipher έχει ως αποτέλεσμα η ακολουθία που χρησιμοποιείται για το XOR με το plaintext να είναι ανεξάρτητη από το ίδιο το plaintext (στην εικόνα

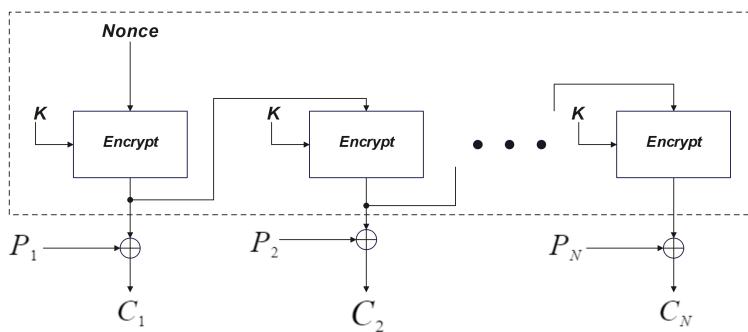


(α) Κρυπτογράφηση

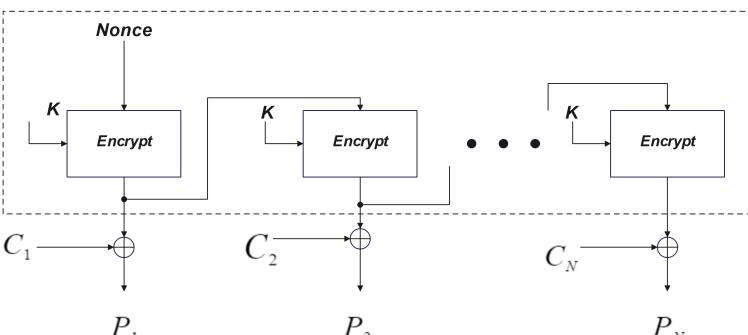


(β) Αποκρυπτογράφηση

Σχήμα 3.9: Cipher Feedback λειτουργία των s-bit.



(α) Κρυπτογράφηση



(β) Αποκρυπτογράφηση

Σχήμα 3.10: Output Feedback λειτουργία.

3.10 αυτό φαίνεται από τις διακεκομμένες γραμμές). Έτσι η OFB λειτουργία είναι χρήσιμη για περιπτώσεις όπου η κρυπτογράφηση πρέπει να είναι σταθερή και δεν πρέπει να επηρεάζεται από το περιεχόμενο των δεδομένων [22].

Η OFB λειτουργία έχει ένα κύριο χαρακτηριστικό πλεονέκτημα, ότι σε περίπτωση που προκύψουν σφάλματα κατά τη μετάδοση του ciphertext, αυτά τα σφάλματα δεν διαδίδονται πέρα από το συγκεκριμένο μπλοκ που επηρεάζεται. Δηλαδή, αν ένα σφάλμα εμφανιστεί σε ένα μπλοκ ciphertext, αυτό θα επηρεάσει μόνο το αντίστοιχο μπλοκ plaintext κατά την αποκρυπτογράφηση, χωρίς να επηρεάσει τα επόμενα μπλοκ [22]. Αυτό προσδίδει στην OFB υψηλή αντοχή σε σφάλματα μετάδοσης.

Βέβαια η OFB λειτουργία είναι αρκετά πιο επιρρεπής σε επιθέσεις τροποποίησης του μηνύματος. Δεδομένου ότι η κρυπτογραφική διαδικασία βασίζεται σε ένα σταθερό ρεύμα εξόδου, αν ένας επιτιθέμενος τροποποιήσει συγκεκριμένα bits του ciphertext, τότε μπορεί να τροποποιήσει αντίστοιχα και το plaintext [22]. Αυτή η αδυναμία την κάνει λιγότερο ασφαλή σε επιθέσεις που στοχεύουν στην αλλοίωση του περιεχομένου του μηνύματος.

Counter

Η CTR λειτουργία αντίθετα με τις άλλες λειτουργίες που αναλύσαμε, χρησιμοποιεί έναν μετρητή (counter) για την κρυπτογράφηση του κάθε block plaintext, ο οποίος αυξάνεται κατά ένα για κάθε επόμενο μπλοκ. Ο μόνος περιορισμός που τίθεται από το πρότυπο NIST SP 800-38A είναι ότι ο μετρητής πρέπει να είναι διαφορετικός για κάθε μπλοκ plaintext που κρυπτογραφείται, έτσι ώστε να μην επαναλαμβάνονται οι τιμές του κατά τη διάρκεια της κρυπτογράφησης [22].

Η βασική λειτουργία της CTR (εικόνα 3.11) είναι ότι για κάθε μπλοκ δεδομένων, ο μετρητής κρυπτογραφείται και το αποτέλεσμα του γίνεται XOR με το μπλοκ plaintext και παράγει το αντίστοιχο μπλοκ ciphertext. Στην αποκρυπτογράφηση, χρησιμοποιείται η ίδια σειρά τιμών του μετρητή και το κάθε ciphertext γίνεται XOR με το αντίστοιχο κρυπτογραφημένο μπλοκ του μετρητή και παράγει το plaintext. Η διαδικασία αυτή δεν απαιτεί αλυσιδωτή επεξεργασία, δηλαδή δεν υπάρχει αλληλεξάρτηση των μπλοκ όπως στις CBC και CFB, καθιστώντας την διαδικασία κρυπτογράφησης και αποκρυπτογράφησης πιο ευέλικτη και ταχύτερη [12].

Δεδομένης μια ακολουθίας τιμών του μετρητή $T_1, T_2 \dots T_N$ μπορούμε να ορίσουμε την CTR λειτουργία ως εξής :

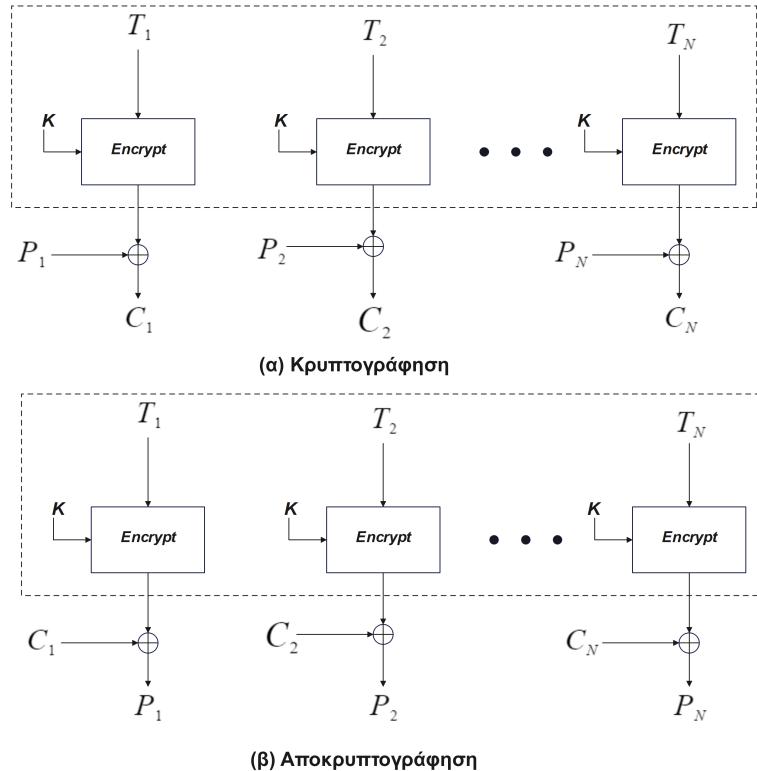
$$\begin{array}{ll} C_j = P_j \oplus E(K, T_j), & j = 1, \dots, N-1 \\ C_N^* = P_N^* \oplus \text{MSB}_u[E(K, T_N)] & \end{array} \quad \left| \quad \begin{array}{ll} P_j = C_j \oplus E(K, T_j), & j = 1, \dots, N-1 \\ P_N^* = C_N^* \oplus \text{MSB}_u[E(K, T_N)] & \end{array} \right.$$

Όπως και στην OFB έτσι και εδώ για το τελευταίο μπλοκ plaintext, το οποίο μπορεί να είναι μεγέθους u bits οπού $u < b$, τα περισσότερο σημαντικά ψηφία του τελευταίου μπλοκ εξόδου χρησιμοποιούνται για την πράξη XOR, και τα υπόλοιπα $u - b$ απορρίπτονται [22].

Μερικά από τα πλεονεκτήματα που προσφέρει η CTR λειτουργία και αναλύονται πλήρως στο [22], είναι τα εξής :

- **Απόδοση σε υλοποιήσεις υλικού και λογισμικού.**
- **Προεπεξεργασία.**
- **Τυχαία προσβασιμότητα.**

- Αποδεδειγμένη ασφάλεια.
- Απλότητα.



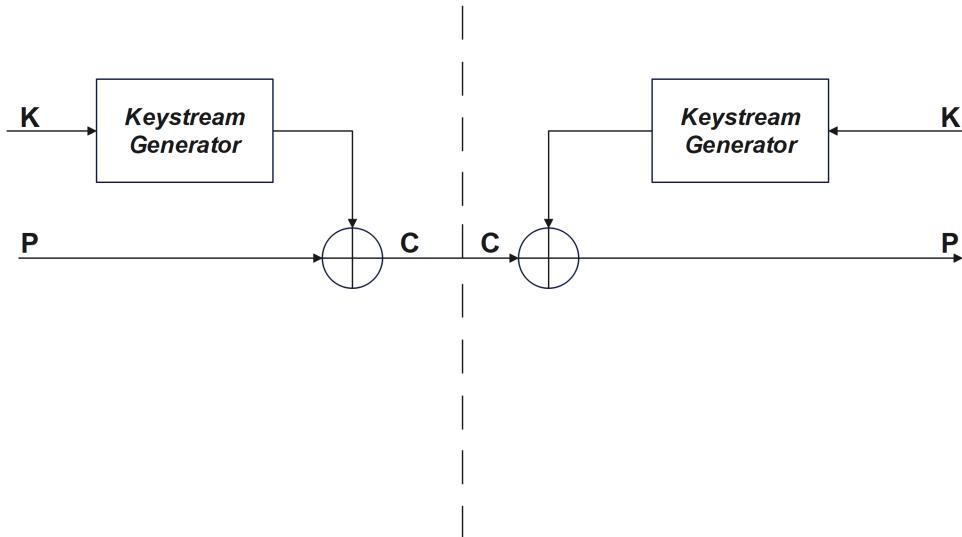
Σχήμα 3.11: Counter λειτουργία.

3.3 Αλγόριθμοι Ροής

Οι αλγόριθμοι ροής αποτελούν μία κατηγορία κρυπταλγορίθμων που κρυπτογραφούν δεδομένα διαδοχικά, επεξεργαζόμενοι κάθε bit του μηνύματος διαδοχικά. Βασικός στόχος των αλγορίθμων ροής είναι η παραγωγή της κλειδοροής, η οποία προστίθεται στο αρχικό κείμενο με τη χρήση της δυαδική πράξης XOR για την παραγωγή του κρυπτογραφημένου κειμένου. Οι ακολουθίες αυτές, αν και ψευδοτυχαίες, πρέπει να προσομοιάζουν την έκβαση ανεξάρτητων επαναλήψεων ενός τυχαίου πειράματος, παρέχοντας έτσι υψηλό επίπεδο ασφάλειας [16].

Η παραγωγή κλειδοροής στηρίζεται σε γεννήτριες ψευδοτυχαίων ακολουθιών, οι οποίες λειτουργούν ως πεπερασμένα αυτόματα με περιορισμένη μνήμη που ανανεώνεται ντετερμινιστικά [12]. Χρησιμοποιώντας τις ίδιες αρχικές συνθήκες, οι δύο οντότητες που επιθυμούν να επικοινωνήσουν μπορούν να παράγουν την ίδια κλειδοροή τόσο για την κρυπτογράφηση όσο και για την αποκρυπτογράφηση. Στην πράξη, αυτό σημαίνει ότι η ίδια γεννήτρια, με τις ίδιες αρχικές συνθήκες, θα παράγει πάντα την ίδια έξοδο, διασφαλίζοντας τη συνέπεια και την ασφάλεια της επικοινωνίας.

Οι αρχιτεκτονικές που χρησιμοποιούνται στους αλγόριθμους ροής περιλαμβάνουν τους Linear Feedback Shift Registers (LFSR) και τους Non-linear Feedback Shift Registers (NFSR) [32]. Οι LFSR είναι απλοί και αποδοτικοί, ενώ οι NFSR προσφέρουν μεγαλύτερη ασφάλεια λόγω της μη γραμμικής συμπεριφοράς τους.



Σχήμα 3.12: Λειτουργία Κρυπταλγόριθμου Ροής.

3.3.1 Linear Feedback Shift Registers

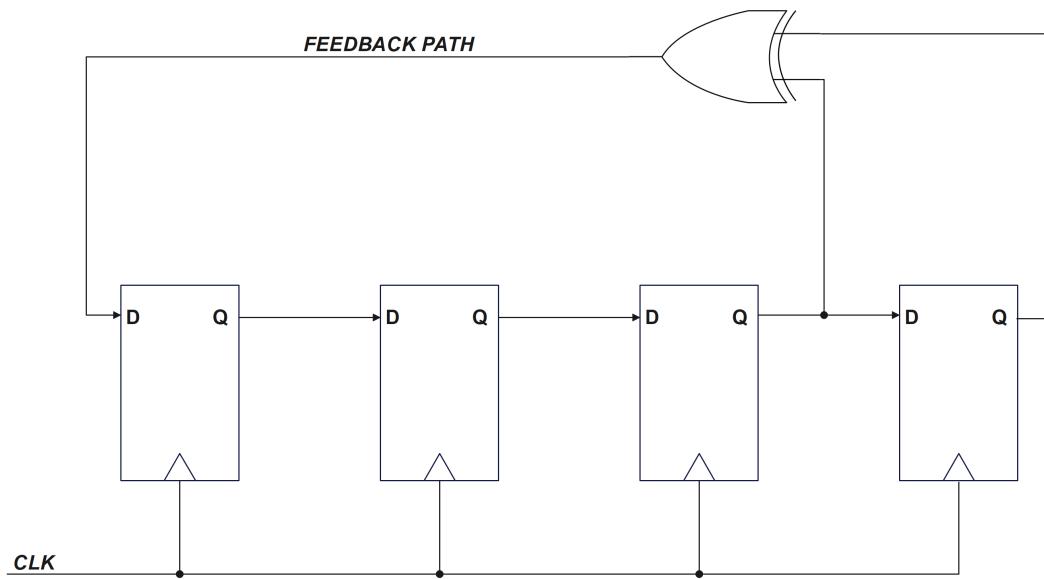
Ο Linear Feedback Shift Register (LFSR) αποτελεί έναν καταχωρητή ολίσθησης με γραμμική ανάδραση, του οποίου η είσοδος βασίζεται σε μια γραμμική συνάρτηση της προηγούμενης κατάστασης. Η πιο κοινή γραμμική συνάρτηση που χρησιμοποιείται είναι η πραξή του λογικού XOR, και έτσι το LFSR είναι συνήθως ένας καταχωρητής ολίσθησης που η είσοδός του καθορίζεται από το XOR κάποιων bits της συνολικής τιμής του καταχωρητή ολίσθησης. Το κάθε νέο bit εισόδου που παράγεται από το XOR εισάγεται στη θέση του πιο αριστερού bit και τα υπόλοιπα bits ολισθαίνουν μία θέση προς τα δεξιά (εικόνα 3.13). Για να επιτευχθούν όλοι οι δυνατοί συνδιασμοί άρα και η μέγιστη περίοδος ενός LFSR μήκους $2^N - 1$, το πολυώνυμο αυτό θα πρέπει να είναι πρωτεύον (primitive polynomial).

Το LFSR μαθηματικά περιγράφεται από ένα πολυώνυμο ανατροφοδότησης, το οποίο ονομάζεται χαρακτηριστικό πολυώνυμο και περιγράφεται από την παρακάτω παράσταση ως:

$$P(x) = x^m + a_{m-1}x^{m-1} + \dots + a_1x + a_0 \quad (3.1)$$

Οπού τα a_i είναι συντελεστές (0 ή 1) που αντιπροσωπεύουν τα taps (bits που συμμετέχουν στην ανάδραση) και το m είναι το μήκος του καταχωρητή ολίσθησης. Υπάρχουν δύο κύριες αρχιτεκτονικές για την υλοποίηση ενός LFSR:

- Αρχιτεκτονική Fibonacci.



Σχήμα 3.13: 4-bit LFSR.

- **Αρχιτεκτονική Galois.**

Στην αρχιτεκτονική Fibonacci, τα taps βρίσκονται εξωτερικά του καταχωρητή και το αποτέλεσμα της λειτουργίας XOR των taps εισάγεται ως το νέο bit στην είσοδο του καταχωρητή. Το χαρακτηριστικό πολυώνυμο σε αυτή την αρχιτεκτονική έχει την μορφή της παράστασης 3.1 όπου τα taps αναπαρίστανται από τους συντελεστές του πολυωνύμου [33].

Στην αρχιτεκτονική Galois, η πράξη XOR πραγματοποιείται παράλληλα, μέσα στον καταχωρητή (αυτή είναι και η κύρια διαφορά από την αρχιτεκτονική Fibonacci), με αποτέλεσμα την αύξηση της ταχύτητας σε υλοποιήσεις hardware, καθώς μειώνεται ο χρόνος διάδοσης του σήματος, προσφέροντας ταχύτερες υλοποιήσεις [33].

3.3.2 Non-linear Feedback Shift Registers

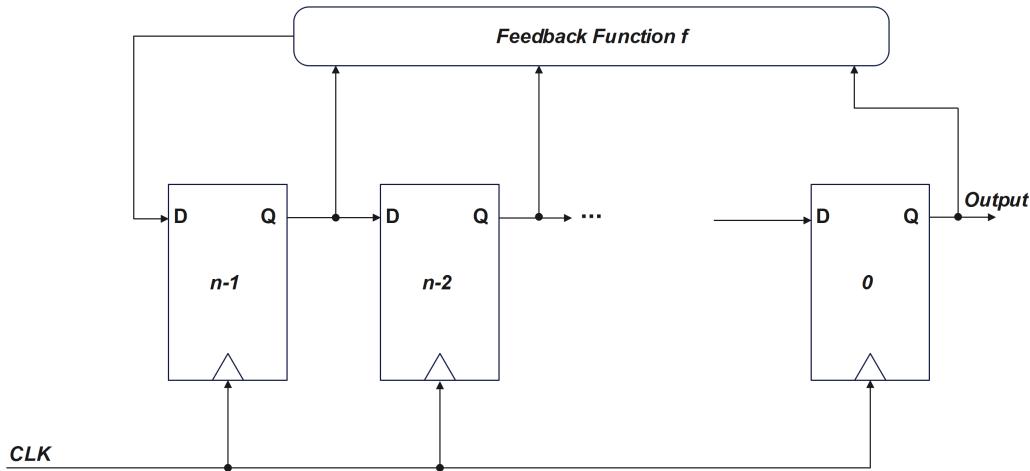
Ο Non-Linear Feedback Shift Register (NFSR) αποτελεί έναν καταχωρητή ολίσθησης, στον οποίο το bit εισόδου καθορίζεται από μια μη γραμμική συνάρτηση της προηγούμενης κατάστασης. Σε αντίθεση με τους LFSRs που περιγράψαμε προηγουμένως, η συνάρτηση ανάδρασης είναι μη γραμμική, γεγονός που τους προσδίδει μεγαλύτερη ανθεκτικότητα σε κρυπτογραφικές επιθέσεις.

Ένας n bit NFSR αποτελείται από n δυαδικά στάδια, καθένα από τα οποία αποθηκεύει ένα bit πληροφορίας. Το σύστημα περιλαμβάνει μια μη γραμμική συνάρτηση Boolean, η οποία ονομάζεται συνάρτηση ανάδρασης, και ένα ρολόι (clock) (εικόνα 3.14). Σε κάθε κύκλο παλμού, το τελευταίο στάδιο του καταχωρητή ενημερώνεται με την τιμή που υπολογίζεται από τη συνάρτηση ανάδρασης, ενώ τα υπόλοιπα στάδια ολισθαίνουν το περιεχόμενο του προηγούμενου

σταδίου τους. Η παραπάνω διαδικασία μαθηματικά αναπαρίσταται ως εξής :

$$\begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{pmatrix} \longrightarrow \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ f(x_0, x_1, \dots, x_{n-1}) \end{pmatrix}$$

οπού x_i είναι η τιμή σταδίου i και η συνάρτηση f είναι μη γραμμική [34].



Σχήμα 3.14: Η δομή ενός n -bit NLFSR [34].

Για να είναι ένας n bit NFSR αντιστρέψιμος πρέπει η συνάρτηση ανάδρασης να έχει τη μορφή :

$$f(x_0, \dots, x_{n-1}) = x_0 \oplus g(x_1, x_2, \dots, x_{n-1})$$

όπου g είναι μια μη γραμμική Boolean συνάρτηση [34].

Μερικοί από τους πιο γνωστούς κρυπτογραφικούς αλγόριθμους που χρησιμοποιούν NFSR είναι οι :

- **Achterbahn.**
- **Grain.**
- **Keeloq.**
- **LIZARD.**
- **Trivium.**
- **VEST.**

4

Message Authentication Codes

4.1 Εισαγωγή

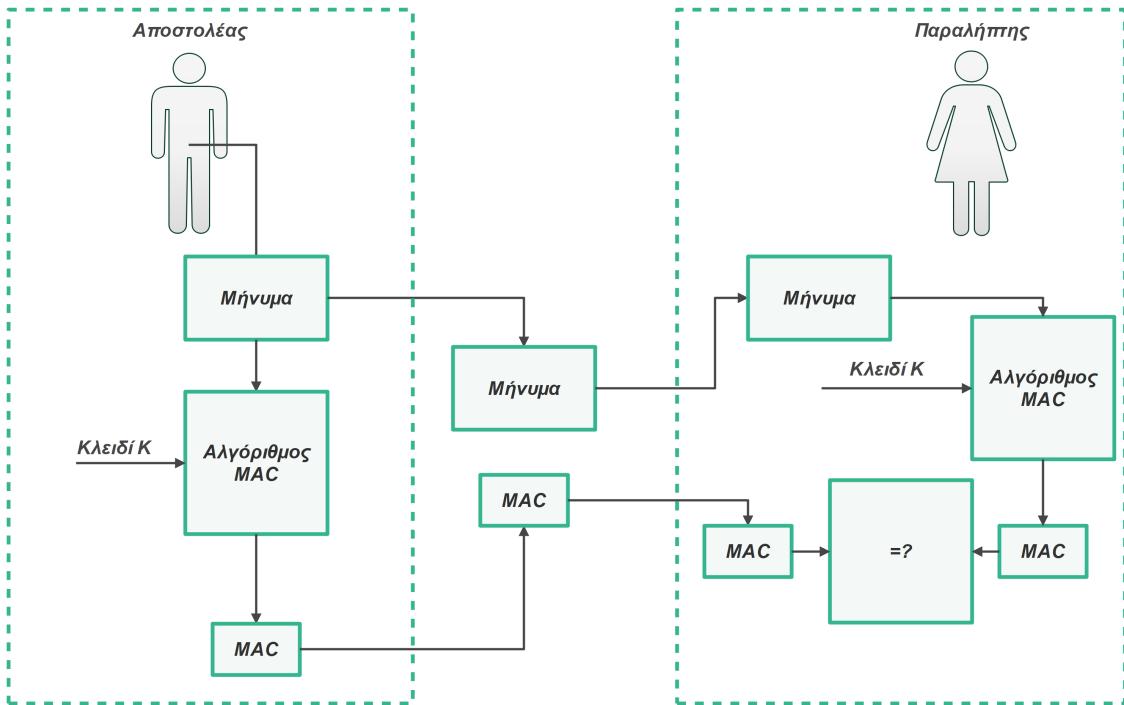
Όπως αναφέραμε και στο Κεφάλαιο δύο, η αυθεντικοποίηση και ακεραιότητα των δεδομένων είναι δύο από τα πιο σημαντικά κριτήρια που λαμβάνονται υπόψιν για την ασφάλεια ενός συστήματος. Οι λεγόμενοι κώδικες αυθεντικοποίησης μηνύματος (Message Authentication Codes - MACs) συμβάλλουν στην διασφάλιση των παραπάνω κριτηρίων [12]. Στην ουσία οι MAC χρησιμοποιούνται για να επιβεβαιώσουν ότι το μήνυμα δεν έχει τροποποιηθεί και πρέρχεται από την αναμενόμενη πηγή. Αυτό το πετυχαίνει μέσω της παραγωγής μιας σύντομης ακολουθίας bits, γνωστή ως «κωδικός αυθεντικοποίησης», που υπολογίζεται από τα δεδομένα και ένα μυστικό κλειδί.

Το βασικό μοντέλο αυθεντικοποίησης του μηνύματος μέσω MAC φαίνεται στην εικόνα 4.1. Ο αποστολέας υπολογίζει τον MAC από το μήνυμα χρησιμοποιώντας το μυστικό κλειδί και τον στέλνει μαζί με το μήνυμα στον παραλήπτη. Ο παραλήπτης με την σειρά του και με τον ίδιο τρόπο υπολογίζει τον δικό του MAC από το μήνυμα που παρέλαβε και συγκρίνει το αποτέλεσμα που υπολόγισε με αυτόν του αποστολέα. Εάν οι δύο τιμές ταιριάζουν, τότε το μήνυμα θεωρείται αυθεντικό και αδιάφθορο. Αν όχι, αυτό σημαίνει ότι είτε το μήνυμα έχει τροποποιηθεί κατά τη μεταφορά του είτε ότι δεν προέρχεται από την αναμενόμενη πηγή.

Οι MACs βρίσκουν εφαρμογή σε πολλές σύγχρονες τεχνολογίες και πρωτόκολλα ασφάλειας, όπως το TLS (Transpot Layer Secyritiy) και το IPsec, ενώ είναι κρίσιμα για την ασφάλεια τραπεζικών συναλλαγών, επικοινωνιών δικτύων, αποθήκευσης δεδομένων. Επιπλέον έχει πολλές εφαρμογές στο Διαδίκτυο των Πραγμάτων [22].

4.2 Cipher-based Message Authentication Code (CMAC)

Ο Cipher-based Message Authentication Code (CMAC) αποτελεί έναν ισχυρό μηχανισμό επαλήθευσης αυθεντικότητας μηνυμάτων, ο οποίος χρησιμοποιεί συμμετρικούς κρυπτογραφι-



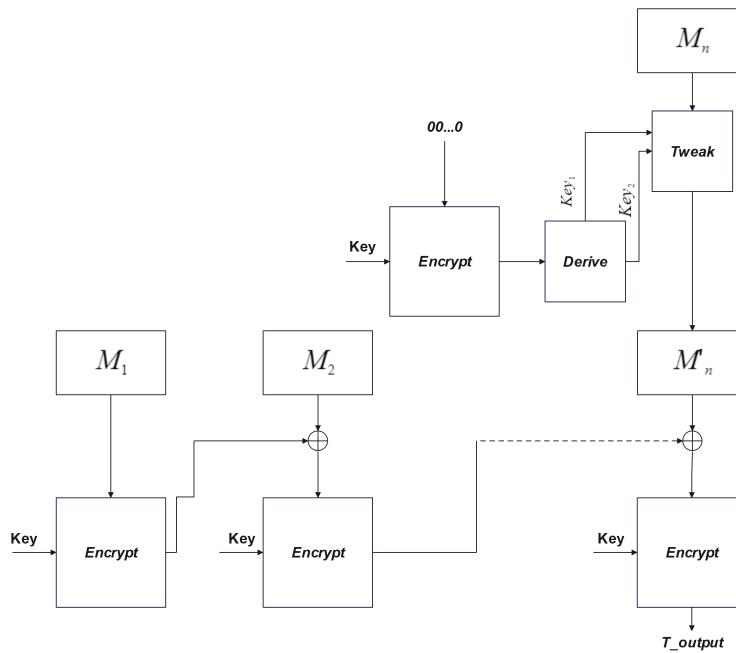
Σχήμα 4.1: Αυθεντικοποίηση μηνύματος μέσω MAC.

κούς αλγόριθμους, όπως ο AES και 3DES, για την παραγωγή ενός MAC [22].

