Πανεπιστήμιο Πατρών
Τμήμα Μηχανικών Η/Υ & Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών
"Ολοκληρωμένα Συστήματα Υλικού και Λογισμικού"

Τίτλος Διπλωματικής Εργασίας

"Ανάπτυξη του πρωτοκόλλου CCMP για ασφαλή ασύρματα δίκτυα 802.11 σε FPGA"

ΛΑΟΥΔΙΑΣ ΧΡΗΣΤΟΣ

Επιβλέπων Καθηγητής: **ΣΕΡΠΑΝΟΣ ΔΗΜΗΤΡΙΟΣ**

Εξεταστική Επιτροπή: **ΓΚΟΥΤΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ**
ΚΟΥΦΟΠΑΥΛΟΥ ΟΔΥΣΣΕΑΣ
ΣΕΡΠΑΝΟΣ ΔΗΜΗΤΡΙΟΣ

Δεκέμβριος 2005

Πρόλογος

Τα ασύρματα δίκτυα που βασίζονται στο πρότυπο IEEE 802.11 είναι σήμερα από τα πλέον δημοφιλή παγκοσμίως. Παρόλη την ευρεία διάδοσή τους υπάρχει σημαντικό πρόβλημα όσον αφορά την ασφάλεια των δεδομένων που διακινούνται εντός του δικτύου. Αρχικά, στο πρότυπο οριζόταν μία μόνο μέθοδος για την ασφάλεια των πληροφοριών, που ονομάζεται WEP (Wired Equivalent Privacy) και βασίζεται στον αλγόριθμο κρυπτογράφησης RC4. Ήδη από το 2000 το WEP έχει αποδειχθεί ανεπαρκές και οι προσπάθειες για την αύξηση του επιπέδου της ασφάλειας οδήγησαν πρόσφατα στο πρότυπο IEEE 802.11i. Το πρότυπο ορίζει μία νέα μέθοδο, που εγγυάται την ασφάλεια των δεδομένων στο MAC επίπεδο. Ονομάζεται CCMP και βασίζεται στον αλγόριθμο κρυπτογράφησης AES (Advanced Encryption Standard).

Το CCMP παρέχει εμπιστευτικότητα, επικύρωση, ακεραιότητα και προστασία από την επανάληψη πακέτων. Βασίζεται στη χρήση του αλγόριθμου κρυπτογράφησης AES σε κατάσταση λειτουργίας CCM. Το CCM συνδυάζει την κατάσταση λειτουργίας CTR (Counter mode) για εμπιστευτικότητα και την CBC (Cipher Block Chaining mode) για επικύρωση και ακεραιότητα. Το CCM προστατεύει την ακεραιότητα τόσο των δεδομένων του πακέτου, όσο και συγκεκριμένων τμημάτων της επικεφαλίδας του πακέτου. Η επεξεργασία που γίνεται στο CCMP από τον αλγόριθμο AES χρησιμοποιεί μέγεθος κλειδιού 128-bit και μέγεθος μπλοκ 128-bit. Μετά την επεξεργασία από το CCMP το μέγεθος του πακέτου έχει επεκταθεί κατά 16 bytes, 8 bytes για την επικεφαλίδα του CCMP και 8 bytes για την ψηφιακή υπογραφή MIC (Message Integrity Code). Τα δεδομένα του πακέτου και το MIC μεταδίδονται κρυπτογραφημένα, αφού προστεθεί η αρχική επικεφαλίδα του πακέτου και η επικεφαλίδα του CCMP.

Στα πλαίσια της διπλωματικής μελετήθηκαν διάφορες αρχιτεκτονικές για την υλοποίηση του συστήματος κρυπτογράφησης και αποκρυπτογράφησης σύμφωνα με το CCMP. Οι αρχιτεκτονικές αυτές παρουσιάζουν διαφορετικά χαρακτηριστικά όσον αφορά την επιφάνεια, την ταχύτητα λειτουργίας και το συνολικό throughput. Η υλοποίηση και ο έλεγχος ορθής λειτουργίας των σχεδιασμών έγινε σε τεχνολογία FPGA Spartan-3 της εταιρίας Xilinx.

Θα ήθελα να ευχαριστήσω τον Δημήτριο Σερπάνο, καθηγητή του Τμήματος Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών, για τη βοήθεια που μου προσέφερε ως επιβλέπων της διπλωματικής εργασίας. Δεν μπορώ επίσης να παραβλέψω τις διευκολύνσεις που μου παρείχε η Εθνική Φρουρά κατά τη συγγραφή της εργασίας, το διάστημα που υπηρετούσα τη στρατιωτική μου θητεία στην Κύπρο.

Πίνακας Περιεχομένων

Πίνακας Σχημάτων

1 Ασφάλεια στα ασύρματα δίκτυα 802.11

1.1 Σύνοψη

Τα πρώτα πέντε χρόνια της ύπαρξής του, το πρότυπο IEEE 802.11 για ασύρματα δίκτυα, καθόριζε μόνο μία μέθοδο για την ασφάλεια των δεδομένων που διακινούνται ανάμεσα στους χρήστες του δικτύου. Η μέθοδος αυτή ονομάζεται Wired Equivalent Privacy (WEP) και όπως δηλώνει η ονομασία της, σκοπός είναι να παρέχει στο ασύρματο δίκτυο ασφάλεια του ίδιου επιπέδου με τα ενσύρματα δίκτυα. Αυτό είναι εν γένει δύσκολο αφού κάποιος αποκτά πιο εύκολα πρόσβαση στο μέσο που χρησιμοποιείται για τις ασύρματες επικοινωνίες, δηλαδή τον αέρα, παρά στα καλώδια ενός ενσύρματου δικτύου. Το πρότυπο IEEE 802.11 ορίζει δύο μεθόδους ασφάλειας: την ανοικτή ασφάλεια (open security) και την ασφάλεια κοινού κλειδιού (shared key). Η δεύτερη μέθοδος βασίζεται στη χρήση ενός κοινού μυστικού κλειδιού και αποτελεί ουσιαστικά το σύστημα WEP, ενώ η πρώτη συνεπάγεται μηδενική ασφάλεια.