Η λειτουργία του CMAC φαίνεται στην εικόνα 4.2. Πρώτα, το μήνυμα χωρίζεται σε μπλοκ δεδομένων σταθερού μήκους, ανάλογα με το μέγεθος του μπλοκ του χρησιμοποιούμενου αλγόριθμου (π.χ., 128 bits για τον AES). Ένα κύριο κλειδί k χρησιμοποιείται για την κρυπτογράφηση κάθε μπλοκ του μηνύματος μέσω της συνάρτησης κρυπτογράφησης $E(K, M_i)$ [22]. Το αποτέλεσμα της κρυπτογράφησης κάθε μπλοκ συνδυάζεται μέσω της πράξης XOR με το επόμενο μπλοκ μηνύματος και εισάγεται στην συνάρτηση κρυπτογράφησης. Σε περίπτωση που το μήνυμα δεν είναι ακέραιο πολλαπλάσιο του μήκους του μπλοκ, το τελευταίο μπλοκ συμπληρώνεται (padding) ώστε να φτάσει το απαιτούμενο μήκος και να συνδυάζεται μέσω της πράξης XOR με το υπο-κλειδί k_2 , διαφορετικά σε περίπτωση που είναι ακέραιο πολλαπλάσιο τότε συνδυάζεται μέσω της πράξης XOR με το υποκλειδί k_1 . Τα υποκλειδιά k_1 και k_2 , παράγονται, με μια συγκεκριμένη διαδικασία, από το κύριο κλειδί k , και τα οποία χρησιμοποιούνται μόνο στην κρυπτογράφηση του τελευταίου μπλοκ μηνύματος [12]. Η έξοδος του αλγορίθμου αποτελείται γενικά από το τελευταίο τμήμα του κρυπτοκειμένου (π.χ. μεγέθους 128 bits για τον AES). Βέβαια, ο CMAC μπορεί να παραμετροποιηθεί προκειμένου η έξοδος να έχει μικρότερο μέγεθος. Σε μια τέτοια περίπτωση, τα αριστερότερα bits του τελευταίου τμήματος είναι αυτά που αποτελούν την έξοδο του αλγορίθμου [12].

4.3 Encrypt-then-MAC

Ένα ακόμα κριτήριο που πρέπει να διασφαλίσουμε είναι τόσο η εμπιστευτικότητα της πληροφορίας όσο και η αυθεντικότητα των δεδομένων. Αυτό το πετυχαίνουμε με την λεγόμενη

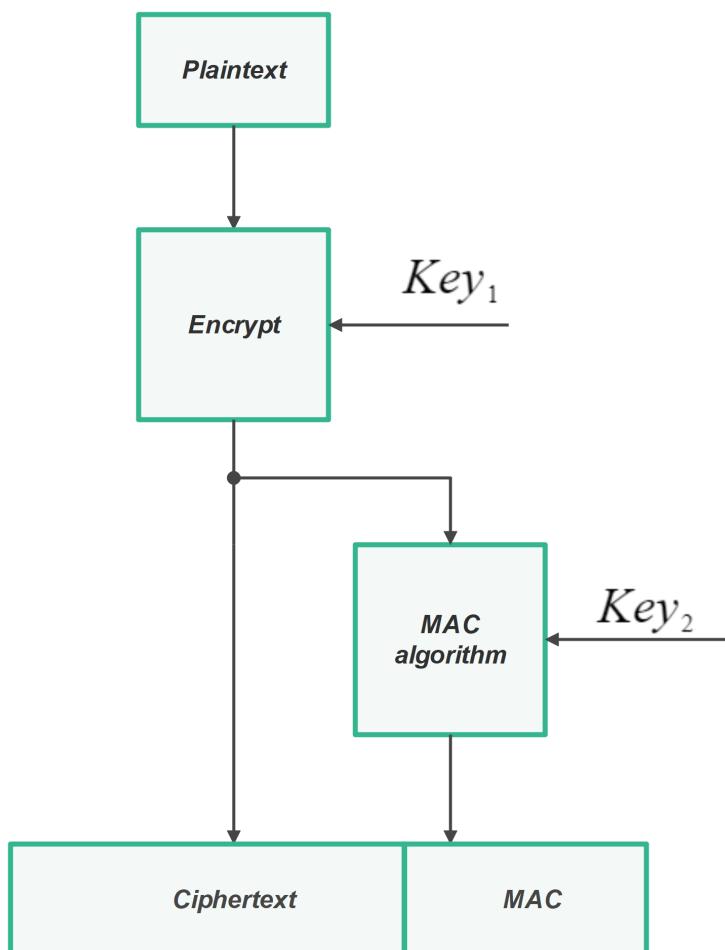


Σχήμα 4.2: Κώδικας αυθεντικοποίησης μηνύματος CMAC.

αυθεντικοποιημένη κρυπτογράφηση. Ένα σχήμα που εφαρμόζει το παραπάνω είναι η μέθοδος Encrypt-then-MAC. Σε αυτή την προσέγγιση, τα δεδομένα πρώτα κρυπτογραφούνται χρησιμοποιώντας έναν συμμετρικό αλγόριθμο κρυπτογράφησης (π.χ., AES) και στη συνέχεια εφαρμόζεται ένας κωδικός αυθεντικοποίησης μηνύματος (MAC) στο κρυπτογραφημένο κείμενο. Αυτός ο συνδυασμός εξασφαλίζει τα δύο κριτήρια που αναφέραμε, καθώς τα δεδομένα παραμένουν κρυπτογραφημένα και ο παραλήπτης μπορεί να επαληθεύσει ότι τα δεδομένα δεν έχουν παραποιηθεί στην διάρκεια της μετάδοσης ή στάλθηκαν από μη – έμπιστη πηγή. Στην μέθοδο αυτή, ο παραλήπτης πρώτα επαληθεύει την αυθεντικότητα του κρυπτογραφημένου μηνύματος χρησιμοποιώντας το MAC και, εφόσον αυτό είναι έγκυρο, προχωρά στην αποκρυπτογράφηση του. Αυτό μειώνει την πιθανότητα επιθέσεων που εκμεταλλεύονται σφάλματα αποκρυπτογράφησης και παρέχει αυξημένη ασφάλεια, αποτρέποντας τη δυνατότητα επίθεσης σε συστήματα που βασίζονται σε κρυπτογραφία [35]. Είναι σημαντικό να αναφερθεί ότι χρησιμοποιούνται διαφορετικά κλειδιά τόσο για την κρυπτογράφηση όσο και για την αυθεντικοποίηση του μηνύματος (εικόνα 4.3) [12].

4.4 Galois/Counter Mode (GCM)

Ο Galois/Counter Mode (GCM) είναι ένας τρόπος λειτουργίας που επιτρέπει τόσο κρυπτογράφηση όσο και αυθεντικοποίηση του μηνύματος, ο οποίος έχει τυποποιηθεί από το NIST και περιγράφεται στο πρότυπο NIST SP 800-38D. Ο GCM σχεδιάστηκε για να είναι παράλληλος επιτρέποντας υψηλό ρυθμό μετάδοσης με χαμηλό κόστος και χαμηλή καθιστέρηση, καθιστώντας τον ιδιαίτερα κατάλληλο για εφαρμογές που απαιτούν γρήγορη επεξεργασία δεδομένων. Η βασική ιδέα πίσω από τον GCM είναι η χρήση ενός παραλλαγμένου τρόπου λειτουργίας του CTR για την κρυπτογράφηση των δεδομένων και ενός πολλαπλασιασμού σε πεδίο Galois για

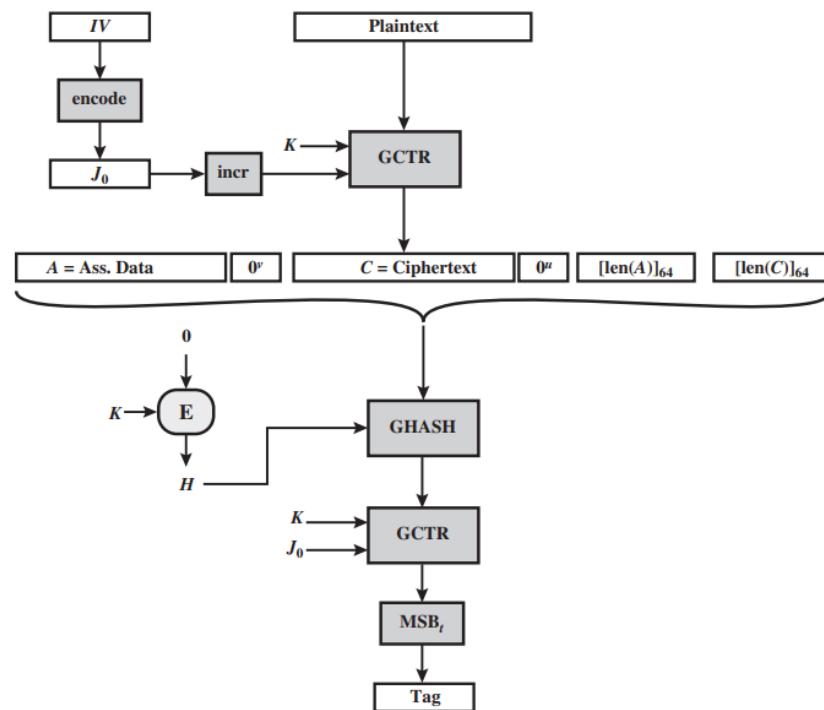


Σχήμα 4.3: Εφαρμογή MAC σε κρυπτογραφημένο μήνυμα.

τη δημιουργία του MAC [22].

Ο GCM χρησιμοποιεί δύο βασικές λειτουργιές, την GHASH που είναι μια κλειδωμένη κατακερματιστική συνάρτηση, και τη λειτουργία GCTR, που αποτελεί την κλασική λειτουργία CTR με τους μετρητές να καθορίζονται από μια απλή λειτουργία αύξησης κατά ένα. Η λειτουργία GHASH λαμβάνει ως είσοδο ένα κλειδί κατακερματισμού H και μια ακολουθία bit X , και παράγει ένα MAC μήκους 128 bit. Η λειτουργία αυτή μπορεί να εκτελεστεί παράλληλα, καθιστώντας την ιδανική για επεξεργασία μεγάλων δεδομένων με υψηλή ταχύτητα [22].

Η διαδικασία υπολογισμού του MAC χρησιμοποιεί αποκλειστικά τον πολλαπλασιασμό σε πεδίο Galois, μια επιλογή που έγινε λόγω της ευκολίας εκτέλεσης αυτής τη λειτουργίας εντός ενός πεδίου Galois και της δυνατότητας εύκολης υλοποίησης σε υλικό [22].



Σχήμα 4.4: Τρόπος λειτουργίας GCM [22].

5

FPGAs & ASICs

5.1 Εισαγωγή στα FPGA

Τα Field-Programmable Gate Arrays (FPGAs) είναι προγραμματιζόμενες λογικές συσκευές που επιτρέπουν στους χρήστες να διαμορφώσουν τη λειτουργία τους μετά την κατασκευή τους, προσφέροντας ευελιξία και δυνατότητα επαναπρογραμματισμού. Αυτή η ευελιξία τα καθιστά ιδανικά για εφαρμογές που απαιτούν προσαρμογή των λογικών κυκλωμάτων σε συγκεκριμένες ανάγκες, όπως η τηλεπικοινωνία, η αυτοκινητοβιομηχανία και οι βιομηχανικές εφαρμογές. Επιπλέον, τα FPGAs χρησιμοποιούνται εκτενώς στην έρευνα, καθώς επιτρέπουν τη γρήγορη ανάπτυξη και δοκιμή πρωτότυπων.

Η βασική αρχιτεκτονική ενός FPGA αποτελείται από τρία κύρια δομικά στοιχεία:

- **Τα Configurable Logic Blocks (CLBs).**
- **Τα μπλοκ εισόδου-εξόδου (Input-Output Block - IOBs).**
- **Την προγραμματιζόμενη διασύνδεση (programmable interconnect).**

Τα CLBs είναι η κύρια μονάδα επεξεργασίας ενός FPGA και μπορούν να διαμορφωθούν για την εκτέλεση διαφόρων συνδυαστικών ή ακολουθιακών λογικών λειτουργιών. Στη πράξη, περιλαμβάνουν δομές όπως οι πίνακες αναζήτησης (Lookup Tables - LUTs) που υλοποιούν λογικές συναρτήσεις και flip-flops για αποθήκευση καταστάσεων. Οι LUTs χρησιμοποιούνται για την εκτέλεση βασικών λογικών πράξεων, ενώ τα flip-flops παρέχουν αποθηκευτική δυνατότητα για συγχρονισμό σημάτων.

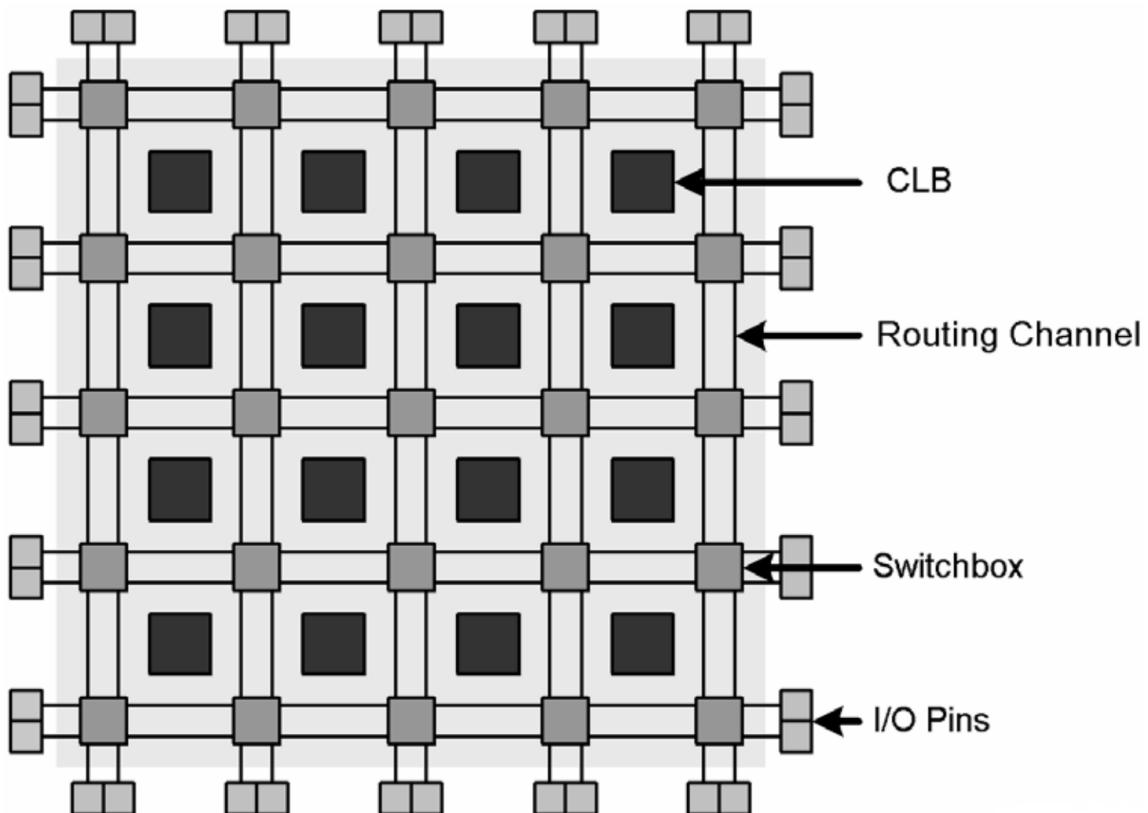
Η προγραμματιζόμενη διασύνδεση αποτελεί ένα σημαντικό στοιχείο των FPGAs, καθώς συνδέει τα CLBs μεταξύ τους και με τα IOBs. Μέσω αυτής της δομής, ο σχεδιαστής μπορεί να επιλέξει πως θα συνδεθούν τα διάφορα μπλοκ μεταξύ τους, ώστε να εκτελούν τις επιθυμητές λογικές λειτουργίες. Η διασύνδεση αυτή γίνεται μέσω ενός προγραμματιζόμενου δικτύου

διακοπτών (matrix switch), το οποίο επιτρέπει τη δημιουργία μιας ευέλικτης τοπολογίας συνδέσεων.

Τα μπλοκ εισόδου-εξόδου (IOBs) είναι υπεύθυνα για την επικοινωνία του FPGA με εξωτερικά συστήματα. Επιτρέπουν στο FPGA να στέλνει και να λαμβάνει δεδομένα από άλλες συσκευές ή αισθητήρες, προσφέροντας παράλληλα προγραμματιζόμενα χαρακτηριστικά για να προσαρμόζεται η συμπεριφορά των σημάτων, όπως το εύρος και η ταχύτητα μεταφοράς.

Ένα σημαντικό πλεονέκτημα των FPGAs είναι η ικανότητα επαναπρογραμματισμού τους. Ο σχεδιασμός ενός FPGA πραγματοποιείται μέσω γλωσσών περιγραφής υλικού, όπως η VHDL και η Verilog, που επιτρέπουν την ακριβή περιγραφή της λογικής συμπεριφοράς του κυκλώματος. Αυτό επιτρέπει στους σχεδιαστές να προσαρμόζουν το FPGA σε διαφορετικές εφαρμογές, καθιστώντας τα εξαιρετικά ευέλικτα για ποικιλία απαιτήσεων.

Στην εικόνα 5.1 φαίνεται η γενική δομή της αρχιτεκτονικής FPGA.



Σχήμα 5.1: Η εσωτερική δομή ενός FPGA [36].

5.2 Εισαγωγή στα ASICs

Τα ASICs (Application-Specific Integrated Circuits) είναι ολοκληρωμένα κυκλώματα σχεδιασμένα για συγκεκριμένες εφαρμογές, σε αντίθεση με τα γενικής χρήσης κυκλώματα όπως οι μικροεπεξεργαστές ή οι μνήμες. Τα ASICs χρησιμοποιούνται σε εφαρμογές που απαιτούν ειδική προσαρμογή του υλικού, όπως τα κυκλώματα για δορυφόρους, αυτοκίνητα ή συσκευές

επικοινωνίας κ.α. Η βασική αρχή πίσω από τα ASICs είναι ότι σχεδιάζονται με τέτοιο τρόπο ώστε να ανταποκρίνονται στις συγκεκριμένες ανάγκες μιας εφαρμογής, προσφέροντας βελτιωμένη απόδοση και αποδοτικότητα σε σχέση με άλλες λύσεις, όπως τα FPGAs.

Η βασική λειτουργία ενός ASIC είναι προσαρμοσμένη στη συγκεκριμένη εφαρμογή που εξυπηρετεί, και αυτό τα καθιστά ιδιαίτερα αποδοτικά σε όρους ταχύτητας, κατανάλωσης ενέργειας και απαιτήσεων χώρου. Οι σχεδιαστές ASICs έχουν την ελευθερία να επιλέξουν τις βέλτιστες αρχιτεκτονικές και να διαμορφώσουν την εσωτερική τους δομή ώστε να ανταποκρίνονται στις επιθυμητές προδιαγραφές. Τα σύγχρονα ASICs μπορούν να περιλαμβάνουν εκατομμύρια λογικές πύλες, ενσωματωμένες, μνήμες, μικροεπεξεργαστές, και άλλα στοιχεία, καθιστώντας τα συστήματα-on-chip (SoCs).

Υπάρχουν διαφορετικοί τύποι ASICs, όπως τα Full-Custom ASICs, όπου όλα τα λογικά κύτταρα και η διάταξη σχεδιάζονται από την αρχή, και τα Standard Cell ASICs, όπου χρησιμοποιούνται προκατασκευασμένα λογικά κύτταρα από βιβλιοθήκες standard cell, εξοικονομώντας χρόνο και πόρους. Η επιλογή της κατάλληλης προσέγγισης εξαρτάται από τις απαιτήσεις της εφαρμογής, καθώς τα Full-Custom ASICs προσφέρουν μεγαλύτερη ευελιξία αλλά με υψηλότερο κόστος και ρίσκο, ενώ τα Standard Cell ASICs επιταχύνουν τη διαδικασία ανάπτυξης.

Τα Gate Array ASICs είναι μια άλλη προσέγγιση, όπου τα τρανζίστορ είναι προκαθορισμένα στη δομή του wafer και ο σχεδιασμός επικεντρώνεται στην προσαρμογή των συνδέσεων μεταξύ αυτών. Αυτή η μέθοδος μειώνει το κόστος και τον χρόνο παραγωγής, αλλά περιορίζει την ευελιξία και την απόδοση σε σύγκριση με τις λύσεις πλήρους προσαρμογής (Full-Custom). Τα ASICs προσφέρουν υψηλές επιδόσεις και μειωμένη κατανάλωση ενέργειας, κάτι που είναι κρίσιμο σε εφαρμογές όπου η ισχύς και η απόδοση είναι βασικές απαιτήσεις, όπως στις ασύρματες επικοινωνίες και τις κινητές συσκευές.

Συνολικά, τα ASICs είναι ιδανική επιλογή για μεγάλες παραγωγές, καθώς το κόστος τους μπορεί να μειωθεί σημαντικά με την αύξηση του όγκου παραγωγής. Ωστόσο, η ανάπτυξη τους απαιτεί εκτεταμένη προκατασκευαστική ανάλυση και προσομοίωση για την αποφυγή σφαλμάτων, καθώς τα λάθη στη φάση κατασκευής μπορούν να είναι εξαιρετικά δαπανηρά.

Στον πίνακα 5.1 φαίνονται οι διαφορές των ASICs από τα FPGAs.

Προγραμματιζόμενο Ολοκληρωμένο Πλέγμα Πολύτον (FPGA)	Ολοκληρωμένο Κύκλωμα Ειδικής Εφαρμογής (ASIC)
Διαδικασία Σχεδίασης	Απλή διαδικασία σχεδίασμού.
Έξοδος	Δεν υπάρχουν μη επαναλαμβανόμενα έξοδα.
Χαρακτηρίζεται ως	Γρηγορότερο προϊόν στην αγορά.
Ταχύτητα	Πιο αργό από το ASIC.
Επαναχρησιμοποίηση και ενελίξια	Επαναχρησιμοποίησμα και ενέλικτο.
Απόβλητα	Μη αποφεύγιμα.
Κατάλληλο για	Όταν οι απαιτούμενες ποσότητες είναι μικρές.
	Όταν οι απαιτούμενες ποσότητες είναι μεγάλες.

Πίνακας 5.1: Διαφορές μεταξύ των FPGAs και ASICs.

5.3 Εισαγωγή στις Γλώσσες Υλικού (HDLs)

Οι γλώσσες περιγραφής υλικού (Hardware Description Languages - HDLs) είναι ειδικές γλώσσες προγραμματισμού που χρησιμοποιούνται για τον ορισμό, την προσομοίωση και την επαλήθευση της συμπεριφοράς ηλεκτρονικών κυκλωμάτων και συστημάτων σε επίπεδο υλικού. Σε αντίθεση με τις παραδοσιακές γλώσσες προγραμματισμού που περιγράφουν τον τρόπο εκτέλεσης ενός αλγόριθμου από έναν επεξεργαστή, οι γλώσσες υλικού επικεντρώνονται στην

περιγραφή της δομής και της λειτουργίας των ίδιων των κυκλωμάτων, όπως είναι τα λογικά κυκλώματα, οι καταχωρητές, οι πολυπλέκτες, και τα άλλα ψηφιακά δομικά στοιχεία. Οι HDLs επιτρέπουν στους σχεδιαστές να περιγράψουν ψηφιακά συστήματα με έναν ακριβή και αυστηρό τρόπο.

Οι δύο πιο γνωστές και χρησιμοποιούμενες γλώσσες περιγραφής υλικού είναι η VHDL, οπού και χρησιμοποιήθηκε στα πλαίσια της διπλωματικής εργασίας, και η Verilog. Αμφότερες οι γλώσσες έχουν αναπτυχθεί για να επιτρέπουν τη δημιουργία, προσομοίωση και σύνθεση κυκλωμάτων σε FPGAs, ASICs και άλλες τεχνολογίες υλικού.

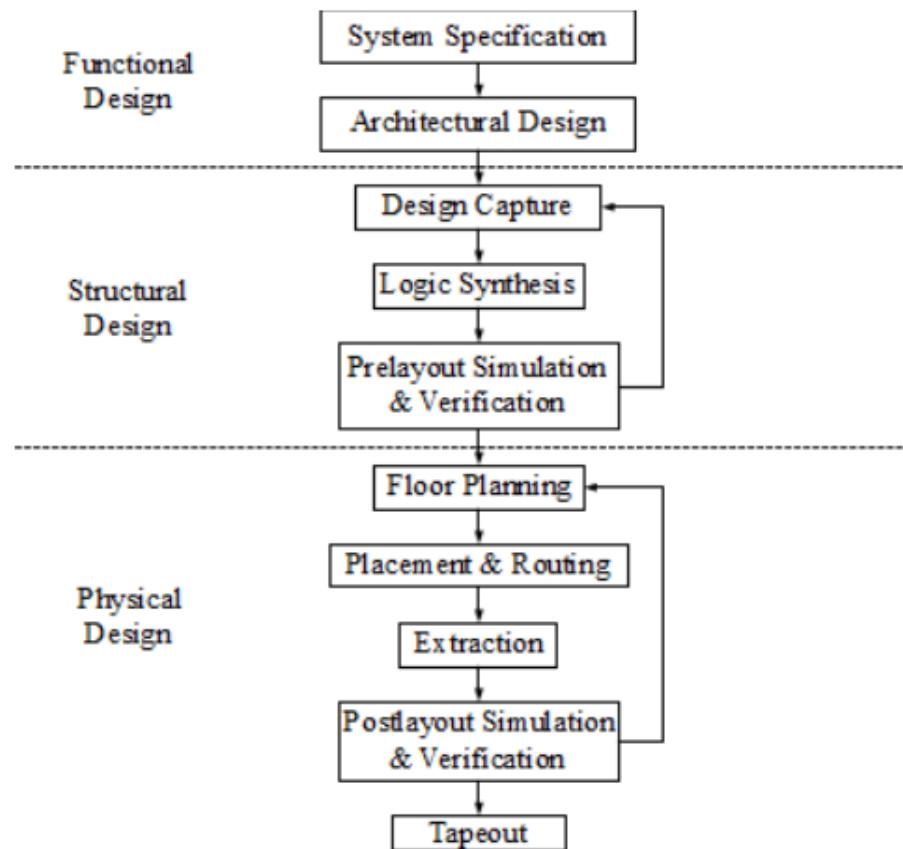
Η VHDL ή αλλιώς VHSIC (Very High Speed Integrated Circuits Program) Hardware Description Language είναι μια γλώσσα περιγραφής υλικού που αναπτύχθηκε αρχικά από το Υπουργείο Άμυνας των Ηνωμένων Πολιτειών για να καλύψει τις ανάγκες περιγραφής και προσομοίωσης πολύπλοκων ψηφιακών συστημάτων. Είναι μια ισχυρή γλώσσα με αρκετά αυστηρή σύνταξη, που επιτρέπει τον σχεδιασμό πολύπλοκων κυκλωμάτων με αυστηρό έλεγχο της λειτουργίας τους. Ένα σημαντικό χαρακτηριστικό της VHDL είναι η δυνατότητα περιγραφής του σχεδιασμού σε διαφορετικά επίπεδα αφαίρεσης, από το πολύ υψηλό επίπεδο (π.χ. συμπεριφορική περιγραφή) μέχρι το επίπεδο πύλης (gate-level).

Η Verilog από την άλλη είναι μια γλώσσα περιγραφής υλικού με πιο απλή και προσβάσιμη σύνταξη σε σύγκριση με την VHDL, και είναι αρκετά δημοφιλής στη βιομηχανία για τον σχεδιασμό και την προσομοίωση ψηφιακών συστημάτων. Εισάγει μια πιο προγραμματιστική προσέγγιση στον σχεδιασμό υλικού, με σύνταξη που μοιάζει αρκετά με αυτήν των παραδοσιακών γλωσσών προγραμματισμού όπως η C. Η Verilog χρησιμοποιείται ευρέως σε πολλά πεδία και είναι αρκετά δημοφιλής για τη σχεδίαση ψηφιακών κυκλωμάτων.

Οι γλώσσες περιγραφής υλικού χρησιμοποιούνται σε όλα τα στάδια της σχεδίασης ενός ψηφιακού κυκλώματος. Από την αρχική σύλληψη της ιδέας και τη λειτουργική περιγραφή του κυκλώματος μέχρι την προσομοίωση και τη σύνθεση για φυσική υλοποίηση. Οι σχεδιαστές μπορούν να περιγράψουν τη συμπεριφορά του κυκλώματος, να δοκιμάσουν εναλλακτικές αρχιτεκτονικές και να επιβεβαιώσουν ότι το κύκλωμα λειτουργεί όπως αναμένεται προτού πρωτήσουν στην κατασκευή του. Στην εικόνα 5.2 φαίνεται η σχεδιαστική ροή ενός ψηφιακού σχεδιασμού.

Μια από τις κύριες δυνατότητες των HDLs είναι ότι επιτρέπουν την προσομοίωση του σχεδίου σε υπολογιστικό περιβάλλον, επιτρέποντας στους μηχανικούς να επαληθεύουν τη λειτουργία του κυκλώματος σε διαφορετικά σενάρια και καταστάσεις πριν την υλοποίηση του στο υλικό. Αυτή η δυνατότητα προσφέρει σημαντικά πλεονεκτήματα στη μείωση του κόστους και του χρόνου ανάπτυξης, καθώς επιτρέπει την ανίχνευση και διόρθωση σφαλμάτων πολύ νωρίτερα στη διαδικασία.

Συνολικά, οι γλώσσες περιγραφής υλικού προσφέρουν στους μηχανικούς τη δυνατότητα να σχεδιάζουν και να επαληθεύουν ψηφιακά κυκλώματα με μεγαλύτερη ευελιξία, ακρίβεια και ταχύτητα.



Σχήμα 5.2: Βήματα ροής ψηφιακού σχεδιασμού [37].

6

Οικογένεια Hummingbird

6.1 Εισαγωγή στην Οικογένεια των Hummingbird Αλγορίθμων

Η οικογένεια των αλγορίθμων Hummingbird αντιπροσωπεύει μια σημαντική εξέλιξη στον τομέα της κρυπτογραφίας για συσκευές με περιορισμένους πόρους, όπως τα RFID tags, οι ασύρματοι αισθητήρες και άλλες ενσωματωμένες συσκευές. Αναπτύχθηκαν για να αντιμετωπίσουν τις προκλήσεις της κρυπτογράφησης και αυθεντικοποίησης σε περιβάλλοντα όπου η υπολογιστική ισχύς, η κατανάλωση ενέργειας και ο χώρος αποθήκευσης είναι αυστηρά περιορισμένα. Ο αρχικός αλγόριθμος Hummingbird σχεδιάστηκε ως ένα υβρίδιο μεταξύ block cipher και stream cipher, επιτυγχάνοντας έτσι την παροχή ισχυρής ασφάλειας με μικρό μέγεθος block και εξαιρετική απόδοση σε περιβάλλοντα χαμηλής ισχύος [38].

Οι αλγόριθμοι αυτοί, οι οποίοι αναπτύχθηκαν αρχικά από τους Engels, Schultz, Schweitzer και Smith, εισήγαγαν καινοτομίες που τους καθιστούν κατάλληλους για εφαρμογές όπου απαιτείται ταχεία απόκριση και χαμηλή κατανάλωση ενέργειας, όπως προβλέπεται στο πρωτόκολλο ISO 18000-6C [39]. Συγκεκριμένα, ο Hummingbird-2, μια εξέλιξη του αρχικού αλγορίθμου, ενσωματώνει βελτιώσεις που τον καθιστούν ανθεκτικό σε πολλές γνωστές κρυπταναλυτικές επιθέσεις. Συνεπώς, αποτελεί έναν από τους πλέον ασφαλείς και αποδοτικούς αλγορίθμους για συσκευές με περιορισμένους πόρους [40].

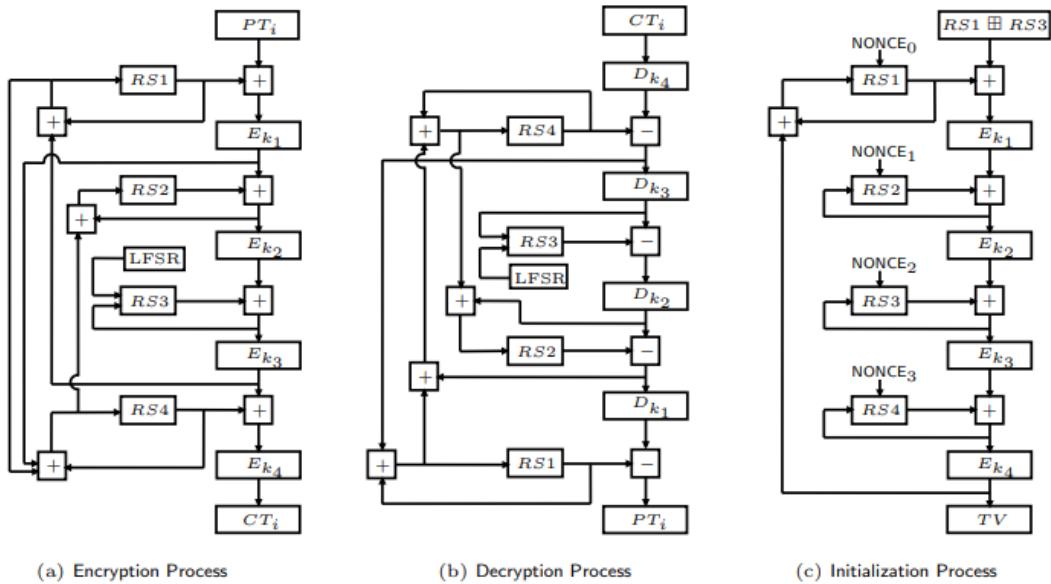
Σε αυτό το κεφάλαιο, θα εξετάσουμε λεπτομερώς τη λειτουργία του αλγορίθμου Hummingbird-1, αναλύοντας τις κρυπτογραφικές του ιδιότητες καθώς και τις ευπάθειες του σε διάφορες επιθέσεις. Στη συνέχεια, θα επικεντρωθούμε στον Hummingbird-2, ο οποίος αποτελεί και το κύριο θέμα αυτής της εργασίας. Θα αναλυθεί πλήρως η αρχιτεκτονική του, η λειτουργία του και το επίπεδο ασφαλείας του, παρέχοντας έτσι μια ολοκληρωμένη κατανόηση της ασφάλειας και της απόδοσης του σε σύγχρονα κρυπτογραφικά περιβάλλοντα.

6.2 Hummingbird-1

Ο αλγόριθμος Hummingbird-1 αντιπροσωπεύει μια κανονικόμο προσέγγιση στην κρυπτογράφηση, συνδυάζοντας χαρακτηριστικά τόσο των μπλοκ αλγορίθμων όσο και των αλγορίθμων ροής. Ο αλγόριθμος αυτός χρησιμοποιεί μπλοκ δεδομένων μεγέθους 16 bit, με ένα κλειδί κρυπτογράφησης μεγέθους 256 bit και εσωτερική κατάσταση 80 bit, που προσφέρει ένα επίπεδο ασφάλειας επαρκές για πολλές ενσωματωμένες εφαρμογές. Η αρχιτεκτονική του Hummingbird-1 περιλαμβάνει τέσσερις 16 bit μπλοκ αλγορίθμους κρυπτογράφησης ($E_{k_1}, E_{k_2}, E_{k_3}, E_{k_4}$), τέσσερις 16 bit καταχωρητές εσωτερικής κατάστασης (RS_1, RS_2, RS_3, RS_4), και ένα 16 bit στάδιο LFSR [38].

Στη διαδικασία κρυπτογράφησης, κάθε μπλοκ κειμένου 16 bit περνά αρχικά από μια πράξη πρόσθεσης $modulo\ 2^{16}$ με το περιεχόμενο του πρώτου καταχωρητή εσωτερικής κατάστασης (RS_1). Το αποτέλεσμα στη συνέχεια κρυπτογραφείται από τον πρώτο block cipher (E_{k_1}). Αυτή η διαδικασία επαναλαμβάνεται διαδοχικά για τους άλλους τρεις block ciphers, με το τελικό αποτέλεσμα να αποτελεί το κρυπτογραφημένο κείμενο (CT_i). Κατά τη διάρκεια αυτής της διαδικασίας, οι εσωτερικοί καταχωρητές ενημερώνονται με τρόπο που δεν μπορεί να προβλεφθεί, βασιζόμενοι στην τρέχουσα κατάσταση τους, τα αποτελέσματα των τριών πρώτων block ciphers και την κατάσταση του LFSR [38].

Η αποκρυπτογράφηση ακολουθεί παρόμοιο μοτίβο αντίστροφο όμως της κρυπτογράφησης. Για την αρχικοποίηση του Hummingbird, χρησιμοποιούνται τέσσερις 16 bit τιμές ($NONCE_i$) που ορίζουν τις αρχικές καταστάσεις των εσωτερικών καταχωρητών. Στη συνέχεια, εκτελούνται τέσσερις διαδοχικές κρυπτογραφήσεις για την παραγωγή του τελικού 16 bit κρυπτογραφημένου κειμένου (TV), το οποίο χρησιμοποιείται για την αρχικοποίηση του LFSR [38].



Σχήμα 6.1: Περιγραφή Top-Level του κρυπτογραφικού αλγορίθμου Hummingbird [38].

S-Boxes και Δομή του 16 bit Block Cipher E_{k_1}

Κεντρικό ρόλο στον Hummingbird-1 παίζουν οι τέσσερις 16 bit block ciphers $E_{k_1}, E_{k_2}, E_{k_3}, E_{k_4}$, οι οποίοι είναι βασισμένοι στα SPN δίκτυα. Κάθε ένας από αυτούς τους αλγορίθμους μπλοκ αποτελείται από τέσσερις γύρους, με έναν τελευταίο γύρο που περιλαμβάνει τα βήματα ανάμιξης κλειδιού και υποκατάστασης. Το τελικό αποτέλεσμα αυτού του γύρου αποτελεί την έξοδο του block cipher, η οποία στη συνέχεια χρησιμοποιείται για την ενημέρωση των εσωτερικών καταχωρητών και την παραγωγή του κρυπτογραφημένου κειμένου. Το κλειδί κρυπτογράφησης των 256 bit διαιρείται σε τέσσερα υποκλειδιά των 64 bit, τα οποία χρησιμοποιούνται στην λογική πράξη XOR μαζί με τα δεδομένα του κάθε ενός από τα τέσσερα block ciphers [38].

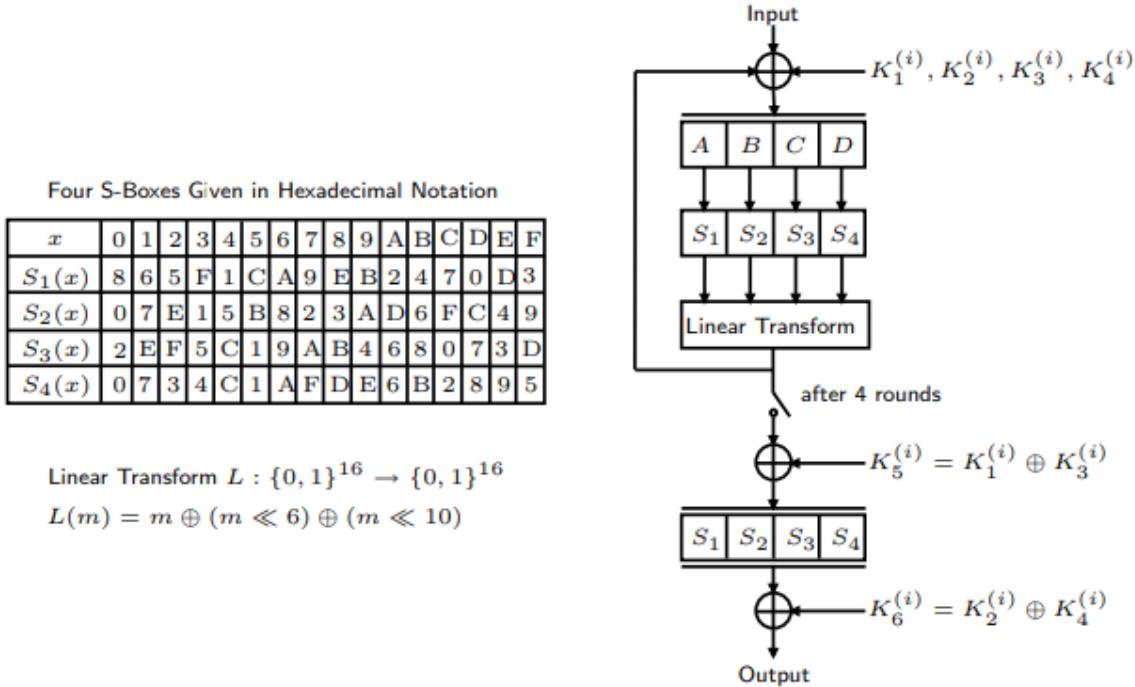
Η υποκατάσταση (substitution) επιτυγχάνεται με τη χρήση τεσσάρων S-boxes, οι οποίες είναι Serpent-type και έχουν εισόδους και εξόδους 4 bit. Κάθε S-box έχει σχεδιαστεί για να προσφέρει ανθεκτικότητα απέναντι σε κοινές επιθέσεις, όπως γραμμική και διαφορική κρυπτανάλυση. Τα S-boxes είναι προσεκτικά επιλεγμένα ώστε να ικανοποιούν κριτήρια όπως η μέγιστη αλγεβρική βαθμίδα και ο ελάχιστος αριθμός μονομερών όρων, προκειμένου να παρέχουν υψηλή ασφάλεια με ταυτόχρονα χαμηλό κόστος υλοποίησης υλικού [38].

Το στρώμα μετάθεσης (permutation layer) υλοποιείται με τη γραμμική μετασχηματιστική συνάρτηση $L(m) = m \oplus (m \lll 6) \oplus (m \lll 10)$, όπου m είναι ένα 16 bit μπλοκ δεδομένων. Αυτή η μετασχηματιστική συνάρτηση εξασφαλίζει την ανάμειξη των bits του μπλοκ, συμβάλλοντας στη διάχυση των bits του κλειδιού και των δεδομένων σε όλο το μπλοκ, κάτι που δυσκολεύει σημαντικά τις επιθέσεις [38].

16 bit Stage LFSR

Ο βασικός ρόλος του LFSR στον Hummingbird-1 είναι η ενημέρωση των εσωτερικών καταχωρητών κατά τη διάρκεια της κρυπτογράφησης. Ο LFSR έχει χαρακτηριστικό πολυώνυμο $f(x) = x^{16} + x^{15} + x^{12} + x^{10} + x^7 + x^3 + 1$, και διασφαλίζει την παραγωγή ψευδοτυχαίων ακολουθιών που είναι κρίσιμες για την ασφάλεια του αλγορίθμου [38].

Κατά την ενημέρωση των εσωτερικών καταχωρητών, ο LFSR εκτελεί ένα βήμα πριν χρησιμοποιηθεί για την ενημέρωση του καταχωρητή RS_2 ενώ το 13o bit του LFSR είναι πάντα ρυθμισμένο για να αποτρέψει τη δημιουργία μηδενικού καταχωρητή, κάτι που θα μπορούσε να αποδυναμώσει την ασφάλεια της κρυπτογράφησης [38].

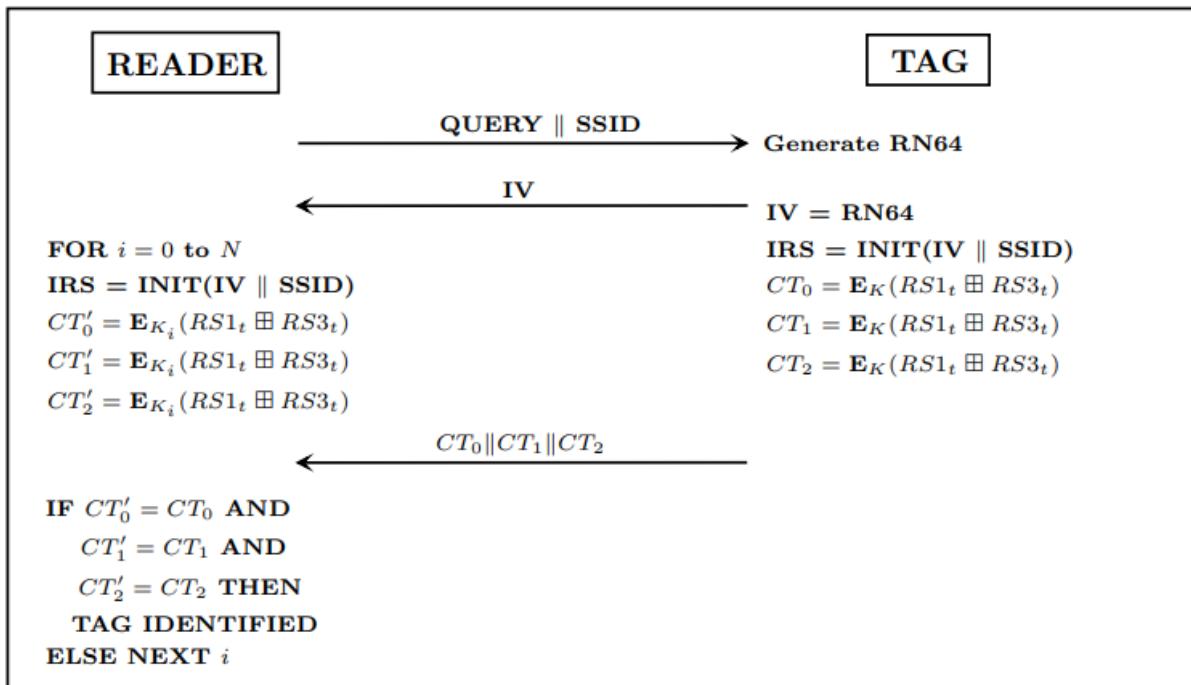


Σχήμα 6.2: Τα 4 S-boxes και η δομή του μπλοκ κρυπτογράφησης στον κρυπτογραφικό αλγόριθμο Hummingbird [38].

Πρωτόκολλο Αυθεντικοποίησης του Hummingbird-1

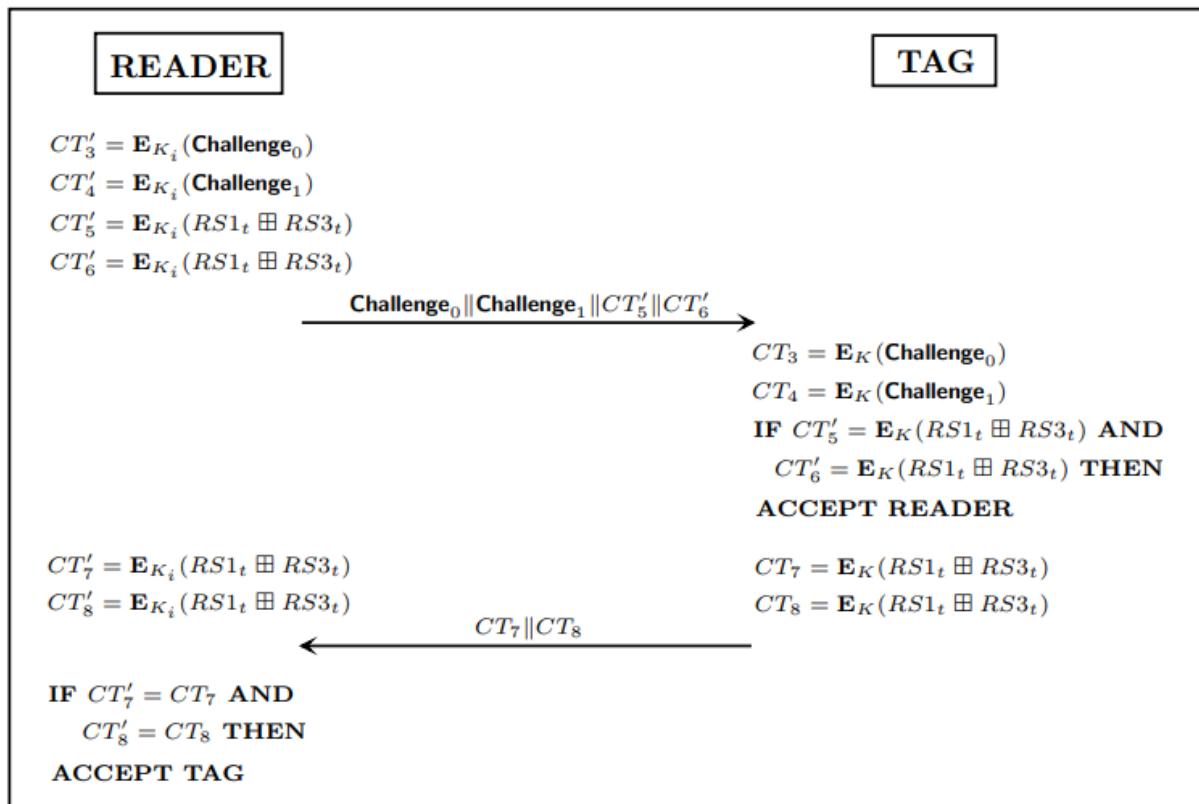
Το πρωτόκολλο αυθεντικοποίησης του Hummingbird-1 σχεδιάστηκε για να εξασφαλίσει ότι τόσο ο αναγνώστης όσο και η ετικέτα RFID μπορούν να αυθεντικοποιήσουν ο ένας τον άλλον πριν την μετάδοση ή την εκτέλεση οποιασδήποτε εντολής. Το πρωτόκολλο αποτελείται από τρεις κύριες φάσεις :

- Αναγνώριση της Ετικέτας με Διατήρηση της Ιδιωτικότητας:** Στην πρώτη φάση, ο αναγνώστης στέλνει ένα αίτημα (*QUERY*) στην ετικέτα μαζί με ένα μοναδικό ID συνεδρίας (*SSID*). Η ετικέτα δημιουργεί τέσσερις τυχαίους αριθμούς (*IVs*) και τους μεταδίδει στον αναγνώστη. Αυτοί οι αριθμοί χρησιμοποιούνται για την αρχικοποίηση των εσωτερικών καταχωρητών της ετικέτας. Στη συνέχεια, η ετικέτα κρυπτογραφεί μια τιμή που συνδυάζει τους εσωτερικούς καταχωρητές και παράγει τρεις κωδικοποιημένους δείκτες (CT_0, CT_1, CT_2), οι οποίοι χρησιμοποιούνται για την αναγνώριση της ετικέτας χωρίς να αποκαλύπτεται η ταυτότητα της [39].



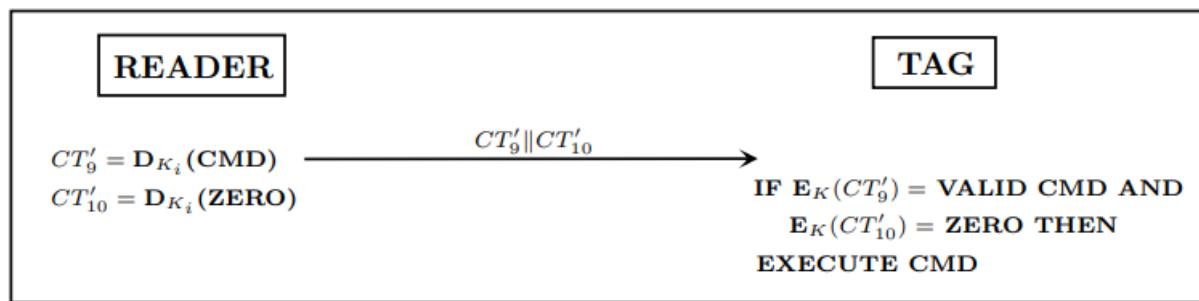
Σχήμα 6.3: To Hummingbird Πρωτόκολλο Ιδιωτικής Ταυτοποίησης [39].

2. **Αμοιβαία Αυθεντικοποίηση μεταξύ Αναγνώστη και Ετικέτας:** Αφού η ετικέτα αναγνωριστεί από τον αναγνώστη, ακολουθεί η φάση της αμοιβαίας αυθεντικοποίησης. Ο αναγνώστης δημιουργεί δύο τυχαίες προκλήσεις ($Challenge_0$ και $Challenge_1$) και τις κρυπτογραφεί χρησιμοποιώντας την τρέχουσα εσωτερική κατάσταση του αλγορίθμου Hummingbird. Στη συνέχεια, ο αναγνώστης υπολογίζει δύο κωδικοποιημένους δείκτες (CT_5 και CT_6) και τους στέλνει μαζί με τις προκλήσεις στην ετικέτα. Η ετικέτα, αφού κρυπτογραφήσει τις προκλήσεις, συγκρίνει τους δείκτες που υπολόγισε με αυτούς που έλαβε από τον αναγνώστη. Αν υπάρχει ταύτιση, η ετικέτα επιβεβαιώνει την αυθεντικότητα του αναγνώστη. Στη συνέχεια, η ετικέτα υπολογίζει και στέλνει δύο νέους δείκτες (CT_7 και CT_8) στον αναγνώστη, ο οποίος με τη σειρά του επιβεβαιώνει την αυθεντικότητα της ετικέτας [39].



Σχήμα 6.4: Το πρωτόκολλο αμοιβαίου ελέγχου ταυτότητας Hummingbird [39].

3. **Εκτέλεση Εντολής:** Στην τελική φάση, η ετικέτα λαμβάνει και εκτελεί μια εντολή που της έχει αποσταλεί από τον αυθεντικοποιημένο αναγνώστη. Η εντολή και μια τιμή ελέγχου (όλα μηδενικά) κρυπτογραφούνται και αποστέλλονται στην ετικέτα. Η ετικέτα αποκρυπτογραφεί τα δεδομένα για να ανακτήσει την εντολή και εάν η εντολή είναι έγκυρη, την εκτελεί [39].



Σχήμα 6.5: Ο αναγνώστης εκδίδει μια εντολή στην ετικέτα [39].

Το πρωτόκωλο αυτό παρέχει έναν αποδοτικό μηχανισμό αμοιβαίας αυθεντικοποίησης, ενώ ταυτόχρονα διασφαλίζει την ιδιωτικότητα της ετικέτας μέσω της χρήσης ψευδοτυχαίων αριθμών και μοναδικών αναγνωριστικών συνεδρίας. Η χρήση του Hummingbird-1 σε αυτό το πρωτόκολλο εξασφαλίζει ότι οι πράξεις κρυπτογράφησης και αυθεντικοποίησης μπορούν να εκτελούνται σε πραγματικό χρόνο ακόμα και σε συσκευές με περιορισμένη υπολογιστική ισχύ, όπως τα RFID tags.

Επιπλέον, η χρήση διαφορετικών προκλήσεων και κωδικοποιημένων δεικτών για κάθε συνεδρία επικοινωνίας μειώνει τον κίνδυνο επαναχρησιμοποίησης δεδομένων και αυξάνει την αντίσταση σε επιθέσεις αναπαραγωγής (replay attacks). Αυτό καθιστά τον Hummingbird-1 κατάλληλο για εφαρμογές που απαιτούν υψηλή ασφάλεια σε περιβάλλοντα με χαμηλούς πόρους.

6.2.1 Ευπάθειες Hummingbird-1

Αν και ο Hummingbird-1, σχεδιάστηκε με στόχο την παροχή ασφάλειας για περιβάλλοντα με περιορισμένους πόρους, παρουσιάζει σημαντικές ευπάθειες, κυρίως λόγω του σχεδιασμού της αρχικοποίησης (initialization) και της δομής του. Αυτές οι ευπάθειες μπορούν να αξιοποιηθούν για την ανάκτηση του μυστικού κλειδιού, και έτσι να θέσουν σε κίνδυνο την ασφάλεια του συστήματος.

Κατά τη διαδικασία της αρχικοποίησης, το σύστημα φορτώνει την τιμή του IV (Initialization Vector) στους εσωτερικούς καταχωρητές της κατάστασης και εκτελεί τέσσερις γύρους κρυπτογράφησης, χρησιμοποιώντας τις μετατοπισμένες τιμές των εσωτερικών καταχωρητών και τα υποκλειδιά. Ωστόσο, έχει παρατηρηθεί ότι η διαφορά στα ανωτέρα bits των καταχωρητών παραμένει αμετάβλητη κατά τη διαδικασία αυτή, γεγονός που οδηγεί σε απώλεια πληροφορίας και αφήνει το σύστημα ευάλωτο σε επιθέσεις με διαφορική ανάλυση [41].

Επιπλέον, το μικρό μέγεθος της κατάστασης των 80 bit καθιστά τον Hummingbird-1 ευάλωτο σε επιθέσεις επιλεγμένου IV (chosen-IV attack). Σε τέτοιες επιθέσεις, ο επιτιθέμενος μπορεί να εκμεταλλευτεί τη μικρή αυτή κατάσταση και να προκαλέσει συγκεκριμένες διαφορές στους καταχωρητές της εσωτερικής κατάστασης, οδηγώντας σε ανάκτηση του μυστικού κλειδιού με πολύ λιγότερες υπολογιστικές προσπάθειες από τις θεωρητικά απαιτούμενες. Συγκεκριμένα χρειάζεται μόλις ένα υπολογιστικό φορτίο πολύ μικρότερο από 2^{64} υπολογιστικές πράξεις και με δεδομένα που δεν ξεπερνούν μερικά MegaByte (MB) [41].

Συγκεκριμένα οι επιθέσεις αυτού του τύπου στον Hummingbird-1, εκμεταλλεύονται τη δομή των 16 bit υποκλειδιών και των λειτουργιών του αλγορίθμου, όπως την αλληλονοχία XOR και των S-boxes, που παρουσιάζουν προβλήματα διάχυσης των διαφορών στα ανώτερα bits. Ειδικότερα, η διαδικασία κρυπτογράφησης των πρώτων 16 bit του μηνύματος μέσω των τεσσάρων 16 bit υποκλειδιών, μπορεί να παρακαμφθεί με τη χρήση επιλεγμένων μηνυμάτων και IVs που εκμεταλλεύονται τη συγκεκριμένη αδυναμία του συστήματος.