Μέχρι το 2000 η δημοτικότητα των ασύρματων δικτύων αυξήθηκε σημαντικά και προσέλκυσαν το ενδιαφέρον της κρυπτογραφικής κοινότητας. Σύντομα πολλοί επιστήμονες διαπίστωσαν αδυναμίες στη συνολική προσέγγιση της ασφάλειας που παρέχει το WEP. Μέχρι το τέλος του 2001, υπήρχαν στο διαδίκτυο εργαλεία που παραβίαζαν οποιαδήποτε δικλείδα ασφάλειας του WEP και μάλιστα σε σύντομο χρονικό διάστημα.

Το WEP αποτελεί για πολλούς χρήστες ακόμη και σήμερα τη μοναδική επιλογή για την προστασία των δεδομένων που ανταλλάσσουν μέσω ενός ασύρματου δικτύου. Μέχρι την πλήρη μετάβαση στο νέο πρωτόκολλο ασφάλειας (CCMP), που απαιτεί την αλλαγή όλων των ασύρματων συσκευών καθώς βασίζεται σε διαφορετικό αλγόριθμο κρυπτογράφησης, υπάρχει η λύση του TKIP, που προσφέρει μεγαλύτερα επίπεδα ασφάλειας απαιτώντας απλά την αναβάθμιση του firmware και πιθανώς του λογισμικού (driver) της συσκευής. Παρόλες τις αδυναμίες του, το WEP είναι πολύ καλύτερη λύση από το να μην υπάρχει καθόλου ασφάλεια, αρκεί να γνωρίζει κανείς τις αδυναμίες του. Άλλωστε οι περισσότερες επιθέσεις βασίζονται στη συλλογή μεγάλου δείγματος

δεδομένων που μεταδίδονται και επομένως για έναν οικιακό χρήστη, όπου τα πακέτα που διακινούνται είναι λίγα, το WEP αποτελεί αρκετά ασφαλή επιλογή.

Αρκετοί είναι αυτοί που ασκούν κριτική στους σχεδιαστές του προτύπου IEEE 802.11, ότι δημιούργησαν το WEP με τέτοιο τρόπο ώστε να έχει τελικά τεράστιες αδυναμίες. Παρόλα αυτά πρέπει να λάβουμε υπόψη ότι τον καιρό που σχεδιάστηκε το WEP ο σκοπός δεν ήταν να παρέχει ασφάλεια στρατιωτικού επιπέδου. Μάλιστα το πρότυπο IEEE 802.11 του 1999 ορίζει ότι οι στόχοι του WEP είναι οι ακόλουθοι [1]:

1. Ισχυρό σε λογικά πλαίσια: Η ασφάλεια που παρέχει ένας αλγόριθμος είναι συνάρτηση της δυσκολίας που επιφέρει για την αποκάλυψη του κλειδιού της κρυπτογράφησης μετά από μία απευθείας επίθεση (brute-force attack). Η δυσκολία αυτή συνδέεται άμεσα με το μήκος του κλειδιού κρυπτογράφησης και τη συχνότητα με την οποία αλλάζει αυτό. Το WEP επιτρέπει την αλλαγή του κλειδιού (K) και του διανύσματος αρχικοποίησης (Initialization Vector - IV).

2. Συγχρονίζεται μόνο του: Το WEP συγχρονίζεται μόνο του, με την έννοια ότι κάθε πακέτο κρυπτογραφείται ξεχωριστά, ώστε έχοντας το κρυπτογραφημένο πακέτο και το κλειδί υπάρχει όλη η απαιτούμενη πληροφορία για την αποκρυπτογράφηση. Δηλαδή αν χαθεί κάποιο πακέτο, αυτό δε θα επηρεάσει την αποκρυπτογράφηση των υπολοίπων πακέτων. Η ιδιότητα αυτή είναι εξαιρετικά σημαντική για τον αλγόριθμο κρυπτογράφησης στο επίπεδο της σύνδεσης (data link), όπου ο ρυθμός απώλειας πακέτων λόγω του μη αξιόπιστου μέσου μετάδοσης είναι μεγάλος.

3. Είναι αποδοτικό: Το WEP βασίζεται στον αλγόριθμο κρυπτογράφησης RC4 και μπορεί να υλοποιηθεί συνολικά πολύ εύκολα, τόσο σε υλικό, όσο και σε λογισμικό.

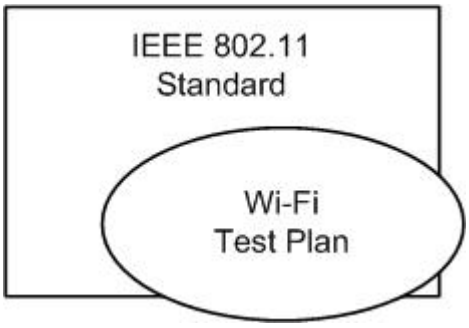
4. Είναι εξαγώγιμο: Έγινε προσπάθεια το WEP να είναι σχεδιασμένο με τέτοιο τρόπο ώστε να εγκριθεί από το Υπουργείο Εμπορίου των ΗΠΑ και να είναι δυνατή η εξαγωγή όσων προϊόντων βασίζονται σε αυτό. Λόγω της αντίληψης που υπήρχε για την κρυπτογραφία τον καιρό

που γινόταν η προτυποποίηση του 802.11, δεν ήταν βέβαιο ότι αυτό θα λάβει πλήρη έγκριση για εξαγωγή προϊόντων εκτός των ΗΠΑ.

5. Είναι προαιρετικό: Η υλοποίηση και η χρήση του WEP από οποιαδήποτε συσκευή ή ασύρματο δίκτυο IEEE 802.11 αντίστοιχα δεν είναι υποχρεωτική.

Το πρότυπο 802.11 ορίζει μόνο τη χρήση κλειδιών μήκους 40 bits. Είναι γνωστό ότι κλειδιά τέτοιου μήκους δεν μπορούν να αντισταθούν σε σοβαρές απευθείας επιθέσεις. Για αυτό το λόγο ουσιαστικά εγκρίθηκε για εξαγωγή εκτός των ΗΠΑ. Το σκεπτικό είναι ότι αν για παράδειγμα μία τράπεζα επιθυμεί να χρησιμοποιεί ασύρματο δίκτυο, τότε θα έχει το δικό της πρωτόκολλο ασφάλειας πάνω από το WEP, όπως αρμόζει σε μία τέτοια εφαρμογή.