Συνοψίζοντας, η ανάλυση του Hummingbird-1 αποκάλυψε κρίσιμες αδυναμίες στη διαδικασία εκκίνησης, στον σχεδιασμό των S-boxes, και στη συνολική ασφάλεια του αλγορίθμου έναντι επιθέσεων με επιλεγμένο IV μήνυμα. Παρότι σχεδιάστηκε για περιβάλλοντα με περιορισμένους πόρους, η περιορισμένη ασφάλεια του αλγορίθμου καθιστά το Hummingbird-1 μη κατάλληλο για εφαρμογές όπου απαιτείται υψηλό επίπεδο ασφάλειας. Αυτές οι αδυναμίες οδήγησαν στην ανάπτυξη του Hummingbird-2, ο οποίος επιχειρεί να διορθώσει τις ευπάθειες αυτές [41].

6.3 Hummingbird-2

6.3.1 Εισαγώγη στον Hummingbird-2

Ο Hummingbird-2, αναπτυγμένος ως απάντηση στις ευπάθειες που αποκαλύφθηκαν κατά την κρυπτανάλυση του Hummingbird-1, αποτελεί μια εξελιγμένη έκδοση του προκατόχου του, σχεδιασμένη να προσφέρει αυξημένη ασφάλεια σε περιβάλλοντα με αυστηρούς περιορισμούς πόρων, όπως τα RFID, οι ασύρματοι αισθητήρες και οι βιομηχανικοί ελεγκτές. Ενσωματώνει χαρακτηριστικά τόσο από τους παραδοσιακούς αλγόριθμους ροής όσο και από τους block ciphers, γεγονός που τον καθιστά μια υβριδική λύση κατάλληλη για περιπτώσεις όπου απαιτείται κρυπτογράφηση μικρών μηνυμάτων, όπως συμβαίνει με τις συσκευές RFID και τους ασύρματους αισθητήρες. Αυτή η υβριδική προσέγγιση προσφέρει τόσο υψηλή απόδοση όσο και ευκολία στην ενσωμάτωση σε διάφορα πρωτόκολλα ασφάλειας [42].

Ο Hummingbird-2 λειτουργεί με μπλοκ 16 bit, κάτι που τον καθιστά αποτελεσματικότερο σε εφαρμογές που χειρίζονται μικρά μηνύματα. Σε σύγκριση με τον Hummingbird-1, ο Hummingbird-2 έχει σχεδιαστεί με βάση τις απαιτήσεις ασφάλειας που επιβλήθηκαν από την κρυπτανάλυση του πρώτου μοντέλου, και είναι ανθεκτικός σε όλες τις γνωστές κρυπταναλυτικές επιθέσεις. Παράλληλα, ο αλγόριθμος έχει σχεδιαστεί ώστε να υλοποιείται με πολύ μικρό αποτύπωμα τόσο σε υλικό (hardware) όσο και σε λογισμικό (software), καθιστώντας τον ιδανικό για εφαρμογές με χαμηλό κόστος και περιορισμένους πόρους [42].

Η εξέλιξη του Hummingbird-2 αντικατοπτρίζει τη συνεχή ανάγκη για βελτίωση των κρυπτογραφικών μεθόδων σε ένα περιβάλλον όπου οι επιθέσεις γίνονται όλο και πιο εξελιγμένες.

6.3.2 Γενική Αρχιτεκτονική Σχεδιασμού

Ο Hummingbird-2 βασίζεται σε ένα μυστικό κλειδι K των 128 bit και σε μια εσωτερική κατάσταση R των 128 bit, η οποία αρχικοποιείται με τη χρήση ενός Διανύσματος Αρχικοποίησης IV των 64 bit. Τα παραπάνω μπορούν να αναπαρασταθούν ως διανύσματα των 16 bit λέξεων, όπως φαίνεται παρακάτω :

$$\begin{aligned} K &= (K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_8), \\ R &= (R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8), \\ IV &= (IV_1, IV_2, IV_3, IV_4). \end{aligned}$$

Σε αντίθεση με τον Hummingbird-1, ο Hummingbird-2 αυξάνει το μέγεθος της εσωτερικής κατάστασης από 80 bit σε 128 bit, βελτιώνοντας σημαντικά την ασφάλεια του συστήματος. Η αύξηση αυτή επιτυγχάνεται με την προσθήκη τεσσάρων νέων καταχωρητών, R_5, R_6, R_7, R_8 , οι οποίοι αναφέρονται ως «καταχωρητές συσσωρευτή» (accumulator registers). Επιπλέον, έχει αφαιρεθεί ο LFSR, ο οποίος υπήρχε στον Hummingbird-1, βελτιώνοντας τη δομή του αλγορίθμου και απλοποιώντας τη διαδικασία επεξεργασίας [42]. Ο Hummingbird-2 έχει κατασκευαστεί εξολοκλήρου από πράξεις των 16 bit λέξεων, οι οποίες είναι :

- Λογική πράξη Αποκλειστικού OR (\oplus).
- Προσθήκη Modulo 2^{16} (\boxplus).
- Μη γραμμική συνάρτηση ανάμειξης $f(x)$.

Μια ακόμα σημαντική διαφορά με τον Hummingbird-1 είναι η βελτιωμένη δομή του *WD16* (πρώην «*Ebox*»), δηλαδή του βασικού μπλοκ μετασχηματισμού, το οποίο πλέον εκτελεί τέσσερις κλήσεις στα S-boxes αντί για πέντε, όπως γινόταν στον Hummingbird-1. Η αλλαγή αυτή αυξάνει την ταχύτητα κρυπτογράφησης του αλγορίθμου χωρίς να θυσιάζεται η ασφάλεια [42]. Επιπλέον, ο Hummingbird-2 έχει βελτιωμένο μηχανισμό αυθεντικοποίησης, ο οποίος προστατεύει από επιθέσεις επέκτασης μηνύματος, ενώ έχει εισαχθεί υποστήριξη για αυθεντικοποίηση μη κρυπτογραφημένων σχετιζόμενων δεδομένων (AEAD mode). Τέλος, ο Hummingbird-2 είναι συμβατός με τις απαιτήσεις χρονισμού του πρωτοκόλλου ISO 18000-6C, που τον καθιστά ιδανικό για εφαρμογές σε RFID και άλλες συσκευές με περιορισμένους πόρους [42].

S-boxes

Στον Hummingbird-2, η επιλογή των S-boxes διαδραματίζει κρίσιμο ρόλο στη διασφάλιση της κρυπτογραφικής ανθεκτικότητας. Τα S-boxes (όπως παρουσιάζονται στον Πίνακα 6.1) επιλέχθηκαν μέσω εξαντλητικής αναζήτησης ανάμεσα σε $16!$ δυνατές μεταθέσεις, με βάση την εξειδικευμένη έρευνα [43]. Σκοπός ήταν η βέλτιστη δυνατή συμπεριφορά τους όσον αφορά την ασφάλεια έναντι διαφορικής και γραμμικής κρυπτανάλυσης, καθώς και η διασφάλιση της πολυπλοκότητας στη δομή του αλγορίθμου.

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$S_1(x)$	7	12	14	9	2	1	5	15	11	6	13	0	4	8	10	3
$S_2(x)$	4	10	1	6	8	15	7	12	3	0	14	13	5	9	11	2
$S_3(x)$	2	15	12	1	5	6	10	13	14	8	3	4	0	11	9	7
$S_4(x)$	15	4	5	8	9	7	2	1	10	3	0	14	6	12	13	11

Πίνακας 6.1: Τα S-boxes που χρησιμοποιούνται στον Hummingbird-2 [42]

Συγκεκριμένα, τα S-boxes ανήκουν σε κατηγορίες που ικανοποιούν τα παρακάτω κριτήρια:

- Διαφορικό όριο (differential bound):** Το ανώτατο όριο πιθανότητας μιας επιτυχούς διαφορικής ανάλυσης έχει οριστεί στο $p \leq 1/4$, ενώ η πιθανότητα επιτυχούς γραμμικής ανάλυσης περιορίζεται στο $\epsilon \leq 1/4$. Επιπλέον, ο αριθμός κλάδων (branch number) έχει οριστεί στο 3, εξασφαλίζοντας ότι οι διαφορές διαχέονται αποτελεσματικά μέσω των S-boxes.[42]
- Ελαχιστοποίηση διαφορικών χαρακτηριστικών:** Υπάρχει ελάχιστος αριθμός διαφορικών χαρακτηριστικών και γραμμικών προσεγγίσεων στα όρια αυτά, μειώνοντας τον κίνδυνο εκμετάλλευσης από επιθέσεις διαφορικής ή γραμμικής κρυπτανάλυσης.[42]
- Γραμμική Ισοδυναμία:** Όλα τα S-boxes ανήκουν σε διαφορετικές κατηγορίες γραμμικής ισοδυναμίας, ενισχύοντας τη δυσκολία ανάλυσης του αλγορίθμου μέσω γραμμικών μεθόδων.[42]
- Απόσταση Hamming:** Τα τέσσερα S-boxes έχουν μεγάλη απόσταση Hamming μεταξύ τους, καθώς και από την ταυτοτική μετάθεση (identity permutation). Αυτό σημαίνει ότι ακόμη και μικρές αλλαγές στα δεδομένα εισόδου προκαλούν σημαντικές αλλαγές στα δεδομένα εξόδου, ενισχύοντας την ασφάλεια του συστήματος.[42]

- **Αλγεβρικός βαθμός:** Ο αλγεβρικός βαθμός όλων των εξόδων, εκτός από μία, είναι 3, εξασφαλίζοντας ότι κάθε bit εξόδου εξαρτάται μη γραμμικά από τον μέγιστο δυνατό αριθμό εισερχόμενων bits. Αυτή η μη γραμμικότητα επιπλέον ενισχύει την αντοχή του αλγορίθμου σε επιθέσεις με χρήση πολυωνυμικών αναλύσεων [42].

Η συγκεκριμένη επιλογή των S-boxes βελτιστοποιεί την ανθεκτικότητα του Hummingbird-2 απέναντι σε κοινές κρυπταναλυτικές επιθέσεις ενώ παράλληλα διατηρεί χαμηλό υπολογιστικό κόστος.

Μη Γραμμική Συνάρτηση Ανάμειξης $f(x)$

Η μη γραμμική συνάρτηση $f(x)$ αποτελεί ένα κρίσιμο στοιχείο στην αρχιτεκτονική του Hummingbird-2. Η δομή της βασίζεται σε αναζητήσεις μετάθεσης σε S-boxes των τεσσάρων bit, οι οποίες εφαρμόζονται σε κάθε nibble (τετράδα bit) της λέξης, και ακολουθείται από έναν γραμμικό μετασχηματισμό για την ενίσχυση της διάχυσης.

Πιο συγκεκριμένα, η λειτουργία $S(x)$ αναφέρεται στον υπολογισμό τεσσάρων S-boxes, $S_1(x_0), S_2(x_1), S_3(x_2)$ και $S_4(x_3)$, όπου κάθε S-box λειτουργεί σε ένα nibble της εισόδου. Ο γραμμικός μετασχηματισμός $L(x)$ ορίζεται ως η εφαρμογή του τελεστή \oplus και της αριστερής κυκλικής ολίσθησης (\lll) κατά 6 και 10 θέσεις, αντίστοιχα.

Η συνολική συνάρτηση $f(x)$ μπορεί να γραφεί ως εξής :

$$\begin{aligned} S(x) &= S_1(x_0)|S_2(x_1)|S_3(x_2)|S_4(x_3), \\ L(x) &= x \oplus (x \lll 6) \oplus (x \lll 10), \\ f(x) &= L(S(x)). \end{aligned}$$

Η αρχιτεκτονική της $f(x)$ φαίνεται στην εικόνα 6.6, όπου παρουσιάζεται η αλληλουχία των σταδίων της μη γραμμικής ανάμειξης και ο τρόπος με τον οποίο τα S-boxes και ο γραμμικός μετασχηματισμός συνεργάζονται για την παραγωγή του τελικού αποτελέσματος.

Συνάρτηση Κλειδωμένης Μετάθεσης $WD16$

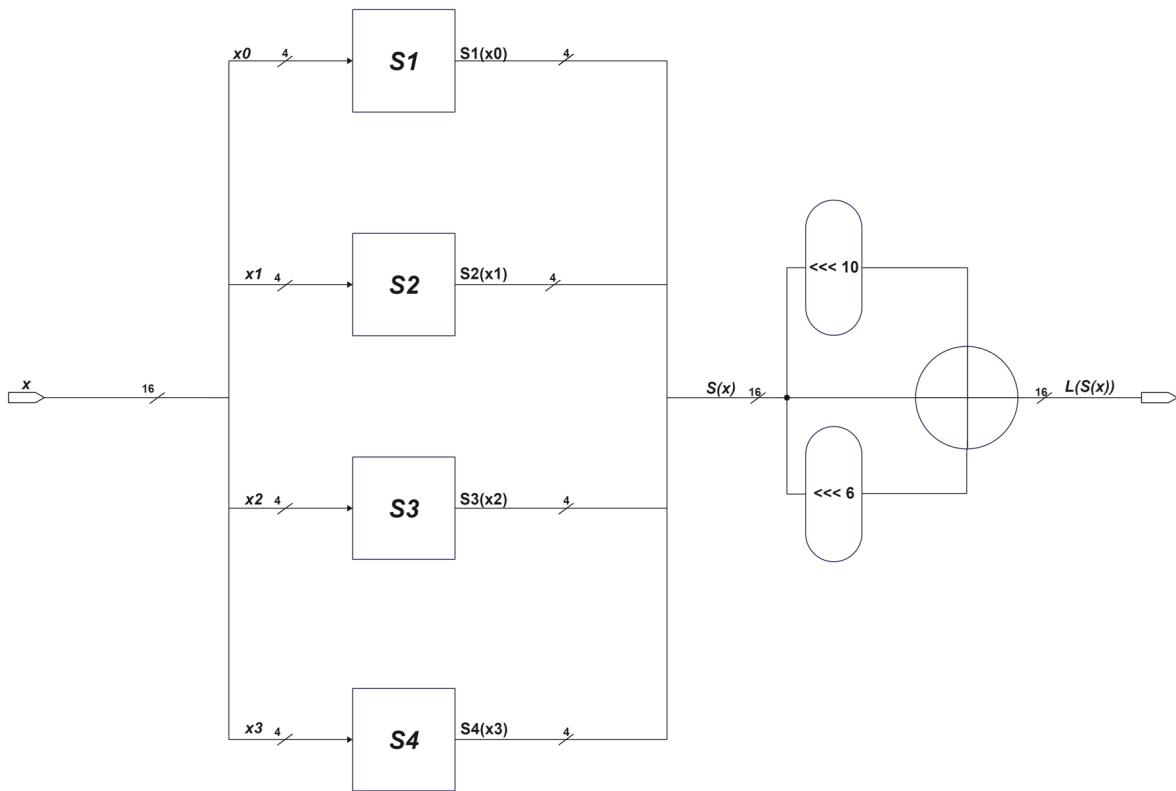
Η συνάρτηση $WD16$ στον Hummingbird-2 είναι μια 16 bit μη γραμμική συναρτησιακή μετασχηματιστική λειτουργία, η οποία βασίζεται στη χρήση της $f(x)$. Η $WD16$ ακολουθεί μια τυπική δομή δικτύου SPN, περιλαμβάνοντας πολλαπλές φάσεις υποκατάστασης και γραμμικής αντιμετάθεσης, όπως συναντάμε σε κλασικά SPN.

Συγκεκριμένα, η $WD16$ ορίζεται ως:

$$WD16(x, a, b, c, d) = f(f(f(f(x \oplus a) \oplus b) \oplus c) \oplus d)$$

Όπως και σε τυπικά SPN, η $WD16$ περιλαμβάνει:

- **Μη γραμμική υποκατάσταση:** Η συνάρτηση $f(x)$ εκτελεί υποκατάσταση μέσω των S-boxes σε κάθε βήμα.
- **Γραμμική ανάμειξη:** Η γραμμική ανάμειξη διασφαλίζει τη διάχυση της πληροφορίας σε όλο το μπλοκ δεδομένων, δημιουργώντας πολύπλοκες σχέσεις μεταξύ των bits.
- **Προσθήκη κλειδιών μέσω XOR:** Σε κάθε βήμα, προστίθεται ένα νέο υποκλειδί μέσω της πράξης XOR με τα υποκλειδιά a, b, c και d , προσθέτοντας σύγχυση και διάχυση στα δεδομένα.



Σχήμα 6.6: Αρχιτεκτονική της μη γραμμικής συνάρτησης ανάμιξης $f(x)$.

Αυτό το δομικό στοιχείο είναι η καρδιά του Hummingbird-2, συμβάλλοντας στην ασφάλεια μέσω των χαρακτηριστικών σύγχυσης και διάχυσης που προσφέρει η δομή SPN. Η αρχιτεκτονική της $WD16$ φαίνεται στην εικόνα 6.7.

Μονάδα Αρχικοποίησης

Η διαδικασία αρχικοποίησης(εικόνα 6.8) του Hummingbird-2, βασίζεται σε τέσσερις γύρους όπου χρησιμοποιείται το IV των 64 bit για να δημιουργήσει την αρχική κατάσταση. Αυτή η διαδικασία έχει σχεδιαστεί για να παρέχει υψηλό επίπεδο ασφάλειας με μικρό υπολογιστικό κόστος, κατάλληλο για περιβάλλοντα με περιορισμένους πόρους.

Η εσωτερική κατάσταση R ορίζεται αρχικά ως εξής:

$$R^{(0)} = (IV_1, IV_2, IV_3, IV_4, IV_1, IV_2, IV_3, IV_4)$$

Για τέσσερις γύρους $i = 0, 1, 2, 3$, η κατάσταση ενημερώνεται μέσω της συνάρτησης $WD16$ και προσθηκών $modulo 2^{16}$, ακολουθούμενη από κυκλικές μετατοπίσεις. Αναλυτικά, η διαδικασία έχει ως εξής:

- **Υπολογισμός των ενδιάμεσων τιμών :**

$$\begin{aligned} t_1 &= WD16(R_1^{(i)} \boxplus (i), K_1, K_2, K_3, K_4) \\ t_2 &= WD16(R_2^{(i)} \boxplus t_1, K_5, K_6, K_7, K_8) \\ t_3 &= WD16(R_3^{(i)} \boxplus t_2, K_1, K_2, K_3, K_4) \\ t_4 &= WD16(R_4^{(i)} \boxplus t_3, K_5, K_6, K_7, K_8) \end{aligned}$$

- **Ενημέρωση των καταχωρητών κατάστασης:** Οι καταχωρητές κατάστασης ενημερώνονται με βάση τα παραπάνω ενδιάμεσα αποτελέσματα χρησιμοποιώντας κυκλικές μετατοπίσεις και προσθήκες *modulo 2¹⁶*:

$$\begin{aligned} R_1^{(i+1)} &= (R_1^{(i)} \boxplus t_4) \lll 3 \\ R_2^{(i+1)} &= (R_2^{(i)} \boxplus t_1) \ggg 1 \\ R_3^{(i+1)} &= (R_3^{(i)} \boxplus t_2) \lll 8 \\ R_4^{(i+1)} &= (R_4^{(i)} \boxplus t_3) \lll 1 \end{aligned}$$

- **Αθροιστική ενημέρωση των καταχωρητών R5 - R8:** Οι καταχωρητές R_5, R_6, R_7, R_8 , ενημερώνονται μέσω των καταχωρητών R_1, R_2, R_3, R_4 :

$$\begin{aligned} R_5^{(i+1)} &= R_5^{(i)} \oplus R_1^{(i+1)} \\ R_6^{(i+1)} &= R_6^{(i)} \oplus R_2^{(i+1)} \\ R_7^{(i+1)} &= R_7^{(i)} \oplus R_3^{(i+1)} \\ R_8^{(i+1)} &= R_8^{(i)} \oplus R_4^{(i+1)} \end{aligned}$$

Η τελική κατάσταση $R^{(4)}$ χρησιμοποιείται για την κρυπτογράφηση του πρώτου μπλοκ δεδομένων.

Η χρήση του *WD16* εξασφαλίζει την πολυπλοκότητα των ενημερώσεων της κατάστασης και τη διασφάλιση της μη επαναληψιμότητας της εσωτερικής κατάστασης. Η διαδικασία αυτή σίναι one-to-one από το IV στις καταχωρημένες καταστάσεις R_1, R_2, R_3, R_4 , αποφεύγοντας έτσι συγκρούσεις nonce.

Οι καταχωρητές R_5, R_6, R_7, R_8 χρησιμοποιούνται για να συσσωρεύουν τις τιμές των καταχωρητών R_1, R_2, R_3, R_4 , μέσω του XOR, προσφέροντας αυξημένη ασφάλεια έναντι πλευρικών επιθέσεων και ενίσχυση του αλγορίθμου σε περιορισμένους πόρους [45].

Η διαδικασία αυτή επιτρέπει τη διατήρηση μιας ισχυρής κατάστασης ασφάλειας κατά την αρχικοποίηση, κάτι που είναι απαραίτητο για την ανθεκτικότητα του *Hummingbird-2* απέναντι σε διάφορες κρυπτανάλυτικές επιθέσεις, ενώ διατηρεί την ευκολία υλοποίησης σε υλικό και λογισμικό.

Μονάδα Μπλοκ 16 bit Κρυπτογράφησης

Η μονάδα κρυπτογράφησης των 16 bit μπλοκ (εικόνα 6.9) του *Hummingbird-2* χρησιμοποιεί την τελική κατάσταση που παράγεται από τη διαδικασία αρχικοποίησης και βασίζεται στη λειτουργία *WD16*. Κάθε λέξη από το κείμενο εισόδου, δηλαδή το μπλοκ δεδομένων προς κρυπτογράφηση, περνά από τέσσερα στάδια *WD16*, δημιουργώντας την κρυπτογραφημένη λέξη.

Αρχικά, για κάθε λέξη του αρχικού κειμένου P_i , το σύστημα ξεκινάει την κρυπτογράφηση καλώντας τη συνάρτηση *WD16* με την πρώτη κατάσταση R_1 και το κείμενο εισόδου. Η πρώτη ενδιάμεση τιμή t_1 προκύπτει ως εξής :

$$t_1 = WD16(R_1^{(i)} \boxplus P_i, K_1, K_2, K_3, K_4)$$

Στη συνέχεια, η δεύτερη τιμή t_2 υπολογίζεται χρησιμοποιώντας την τιμή του t_1 , την κατάσταση R_2 και τα τροποποιημένα υποκλειδιά K_5, K_6, K_7, K_8 , τα οποία τροποποιούνται με τις τιμές των

καταχωρητών R_5 έως R_8 αντίστοιχα:

$$t_2 = WD16(R_2^{(i)} \boxplus t_1, K_5 \oplus R_5^{(i)}, K_6 \oplus R_6^{(i)}, K_7 \oplus R_7^{(i)}, K_8 \oplus R_8^{(i)})$$

Η διαδικασία συνεχίζεται με την επεξεργασία της τρίτης κατάστασης R_3 και της τιμής t_2 , καθώς και των υποκλειδών K_1, K_2, K_3, K_4 που συνδυάζονται με τις τιμές των καταχωρητών R_5 έως R_8 :

$$t_3 = WD16(R_3^{(i)} \boxplus t_2, K_1 \oplus R_5^{(i)}, K_2 \oplus R_6^{(i)}, K_3 \oplus R_7^{(i)}, K_4 \oplus R_8^{(i)})$$

Τέλος, η λέξη κρυπτογράφησης C_i προκύπτει από την τέταρτη κατάσταση R_4 και την τιμή t_3 , μαζί με τα υποκλειδιά K_5, K_6, K_7, K_8 και την αρχική κατάσταση R_1 :

$$C_i = WD16(R_4^{(i)} \boxplus t_3, K_5, K_6, K_7, K_8) \boxplus R_1^{(i)}$$

Αφού ολοκληρωθεί η κρυπτογράφηση μιας λέξης, η εσωτερική κατάσταση ενημερώνεται ώστε να είναι έτοιμη για την κρυπτογράφηση της επόμενης λέξης. Η διαδικασία ενημέρωσης των καταχωρητών R περιλαμβάνει την προσθήκη *modulo* 2^{16} των ενδιάμεσων τιμών t_1, t_2, t_3 στις αντίστοιχες καταστάσεις:

$$\begin{aligned} R_1^{(i+1)} &= R_1^{(i)} \boxplus t_3 \\ R_2^{(i+1)} &= R_2^{(i)} \boxplus t_1 \\ R_3^{(i+1)} &= R_3^{(i)} \boxplus t_2 \\ R_4^{(i+1)} &= R_4^{(i)} \boxplus R_1^{(i)} \boxplus t_3 \boxplus t_1 \end{aligned}$$

Επιπλέον, οι καταχωρητές R_5, R_6, R_7, R_8 , ενημερώνονται με τις τιμές R_1, R_2, R_3, R_4 χρησιμοποιώντας πράξεις XOR:

$$\begin{aligned} R_5^{(i+1)} &= R_5^{(i)} \oplus R_1^{(i+1)} \\ R_6^{(i+1)} &= R_6^{(i)} \oplus R_2^{(i+1)} \\ R_7^{(i+1)} &= R_7^{(i)} \oplus R_3^{(i+1)} \\ R_8^{(i+1)} &= R_8^{(i)} \oplus R_4^{(i+1)} \end{aligned}$$

Μονάδα stream cipher Λειτουργίας για Κωδικοποίηση Πεδίων Μικρού Σταθερού Μήκους

Η μονάδα stream cipher λειτουργίας στον Hummingbird-2 έχει σχεδιαστεί για να παρέχει κωδικοποίηση πεδίων μικρού σταθερού μήκους χωρίς αύξηση του μεγέθους του μηνύματος. Αυτό είναι ιδιαίτερα χρήσιμο όταν απαιτείται η μετάδοση μικρών μηνυμάτων, των οποίων το μήκος μπορεί να κυμαίνεται από 1 έως 15 bits. Ο βασικός στόχος αυτής της μεθόδου είναι η αποτελεσματική κωδικοποίηση τέτοιων μηνυμάτων χωρίς επιπλέον αύξηση των δεδομένων, κάτι που είναι κρίσιμο σε περιβάλλοντα με περιορισμένους πόρους και απαιτήσεις για συμπαγή κρυπτογράφηση.

Έστω ότι το μήνυμα προς μετάδοση είναι το x , το οποίο αποτελείται από $1 \leq n \leq 15$ bits. Το κρυπτογραφημένο μήνυμα προκύπτει από τα n λιγότερα σημαντικά bits της τιμής $x \oplus E(0)$, οπού η συνάρτηση $E(0)$ αναπαριστά την κρυπτογράφηση της λέξης 0 χρησιμοποιώντας την εσωτερική κατάσταση του αλγορίθμου τη δεδομένη στιγμή. Η λειτουργία XOR μεταξύ του μηνύματος και του (0) εξασφαλίζει την κρυπτογράφηση του μικρού πεδίου χωρίς την ανάγκη για αύξηση του μεγέθους [45].

Εάν απαιτείται διασφάλιση της ακεραιότητας του μηνύματος, η εσωτερική κατάσταση του αλγορίθμου μπορεί να ενημερωθεί περαιτέρω μέσω της κρυπτογράφησης του ίδιου του μηνύματος x χρησιμοποιώντας την εσωτερική κατάσταση από την κρυπτογράφηση που προέκυψε από την $E(0)$, κάτι που παρέχει έναν επιπλέον μηχανισμό για την επαλήθευση του περιεχομένου του μηνύματος κατά τη διαδικασία αποκρυπτογράφησης. Να σημειώθει εδώ ότι η κρυπτογράφηση του ίδιου μηνύματος x δεν παράγει κάποια έξοδο στο C_i [45].

Μονάδα Υπολογισμού MAC

Η μονάδα υπολογισμού του MAC στον Hummingbird-2 έχει σχεδιαστεί για να εξασφαλίζει την ακεραιότητα και την αυθεντικότητα των μηνυμάτων. Ο υπολογισμός του MAC πραγματοποιείται με βάση τη δομή της εσωτερικής κατάστασης του αλγορίθμου και τη χρήση του κρυπτογραφικού μηχανισμού για την παραγωγή ενός κωδικού αυθεντικοποίησης (authentication tag) [45].

Η διαδικασία υπολογισμού του MAC ξεκινάει μετά την ολοκλήρωση της κρυπτογράφησης ή της επεξεργασίας του μηνύματος. Ουσιαστικά, η ακολουθία υπολογισμού του MAC πραγματοποιεί μια τελική επικύρωση του μηνύματος, χρησιμοποιώντας την εσωτερική κατάσταση του αλγορίθμου πριν από την παραγωγή της έξόδου. Συγκεκριμένα, πριν την παραγωγή οποιουδήποτε αποτελέσματος, η εσωτερική κατάσταση του αλγορίθμου Hummingbird-2 ανανεώνεται τρεις φορές χωρίς να παράγεται έξοδος, ως εξής:

1. Η πρώτη κλήση της E πραγματοποιείται στο $IV_1 \oplus R_1 \oplus R_3 \oplus n$, όπου το n είναι ο αριθμός των λέξεων του μηνύματος (μέχρι 8 λέξεις).
2. Η δεύτερη κλήση πραγματοποιείται στο $IV_2 \oplus R_1 \oplus R_3$.
3. Η τρίτη κλήση πραγματοποιείται στο $IV_3 \oplus R_1 \oplus R_3$.