Το πρότυπο 802.11 είναι ογκώδες και πολύπλοκο. Περιέχει αρκετές ασάφειες και επιπλέον κάποια χαρακτηριστικά δεν είναι υποχρεωτικό να υλοποιηθούν, με αποτέλεσμα οι διάφοροι κατασκευαστές να καταλήγουν σε υλοποιήσεις που δεν είναι συμβατές μεταξύ τους. Για να αποφευχθούν τα προβλήματα διασύνδεσης κάποιοι μεγάλοι κατασκευαστές δημιούργησαν τον οργανισμό Wi-Fi Alliance, με σκοπό την καθιέρωση συγκεκριμένων προδιαγραφών που πρέπει να πληρούν όλα ανεξαιρέτως τα προϊόντα 802.11. Προκειμένου να πιστοποιηθεί κάποιο προϊόν θα πρέπει να περάσει το σχέδιο δοκιμής (Test Plan) της Wi-Fi Alliance. Οι προδιαγραφές που ορίζονται από το σχέδιο δοκιμής δε συμβαδίζουν πλήρως με το πρότυπο IEEE 802.11. Κάποια απαιτούμενα χαρακτηριστικά διατηρούνται, ενώ προστίθενται κάποια επιπλέον, όπως φαίνεται στο Σχήμα 1.1.



Σχήμα 1.1 Σχέση ανάμεσα στο πρότυπο 802.11 και την πιστοποίηση Wi-Fi.

Όταν άρθηκαν οι περιορισμοί για τις εξαγωγές προϊόντων κρυπτογραφίας, έγιναν κάποιες τροποποιήσεις, εκτός προτύπου που οδήγησαν στη χρησιμο-

ποίηση κλειδιών μήκους 104 bits. Η επέκταση του κλειδιού υιοθετήθηκε από την προδιαγραφή Wi-Fi το 1999 και η χρήση του έγινε επιβεβλημένη. Δυστυχώς, το γεγονός αυτό δε συνέβαλε σημαντικά στην αύξηση του επιπέδου της ασφάλειας και σύντομα αναζητήθηκαν λύσεις για τη βελτίωση της κατάστασης.

1.2 Wi-Fi Protected Access

Το σύνολο των προδιαγραφών Wi-Fi προέκυψε μετά την ολοκλήρωση του προτύπου IEEE 802.11. Οι μεγαλύτεροι κατασκευαστές που συμμετείχαν στην ομάδα Wi-Fi, αποφάσισαν ότι η ασφάλεια είναι τόσο σημαντική για τον τελικό χρήστη, ώστε έπρεπε να κινηθούν το συντομότερο για να αντικαταστήσουν το WEP. Από την άλλη πλευρά, κατέληξαν στο συμπέρασμα ότι οι χρήστες δεν ήταν προετοιμασμένοι να πετάξουν στα άχρηστα όλο τον εξοπλισμό τους, προκειμένου να τον αντικαταστήσουν με νέα προϊόντα που προσφέρουν μεγαλύτερη ασφάλεια. Αποφάσισαν ότι μία απλή αναβάθμιση του λογισμικού της συσκευής ήταν πιο λογική τη δεδομένη χρονική στιγμή. Για να αντιμετωπιστεί αυτή η ανάγκη η ομάδα εργασίας i (Task Group i - TGi), που αποτελεί παρακλάδι της ομάδας η οποία ασχολείται με το πρότυπο 802.11, άρχισε να αναπτύσσει μία λύση για την αύξηση του επιπέδου της παρεχόμενης ασφάλειας, με βάση τις δυνατότητες των Wi-Fi συσκευών που υπήρχαν διαθέσιμες στην αγορά. Η προσπάθεια αυτή οδήγησε στον ορισμό του Πρωτοκόλλου Ακεραιότητας Προσωρινού Κλειδιού (Temporal Key Integrity Protocol - TKIP), το οποίο εξακολουθεί να βασίζεται στον αλγόριθμο κρυπτογράφησης RC4. Το TKIP διατηρείται ως εναλλακτική λύση στην επέκταση του προτύπου IEEE 802.11 για την ασφάλεια, που βασίζεται στον αλγόριθμο κρυπτογράφησης AES, το οποίο αντικαθιστά οριστικά το WEP.

Η ανάπτυξη του TKIP επέτρεψε την αναβάθμιση των συστημάτων που ήταν διαθέσιμα εκείνη τη στιγμή, αλλά η βιομηχανία δεν μπορούσε να περιμένει μέχρι να ολοκληρωθεί η χρονοβόρα διαδικασία του ορισμού του νέου προτύπου για τη βελτίωση της ασφάλειας. Για το λόγο αυτό, η Wi-Fi Alliance υιοθέτησε μια διαφορετική προσέγγιση ορίζοντας ένα υποσύνολο του νέου προτύπου, το οποίο περιείχε μόνο τις προδιαγραφές για τη χρήση του TKIP. Το υποσύνολο

αυτό ονομάζεται Wi-Fi Protected Access (WPA). Ήδη πολλοί μεγάλοι κατασκευαστές δικτυακών συσκευών για ασύρματα δίκτυα έχουν ετοιμάσει αναβαθμίσεις για το λογισμικό και το firmware, ώστε τα υπάρχοντα προϊόντα να υποστηρίζουν το WPA. Τα πιο καινούργια προϊόντα φτάνουν στα χέρια του καταναλωτή με ενσωματωμένη τη δυνατότητα υποστήριξης του WPA.

Στην ιστορία της τεχνολογίας δεν είναι λίγα τα παραδείγματα όπου η βιομηχανία προχώρησε ένα βήμα πιο πέρα από τις προδιαγραφές που ορίζουν τα πρότυπα. Αυτό συνήθως οδήγησε στην παράλληλη ύπαρξη στην αγορά συσκευών που είναι μεταξύ τους ασύμβατες. Ευτυχώς στην περίπτωση της Wi-Fi Alliance δεν ίσχυσε αυτός ο κανόνας και πλέον όλοι σχεδόν οι κατασκευαστές προσαρμόστηκαν στις προδιαγραφές που ορίζει το WPA.

1.3 Robust Secure Network

Η προσθήκη στο πρότυπο IEEE 802.11 που ορίζει την ασφάλεια της επόμενης γενιάς για τα ασύρματα δίκτυα ονομάζεται IEEE 802.11i. Η ομάδα εργασίας TGi είναι επιφορτισμένη με τον καθορισμό των προδιαγραφών. Το πρότυπο εκδόθηκε τελικά το 2004 [2].

Το πρότυπο IEEE 802.11i ορίζει ένα νέο τύπο ασύρματου δικτύου, το οποίο ονομάζεται Δίκτυο Ανθεκτικής Ασφάλειας (Robust Secure Network - RSN). Όσον αφορά τα περισσότερα χαρακτηριστικά λειτουργίας του ασύρματου δικτύου, αυτά είναι τα ίδια με τα συνηθισμένα δίκτυα που βασίζουν την ασφάλειά τους στο WEP. Παρόλα αυτά, προκειμένου να γίνει δεκτός ένας χρήστης σε κάποιο δίκτυο RSN, θα πρέπει η ασύρματη συσκευή του να υποστηρίζει πληθώρα νέων δυνατοτήτων, που αρχίζουν από την επικύρωση και τη διαχείριση κλειδιών σε υψηλό επίπεδο και φτάνουν μέχρι την κρυπτογράφηση και την επικύρωση των δεδομένων που διακινούνται σε MAC επίπεδο.

Σε ένα πραγματικό δίκτυο RSN επιτρέπεται η πρόσβαση μόνο σε συσκευές που ικανοποιούν τα απαιτούμενα κριτήρια και επιβάλλονται αρκετοί περιορισμοί ασφάλειας στη διαδικασία. Παρόλα αυτά, επειδή αρκετοί καταναλωτές χρειάζονται χρόνο για να αναβαθμίσουν τον εξοπλισμό τους και χρησιμοποιούν συσκευές που δεν υποστηρίζουν τις νέες δυνατότητες, το πρότυπο

IEEE 802.11i ορίζει το Δίκτυο Μεταβατικής Ασφάλειας (Transitional Security Network - TSN). Τα δίκτυα TSN υποστηρίζουν τόσο τις δυνατότητες του RSN, δηλαδή το CCMP ή εναλλακτικά το TKIP, όσο και το WEP. Οι συσκευές που γίνονται δεκτές σε ένα δίκτυο TSN, μπορούν να λειτουργήσουν παράλληλα για όλα τα προηγούμενα συστήματα ασφάλειας.

1.4 Διαφορές ανάμεσα στο RSN και το WPA

Το RSN και το WPA χρησιμοποιούν παρόμοια αρχιτεκτονική και προσεγγίζουν το θέμα της ασφάλειας με κοινό τρόπο. Το WPA είναι ένα υποσύνολο των δυνατοτήτων και επικεντρώνεται κυρίως στην υλοποίηση του δικτύου με συγκεκριμένο τρόπο, ενώ το RSN επιτρέπει μεγαλύτερη ευελιξία στην υλοποίηση. Το RSN θέτει ως απαραίτητη προϋπόθεση τη χρήση του πρωτοκόλλου CCMP, που βασίζεται στον αλγόριθμο κρυπτογράφησης AES, με εναλλακτική λύση το TKIP, ενώ το WPA επικεντρώνεται στο TKIP. Επειδή σήμερα το WEP είναι η λύση που χρησιμοποιούν οι περισσότεροι οργανισμοί και επιχειρήσεις, η πιο λογική προσέγγιση είναι η αναβάθμιση στο WPA και η σταδιακή αντικατάσταση των συστημάτων προς τη λύση του RSN καθώς όλο και περισσότερα προϊόντα θα εμφανίζονται στην αγορά. Με τον τρόπο αυτό θα γίνει ομαλά η οριστική μετάβαση στο 802.11i. Το WPA καλύπτει τις αυξημένες ανάγκες σε ασφάλεια όσων χρησιμοποιούν ασύρματα δίκτυα 802.11 με το σημερινό εξοπλισμό και το RSN επιτρέπει μακροπρόθεσμα μεγαλύτερη ευελιξία.

Το RSN και το WPA χρησιμοποιούν παρόμοια αρχιτεκτονική κάτω από την οποία λειτουργούν τα πρωτόκολλα ασφάλειας που βασίζονται στους αλγόριθμους AES και RC4 αντίστοιχα. Η αρχιτεκτονική αυτή καλύπτει διαδικασίες όπως η επικύρωση σε υψηλό επίπεδο, η διανομή του κλειδιού κρυπτογράφησης και η ανανέωση του κλειδιού, που είναι κοινές στο RSN και το WPA. Η αρχιτεκτονική του RSN είναι πολύ διαφορετική σε σχέση με το WEP και πιο πολύπλοκη. Παρόλα αυτά το πρώτο αποτελεί μία εξαιρετική λύση για την ασφάλεια, η οποία μπορεί να εφαρμοστεί σε μεγάλα δίκτυα. Ένα από τα μεγαλύτερα προβλήματα του WEP είναι ότι στην πράξη δεν είναι εύκολη η διαχείριση της διανομής των κλειδιών, όταν οι χρήστες ξεπεράσουν τις μερικές

δεκάδες. Το πρόβλημα αυτό επιλύεται με αποδοτικό τρόπο τόσο από το RSN όσο και από το WPA.

Το RSN μπορεί να εφαρμοστεί είτε σε ένα δομημένο ασύρματο δίκτυο (infrastructure mode), όπου υπάρχουν σημεία πρόσβασης (Access Point - AP), είτε σε ένα τοπικό δίκτυο χωρίς AP (ad-hoc mode). Αντίθετα, το WPA δεν μπορεί να εφαρμοστεί σε ad-hoc ασύρματα δίκτυα.

Κανείς δεν μπορεί να ισχυριστεί ότι ένα σύστημα ασφάλειας είναι απαραβίαστο. Παρόλα αυτά και οι δύο πρόσφατες προσεγγίσεις αναπτύχθηκαν με τη συμβολή ειδικών σε θέματα ασφάλειας και εξειδικευμένους κρυπτογράφους, οι οποίοι τα μελέτησαν με μεγαλύτερη σχολαστικότητα σε σχέση με το WEP. Η μελέτη και η ανάλυση του WEP έγινε μόνο αφού τα προϊόντα που βασίζονταν σε αυτό είχαν ήδη κατακλύσει την αγορά και ήταν πλέον αργά για οποιεσδήποτε διορθώσεις. Ο προσεκτικός σχεδιασμός του RSN και του WPA εγγυάται ότι θα είναι σίγουρα πολύ πιο δύσκολο να παραβιαστούν στο άμεσο μέλλον και θα αποδειχθούν περισσότερο ανθεκτικά από το WEP.