Εδώ, τα R_1 και R_3 αναφέρονται στην περιεχόμενη κατάσταση των αντίστοιχων καταχωρητών ακριβώς πριν από κάθε κλήση της συνάρτησης E .

Μετά την ολοκλήρωση αυτών των τριών βημάτων, το MAC δημιουργείται μέσω της παρακάτω διαδικασίας:

- Ο πρώτος κωδικός αυθεντικοποίησης T_1 υπολογίζεται ως $T_1 = E(IV_4 \oplus R_1 \oplus R_3)$.
- Οι επόμενες τιμές του κωδικού αυθεντικοποίησης T_i , υπολογίζονται με την κλήση της συνάρτησης E στα $R_1 \oplus R_3$ για κάθε $i = 2, 3, \dots, n$.

Αυτή η διαδικασία υπολογισμού του MAC εξασφαλίζει ότι το παραγόμενο authentication tag είναι απόλυτα συνδεδεμένο με το περιεχόμενο του μηνύματος και την εσωτερική κατάσταση του αλγορίθμου τη δεδομένη στιγμή, παρέχοντας έτσι προστασία έναντι επιθέσεων που στοχεύουν στη μεταβολή του μηνύματος χωρίς ανίχνευση.

Μονάδα Μπλοκ Αποκρυπτογράφησης 16 bit

Η μονάδα Αποκρυπτογράφησης των 16 bit μπλοκ, είναι θεμελιώδης για την ανάκτηση του plaintext από το ciphertext. Αυτή η μονάδα λειτουργεί ως η αντίστροφη διαδικασία της E . Για να επιτευχθεί αυτό, είναι απαραίτητο να αντιστρέψουμε τόσο το δομικό στοιχείο $WD16$ όσο και

τις προσθέσεις $modulo\ 2^{16}$ που εφαρμόζονται στο plaintext, αντικαθιστώντας τες με αφαιρέσεις $modulo\ 2^{16}$ (\ominus).

Για να αντιστρέψουμε το $WD16$, πρέπει πρώτα να αντιστρέψουμε τη συνάρτηση $f(x)$ (εικόνα 6.10), η οποία με τη σειρά της απαιτεί την αντιστροφή των S-boxes. Όπως αναφέραμε στο κεφάλαιο 3, τα S-boxes πρέπει να είναι αντιστρέψιμα ώστε να διασφαλιστεί η ομαλή λειτουργία της αποκρυπτογράφησης. Στον Hummingbird-2, τα αντιστρέψιμα S-boxes παρουσιάζονται στον πίνακα 6.2, προσφέροντας την απαραίτητη δομή για την επιτυχή αποκρυπτογράφηση

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$S_1^{-1}(x)$	11	5	4	15	12	6	9	0	13	3	14	8	1	10	2	7
$S_2^{-1}(x)$	9	2	15	8	0	12	3	6	4	13	1	14	7	11	10	5
$S_3^{-1}(x)$	12	3	0	10	11	4	5	15	9	14	6	13	2	7	8	1
$S_4^{-1}(x)$	10	7	6	9	1	2	12	5	3	4	8	15	13	14	11	0

Πίνακας 6.2: Τα S-boxes που χρησιμοποιούνται στην αποκρυπτογράφηση στον Hummingbird-2.

Ωστόσο, η αντιστροφή της $f(x)$ δεν είναι απλή, κυρίως λόγω των σύνθετων λειτουργιών που περιλαμβάνει. Συγκεκριμένα, η συνάρτηση ανάμειξης $f(x)$ αποτελείται από συνδυασμούς αριστερών κυκλικών μετατοπίσεων και πράξεων XOR. Παρόλο που η πράξη XOR μπορεί να αντιστραφεί εύκολα (χρησιμοποιώντας την ίδια πράξη XOR) και η αντιστροφή των αριστερών κυκλικών ολισθήσεων γίνεται μέσω δεξιών κυκλικών μετατοπίσεων, οι αλληλεπιδράσεις μεταξύ τους απαιτούν προσεκτική διαχείριση των ενδιάμεσων αποτελεσμάτων σε πολλαπλούς βρόχους, ώστε να εξασφαλιστεί η ορθή αντιστροφή της διαδικασίας. Πιο συγκεκριμένα:

- Κυκλικές Μετατοπίσεις και XOR:** Οι κυκλικές ολισθήσεις κατά 6 και 10 bits αλλάζουν τις θέσεις των bits, ενώ οι πράξεις XOR αναμειγνύουν bits από διάφορα μέρη της εισόδου. Για να «αναιρεθεί» αυτή η ανάμειξη, πρέπει να «ξετυλιχτούν» προσεκτικά οι λειτουργίες στην αντίθετη κατεύθυνση. Όμως, επειδή οι πράξεις XOR είναι σωρευτικές και επηρεάζουν όλα τα bits, η αντιστροφή μιας πράξης XOR εξαρτάται από την κατάσταση των άλλων bits. Αυτός είναι ο λόγος που απαιτούνται πολλαπλές επαναλήψεις της αντίστροφης λειτουργίας για να επιτευχθεί η σωστή αρχική τιμή.
- Πολλαπλές Επαναλήψεις:** Ο βρόχος που χρειάζεται να τρέξει τρείς φορές υποδηλώνει ότι η αντιστροφή αυτής της μετασχηματισμένης λειτουργίας απαιτεί διαδοχικές προσεγγίσεις της εισόδου, ανακτώντας σταδιακά την αρχική κατάσταση του αναμεμιγμένου κειμένου. Οι διαδοχικές επαναλήψεις βελτιώνουν την εκτίμηση της αρχικής τιμής χρησιμοποιώντας ενδιάμεσες τιμές από την αντίστροφη διαδικασία.

Τα παραπάνω εκφράζονται ως εξής :

$$\begin{aligned} f^{-1}(y) &= L_i^{-1}(S^{-1}(y)), \\ L_i^{-1}(x_{new}) &= y \oplus (x_{old} \ggg 6) \oplus (x_{old} \ggg 10) \\ word_{output} &= S_1^{-1}(x_{new_0}) | S_2^{-1}(x_{new_1}) | S_3^{-1}(x_{new_2}) | S_4^{-1}(x_{new_3}). \end{aligned}$$

όπου y είναι η είσοδος, x_{old} είναι η τρέχουσα τιμή του x από την προηγούμενη επανάληψη, x_{new} είναι η ενημερωμένη τιμή μετά από κάθε επανάληψη και το $word_{output}$ η τελική έξοδος.

Αφού αντιστρέψαμε την λειτουργία της $f(x)$ τότε η αντιστροφή της $WD16$ (εικόνα 6.11) είναι απλή και εκφράζεται ως:

$$WD16^{-1}(y, a, b, c, d) = f^{-1}(f^{-1}(f^{-1}(f^{-1}(y \oplus a) \oplus b) \oplus c) \oplus d)$$

Οπότε τώρα για την αποκρυπτογράφηση ενός 16 bit μπλοκ (εικόνα 6.12), η διαδικασία ακολουθεί μια αντίστροφη πορεία από την κρυπτογράφηση, αξιοποιώντας την αντίστροφη του $WD16$ και της $f(x)$. Έτσι, το ciphertext C_i αποκρυπτογραφείται ως εξής:

1. Υπολογίζουμε το ενδιάμεσο αποτέλεσμα u_3 χρησιμοποιώντας την $WD16^{-1}$:

$$u_3 = WD16^{-1}(C_i \boxminus R_1^{(i)}, K_8, K_7, K_6, K_5)$$

2. Έπειτα υπολογίζουμε το u_2 χρησιμοποιώντας τη δεύτερη $WD16^{-1}$:

$$u_2 = WD16^{-1}(u_3 \boxminus R_4^{(i)}, K_4 \boxplus R_8^{(i)}, K_3 \boxplus R_7^{(i)}, K_2 \boxplus R_6^{(i)}, K_1 \boxplus R_5^{(i)})$$

3. Το επόμενο βήμα είναι να υπολογίσουμε το u_1 :

$$u_1 = WD16^{-1}(u_2 \boxminus R_3^{(i)}, K_8 \boxplus R_8^{(i)}, K_7 \boxplus R_7^{(i)}, K_6 \boxplus R_6^{(i)}, K_5 \boxplus R_5^{(i)})$$

4. Τέλος, αποκρυπτογραφούμε το P_i :

$$P_i = WD16^{-1}(u_1 \boxminus R_2^{(i)}, K_4, K_3, K_2, K_1) \boxminus R_1^{(i)}$$

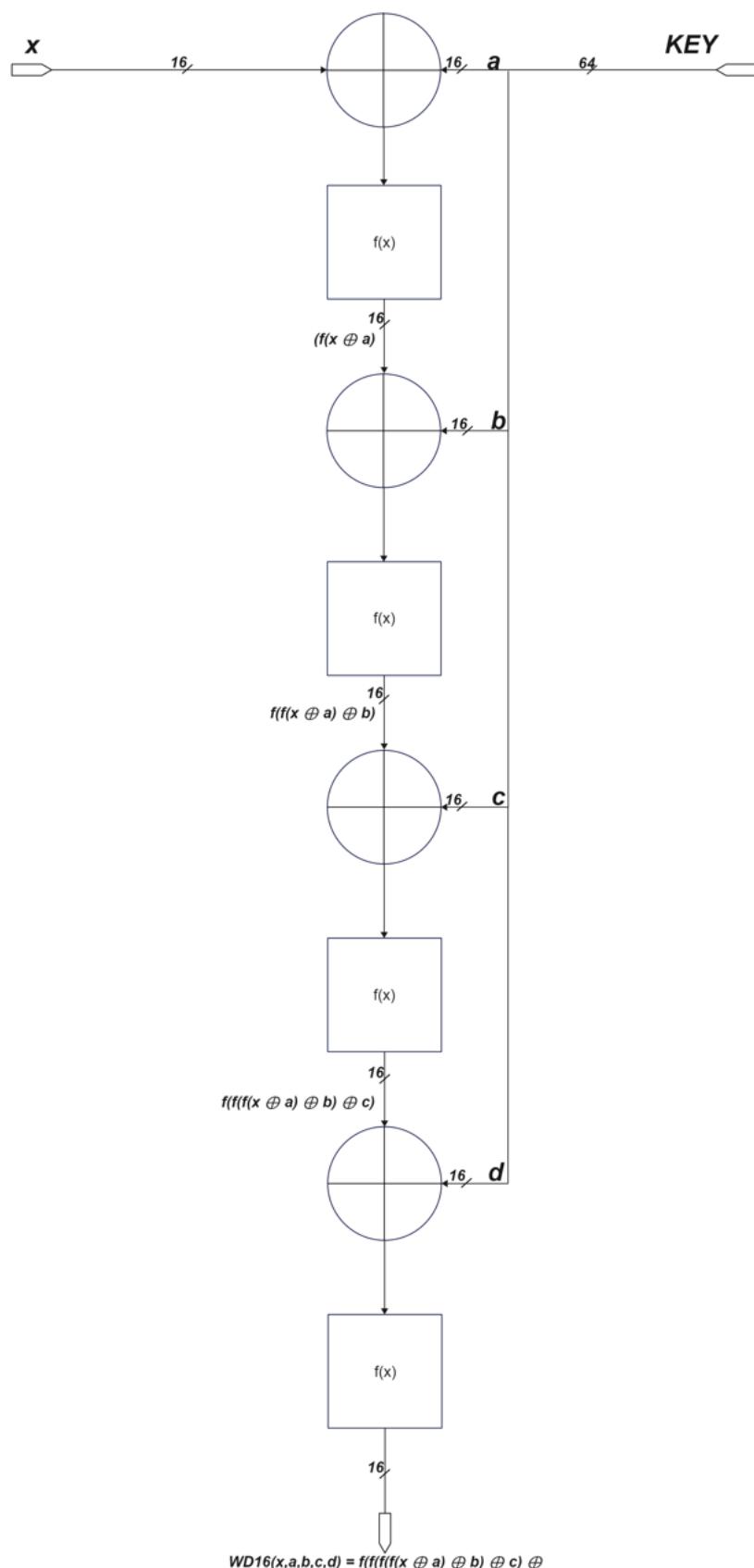
Μετά την ολοκλήρωση της αποκρυπτογράφησης, η εσωτερική κατάσταση ενημερώνεται όπως και κατά την κρυπτογράφηση όπου t_3, t_2, t_1 ισούνται με:

$$t_3 = u_3 \boxminus R_4^{(i)}$$

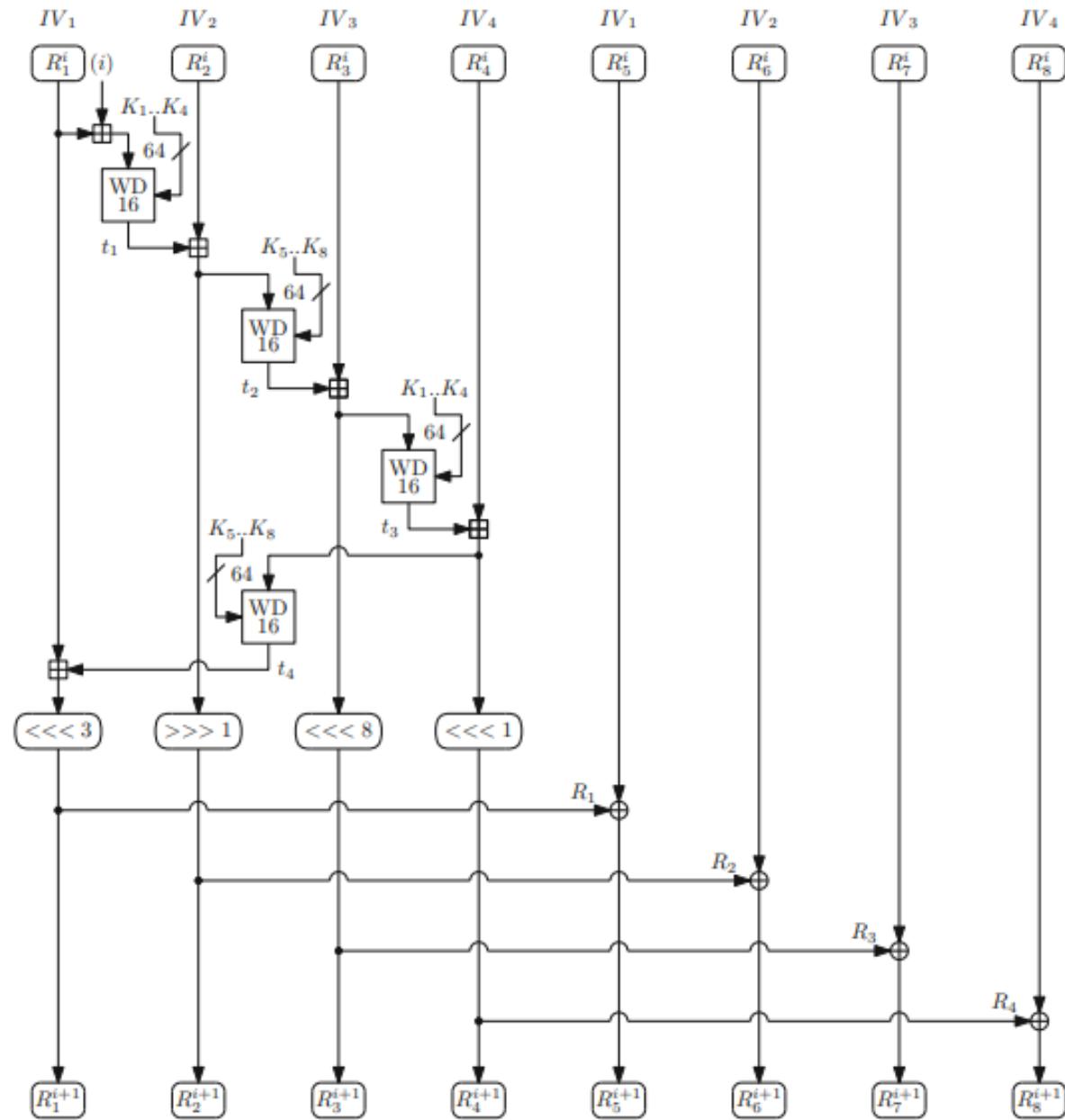
$$t_2 = u_2 \boxminus R_3^{(i)}$$

$$t_1 = u_1 \boxminus R_2^{(i)}$$

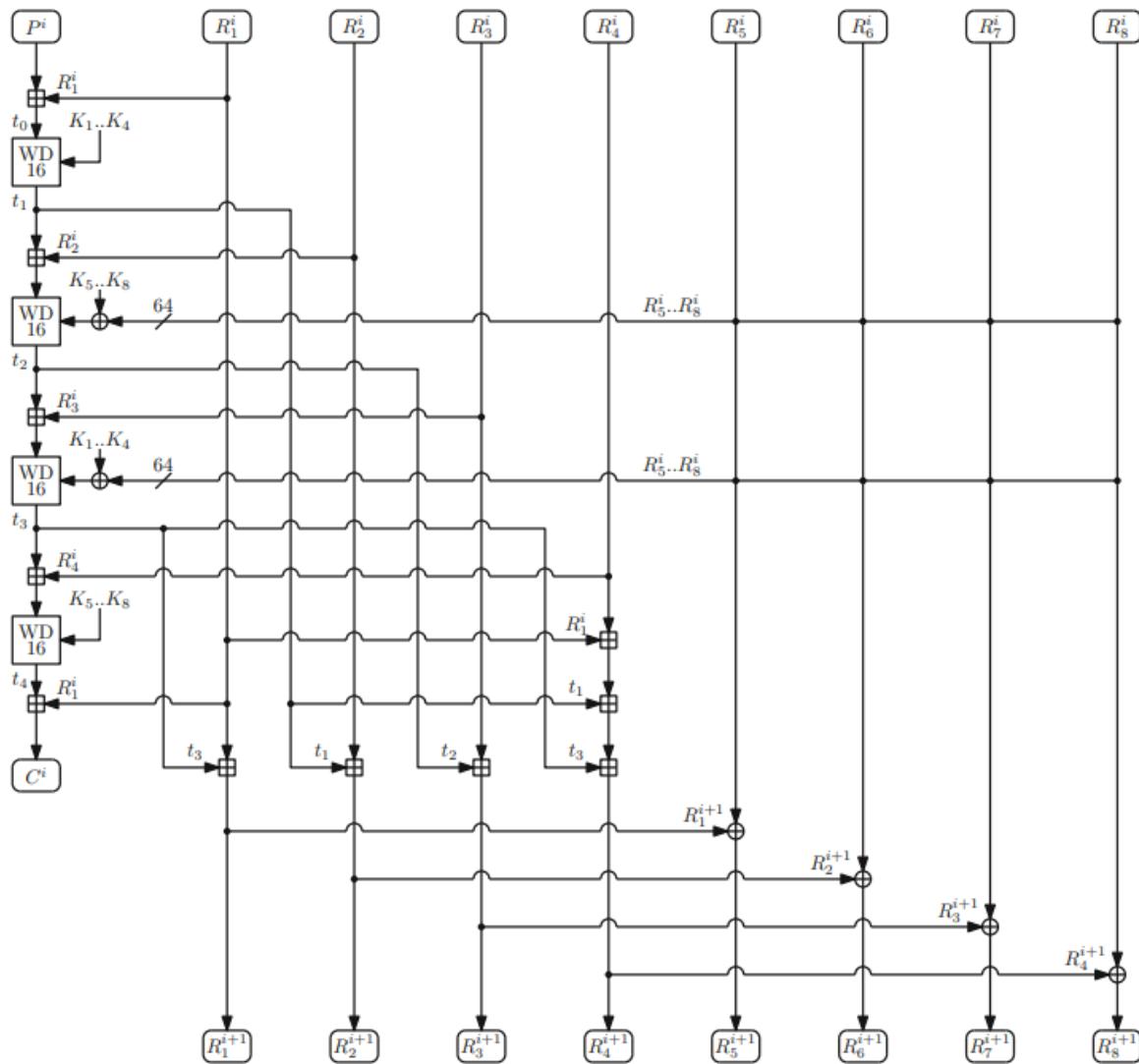
Σημαντικό είναι να αναφέρουμε πως η διαδικασία Αρχικοποίησης και Αυθεντικοποίησης παραμένουν ίδιες.



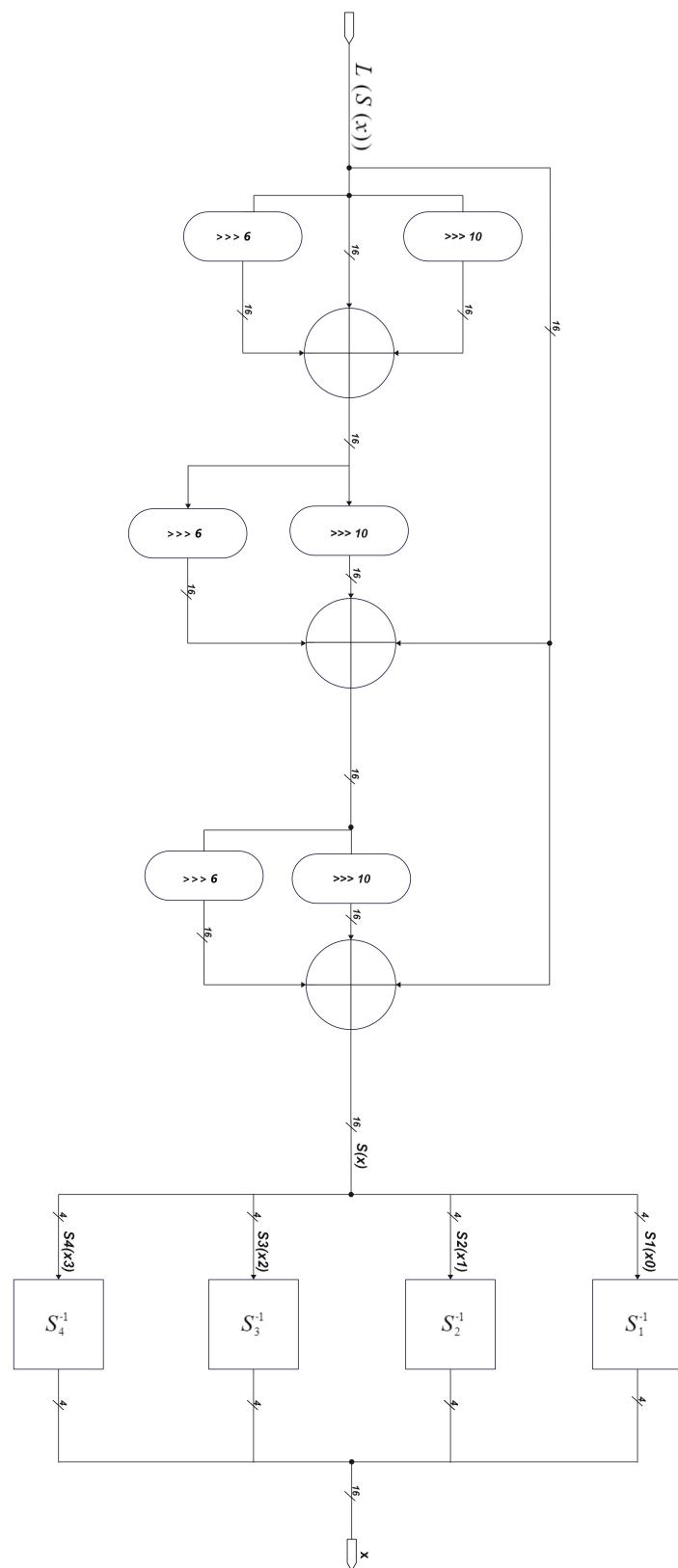
Σχήμα 6.7: Αρχιτεκτονική της συνάρτησης $WD16$.



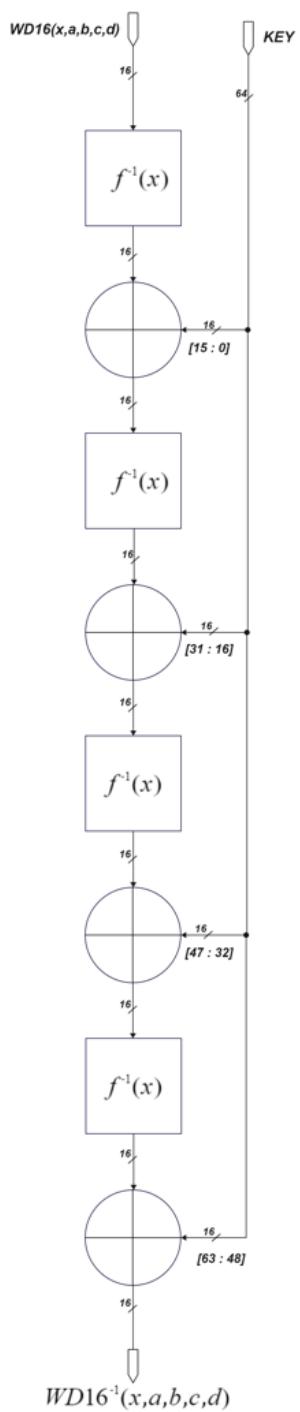
Σχήμα 6.8: Αρχιτεκτονική Μονάδας Αρχικοποιήσης [44].



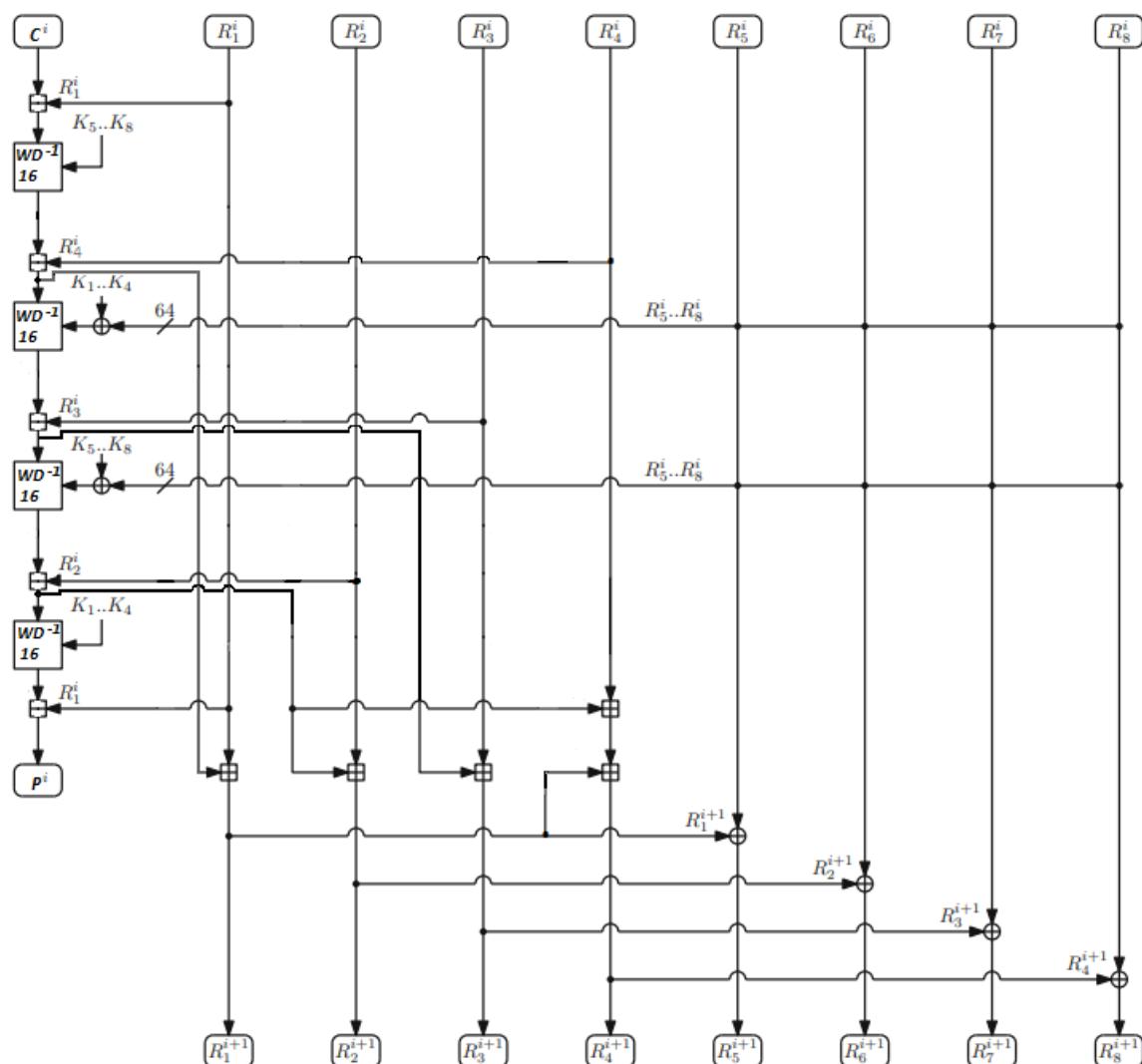
Σχήμα 6.9: Αρχιτεκτονική Μονάδας μπλοκ 16 bit Κρυπτογράφησης [44].