1.5 Πλαίσιο ασφάλειας

Η ομάδα εργασίας TGi του IEEE 802.11i είχε δύο σκοπούς να επιτύχει κατά το σχεδιασμό του νέου προτύπου ασφάλειας: να δημιουργήσει μία λύση η οποία θα εφαρμόζεται εύκολα σε μεγάλα ασύρματα δίκτυα και θα παρέχει προστασία απέναντι σε όλες τις γνωστές παθητικές και ενεργητικές επιθέσεις κρυπτανάλυσης. Από την αρχή είχε αποφασιστεί ότι το νέο πρότυπο θα αντικαταστήσει οριστικά το WEP και επομένως οι ειδικοί που ανέλαβαν το έργο της προτυποποίησης άρχισαν από το μηδέν. Η πρώτη και πιο σημαντική αλλαγή που εφάρμοσαν ήταν ο διαχωρισμός της διαδικασίας επικύρωσης (authentication) του χρήστη, από τη διαδικασία προστασίας των μηνυμάτων (message protection), δηλαδή την παροχή εμπιστευτικότητας (confidentiality) και ακεραιότητας (integrity). **Επικύρωση** είναι η διαδικασία με την οποία κάποιος αποδεικνύει ότι είναι νόμιμος χρήστης του δικτύου. Παρατηρείστε ότι ο όρος υπονοεί και την αντίστροφη διαδικασία, ότι δηλαδή το δίκτυο, στην πράξη το AP με το οποίο επικοινωνεί η ασύρματη συσκευή, πιστοποιεί την ταυτότητά του. **Εμπιστευτικότητα** είναι η προστασία του απορρήτου των

μηνυμάτων που διακινούνται εντός του δικτύου. Είναι εφικτή μέσω ενός αλγορίθμου κρυπτογράφησης και ενός μυστικού κλειδιού, που αναλαμβάνουν σε συνδυασμό να μετατρέψουν τα πραγματικά δεδομένα που ανταλλάσσονται σε δεδομένα που μοιάζουν τυχαία για κάποιον εξωτερικό παρατηρητή. **Ακεραιότητα** είναι η παροχή μίας ψηφιακής υπογραφής η οποία πιστοποιεί ότι το μήνυμα που φτάνει στον παραλήπτη προέρχεται όντως από τον νόμιμο αποστολέα και ότι δεν έχουν τροποποιηθεί τα αρχικά δεδομένα σε καμία περίπτωση. Ο διαχωρισμός της επικύρωσης του χρήστη από την προστασία των μηνυμάτων επιτρέπει την εύκολη μεταφορά του συστήματος ασφάλειας σε δίκτυα μεγάλων οργανισμών. Παρόλα αυτά τα δύο μέρη πρέπει να είναι αλληλένδετα σε ένα ενιαίο πλαίσιο ασφάλειας. Η ραχοκοκαλιά του πλαισίου ασφάλειας είναι το κλειδί της κρυπτογράφησης.

1.6 Κλειδιά

Η ασφάλεια εξαρτάται σε μεγάλο βαθμό από τα μυστικά κλειδιά. Στο RSN το πλαίσιο της ασφάλειας βασίζεται στην κατοχή κλειδιών περιορισμένου χρόνου. Σε αντίθεση με το WEP, στο RSN υπάρχουν πολλά διαφορετικά κλειδιά, τα οποία αποτελούν μέρος μίας ιεραρχίας κλειδιών (key hierarchy). Τα περισσότερα από αυτά τα κλειδιά δεν είναι γνωστά πριν την ολοκλήρωση της διαδικασίας επικύρωσης. Στην πραγματικότητα, τα κλειδιά προκύπτουν σε πραγματικό χρόνο μόλις δημιουργηθεί το πλαίσιο ασφάλειας, αμέσως μετά την επικύρωση. Επειδή προκύπτουν σε πραγματικό χρόνο, ονομάζονται προσωρινά κλειδιά (temporal keys). Αυτά τα προσωρινά κλειδιά μπορεί να ανανεωθούν, αλλά πάντα καταστρέφονται όταν κλείσει το πλαίσιο ασφάλειας. Επομένως, μετά την επιτυχημένη επικύρωση ο χρήστης έχει τη δυνατότητα να παραλάβει ή να δημιουργήσει ο ίδιος τα κλειδιά που χρειάζονται για την κρυπτογράφηση και την προστασία της ακεραιότητας των δεδομένων.

Η επικύρωση βασίζεται σε κάποια κοινή μυστική πληροφορία, η οποία δεν μπορεί να δημιουργηθεί με αυτόματο τρόπο. Πρέπει να δημιουργηθεί ένα κλειδί για την επικύρωση από κάποια έμπιστη πηγή και να αποδοθεί στο χρήστη με τέτοιο τρόπο ώστε να μη μπορεί αυτό να κλαπεί ή να αντιγραφεί. Επομένως, αυτός που παρέχει το κλειδί της επικύρωσης πρέπει να είναι βέβαιος

για την ταυτότητα του παραλήπτη του κλειδιού. Η βάση των μεθόδων επικύρωσης είναι ότι ο χρήστης που επιδιώκει την επικύρωση κατέχει εκ των προτέρων μία πληροφορία, η οποία ονομάζεται κύριο κλειδί (master key). Είναι απαραίτητο το κύριο κλειδί να χρησιμοποιηθεί με τέτοιο τρόπο, ώστε να αποτραπεί η αποκάλυψή του. Γενικά, το κύριο κλειδί δε χρησιμοποιείται ποτέ απευθείας για οποιαδήποτε διαδικασία ασφάλειας, πέραν της επικύρωσης σε υψηλό επίπεδο. Αντίθετα, χρησιμοποιείται ώστε να προκύψουν τα προσωρινά κλειδιά. Το WEP παραβιάζει αυτό τον κανόνα, καθώς χρησιμοποιεί το κύριο κλειδί τόσο για την επικύρωση, όσο και για την κρυπτογράφηση.

Επομένως, υπάρχουν δύο τύποι κλειδιών. Το κύριο κλειδί αποτελεί απόδειξη της ταυτότητας κάποιου και τα προσωρινά κλειδιά που προκύπτουν από το κύριο κλειδί, ώστε να χρησιμοποιηθούν στο πρωτόκολλο ασφάλειας. Αυτή η αρχή ακολουθείται στο σχεδιασμό του RSN.

1.7 Επίπεδα ασφάλειας

Όταν το WEP εμφανίστηκε στην αγορά υποσχόμενο την ασφάλεια των ασύρματων επικοινωνιών σε δίκτυα 802.11, δημιούργησε αίσθηση το γεγονός ότι όλα τα θέματα ασφάλειας ήταν συμπιεσμένα στο ίδιο πακέτο. Όλα τα μέτρα και οι δικλείδες ασφάλειας, από την επικύρωση του χρήστη μέχρι την κρυπτογράφηση των δεδομένων, ορίζονταν μέσα σε ένα μόνο πρότυπο. Εκτός από τους τεχνικούς λόγους για τους οποίους απέτυχε το WEP, η προσέγγιση αυτή δημιούργησε τεράστια προβλήματα επεκτασιμότητας. Το WEP δεν είναι δυνατό να εφαρμοστεί με αποδοτικό τρόπο σε ένα δίκτυο με περισσότερες από μερικές δεκάδες συσκευές. Ορισμένες λειτουργίες, όπως αυτή της κρυπτογράφησης, είναι πολύ συγκεκριμένες και σχετίζονται με το υλικό της συσκευής Wi-Fi, που αναλαμβάνει τελικά την επικοινωνία. Άλλα θέματα παρόλα αυτά, όπως ποιος επιτρέπεται να έχει πρόσβαση στο δίκτυο, είναι εξίσου σημαντικά και απαιτούν συνολικότερη αντιμετώπιση για ολόκληρο το δίκτυο.

Για το λόγο αυτό είναι απαραίτητη η διάκριση και υλοποίηση επιπέδων διαχείρισης στο σύστημα ασφάλειας που χρησιμοποιείται. Όσον αφορά την ασφάλεια των ασύρματων τοπικών δικτύων 802.11, υπάρχουν τρία διακριτά επίπεδα ασφάλειας [3]. Στην πραγματικότητα αυτά τα επίπεδα δε συναντώνται

μόνο στα ασύρματα δίκτυα, αλλά εφαρμόζονται σε οποιοδήποτε σύστημα ασφάλειας σε ένα τοπικό δίκτυο. Το πλεονέκτημα αυτής της προσέγγισης είναι ότι η λύση του RSN μπορεί να στηριχθεί στις υπάρχουσες αρχιτεκτονικές ασφάλειας και στα ήδη δοκιμασμένα πρότυπα.

Τα τρία επίπεδα της ασφάλειας είναι:

- Επίπεδο ασύρματου δικτύου
- Επίπεδο ελέγχου πρόσβασης
- Επίπεδο επικύρωσης

Το **επίπεδο ασύρματου δικτύου** (Wireless LAN Layer) είναι επιφορτισμένο με την αποστολή και λήψη των δεδομένων, την αποστολή μηνυμάτων που περιλαμβάνουν τις δυνατότητες της συσκευής και την αποδοχή αιτήσεων για την είσοδο στο δίκτυο. Είναι επίσης υπεύθυνο για την κρυπτογράφηση και αποκρυπτογράφηση των πραγματικών δεδομένων, μόλις δημιουργηθεί το πλαίσιο ασφάλειας.

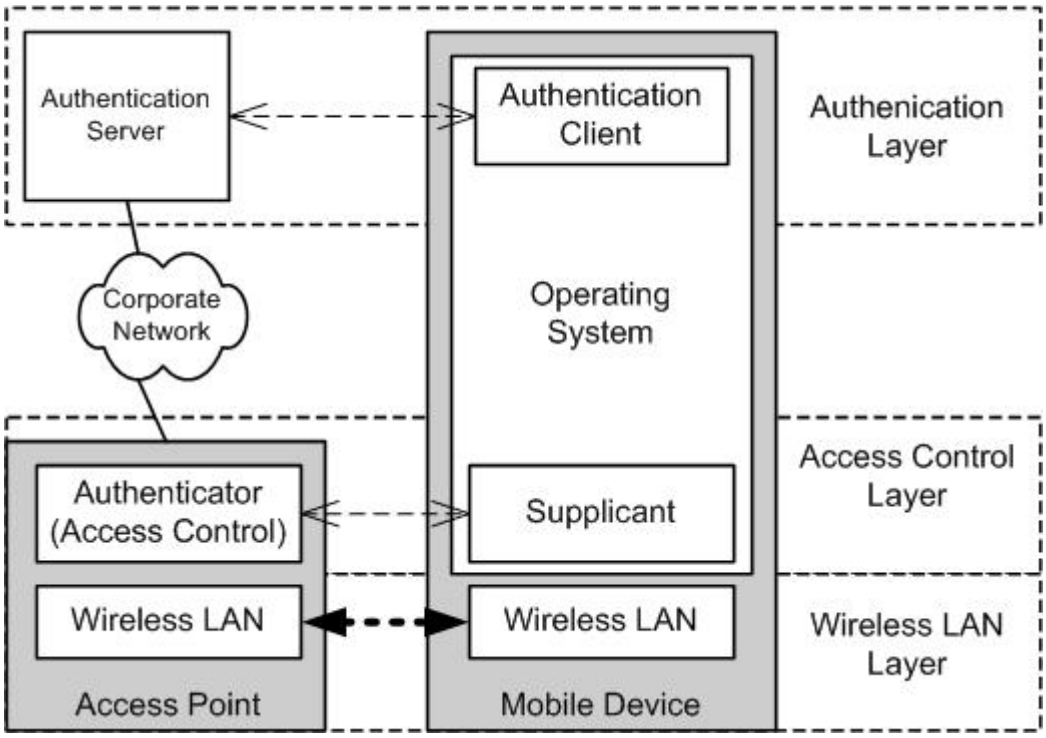
Το **επίπεδο ελέγχου πρόσβασης** (Access Control Layer) βρίσκεται στη μέση και συντονίζει τις ενέργειες για τη δημιουργία του πλαισίου ασφάλειας. Πρέπει να εμποδίσει την ανταλλαγή δεδομένων μεταξύ των νόμιμων χρηστών του δικτύου και ενός μη εξουσιοδοτημένου χρήστη. Επιπλέον, επικοινωνεί με το επίπεδο επικύρωσης ώστε να γνωρίζει πότε να δημιουργήσει το πλαίσιο ασφάλειας και συμμετέχει στην κατασκευή των προσωρινών κλειδιών, που σχετίζονται με την κάθε επικοινωνία.

Το ανώτερο επίπεδο είναι το **επίπεδο επικύρωσης** (Authentication Layer). Σε αυτό το επίπεδο λαμβάνονται οι αποφάσεις για την αποδοχή ή την απόρριψη των αιτήσεων που γίνονται για την είσοδο στο δίκτυο.