Σχήμα 6.10: Αρχιτεκτονική της $f^{-1}(x)$.



Σχήμα 6.11: Αρχιτεκτονική της $WD16^{-1}$.



Σχήμα 6.12: Αρχιτεκτονική Μονάδας μπλοκ των 16 bit Αποκρυπτογράφησης.

7

Βελτιστοποίησεις του Σχεδιασμού του Hummingbird-2

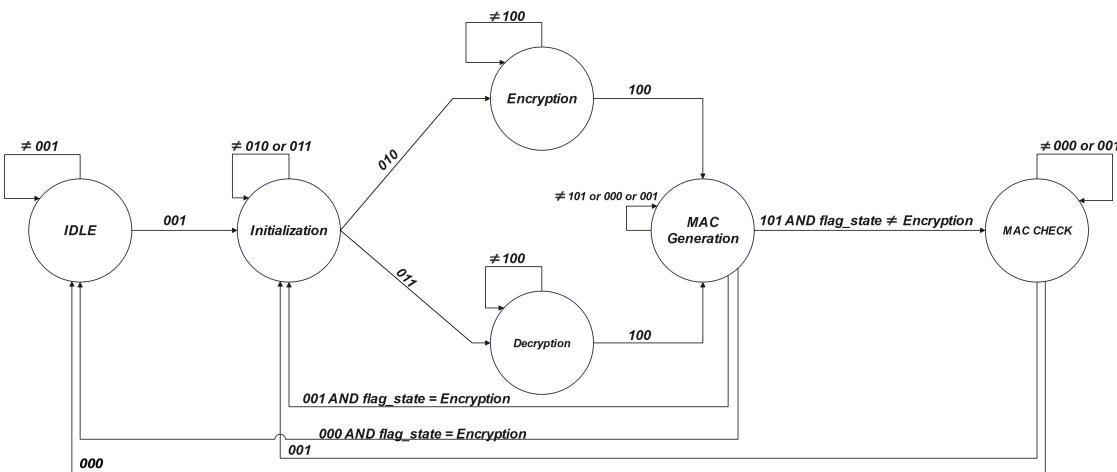
7.1 Προτεινόμενος Σχεδιασμός

Στα πλαίσια της παρούσας εργασίας, σχεδιάστηκε πλήρως ένας επεξεργαστής Hummingbird-2 για block μεγέθους 128 bits, ο οποίος περιέχει τις ακόλουθες βασικές λειτουργίες :

- **Αρχικοποίηση (Initialization).**
- **Κρυπτογράφηση (Encryption).**
- **Λειτουργία stream cipher με ενσωματωμένη ακεραιότητα (Stream cipher mode with integrated integrity).**
- **Αυθεντικοποίηση (Authentication).**
- **Μονάδα ελέγχου ισότητας των δύο MAC (MAC verification unit).**

Τα παραπάνω πέντε στοιχεία αποτελούν τον πυρήνα του συστήματος μας. Επιπλέον για την ομαλή λειτουργία και την απόδοση της σωστής λειτουργίας κάθε φορά δημιουργήθηκαν:

- **Μηχανή πεπερασμένων καταστάσεων (FSM):** Μια μονάδα finite state machine (FSM) η οποία καθορίζει τη ροή των λειτουργιών του συστήματος και τη διαχείριση των διαφορετικών φάσεων (εικόνα 7.1).
- **Μονάδα διαχείρισης σημάτων ελέγχου:** Μια μονάδα πλήρως συνδυαστικής λογικής που διαχειρίζεται τα σήματα ελέγχου των μπλοκ, καθώς και το μήκος και τον αριθμό των λέξεων δεδομένων.



Σχήμα 7.1: Διάγραμμα μηχανής πεπερασμένων καταστάσεων.

Συγκεκριμένα, ο σχεδιασμός αποτελείται από 11 μπλοκ κρυπτογράφησης. Το πρώτο μπλοκ χρησιμοποιείται αποκλειστικά για την πρώτη φάση της αυθεντικοποίησης, ενώ τα επόμενα δύο μπλοκ υποστήριζουν τόσο τις επόμενες δύο φάσεις της αυθεντικοποίησης όσο και τη λειτουργία σε mode stream cipher και την διασφάλιση της ακεραιότητας όπου απαιτείται. Τα υπόλοιπα οκτώ μπλοκ κρυπτογράφησης χρησιμοποιούνται για την παραγωγή τόσο του κρυπτογραφημένου κειμένου όσο και των ετικετών αυθεντικοποίησης.

Επιπλέον, έχουν σχεδιαστεί οκτώ μπλοκ αποκρυπτογράφησης, τα οποία χρησιμοποιούνται αποκλειστικά για την αποκρυπτογράφηση των δεδομένων και την παραγωγή του αρχικού κειμένου.

7.2 Βελτιστοποίηση Σχεδιασμού στο Κόστος Υλικού

Όπως αναφέρθηκε στο κεφάλαιο 6, τόσο η κρυπτογράφηση όσο και η αυθεντικοποίηση υλοποιούνται μέσω των ίδιων μπλοκ κρυπτογράφησης. Αυτή η αρχιτεκτονική προσέγγιση αποτελεί σημαντικό πλεονέκτημα, καθώς εκμεταλλευόμενοι την κοινή χρήση των μπλοκ κρυπτογράφησης, καταφέραμε να μειώσουμε σημαντικά την επιφάνεια που καταλαμβάνει ο συνολικός σχεδιασμός. Αντί να υλοποιηθούν ξεχωριστές μονάδες για την αυθεντικοποίηση και την κρυπτογράφηση, οπού η καθεμία θα απαιτούσε τον δικό της αριθμό μπλοκ κρυπτογράφησης (οκτώ για την κρυπτογράφηση και έντεκα για την αυθεντικοποίηση), σχεδιάσαμε ένα ενιαίο σύστημα που αξιοποιεί κοινά μπλοκ κρυπτογράφησης. Αυτή η βελτιστοποίηση μείωσε τον συνολικό αριθμό των απαιτούμενων μπλοκ από δεκαεννέα σε μόλις έντεκα.

Επιπρόσθετα, αντί να δημιουργήσουμε δύο ξεχωριστά μπλοκ κρυπτογράφησης για τη λειτουργία του stream cipher και την ακεραιότητα, χρησιμοποιήσαμε δύο από τα υπάρχοντα έντεκα μπλοκ, περιορίζοντας έτσι ακόμη περισσότερο το χώρο που καταλαμβάνει το σύστημα. Αντές οι βελτιστοποίησεις οδήγησαν σε συνολική μείωση της επιφάνειας (αρεα) που καταλαμβάνουν τα μπλοκ κρυπτογράφησης κατά περίπου 48%.

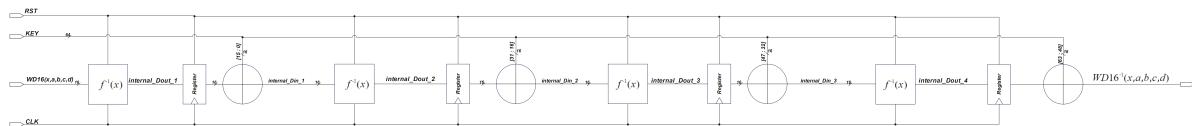
Παρά τα πλεονεκτήματα της κοινής χρήσης των μπλοκ κρυπτογράφησης, ένα μειονέκτημα

του Hummingbird-2 είναι ότι η διαδικασία της αποκρυπτογράφησης δεν μπορεί να εκμεταλλευτεί τα ήδη υπάρχοντα μπλοκ κρυπτογράφησης για τη μείωση του κόστους υλικού. Η αποκρυπτογράφηση απαιτεί τη χρήση αποκλειστικών μπλοκ αποκρυπτογράφησης, γεγονός που εμποδίζει την περαιτέρω μείωση του συνολικού κόστους υλικού.

7.3 Βελτιστοποίηση Σχεδιασμού στην Συχνότητα Ρολογιού

Ο αλγόριθμος Hummingbird-2 σχεδιάστηκε με κύριο στόχο την ελαχιστοποίηση της κατανάλωσης ενέργειας και της επιφάνειας του υλικού. Ωστόσο, παράλληλα με αυτούς τους στόχους, επιδιώξαμε να αυξήσουμε και τη συχνότητα του ρολογιού όσο το δυνατόν περισσότερο, προκειμένου να βελτιώσουμε την απόδοση. Παρόλο που υπάρχει ενδογενής ανταγωνισμός μεταξύ αυτών των παραμέτρων (ενέργεια, επιφάνεια και συχνότητα), καταφέραμε να διατηρήσουμε τις ισορροπίες και να επιτύχουμε ικανοποιητικά αποτελέσματα μέσω της χρήσης τεχνικών pipe-lining όπου κρίθηκε απαραίτητο.

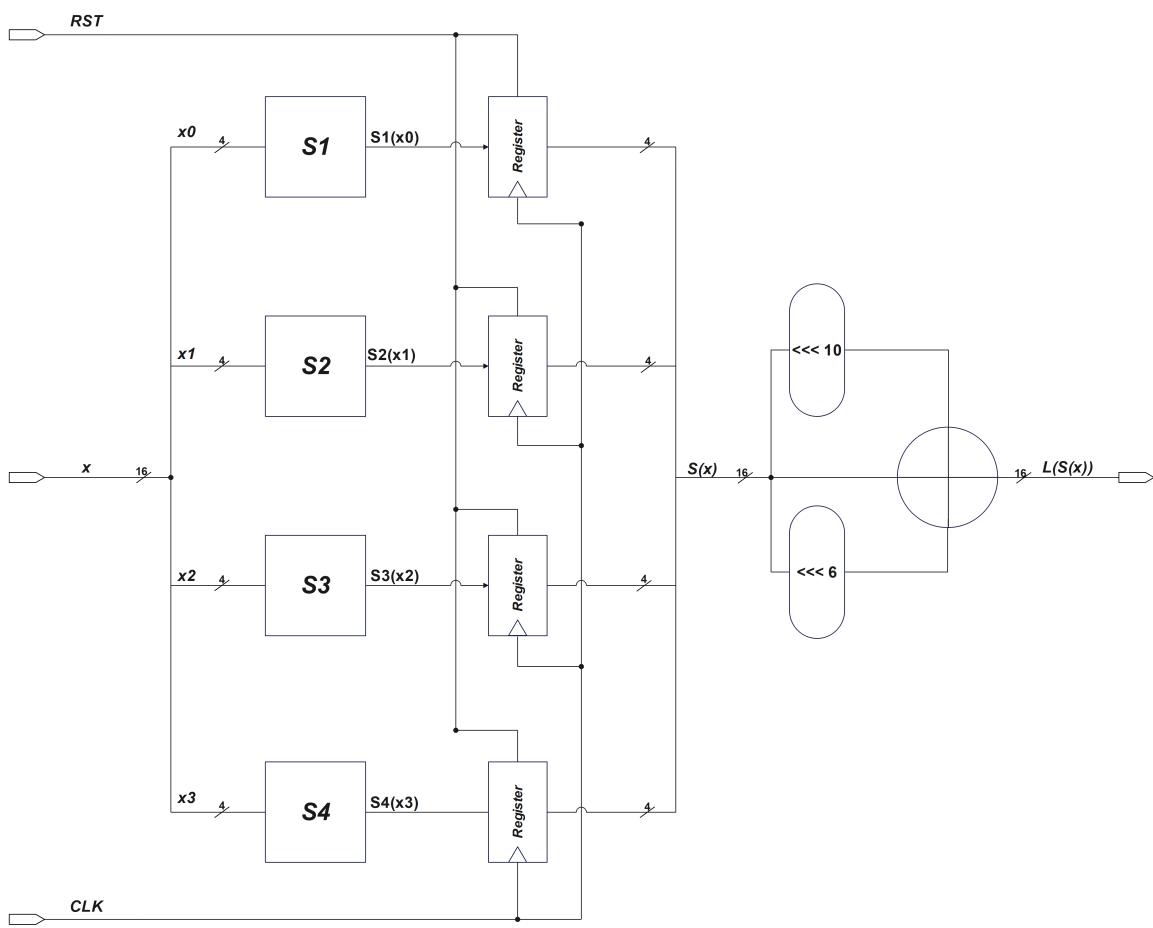
Η χαμηλή αρχική συχνότητα του αλγορίθμου οφειλόταν σε ορισμένες καθυστερήσεις που προκύπτουν από τις μη γραμμικές μετατροπές στα S-boxes, τις πράξεις αρχικοποίησης των καταχωρητών κατάστασης R , καθώς και τις πράξεις στα μπλοκ κρυπτογράφησης και αποκρυπτογράφησης για την ενημέρωση αυτών των καταχωρητών. Για να αντιμετωπίσουμε αυτές τις καθυστερήσεις και να αυξήσουμε τη συχνότητα, εφαρμόστηκαν τεχνικές pipe-lining στην μονάδα $WD16^{-1}$ (εικόνα 7.2), λόγω της πιο σύνθετης διαδικασίας που περιλαμβάνει η συνάρτηση $f^{-1}(x)$ σε σύγκριση με τη $WD16$, που χρησιμοποιεί την απλούστερη συνάρτηση $f(x)$.



Σχήμα 7.2: Αρχιτεκτονική $WD16^{-1}$ με pipe-lining.

Συγκεκριμένα, στη μονάδα $f(x)$ οι έξοδοι των S-boxes αποθηκεύονται προσωρινά σε καταχωρητές 4 bit πριν περάσουν από τη γραμμική συνάρτηση ανάμειξης (εικόνα 7.3). Αυτή η τεχνική εξασφαλίζει ότι τα δεδομένα προωθούνται στην επόμενη φάση επεξεργασίας σε κάθε κύκλο ρολογιού, ενώ ταυτόχρονα εισάγονται νέα δεδομένα για επεξεργασία. Με αυτό τον τρόπο, μειώνονται τα μήκη των κρίσιμων διαδρομών (critical paths), επιτρέποντας την εκτέλεση περισσότερων εντολών ανά κύκλο ρολογιού. Αντίστοιχη τεχνική εφαρμόστηκε και στη $f^{-1}(x)$, με τη διαφορά ότι στην περίπτωση αυτή εφαρμόστηκε pipe-lining τόσο στις γραμμικές αποκωδικοποίησεις όσο και στις εξόδους των S^{-1} -boxes, επιτυγχάνοντας παρόμοια αποτελέσματα με τη μονάδα κρυπτογράφησης (εικόνα 7.4).

Σχετικά με τον υπολογισμό των καταχωρητών κατάστασης R , οι πράξεις προσθέσεων *modulo* 2^{16} , οι κυκλικές μετατοπίσεις και οι πράξεις XOR, που λαμβάνουν χώρα κατά την αρχικοποίηση, διαχωρίστηκαν σε τέσσερα στάδια (εικόνα 7.5) για την αύξηση της συχνότητας λειτουργίας. Στο πρώτο στάδιο πραγματοποιούνται οι προσθέσεις, και τα αποτελέσματα αποθηκεύονται σε καταχωρητές. Στην συνέχεια, τα αποτελέσματα προωθούνται στο δεύτερο στάδιο για τον υπολογισμό των πράξεων *mod* 2^{16} . Ακολουθεί το τρίτο στάδιο, όπου πραγματοποιούνται οι κυκλικές μετατοπίσεις, και τέλος, στο τέταρτο στάδιο, εκτελούνται οι πράξεις XOR και τα δεδομένα

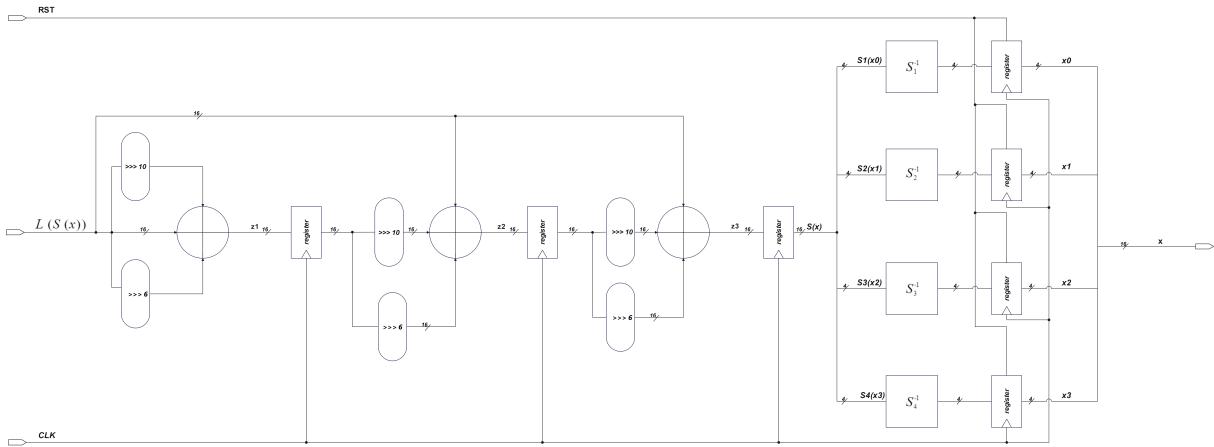


Σχήμα 7.3: Αρχιτεκτονική $f(x)$ με pipe-lining.

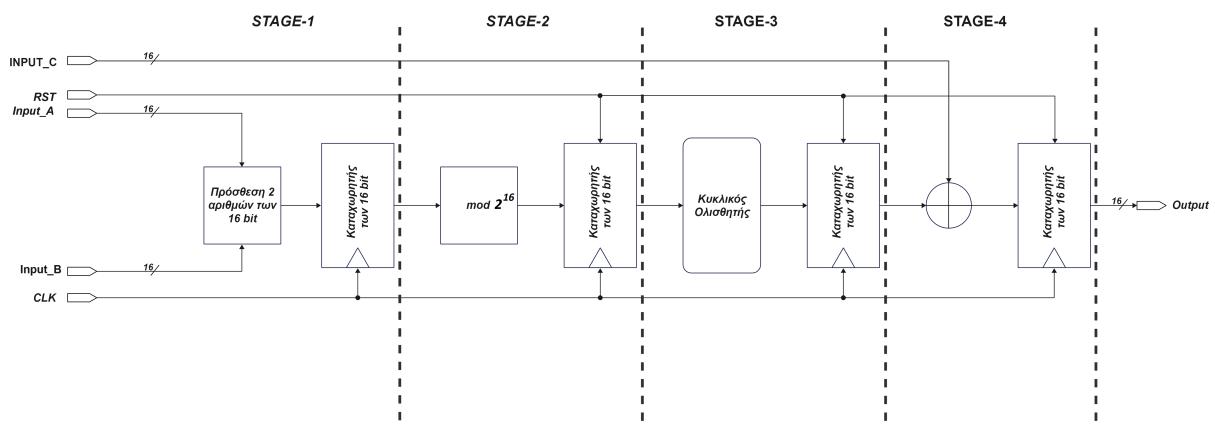
αποθηκεύονται στους αντίστοιχους καταχωρητές.

Για τις πιο απλές πράξεις, όπως οι προσθέσεις $modulo\ 2^{16}$, εφαρμόστηκε η τεχνική με δύο στάδια pipe-lining (εικόνα 7.6). Στην αρχικοποίηση χρησιμοποιήθηκε η τεχνική των τεσσάρων και των δύο σταδίων, ενώ στη μονάδα κρυπτογράφησης και αποκρυπτογράφησης χρησιμοποιήθηκαν δύο στάδια μόνο στα μονοπάτια που επηρεάζουν τους καταχωρητές κατάστασης R .

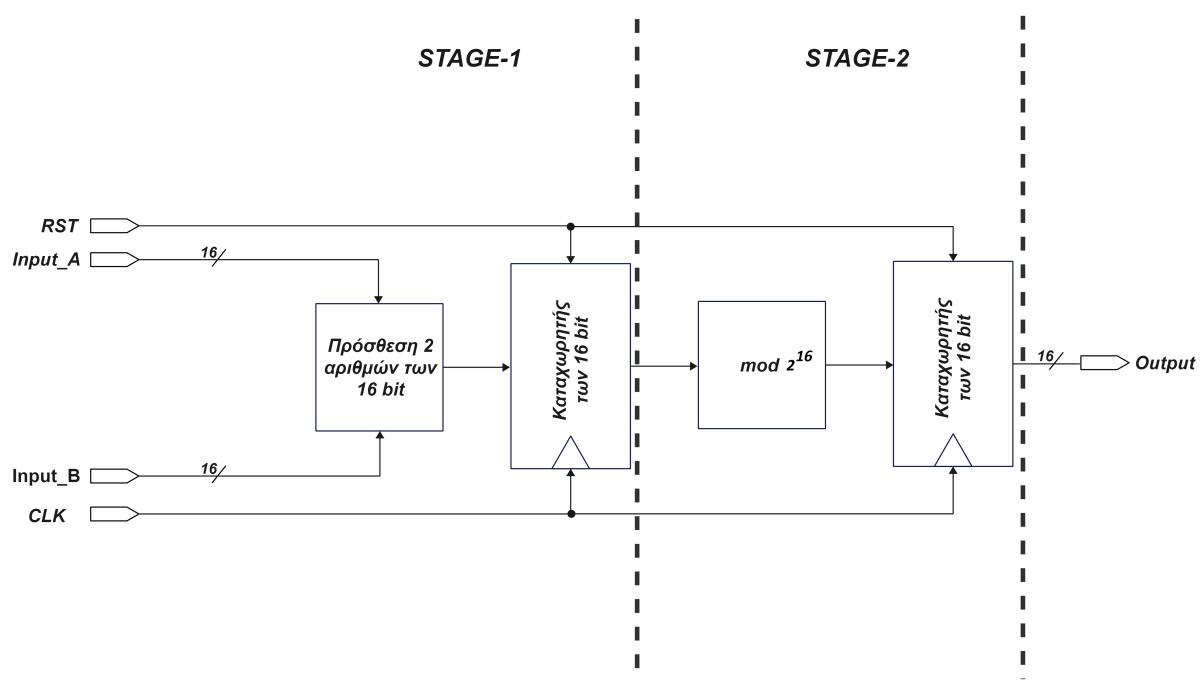
Αυτές οι τεχνικές pipe-lining κατέστησαν δυνατή την αύξηση της συχνότητας λειτουργίας χωρίς να θυσιαστεί η αποδοτικότητα ή να αυξηθεί σημαντικά η επιφάνεια του σχεδιασμού. Συγκεκριμένα, η αρχική συχνότητα του σχεδιασμού, χωρίς την παραπάνω εφαρμογή των προαναφερθεισών τεχνικών, ανερχόταν στα $34,5\ MHz$. Μέσω της βελτίωσης της διαδικασίας, επιτύχαμε την αύξηση της συχνότητας στην τελική τιμή των $243,902\ MHz$, δηλαδή μια αύξηση της τάξεως του 606.96%.



Σχήμα 7.4: Αρχιτεκτονική $f^{-1}(x)$ με pipe-lining.



Σχήμα 7.5: Αρχιτεκτονική pipe-lining τεσσάρων σταδίων.



Σχήμα 7.6: Αρχιτεκτονική pipe-lining δύο σταδίων.

8

Προσομοίωση και Υλοποίηση Προτεινόμενου Σχεδιασμού

8.1 Εξομοίωση Λειτουργίας Σχεδιασμού

Προκειμένου να διασφαλιστεί ότι ο σχεδιασμός μας λειτουργεί ορθά και ικανοποιεί τα καθορισμένα κριτήρια, είναι απαραίτητη η εξομοίωση του μέσω ενός testbench. Το testbench αποτελεί την πιο αξιόπιστη μέθοδο για την επαλήθευση της ορθής λειτουργίας του συστήματος. Για την εξομοίωση του σχεδιασμού χρησιμοποιήθηκε το εργαλείο ModelSim-INTEL FPGA EDITION 10.5b της εταιρείας Intel.

Αρχικά, όλα τα απαραίτητα αρχεία του σχεδιασμού πρέπει να μεταγλωττιστούν (compile), ώστε να καθοριστεί η ιεραρχία και να μετατραπούν σε εκτελέσιμο κώδικα μηχανής. Στη συνέχεια, εκτελούμε τη διαδικασία προσομοίωσης (Simulation) μέσω του ModelSim, η οποία επιτρέπει την παρακολούθηση των καταστάσεων των εισόδων, εξόδων, σημάτων και μεταβλητών του κυκλώματος κατά τη διάρκεια της προσομοίωσης. Για να επιτευχθεί αυτό, στο αρχείο testbench ορίζονται οι τιμές των εισόδων και η περίοδος του ρολογιού, ενώ γίνεται και έλεγχος των εξόδων ως προς την ορθότητά τους.

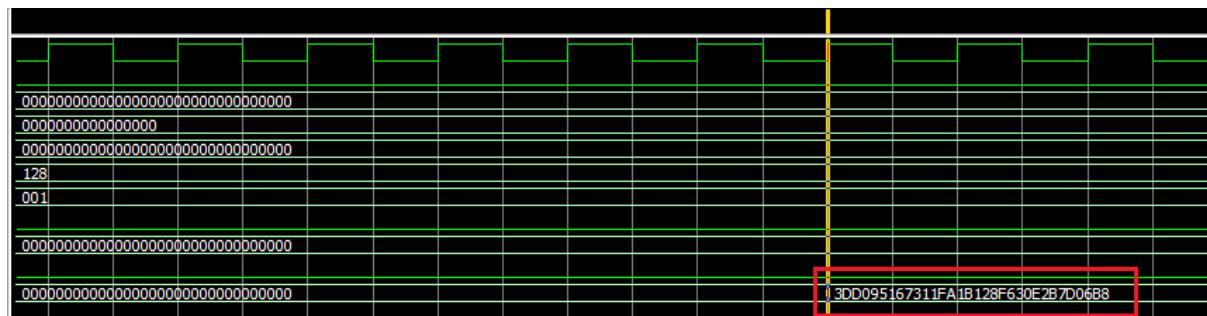
Στον πίνακα 8.1 παρουσιάζονται τα διανύσματα δοκιμής (test vectors), τα οποία χρησιμοποιούνται για την επαλήθευση της ορθής λειτουργίας του συστήματος. Σημειώνεται ότι ο Hummingbird-2 επεξεργάζεται τα δεδομένα σε μορφή little-endian. Αυτό σημαίνει ότι η πρώτη λέξη των 16 bit του απλού κειμένου στο δεύτερο test vector είναι 0x1100.

Αρχικά, ρυθμίζουμε το κύκλωμα μέσω του σήματος *rst*, ώστε να βρίσκεται σε γνωστή κατάσταση. Μετά από μία περίοδο, απενεργοποιούμε το σήμα *rst* και θέτουμε το σήμα *mode* στην τιμή “001”, ώστε να ξεκινήσει η αρχικοποίηση των καταχωρητών *R*. Οι τιμές των εισόδων *text*, *key* και *iv* καθορίζονται σύμφωνα με το πρώτο test vector. Επιπλέον, το σήμα *integrity* ορίζεται στην τιμή ‘0’, καθώς σε αυτή την περίπτωση, που δεν απαιτείται η χρήση του stream cipher, η ακεραιότητα δεν είναι απαραίτητη. Το μήκος του κειμένου *text_len* τίθεται ίσο με 128,

	<i>first test vector</i>	<i>second test vector</i>
<i>Secret key</i>	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10
<i>IV /Nonce</i>	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	12 34 56 78 9A BC DE F0
<i>Plaintext</i>	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF
<i>Ciphertext</i>	C4 EF 87 A8 4F 05 A9 91 57 46 44 81 6E 25 3A CF	5B D1 F8 AD 23 14 20 F4 BA B1 54 C2 45 29 3D 38
<i>MAC</i>	BA ED 40 F0 67 B0 E1 3C 76 F3 59 41 A2 B2 D1 35	C4 F6 74 C0 F6 4B 21 E7 37 24 DC 76 A6 6C 39 19

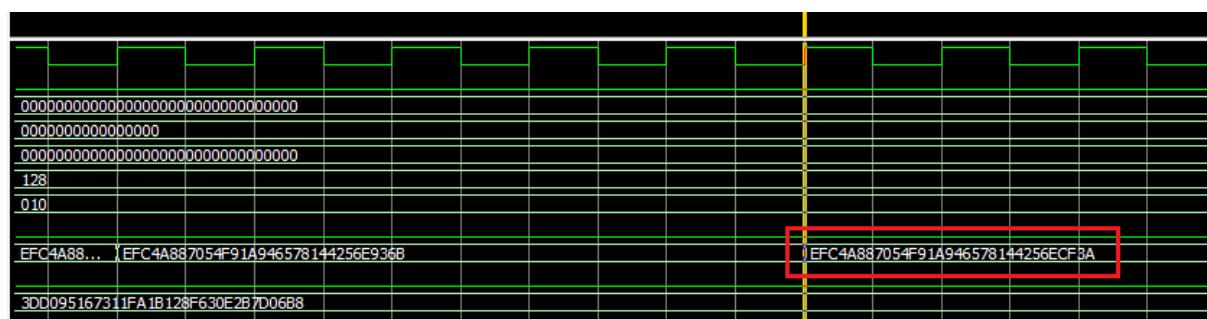
Πίνακας 8.1: Τα test vectors που χρησιμοποιήθηκαν [42].

που είναι και το μήκος του απλού κειμένου. Μετά από 136 κύκλους ρολογιού, οι καταχωρητές R έχουν αρχικοποιηθεί με τις κατάλληλες τιμές (εικόνα 8.1).



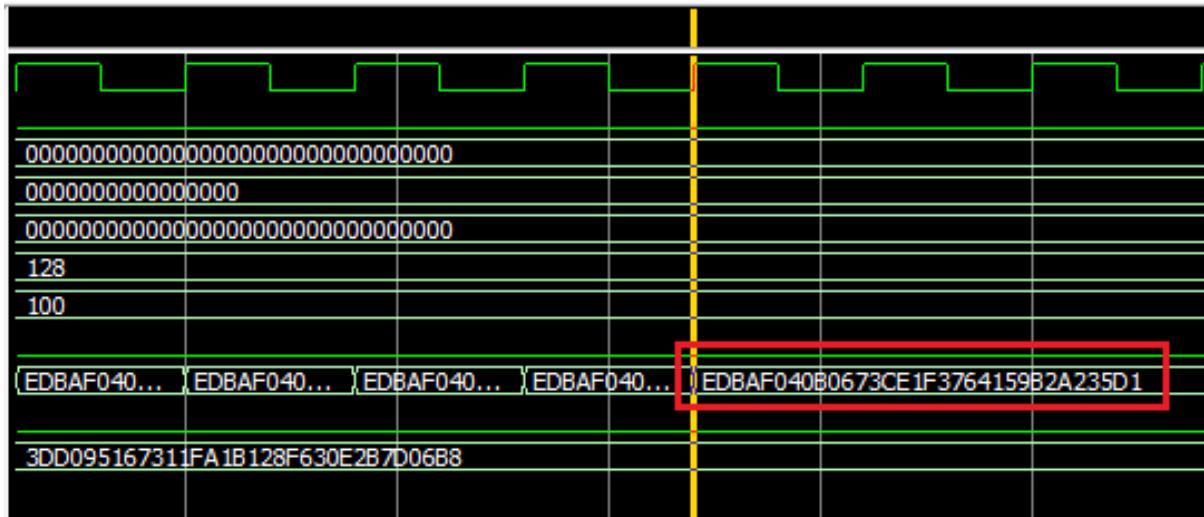
Σχήμα 8.1: Αποτελέσματα αργικοποίησης Hummingbird-2.

Στη συνέχεια το σήμα *mode* τίθεται στην τιμή “010” για να ξεκινήσει η διαδικασία κρυπτογράφησης του plaintext (είσοδος *text*). Μετά από 159 κύκλους ρολογιού, το κείμενο έχει κρυπτογραφηθεί επιτυχώς, όπως φαίνεται και στην εικόνα 8.2, η οποία επιβεβαιώνει το αποτέλεσμα του πίνακα 8.1. Το επόμενο βήμα είναι η παραγωγή του MAC του μηνύματος. Το σήμα *mode* τίθεται στην τιμή “100”, ώστε να ξεκινήσει η διαδικασία αυθεντικοποίησης. Μετά από 214 κύκλους ρολογιού, ο κωδικός αυθεντικοποίησης έχει παραχθεί (εικόνα 8.3) και είναι ταυτόσημος με αυτόν που αναφέρεται στον πίνακα 8.1.

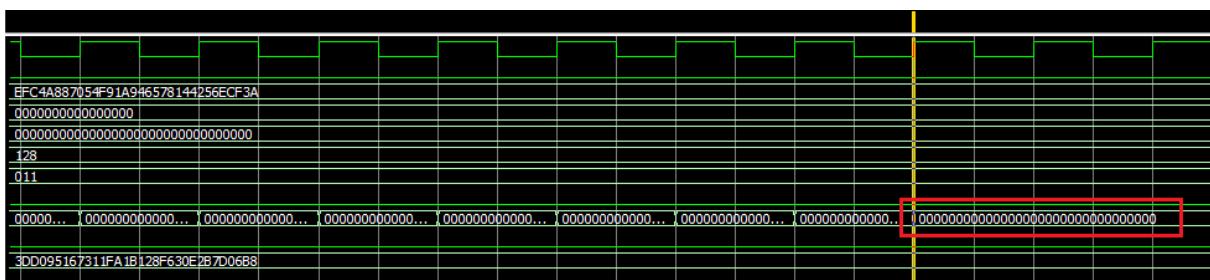


Σχήμα 8.2: Αποτελέσματα κρυπτογράφησης Hummingbird-2.

Μετά την ολοκλήρωση της κρυπτογράφησης, προχωρούμε στην αποκρυπτογράφηση. Η διαδικασία ξεκινά με την αρχικοποίηση των καταχωρητών R , όπως έγινε και στην κρυπτογράφηση, θέτοντας το σήμα *mode* στην τιμή “001”. Όταν ολοκληρωθεί αυτή η φάση, προχωράμε στην αποκρυπτογράφηση θέτοντας το σήμα *mode* στην τιμή “011”, χρησιμοποιώντας αυτή τη φορά το κρυπτογραφημένο κείμενο ως είσοδο. Μετά από 456 κύκλους ρολογιού, το κρυπτοκείμενο έχει αποκρυπτογραφηθεί επιτυχώς (εικόνα 8.4).

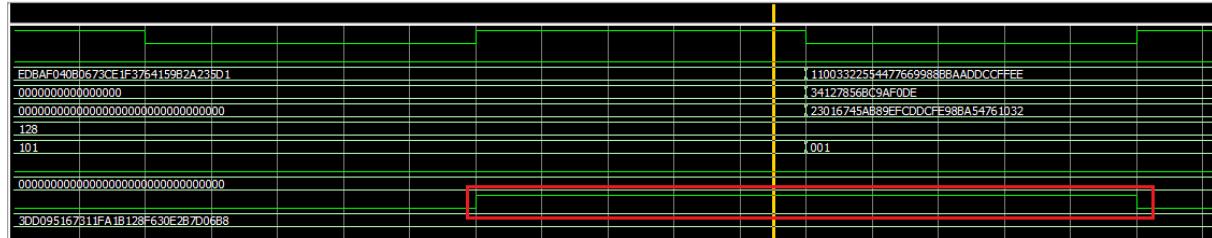


Σχήμα 8.3: Αποτελέσματα αυθεντικοποίησης Hummingbird-2.



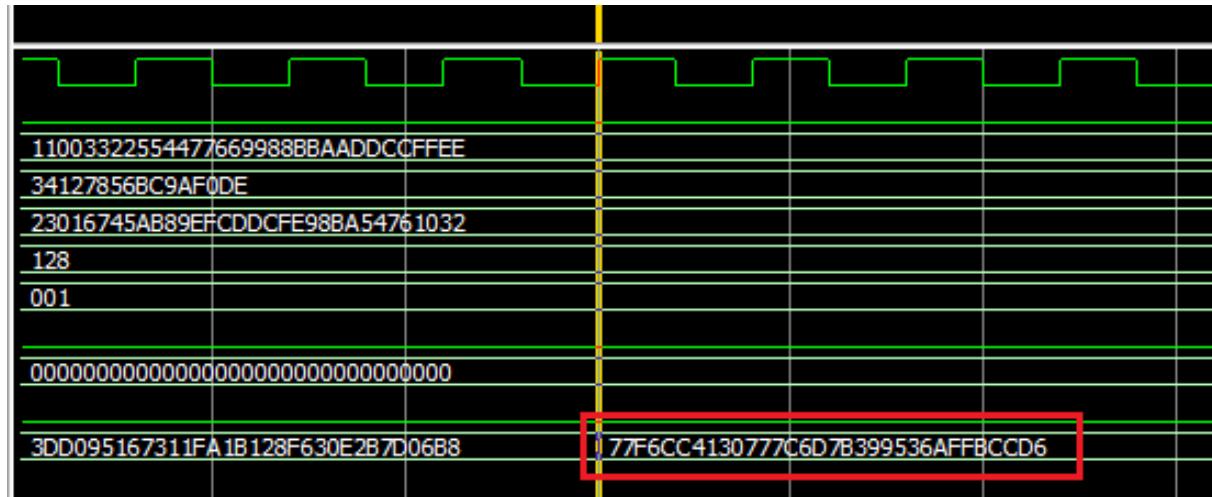
Σχήμα 8.4: Αποτελέσματα αποκρυπτογράφησης Hummingbird-2.

Στη συνέχεια, παράγεται το αντίστοιχο MAC, το οποίο είναι ταυτόσημο με αυτό που δημιουργήθηκε κατά την κρυπτογράφηση. Το τελικό βήμα, αφού ολοκληρωθεί η αποκρυπτογράφηση, είναι η επαλήθευση ότι το MAC του παραλήπτη είναι ταυτόσημο με αυτό του αποστολέα. Το σήμα *mode* τίθεται στην τιμή “101” και η είσοδος *text* λαμβάνει την τιμή του MAC του αποστολέα. Στη συνέχεια, συγκρίνουμε την είσοδο με την έξοδο. Εάν είναι ίσες, το σήμα *mac_flag_equal* γίνεται ‘1’. Σε διαφορετική περίπτωση, παραμένει ‘0’. Όπως φαίνεται και στην εικόνα 8.5, το σήμα ενεργοποιήθηκε μετά από 2 κύκλους ρολογιού, επιβεβαιώνοντας την αυθεντικοποίηση του μηνύματος.

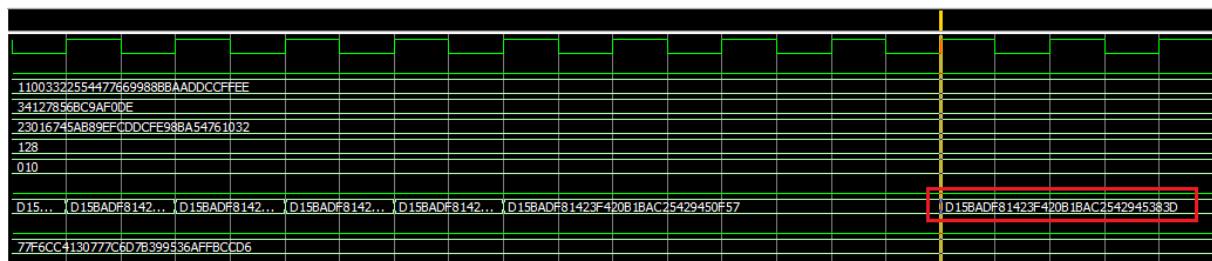


Σχήμα 8.5: Αποτελέσματα ελέγχου αυθεντικοποίησης του μηνύματος Hummingbird-2.

Η ίδια διαδικασία ακολουθείται και για το δεύτερο test vector του πίνακα 8.1. Στις εικόνες 8.6 έως 8.10, αποδεικνύεται η ορθότητα των αποτελεσμάτων.



Σχήμα 8.6: Αποτελέσματα αρχικοποίησης Hummingbird-2 για το δεύτερο test vector.



Σχήμα 8.7: Αποτελέσματα κρυπτογράφησης Hummingbird-2 για το δεύτερο test vector.

Σχήμα 8.8: Αποτελέσματα αυθεντικοποίησης Hummingbird-2 για το δεύτερο test vector.

Σχήμα 8.9: Αποτελέσματα αποκρυπτογράφησης Hummingbird-2 για το δεύτερο test vector.

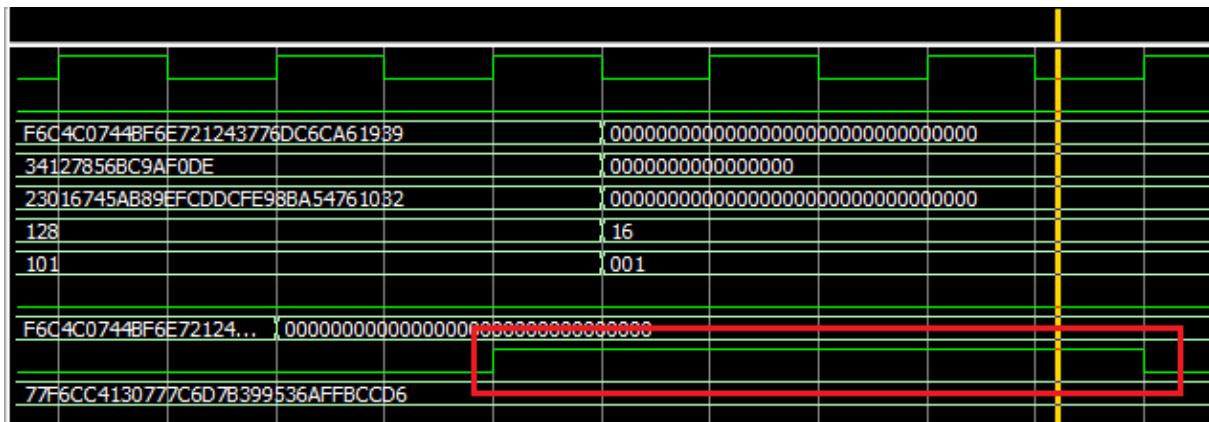
8.2 Υλοποίηση Σχεδιασμού

Μετά την επιβεβαίωση της ορθής λειτουργίας του σχεδιασμού μας, το επόμενο βήμα είναι η υλοποίηση του με λογικές πύλες. Με αυτό εννοούμε ότι, αν και ο σχεδιασμός περιγράφεται σε συμπεριφορικό επίπεδο (behavioral description), το οποίο είναι ταχύτερο και πιο εύκολο για τον αρχικό σχεδιασμό, πρέπει να προχωρήσουμε σε λεπτομερή περιγραφή σε επίπεδο πυλών. Η συμπεριφορική περιγραφή είναι προτιμότερη για την αρχική φάση του σχεδιασμού, καθώς η περιγραφή του κυκλώματος πύλη προς πύλη (gate-level) θα ήταν ιδιαίτερα χρονοβόρα και απαιτητική.

Για να πετύχουμε αυτή τη μετάβαση με γρήγορο και αποτελεσματικό τρόπο, χρησιμοποιούνται εργαλεία σύνθεσης (synthesis tools). Αυτά τα εργαλεία μετατρέπουν το σχεδιασμό από το επίπεδο RTL (register transfer level) σε επίπεδο πυλών (gate level). Η διαδικασία της σύνθεσης βασίζεται σε χρονικούς περιορισμούς και εκτελεί βελτιστοποιήσεις, λαμβάνοντας υπόψη τη χρήση μνήμης και τη επίδοση του κυκλώματος. Μετά τη σύνθεση ακολουθεί η διαδικασία της υλοποίησης (implementation), κατά την οποία το εργαλείο τοποθετεί (place) και δρομολογεί (route) τον σχεδιασμό στις συσκευές της Xilinx. Επιπλέον, παράγεται το bitstream και το device image από τα συνθετοποιημένα netlists.

Για τη διαδικασία σύνθεσης στην παρούσα εργασία, χρησιμοποιήθηκε το εργαλείο Vivado WebPACK Edition 2019.2 της Xilinx, το οποίο παρέχει ένα φιλικό περιβάλλον εργασίας και επιτρέπει την εκτέλεση όλων των παραπάνω διαδικασιών με απλό τρόπο.

Ωστόσο, για να επιτύχουμε τα βέλτιστα αποτελέσματα και να εξασφαλίσουμε ότι ο σχεδιασμός ικανοποιεί τις απαιτούμενες προδιαγραφές, είναι απαραίτητο να οριστεί ένα αρχείο περιορισμών (constraint file), γνωστό ως XDC (Xilinx Design Constraints). Αυτό το αρχείο περιλαμβάνει περιορισμούς που αφορούν την απόδοση, την κατανάλωση ενέργειας, την επιφάνεια που καταλαμβάνει το κύκλωμα, καθώς και άλλες βελτιστοποιήσεις, ώστε να καθοδηγηθεί



Σχήμα 8.10: Αποτελέσματα ελέγχου αυθεντικοποίησης του μηνύματος Hummingbird-2 για το δεύτερο test vector.

το εργαλείο για την επίτευξη καλύτερων αποτελεσμάτων σε διάφορους τομείς.

Αφού ολοκληρώθουν τόσο η σύνθεση όσο και η υλοποίηση, μπορούμε να αναλύσουμε τα αποτελέσματα και να εξετάσουμε τη συνθετοποιημένη σχεδίαση του κυκλώματος. Πιο συγκεκριμένα, το παράθυρο “Reports” του Vivado παρέχει μια λίστα αναφορών για διάφορα ζητήματα, όπως η επιφάνεια που καταλαμβάνει το κύκλωμα, η συχνότητα λειτουργίας, η κατανάλωση ενέργειας κ.α. Σημαντικό είναι να τονιστεί ότι ο τελικός χρονισμός του κυκλώματος πρέπει να λαμβάνεται υπόψη μετά την υλοποίηση, καθώς σε αυτό το στάδιο περιλαμβάνονται και οι πραγματικές καθυστερήσεις λόγω της δρομολόγησης των σημάτων.

Τα αποτελέσματα του σχεδιασμού παρουσιάζονται στον πίνακα 8.2, ο οποίος προέκυψε από τις αναφορές που παρήγθησαν μετά την ολοκλήρωση της διαδικασίας “Implementation”. Συγκεκριμένα, οι μετρήσεις που αφορούν την επιφάνεια και το χρησιμοποιημένο υλικό προέρχονται από την αναφορά “Utilization Report”. Για τις μετρήσεις που σχετίζονται με το χρονισμό και το καθορισμό της μέγιστης συχνότητας λειτουργίας, ορίσαμε έναν περιορισμό ρολογιού στο αρχείο περιορισμών και, μέσω του υπολογισμού του “Worst Negative Slack”(WNS), υπολογίσαμε την τελική συχνότητα λειτουργίας του κυκλώματος ως εξής :

$$f_{max} = \frac{1}{T_{constraint} - WNS}$$

Αυτή η τιμή μπορεί να βρεθεί στην αναφορά χρονισμού (“Timing Report”) του Vivado, η οποία περιέχει επίσης τις μετρήσεις για το κρίσιμο μονοπάτι (critical path).

Αλγόριθμος	LUTs/FFs	Slices	Frequency(MHz)	Device part	Device Family
Hummingbird-2	16857/17623	4887	243,902	xc7a200tffg1156-3	Artix-7

Πίνακας 8.2: Τα αποτελέσματα της υλοποίησης του Hummingbird-2.

8.3 Σύγκριση Σχεδιασμού Hummingbird-2 με Αντίστοιχους Αλγορίθμους

Πριν προχωρήσουμε στην σύγκριση, παραθέσαμε την υλοποίηση ενός ολοκληρωμένου συστήματος για τον Hummingbird-2, το οποίο περιλαμβάνει την πλήρη λειτουργικότητα του με block size 128 bits. Για να διασφαλίσουμε μια ακριβή σύγκριση με άλλους αλγορίθμους, οι οποίοι υλοποιούν κυρίως τις κρυπτογραφικές μονάδες, παρουσιάζουμε τα αποτελέσματα και για την υλοποίηση της κρυπτογράφησης μόνο με block size των 16 bits (πίνακας 8.3). Αυτή η προσέγγιση μας δίνει ένα σωστό μέτρο για τη σύγκριση των επιδόσεων.

Επιπλέον, επιλέξαμε να συγκρίνουμε pipe-line αρχιτεκτονικές, με ορισμένες εξαιρέσεις όπου αυτό δεν ήταν δυνατό. Οι pipe-line αρχιτεκτονικές γενικά προσφέρουν αυξημένο throughput και αποδοτικότητα σε σύγκριση με τις παραδοσιακές αρχιτεκτονικές, επειδή επιτρέπουν τη συνεχή ροή δεδομένων μέσω των σταδίων επεξεργασίας, με ελάχιστη καθυστέρηση.

Για τον υπολογισμό του throughput χρησιμοποιήθηκε η παρακάτω φόρμουλα:

$$\text{Throughput} = \frac{\text{Αριθμός bit ανά έξοδο}}{\text{Κύκλοι ρολογιού για κάθε έξοδο}} \times \text{Συχνότητα ρολογιού (MHz)}$$

Η αποδοτικότητα (efficiency) αποτελεί έναν σημαντικό δείκτη αξιολόγησης για τις υλοποίησεις ψηφιακών σχεδιασμών σε FPGA. Αντικατοπτρίζει το βαθμό στον οποίο οι διαθέσιμοι υλικοί πόροι του συστήματος, όπως τα slices και τα logic elements, αξιοποιούνται για την επίτευξη του επιθυμητού throughput. Ειδικότερα, η αποδοτικότητα ορίζεται ως εξής:

$$\text{Efficiency} = \frac{\text{Throughput (Mbps)}}{\text{Πλήθος των Slices}}$$

Αυτός ο δείκτης υποδεικνύει πόσο αποτελεσματικά μετατρέπονται οι υλικοί πόροι σε πραγματική επεξεργαστική ισχύ. Μια υψηλή τιμή αποδοτικότητας μας δείχνει ότι το σύστημα καταφέρνει να επιτυγχάνει υψηλό throughput με χαμηλή κατανάλωση πόρων, γεγονός που είναι ιδιαίτερα σημαντικό σε εφαρμογές όπου η εξοικονόμιση πόρων είναι καθοριστικός παράγοντας. Αντίθετα, χαμηλή αποδοτικότητα μπορεί να σημαίνει υπερβολική χρήση υλικού για τη δεδομένη απόδοση, κάτι που μπορεί να μην είναι αποδεκτό σε περιβάλλοντα με περιορισμένους πόρους.

Ο χρόνος critical path για την πρώτη έξοδο του Hummingbird-2 με block size 16 bits είναι 19 κύκλοι ρολογιού. Αυτό σημαίνει ότι απαιτούνται 19 κύκλοι για να παραχθεί η πρώτη έξοδος του συστήματος. Ωστόσο, χάρη στη χρήση pipe-line αρχιτεκτονικής, οι επόμενες έξοδοι παράγονται σε 1 κύκλο ρολογιού, γεγονός που επιτρέπει την ταυτόχρονη επεξεργασία πολλαπλών σταδίων και μειώνει τον συνολικό χρόνο επεξεργασίας μετά την αρχική καθυστέρηση.

Για το πλήρες σύστημα Hummingbird-2 με block size 128 bits, ο χρόνος critical path για την παραγωγή της πρώτης εξόδου είναι 159 κύκλοι ρολογιού. Αυτό συμβαίνει λόγω της μεγαλύτερης πολυπλοκότητας και του χρόνου που απαιτείται για να ολοκληρωθεί η επεξεργασία των δεδομένων. Παρόλα αυτά, επειδή υπάρχει εξάρτηση από το R_{input} του επόμενου μπλοκ, το οποίο εξαρτάται από το R_{output} του προηγούμενου μπλοκ, και το R υπολογίζεται κάθε 16 κύκλους ρολογιού, η κάθε επόμενη έξοδος των 128 bits θα είναι έτοιμη σε 16 κύκλους ρολογιού.

Αυτός ο τρόπος επεξεργασίας, με τη χρήση pipeline και την εξάρτηση του R , βελτιστοποιεί τον χρόνο επεξεργασίας, επιτρέποντας τη συνεχή ροή δεδομένων στο σύστημα. Τα αποτελέσματα των παραπάνω φαίνονται στον πίνακα 8.3.

<i>Αλγόριθμος</i>	<i>key size</i>	<i>block size</i>	<i>LUTs/FFs</i>	<i>Slices</i>	<i>Device part</i>	<i>Device Family</i>
HB-2(pipe-line)	128	128	16857/17623	4887	xc7a200tffg1156-3	Artix-7
HB-2(encryption/pipe-line)	128	16	612/485	219	xc7a200tffg1156-3	Artix-7
Phelix [46]	256	-	-	1402	xc3s200ft256-5	Spartan-3
Iceberg (full pipe-line) [47]	128	64	-	6808	-	Virtex-2
Trivium (basic) [48]	80	-	200/257	71	XC7S50FGGA484-1	Spartan-7
Grain-128 [49]	128	-	-	50	-	Spartan-3
PRESENT-80 (pipe-line) [50]	80	64	170/153	48	xc6slx16-3csg324	Spartan-6
PRESENT-128 (pipe-line) [50]	128	64	220/201	61	xc6slx16-3csg324	Spartan-6
KATAN-32 (unrolled) [51]	80	32	-	-	xc6vlx240t-2ff1156	Virtex-6
KATAN-64 (unrolled) [51]	80	64	-	-	xc6vlx240t-2ff1156	Virtex-6
AES-128 (full pipe-line) [52]	128	128	9481/6787	2794	-	Artix-7
SIMECK 32/64 (two-stage pipe-line) [53]	64	32	164/89	50	-	Artix-7

Πίνακας 8.3: Σύγκριση αποτελεσμάτων στο κόστος υλικού της υλοποίησης του Hummingbird-2 (HB-2) της εργασίας με άλλες υλοποίησεις κρυπταλγόριθμων.

<i>Αλγόριθμος</i>	<i>Frequency(MHz)</i>	<i>Throughput(Mbps)</i>	<i>Efficiency</i>
HB-2(pipe-line)	243,902	1.951,21	0,3
HB-2(encryption/pipe-line)	317,46	5.079,36	23,19
Phelix [46]	46	1472	1,05
Iceberg (full pipe-line) [47]	297	19008	2,79
Trivium (basic) [48]	416	416	5,86
Grain-128 [49]	196	196	3,92
PRESENT-80 (pipe-line) [50]	257,4	123,86	2,58
PRESENT-128 (pipe-line) [50]	210,66	99,13	1,62
KATAN-32 (unrolled) [51]	24,27	720	-
KATAN-64 (unrolled) [51]	21,14	1260	-
AES-128 (full pipe-line) [52]	219	28000	10
SIMECK 32/64 (two-stage pipe-line) [53]	388,52	24,39	0,48

Πίνακας 8.4: Σύγκριση αποτελεσμάτων στην συχνότητα/throughput/efficiency της υλοποίησης του Hummingbird-2 (HB-2) της εργασίας με άλλες υλοποίησεις κρυπταλγόριθμων.

Όπως βλέπουμε από τις συγκρίσεις, η υλοποίηση του Hummingbird-2 (encryption pipe-line) επιτυγχάνει αρκετά μεγάλο throughput και αποδοτικότητα, καθιστώντας τον έναν από τους πιο αποδοτικούς στην κατηγορία του. Η χρήση rpipe-line και η χαμηλή κατανάλωση πόρων συμβάλλουν σε αυτή την αποδοτικότητα, η οποία είναι αξιοσημείωτη σε σύγκριση με άλλους αλγορίθμους.

Για παράδειγμα, ο Phelix, λειτουργεί σε χαμηλότερη συχνότητα και έχει χαμηλότερο throughput και efficiency, γεγονός που δείχνει ότι δεν εκμεταλλεύεται τους πόρους τόσο αποδοτικά όσο ο Hummingbird-2.

Ο Iceberg, με λίγο χαμηλότερη συχνότητα, καταφέρνει μεν μεγαλύτερο throughput αλλά χρησιμοποιεί πολύ περισσότερους πόρους με αποτέλεσμα να έχει πολύ χαμηλότερο efficiency. Ο Trivium, παρόλο που λειτουργεί σε υψηλότερη συχνότητα, επιτυγχάνει χαμηλότερο throughput.

Οι αλγόριθμοι PRESENT-80 και PRESENT-128 λειτουργούν σε υψηλές συχνότητες, αλλά το throughput και το efficiency τους παραμένουν χαμηλότερα.

Τέλος, ο AES-128 πετυχαίνει πολύ υψηλό throughput, αλλά χρησιμοποιεί πολλούς πόρους και έχει χαμηλότερο efficiency.

Συμπερασματικά, ο Hummingbird-2 ξεχωρίζει κυρίως λόγω της υψηλής αποδοτικότητας του, ειδικά στην υλοποίηση pipe-line. Παρόλο που το throughput του δεν είναι το υψηλότερο, η πολύ χαμηλή κατανάλωση πόρων τον καθιστά μία από τις πιο αποδοτικές λύσεις για εφαρμογές με περιορισμένους πόρους.

8.4 Σύγκριση Αποτελεσμάτων με Άλλες Υλοποιήσεις της Οικογένειας Hummingbird

Στον πίνακα 8.5 γίνεται μια σύγκριση διάφορων υλοποιήσεων της οικογένειας Hummingbird. Δυστυχώς, δεν εντοπίστηκε καμία υλοποίηση του Hummingbird-2 σε κάποιο επιστημονικό άρθρο που να αναφέρεται σε άλλες αρχιτεκτονικές υλοποιήσης σε υλικό. Για να έχουμε κάποιο μέτρο σύγκρισης, συμπεριλάβαμε διάφορες υλοποιήσεις του Hummingbird-1.

Πρέπει να επισημανθεί ότι και εδώ οι υλοποιήσεις που μελετήθηκαν για σύγκριση επικεντρώνονται κυρίως στην υλοποίηση της μονάδας κρυπτογράφησης, χωρίς να αναφέρονται σε πλήρεις σχεδιασμούς όπως αυτός της παρούσας διπλωματικής. Αυτό οδηγεί σε διαφοροποιήσεις στα αποτελέσματα, καθώς δεν βρίσκονται στην ίδια κλίμακα σύγκρισης.