Το επίπεδο ασύρματου δικτύου βρίσκεται στην ασύρματη συσκευή, που αποτελεί μέρος του AP. Συνήθως, το επίπεδο ελέγχου πρόσβασης υλοποιείται εξ ολοκλήρου στο AP. Παρόλο που σε μικρά συστήματα το επίπεδο επικύρωσης μπορεί να βρίσκεται επίσης στο AP, σε μεγαλύτερα δίκτυα το επίπεδο αυτό υλοποιείται από ένα εξυπηρετητή επικύρωσης (authentication server). Ο εξυπηρετητής αυτός είναι ξεχωριστός από τα AP. Το πλεονέκτημα της χρήσης

ενός τέτοιου εξυπηρετητή είναι η δυνατότητα κεντρικής διαχείρισης της βάσης δεδομένων των χρηστών. Επιλύεται, δηλαδή με αποδοτικό τρόπο η διαχείριση κλειδιών του WEP και γίνεται πιο εύκολη η υλοποίηση ασύρματων δικτύων, ώστε να συνυπάρχουν με τα υπόλοιπα ενσύρματα δίκτυα αρμονικά κάτω από το ίδιο σύστημα ασφάλειας.

Σε μία ασύρματη συσκευή υπάρχουν τα ίδια επίπεδα. Το επίπεδο ασύρματου δικτύου υλοποιείται στην Wi-Fi κάρτα και τους αντίστοιχους οδηγούς της συσκευής. Το επίπεδο ελέγχου πρόσβασης και επικύρωσης συνήθως υλοποιούνται από το λειτουργικό σύστημα. Παρατηρείστε ότι είναι εξίσου σημαντικό η ασύρματη συσκευή να επιβεβαιώσει ότι προσπαθεί να εισέλθει στο επιθυμητό δίκτυο και όχι σε κάποιο ψεύτικο δίκτυο. Στο Σχήμα 1.2 φαίνεται η διασύνδεση μεταξύ των επιπέδων ασφάλειας και ένα τυπικό παράδειγμα του τρόπου που λειτουργούν αυτά. Παρατηρείστε ότι στο Σχήμα 1.2 ο όρος supplicant αναφέρεται στο τμήμα εκείνο του λειτουργικού συστήματος της συσκευής που πραγματοποιεί την αίτηση για είσοδο στο ασύρματο δίκτυο.



Σχήμα 1.2 Διασύνδεση των επιπέδων ασφάλειας.

Τα πρότυπα του IEEE 802.11 καλύπτουν μόνο τα ασύρματα τοπικά δίκτυα και αυτή η ομάδα εργασίας δεν καθορίζει τη συμπεριφορά συστημάτων εκτός της συγκεκριμένης περιοχής. Αυτό δημιουργεί προβλήματα κατά το σχεδιασμό

συστημάτων που απαιτούν τη συνεργασία ετερογενών υποσυστημάτων σε διάφορα επίπεδα. Για αυτό το λόγο το αρχικό πρότυπο για την ασφάλεια, δηλαδή το WEP, προσπάθησε να επιλύσει όλα τα θέματα ασφάλειας στο επίπεδο ασύρματου δικτύου. Κατά το σχεδιασμό του RSN, αυτό το πρόβλημα ξεπεράστηκε ενσωματώνοντας πρότυπα εκτός του IEEE 802.11, κυρίως για τον έλεγχο της πρόσβασης και της επικύρωσης.

Για το επίπεδο ελέγχου πρόσβασης υπάρχει το πρότυπο 802.1X που επιλέχθηκε ως το πλέον κατάλληλο και συμπεριλήφθηκε στις προδιαγραφές του RSN σχεδόν με καθολική αποδοχή, αφού τροποποιήθηκε σε μερικά σημεία για να ικανοποιεί όλες τις απαιτήσεις ασφάλειας της ομάδας εργασίας TGi.

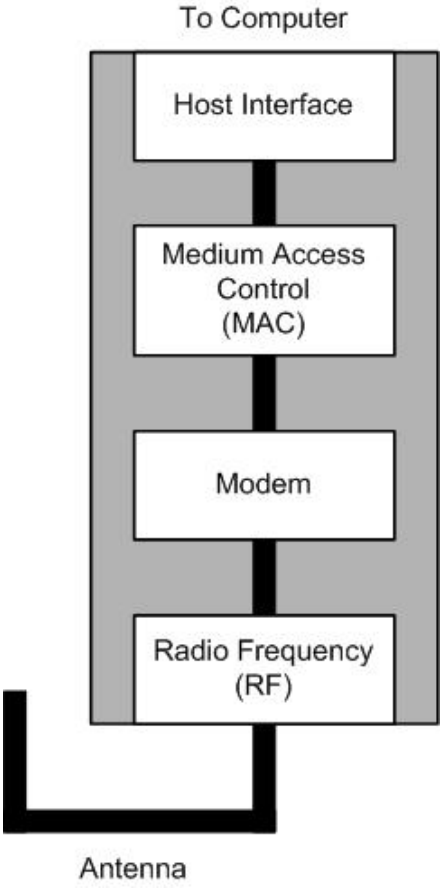
Για το επίπεδο της επικύρωσης υπάρχουν πολλές διαφορετικές εναλλακτικές λύσεις. Άλλωστε ο σκοπός της υλοποίησης της επικύρωσης σε ανώτερο επίπεδο, είναι να διευκολύνει οργανισμούς και επιχειρήσεις να ολοκληρώσουν τη διαδικασία της επικύρωσης στο σύστημα ασφάλειας που ήδη διαθέτουν. Τελικά, στις προδιαγραφές του RSN δεν ορίζεται ρητά το πρότυπο που ακολουθεί ο εξυπηρετητής για την επικύρωση των χρηστών.

Σκοπός αυτής της διπλωματικής εργασίας δεν είναι η ανάλυση όλων των επιπέδων που συμβάλλουν στην ολοκληρωμένη λύση ασφάλειας που ορίζει το πρότυπο IEEE 802.11i, αλλά η μελέτη, ανάλυση και υλοποίηση του πρωτοκόλλου ασφάλειας που ορίζεται για το επίπεδο ασύρματου δικτύου και αποτελεί μέρος του RSN.