Παρόλα αυτά, για λόγους πληρότητας, σε αυτή την εργασία παραθέτουμε τα αποτελέσματα τόσο της υλοποίησης ολόκληρου του σχεδιασμού του Hummingbird-2 με και χωρίς τεχνικές pipe-line, όσο και των υλοποιήσεων που αφορούν μόνο την κρυπτογράφηση (μπλοκ των 16 bit), πάλι με και χωρίς την αρχιτεκτονική pipe-line. Τέλος, παρουσιάζονται και τα αποτελέσματα του Hummingbird-1 σε διαφορετικές αρχιτεκτονικές υλοποίησης.

Αλγόριθμος	key size	block size	LUTs/FFs	Slices	Frequency(MHz)	Device part	Device Family
HB-2(pipe-line αρχιτεκτονική)	128	128	16857/17623	4887	243,902	xc7a200tffg1156-3	Artix-7
HB-2(χωρίς αρχιτεκτονική pipe-line)	128	128	17871/5814	5066	34,5	xc7a200tffg1156-3	Artix-7
HB-2(encryption/pipe-line)	128	16	612/485	219	317,46	xc7a200tffg1156-3	Artix-7
HB-2(encryption/ χωρίς pipe-line)	128	16	704/64	223	137	xc7a200tffg1156-3	Artix-7
HB-1(encryption/coprocessor approach) [54]	256	16	80/80	40	260,8	XC3S200-5	Spartan-3
HB-1(encryption/High Speed) [55]	256	16	3504/4242	4242	152,905	xc5v1x20t-2-ff-323	Virtex-5
HB-1(encryption) [56]	256	16	473/120	273	40,1	xc3s200-5ft256	Spartan-3

Πίνακας 8.5: Σύγκριση αποτελεσμάτων της υλοποίησης του Hummingbird-2 (HB-2) της εργασίας με άλλες υλοποιήσεις της οικογένειας Hummingbird.

Από τα δεδομένα του πίνακα, παρατηρούμε ότι οι υλοποιήσεις του Hummingbird-2, ειδικά με την αρχιτεκτονική pipe-line, εμφανίζουν σημαντικά καλύτερες επιδόσεις σε σχέση με τον Hummingbird-1, τόσο σε ταχύτητα όσο και σε αποδοτικότητα. Η χρήση τεχνικών pipe-line στον Hummingbird-2 αυξάνει την συχνότητα λειτουργίας και μειώνει τη χρήση πόρων σε μονάδες κρυπτογράφησης, ενώ διατηρεί την απαιτούμενη ασφάλεια. Αντίθετα, ο Hummingbird-1, παρά την αποδοτικότητά του σε εφαρμογές χαμηλών πόρων, έχει αποδειχθεί ευάλωτος σε επιθέσεις με διαφορική ανάλυση και επιλεγμένου IV. Οι αδυναμίες αυτές έχουν αντιμετωπιστεί στον σχεδιασμό του Hummingbird-2, γεγονός που τον καθιστά πιο ασφαλή, αλλά αναπόφευκτα αυξάνει τις απαιτήσεις σε επιφάνεια και σε κάποιες περιπτώσεις μειώνει την αποδοτικότητα. Ωστόσο, ο συνολικός σχεδιασμός του Hummingbird-2 επιτυγχάνει μια καλύτερη ισορροπία μεταξύ ασφάλειας και απόδοσης, καθιστώντας τον προτιμότερο για περιβάλλοντα με περιορισμένους πόρους που απαιτούν ικανοποιητική ταχύτητα και αντοχή σε επιθέσεις.

9

Συμπεράσματα και Μελλοντικές Βελτιστοποιήσεις

9.1 Σύνοψη

Με γνώμονα την ταχεία εξέλιξη της τεχνολογίας στον τομέα του Internet of Things (IoT), είναι αναμενόμενο ότι στο άμεσο μέλλον οι IoT συσκευές θα αποτελούν αναπόσπαστο κομμάτι της καθημερινότητας του ανθρώπου. Πολλές καθημερινές δραστηριότητες θα εκτελούνται απλώς με το πάτημα ενός κουμπιού, με τη λειτουργία να ανατίθεται σε μικρές και αποδοτικές συσκευές. Ωστόσο, η επικοινωνία αυτών των συστημάτων, καθώς και οι περιορισμοί σε ενέργεια και πόρους, καθιστούν δύσκολη την υιοθέτηση ισχυρών μηχανισμών ασφαλείας. Για τον λόγο αυτό, απαιτούνται νέα, εξελιγμένα πρωτόκολλα ασφαλείας, τα οποία προσαρμόζονται στις συγκεκριμένες προκλήσεις χωρίς να υποβαθμίζουν την ασφάλεια των συστημάτων.

Μέσω υλοποιήσεων σε υλικό (hardware), μπορούμε να εξασφαλίσουμε την επιθυμητή ασφάλεια για τα συστήματα IoT. Ένα από τα κατάλληλα μέσα για την επίτευξη αυτών των στόχων είναι τα ενσωματωμένα συστήματα (embedded systems), τα οποία είναι κατάλληλα για εφαρμογές IoT, καθώς χαρακτηρίζονται από ευελιξία, προγραμματιζόμενη λειτουργία και μικρό μέγεθος.

Στα πλαίσια της παρούσας διπλωματικής εργασίας, παρουσιάζεται η υλοποίηση ενός ολοκληρωμένου συστήματος που βασίζεται στον ultra-lightweight κρυπτογραφικό αλγόριθμο Hummingbird-2, ο οποίος είναι κατάλληλος για συσκευές με περιορισμένους πόρους, όπως οι RFID συσκευές. Η σχεδιαστική προσέγγιση που ακολουθήθηκε εξασφαλίζει μια καλή ισορροπία ανάμεσα στη μέγιστη συχνότητα λειτουργίας και τη μειωμένη κατανάλωση υλικού. Μετά από σύγκριση με άλλους κρυπτογραφικούς αλγορίθμους, είναι εμφανές ότι ο Hummingbird-2 επιτυγχάνει μια αξιόλογη ισορροπία μεταξύ κόστος υλικού, ταχύτητας και ασφάλειας.

9.2 Ευπάθειες του Hummingbird-2

Ο αλγόριθμος Hummingbird-2 σχεδιάστηκε για να αντιμετωπίσει τις επιθέσεις που υπήρχαν στον προκάτοχό του, Hummingbird-1, ωστόσο παραμένει ευάλωτος σε μια επίθεση γνωστή ως κρυπτανάλυση σχετικού κλειδιού (related-key cryptanalysis) [44]. Αυτή η επίθεση στοχεύει στην ανάκτηση δύο 128 bit μυστικών κλειδιών τα οποία παρουσιάζουν μικρή διαφορά $\delta = K_i \oplus K'_i$ στα bit τους. Τα κλειδιά αυτά, γνωστά ως «υψηλά συσχετισμένα κλειδιά» (highly correlated keys), έχουν μεγάλη πιθανότητα να παράγουν το ίδιο ciphertext, διευκολύνοντας έτσι την εκτέλεση της επίθεσης. Η πολυπλοκότητα της κύριας επίθεσης, η οποία βασίζεται σε επιλεγμένο IV, είναι της τάξης των 2^{64} πράξεων, ενώ τα πρώτα 64 bits του κλειδιού μπορούν να ανακτηθούν ανεξάρτητα με πολυπλοκότητα 2^{36} πράξεων [43].

Ο Hummingbird-2 συνδυάζει χαρακτηριστικά από block ciphers και stream ciphers, στηριζόμενος σε μια 16 bit αρχιτεκτονική με τη χρήση της *WD16*. Η αρχιτεκτονική αυτή περιλαμβάνει τέσσερις γύρους αντικατάστασης και αντιμετάθεσης, με την προσθήκη κλειδιού, S-boxes και μια γραμμική συνάρτηση ανάμειξης σε κάθε γύρο. Αυτός ο σχεδιασμός, παρότι επιτρέπει την ύπαρξη κλειδιών με παρόμοιες ιδιότητες, καθιστά τον αλγόριθμο ευάλωτο στην κρυπτανάλυση σχετικού κλειδιού [44].

Η επίθεση εκμεταλλεύεται τη δυνατότητα δύο συσχετισμένων κλειδιών, τα οποία διαφέρουν σε συγκεκριμένες θέσεις, να παράγουν το ίδιο ciphertext με πιθανότητα $\frac{1}{4}$. Ο επιτιθέμενος μπορεί να χρησιμοποιήσει αυτή την ιδιότητα, επιλέγοντας κατάλληλα *IVs*, για να συγκρίνει τα αποτελέσματα κρυπτογράφησης και αποκρυπτογράφησης και να βρει τη διαφορά των κλειδιών. Αρχικά, ο επιτιθέμενος αναζητά μια τιμή *IV* που προκαλεί σύγκρουση στην κατάσταση (state collision) του αλγορίθμου μετά από τέσσερις γύρους αρχικοποίησης, κάτι που επιτυγχάνεται με περίπου 2^{16} διαφορετικές τιμές *IV* [44]. Μετά την εύρεση της σύγκρουσης, χρησιμοποιεί αυτή την πληροφορία για να ανακτήσει τα πρώτα 64 bit του κλειδιού μέσω εξαντλητικής αναζήτησης. Αφού βρεθεί το πρώτο μέρος του κλειδιού, ο επιτιθέμενος μπορεί να επιτεθεί στο υπόλοιπο μέρος του κλειδιού μέσω μιας απλής μεθόδου εξάντλησης, ολοκληρώνοντας την επίθεση.

Η επίθεση καθίσταται δυνατή λόγω της απλότητας του key schedule και της αλγεβρικής δομής του Hummingbird-2. Η σχεδίαση βασίζεται σε μια 16 bit αρχιτεκτονική και στην προσθετική ανάμειξη, χαρακτηριστικά που τον καθιστούν ευάλωτο σε σχετικές επιθέσεις κλειδιών [44]. Παρόλο που αυτός ο σχεδιασμός απλοποιεί την υλοποίηση σε περιβάλλοντα με περιορισμένους πόρους, όπως τα συστήματα RFID, επιτρέπει την εκμετάλλευση των προαναφερθέντων αδυναμιών.

9.3 Μελλοντική Εργασία

Στο προηγούμενο κεφάλαιο παρουσιάστηκαν διάφορες υλοποιήσεις κρυπτογραφικών αλγορίθμων από τη διεθνή βιβλιογραφία, με στόχο την ασφάλεια συσκευών περιορισμένων σε πόρους, όπως τα ενσωματωμένα συστήματα και οι IoT εφαρμογές. Αυτές οι υλοποιήσεις επικεντρώνονται στη διασφάλιση της εμπιστευτικότητας και της ακεραιότητας των δεδομένων, ενώ ταυτόχρονα λαμβάνουν υπόψη τους περιορισμούς σε κατανάλωση ενέργειας και μέγεθος υλικού. Παρόλο που οι υπάρχουσες υλοποιήσεις του Hummingbird-2 παρουσιάζουν σημαντικά πλεονεκτήματα, υπάρχουν περιθώρια για περαιτέρω βελτιώσεις που θα μπορούσαν να βελτιώσουν τόσο την απόδοση όσο και την ασφάλεια του συστήματος. Για την περαιτέρω μελέτη και

την ανάπτυξη του προτεινόμενο συστήματος, προτείνονται δύο προτάσεις.

Η πρώτη πρόταση εστιάζει στη μείωση της επιφάνειας του κυκλώματος και της κατανάλωσης ενέργειας. Σύμφωνα με τη σχετική βιβλιογραφία [57], η χρήση μιας αρχιτεκτονικής βασισμένης σε επαναληπτικούς γύρους (round-based architecture) θα μπορούσε να μειώσει σημαντικά την πολυπλοκότητα και την κατανάλωση πόρων του συστήματος. Συγκεκριμένα, προτείνεται η αντικατάσταση των τεσσάρων παράλληλων συναρτήσεων $f(x)$ με μία μόνο συνάρτηση, η οποία θα εκτελείται επαναληπτικά για τον καθορισμένο αριθμό γύρων. Αυτή η προσέγγιση μειώνει το κόστος σε υλικό και βελτιώνει την ενεργειακή αποδοτικότητα του συστήματος, καθιστώντας το πιο κατάλληλο για εφαρμογές με αυστηρούς περιορισμούς σε πόρους, όπως τα συστήματα RFID.

Η δεύτερη πρόταση αφορά τη βελτίωση της ασφάλειας του αλγορίθμου, και συγκεκριμένα την προστασία από την επίθεση σχετικού κλειδιού που αναλύθηκε στο [44]. Η επίθεση αυτή αποτελεί σημαντική απειλή για την ακεραιότητα του Hummingbird-2, καθώς εκμεταλλεύεται τη δομή του αλγορίθμου για ανακτήσει τα μυστικά κλειδιά. Για την αντιμετώπιση αυτής της αδυναμίας, προτείνεται η αντικατάσταση της υπάρχουσας μονάδας $WD16(x, k_1, k_2, k_3, k_4)$ με τη συνάρτηση “S-boxless” $\chi_\nu(x, k_1, k_2, k_3, k_4)$. Η συνάρτηση αυτή όχι μόνο εξαλείφει την ευπάθεια σε επιθέσεις σχετικού κλειδιού, αλλά επίσης μειώνει τις απαιτήσεις σε υλικό, καθιστώντας το σύστημα πιο αποδοτικό. Αυτή η αλλαγή στη δομή του αλγορίθμου θα βελτιώσει την ανθεκτικότητα του συστήματος σε επιθέσεις και θα ενισχύσει την ασφάλεια του ευρύ φάσμα εφαρμογών.

Συμπερασματικά, οι προτεινόμενες βελτιώσεις αναμένεται να συμβάλουν ουσιαστικά στην περαιτέρω ανάπτυξη του Hummingbird-2, προσφέροντας έναν πιο αποδοτικό και ασφαλή αλγόριθμο για εφαρμογές σε περιβάλλοντα με περιορισμένους πόρους.

Βιβλιογραφία

- [1] K. Καλοβρέκτης, *Βασικές Δομές Ενσωματωμένων Συστημάτων*. Βαρβαρήγου Εκδόσεις, 2021, pp. 639–640, ISBN: 978-960-7996-80-0.
- [2] M. Weiser, “The Computer for the 21st Century”, *Scientific American*, vol. 265, no. 3, pp. 94–104, 1991.
- [3] B. B. Gupta and M. Quamara, “An overview of Internet of Things (IoT): Architectural aspects, challenges, and protocols”, *Concurrency and Computation: Practice and Experience*, vol. 32, no. 21, e4946, 2020. DOI: 10.1002/cpe.4946.
- [4] K. Ashton, “That ‘Internet of Things’ Thing”, *RFID Journal*, 2011.
- [5] P. Tadejko, “Application of Internet of Things in Logistics - Current Challenges”, *Economics and Management*, vol. 7, no. 4, pp. 54–64, 2015. DOI: 10.12846/J.EM.2015.04.07.
- [6] N. N. Thilakarathne and R. Samarasignhe, “IoT Security: Overview, Challenges and Countermeasures”, in *Annual Research Symposium 2022, Faculty of Technology, University of Colombo*, Colombo, Sri Lanka, 2022.
- [7] N. M. Karie, N. M. Sahri, W. Yang, C. Valli and V. R. Kebande, “A Review of Security Standards and Frameworks for IoT-Based Smart Environments”, *IEEE Access*, vol. 9, pp. 121 975–121 995, 2021. DOI: 10.1109/ACCESS.2021.3109886.
- [8] N. Sklavos, I. D. Zaharakis, A. Kameas and A. Kalapodi, “Security & Trusted Devices in the Context of Internet of Things (IoT)”, in *2017 Euromicro Conference on Digital System Design (DSD)*, 30 Aug.-1 Sept. 2017, Vienna, Austria, IEEE, 2017, ISBN: 978-1-5386-2146-2. DOI: 10.1109/DSD.2017.81. Available at: <https://doi.org/10.1109/DSD.2017.81>.
- [9] S. Zeadally, A. K. Das and N. Sklavos, “Cryptographic technologies and protocol standards for Internet of Things”, *Internet of Things*, vol. 14, p. 100 075, 2021. DOI: 10.1016/j.iot.2019.100075.
- [10] A. B. Pawar and S. Ghumre, “A Survey on IoT Applications, Security Challenges and Counter Measures”, in *Proceedings of the 2016 International Conference on Computing, Analytics and Security Trends (CAST)*, Pune, India, 2016, pp. 1–6. DOI: 10.1109/CAST.2016.7914975.
- [11] N. Tewari and G. Datt, “A Systematic Review of Security Issues and Challenges with Futuristic Wearable Internet of Things (IoTs)”, in *Proceedings of the 2021 International Conference on Technological Advancements and Innovations (ICTAI)*, Bimtal, India, 2021, pp. 1–6. DOI: 10.1109/ICTAI53825.2021.9637433.

- [12] Σ. Κάτσικας, Σ. Γκιρτζάλης and K. Λαμπρινούδάκης, *Εφαρμοσμένη Κρυπτογραφία*, 1η. Αθήνα: NewTech Pub, 2021, pp. 232–275.
- [13] W. A. Kotas, “A Brief History of Cryptography”, University of Tennessee, 2000. Available at: https://trace.tennessee.edu/utk%5C_chanhonoproj/398.
- [14] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal and B. Sikdar, “A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures”, *IEEE Access*, vol. 7, pp. 82 721–82 743, 2019. DOI: 10.1109/ACCESS.2019.2924045.
- [15] J. Saleem, M. Hammoudeh, U. Raza, B. Adebisi and R. Ande, “IoT Standardisation - Challenges, Perspectives and Solution”, in *Proceedings of the 2018 International Conference on Future Networks and Distributed Systems (ICFNDS)*, Amman, Jordan, 2018. DOI: 10.1145/3231053.3231103.
- [16] M. Burmester, Σ. Γκιρτζάλης, Σ. Κάτσικας and B. Χρυσικόπουλος, *Σύγχρονη Κρυπτογραφία: Θεωρία και Εφαρμογές*, Έκδοση: 1η. Αθήνα: Παπασωτηρίου, 2011, ISBN: 9789607182760.
- [17] Q. Zhang, “An Overview and Analysis of Hybrid Encryption: The Combination of Symmetric Encryption and Asymmetric Encryption”, in *2021 2nd International Conference on Computing and Data Science (CDS)*, Stanford, CA, USA: IEEE, 2021, pp. 616–622. DOI: 10.1109/CDS52072.2021.00111.
- [18] S. Zeadally, A. K. Das and N. Sklavos, “Cryptographic Technologies and Protocol Standards for Internet of Things”, *Internet of Things*, vol. 14, 2021. DOI: 10.1016/j.iot.2019.100075. Available at: <https://www.sciencedirect.com/science/article/pii/S2542660519301799>.
- [19] A. B. Pawar and S. Ghambre, “A Survey on IoT Applications, Security Challenges and Counter Measures”, in *2016 International Conference on Computing, Analytics and Security Trends (CAST)*, CAST, Pune, India: College of Engineering Pune, 2016, pp. 294–299.
- [20] A. Balte, A. Kashid and B. Patil, “Security Issues in Internet of Things (IoT): A Survey”, *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 5, no. 4, pp. 450–456, 2015, ISSN: 2277-128X. Available at: <http://www.ijarcse.com>.
- [21] S. Mishra, S. Sahoo and B. Mishra, “Addressing Security Issues and Standards in Internet of Things”, in IGI Global, 2020, chap. 10, pp. 224–257. DOI: 10.4018/978-1-5225-5793-7.ch010. Available at: <https://www.researchgate.net/publication/344348551>.
- [22] W. Stallings, *Cryptography and Network Security: Principles and Practice*, 6th. Upper Saddle River, NJ, USA: Pearson Education, 2014, ISBN: 978-0-13-335469-0.
- [23] E. Dawson and L. Nielsen, “Automated Cryptanalysis of XOR Plaintext Strings”, *Cryptologia*, vol. 20, no. 2, pp. 165–181, 1996. DOI: 10.1080/0161119960898470.
- [24] L. R. Knudsen, “Block Ciphers --- A Survey”, *COSIC'97 Course, LNCS 1528*, pp. 18–48, 1998.
- [25] G. Hatzivasilis, K. Fysarakis, I. Papaefstathiou and C. Manifavas, “A review of lightweight block ciphers”, *Journal of Cryptographic Engineering*, vol. 8, pp. 141–184, 2018. DOI: 10.1007/s13389-017-0160-y. Available at: <https://link.springer.com/article/10.1007/s13389-017-0160-y>.

- [26] Y. Dodis, J. Katz, J. Steinberger, A. Thiruvengadam and Z. Zhang, *Provable Security of Substitution-Permutation Networks*, Cryptology ePrint Archive, Paper 2017/016, 2017. Available at: <https://eprint.iacr.org/2017/016>.
- [27] K. H. Eltartor, I. B. M. Algharbawi and R. M. I. Murad, “Design of a Practical Substitution Permutation Network Cryptosystem”, *Journal of Al-Aqsa University*, vol. 10, no. S.E. Pp. 462–475, 2006.
- [28] E. Miles and E. Viola, “Substitution-Permutation Networks, Pseudorandom Functions, and Natural Proofs”, *Journal of the ACM (JACM)*, vol. 62, no. 6, pp. 1–29, 2015. DOI: 10.1145/2792978. Available at: <https://doi.org/10.1145/2792978>.
- [29] A. N. Tentu, “A Review on Evolution of Symmetric Key Block Ciphers and Their Applications”, *IETE Journal of Education*, vol. 61, no. 1, pp. 34–46, 2020. DOI: 10.1080/09747338.2020.1769508. Available at: <https://doi.org/10.1080/09747338.2020.1769508>.
- [30] J. Lu, Y. Liu, T. Ashur, B. Sun and C. Li, “Rotational-XOR Cryptanalysis of Simon-like Block Ciphers”, in *Information Security and Privacy - 25th Australasian Conference, ACISP 2020, Proceedings*, J. K. Liu and H. Cui, επιμελητές, σειρά Lecture Notes in Computer Science (LNCS), vol. 12248, Berlin, Heidelberg: Springer, 2020, pp. 105–124. DOI: 10.1007/978-3-030-55304-3_6. Available at: https://doi.org/10.1007/978-3-030-55304-3_6.
- [31] R. E. Field and B. C. Jones, “Using carry-truncated addition to analyze add-rotate-xor hash algorithms”, *Journal of Mathematical Cryptology*, vol. 7, no. 1, pp. 97–110, 2013. DOI: 10.1515/jmc-2012-0019.
- [32] M. Robshaw, ``Stream Ciphers'', RSA Laboratories, Redwood City, CA, Technical Report TR-701, v. 2.0, 1995. Available at: <https://www.rsa.com/rsalabs>.
- [33] F. M. A. Eljadi and I. F. T. A. Shaikhli, “Dynamic linear feedback shift registers: A review”, in *2014 The 5th International Conference on Information and Communication Technology for The Muslim World (ICT4M)*, Kuching, Malaysia: IEEE, 2014, pp. 1–6, ISBN: 978-1-4799-6242-6. DOI: 10.1109/ICT4M.2014.7020598.
- [34] E. Dubrova, “On Constructing Secure and Hardware-Efficient Invertible Mappings”, 2015.
- [35] M. Bellare and C. Namprempre, *Authenticated Encryption: Relations among notions and analysis of the generic composition paradigm*, Cryptology ePrint Archive, Paper 2000/025, 2000. Available at: <https://eprint.iacr.org/2000/025>.
- [36] A. Quarter and A. Sharma, “Development of a Place and Route Tool for the RaPiD Architecture”,
- [37] P. Bouklis, L. Katselas and A. Hatzopoulos, “Design of Digital Circuit for a Passive RFID Tag”, 2015.
- [38] D. Engels, X. Fan, G. Gong, H. Hu and E. Smith, “Hummingbird: Ultra-Lightweight Cryptography for Resource-Constrained Devices”, vol. 6054, 2010, pp. 3–18, ISBN: 978-3-642-14991-7. DOI: 10.1007/978-3-642-14992-4_2.
- [39] D. Engels, X. Fan, G. Gong, H. Hu and E. Smith, “Ultra-lightweight cryptography for low-cost RFID tags: hummingbird algorithm and protocol”, *Centre for Applied Cryptographic Research (CACR) Technical Reports*, vol. 29, 2009.

- [40] D. Engels, M.-J. Saarinen, P. Schweitzer and E. Smith, “The Hummingbird-2 Lightweight Authenticated Encryption Algorithm”, vol. 2011, 2011, pp. 19–31, ISBN: 978-3-642-25285-3. DOI: 10.1007/978-3-642-25286-0_2.
- [41] M.-J. Saarinen, “Cryptanalysis of Hummingbird-1”, vol. 6733, 2011, pp. 328–341, ISBN: 978-3-642-21701-2. DOI: 10.1007/978-3-642-21702-9_19.
- [42] D. Engels, M.-J. Saarinen, P. Schweitzer and E. Smith, “The Hummingbird-2 Lightweight Authenticated Encryption Algorithm”, vol. 2011, 2011, pp. 19–31, ISBN: 978-3-642-25285-3. DOI: 10.1007/978-3-642-25286-0_2.
- [43] M.-J. Saarinen, “Cryptographic Analysis of All 4×4 -Bit S-Boxes”, vol. 2011, 2011, pp. 118–133, ISBN: 978-3-642-28495-3. DOI: 10.1007/978-3-642-28496-0_7.
- [44] M.-J. O. Saarinen, *Related-key Attacks Against Full Hummingbird-2*, Cryptology ePrint Archive, Paper 2013/070, 2013. Available at: <https://eprint.iacr.org/2013/070>.
- [45] Q. Chai and G. Gong, *A Cryptanalysis of HummingBird-2: The Differential Sequence Analysis*, Cryptology ePrint Archive, Paper 2012/233, 2012. Available at: <https://eprint.iacr.org/2012/233>.
- [46] G. S. Kris Gaj and R. Bachimanchi, “Comparison of hardware performance of selected Phase II eSTREAM candidates”, in *Proceedings of the International Conference on Field Programmable Logic and Applications (FPL)*, George Mason University, ECE Department, 2008.
- [47] F.-X. Standaert, G. Piret, G. Rouvroy and J.-J. Quisquater, “FPGA implementations of the ICEBERG block cipher.” *Integration*, vol. 40, pp. 20–27, 2007.
- [48] B. Li, M. Liu and D. Lin, “FPGA implementations of Grain v1, Mickey 2.0, Trivium, Lizard and Plantlet”, *Microprocessors and Microsystems*, vol. 78, p. 103210, 2020. DOI: 10.1016/j.micpro.2020.103210.
- [49] L. Pyrgas and P. Kitsos, “Compact Hardware Architectures of Enocoro-128v2 Stream Cipher for Constrained Embedded Devices”, *Electronics*, vol. 9, p. 1505, 2020. DOI: 10.3390/electronics9091505.
- [50] C. Lara-Nino, A. Díaz-Pérez and M. Morales-Sandoval, “Lightweight Hardware Architectures for the Present Cipher in FPGA”, *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. PP, pp. 1–12, 2017. DOI: 10.1109/TCSI.2017.2686783.
- [51] P. Maene and I. Verbauwhede, “Single-Cycle Implementations of Block Ciphers”, vol. 9542, 2016, pp. 131–147, ISBN: 978-3-319-29077-5. DOI: 10.1007/978-3-319-29078-2_8.
- [52] L. Daoud, F. Hussein and N. Rafla, “High-Level Synthesis Optimization of AES-128/192/256 Encryption Algorithms”, *International Journal of Computers and Their Applications*, vol. 26, pp. 129–136, Σεπτ. 2019.
- [53] R. Alharbi, H. Tariq, A. Aljaedi and A. Aljuhni, “Latency-Aware Accelerator of SIMECK Lightweight Block Cipher”, *Applied Sciences*, vol. 13, p. 161, 2022. DOI: 10.3390/app13010161.

- [54] İ. San and N. At, *Enhanced FPGA Implementation of the Hummingbird Cryptographic Algorithm*, Cryptology ePrint Archive, Paper 2010/586, 2010. Available at: <https://eprint.iacr.org/2010/586>.
- [55] N. Arora and Y. Gigras, “FPGA Implementation of Low Power and High Speed Hummingbird Cryptographic Algorithm”, *International Journal of Computer Applications*, vol. 92, 2014. DOI: 10.5120/16097-5423.
- [56] X. Fan, G. Gong, K. Lauffenburger and T. Hicks, “FPGA implementations of the Hummingbird cryptographic algorithm”, in *2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, 2010, pp. 48–51. DOI: 10.1109/HST.2010.5513116.
- [57] Y. Rekha, S. C. Yellamraju and M. Kamaraju, “Design and Implementation of Low Power, Light Weight Cryptographic Core using Hummingbird2 Algorithm”, *International Journal of Scientific and Engineering Research*, vol. 4, 2020.