Για να γίνει κατανοητό το πρωτόκολλο ασφάλειας στο χαμηλότερο από τα τρία επίπεδα πρέπει να δούμε προσεκτικά το εσωτερικό των Wi-Fi συστημάτων και ειδικά της ασύρματης συσκευής Wi-Fi. Στο Σχήμα 1.3 φαίνεται μία Wi-Fi κάρτα, η οποία αποτελείται από τα ακόλουθα τμήματα:

- Τμήμα ραδιοφωνικής συχνότητας (Radio Frequency - RF)
- Τμήμα modem
- Τμήμα MAC (Medium Access Control)
- Τμήμα διασύνδεσης με τον Η/Υ, πχ USB

Σε γενικές γραμμές το τμήμα RF ασχολείται με την λήψη και αποστολή ραδιοφωνικών σημάτων μέσω της κεραίας. Το modem ασχολείται με την ανακατασκευή των δεδομένων από τα σήματα που παραλήφθηκαν και το αντίστροφο, δηλαδή τη μορφοποίηση των δεδομένων ώστε να είναι κατάλληλα για αποστολή. Το τμήμα MAC ασχολείται με θέματα πρωτοκόλλου, όπως ο τεμαχισμός των δεδομένων (fragmentation) και η κρυπτογράφησή τους ανάλογα με τον αλγόριθμο που χρησιμοποιείται. Σήμερα υπάρχει η τάση όλα τα τμήματα να ενσωματωθούν σε ένα μόνο ολοκληρωμένο κύκλωμα (IC), το οποίο μελλοντικά θα περιέχει και το τμήμα RF, οδηγώντας σε μία ολοκληρωμένη λύση SoC (System-on-Chip).



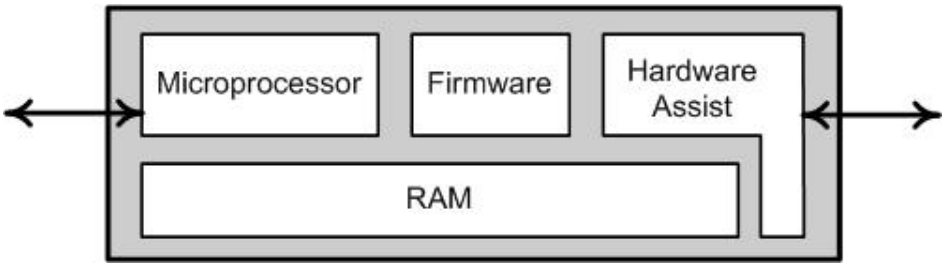
Σχήμα 1.3 Τα τμήματα μίας Wi-Fi συσκευής.

Το τμήμα που αποτελεί αντικείμενο μελέτης αυτής της διπλωματικής είναι το τμήμα MAC, κυρίως το κομμάτι που υλοποιεί το πρωτόκολλο ασφάλειας. Το τμήμα MAC είναι αυτό που υλοποιεί το μεγαλύτερο μέρος του προτύπου 802.11. Από την πλευρά της διασύνδεσης με τον Η/Υ, λαμβάνει από το λειτουργικό σύστημα πακέτα δεδομένων για μετάδοση και εντολές για ενέργειες, όπως η αναζήτηση ενός νέου AP ή η αποστολή μηνύματος για τη σύνδεση με

το AP. Παραδίδει επίσης στον Η/Υ πακέτα δεδομένων που παραλαμβάνει. Από την πλευρά του modem, το τμήμα MAC παραδίδει ακολουθίες από bits που περιέχουν όλα τα αναγκαία πακέτα ελέγχου και δεδομένων του IEEE 802.11. Σε αυτή την πλευρά γίνεται η κρυπτογράφηση και αποκρυπτογράφηση των πακέτων δεδομένων, σύμφωνα με το πρωτόκολλο ασφάλειας που υλοποιείται.

Επειδή οι λειτουργίες του επιπέδου MAC είναι αρκετά πολύπλοκες, όλες οι υλοποιήσεις βασίζονται σε ένα μικροεπεξεργαστή, που είναι ολοκληρωμένος στο ίδιο IC (embedded microprocessor). Ο μικροεπεξεργαστής είναι προγραμματισμένος ώστε να χειρίζεται όλες τις λειτουργίες χρονισμού και μορφοποίησης για τον έλεγχο του προτύπου. Γενικά, ο μικροεπεξεργαστής δεν είναι αρκετά ισχυρός και ορισμένες λειτουργίες είναι πολύ ταχείς για να τις χειριστεί. Για το λόγο αυτό, το τμήμα MAC υλοποιείται ως συνδυασμός firmware και hardware, όπως φαίνεται στο Σχήμα 1.4.

Η μονάδα Hardware Assist είναι αυτή που βοηθάει το μικροεπεξεργαστή στη Wi-Fi κάρτα να επιτύχει το ρυθμό επεξεργασίας δεδομένων (throughput) των 54 Mbps, που είναι το θεωρητικό όριο στα πρότυπα IEEE 802.11a/g. Πρόκειται για ψηφιακό κύκλωμα που σχεδιάζεται ώστε να εκτελεί τους αναγκαίους υπολογισμούς ταχύτατα και να καταναλώνει όσο το δυνατόν λιγότερη ισχύ. Αν όλες οι λειτουργίες του πρωτοκόλλου MAC εκτελούνταν από το μικροεπεξεργαστή, τότε θα ήταν εύκολη η αλλαγή του συστήματος ασφάλειας με την αναβάθμιση του firmware. Επειδή η κρυπτογράφηση και η αποκρυπτογράφηση απαιτούν σημαντικούς υπολογιστικούς πόρους, η υλοποίηση των πρωτοκόλλων ασφάλειας βασίζεται στη λειτουργία της μονάδας Hardware Assist. Φυσικά, η μονάδα αυτή δεν μπορεί να αλλάξει τη λειτουργία της μετά την κατασκευή της κάρτας.



Σχήμα 1.4 Το εσωτερικό του MAC κυκλώματος.

Τώρα γίνεται κατανοητό ποιος είναι ο σκοπός του TKIP. Στις πρώτες συσκευές, που υποστήριζαν μόνο το WEP, η μονάδα Hardware Assist ουσιαστικά υλοποιούσε τον αλγόριθμο κρυπτογράφησης RC4. Αντίθετα, στις νέες συσκευές, που είναι συμβατές με το RSN, υλοποιεί τον αλγόριθμο AES. Επομένως, το TKIP, αποτελεί την προσπάθεια των σχεδιαστών να υλοποιήσουν ένα σύστημα που θα παρέχει πραγματική ασφάλεια, χρησιμοποιώντας την υπάρχουσα υλοποίηση του αλγόριθμου RC4 της συσκευής.

Πριν προχωρήσουμε στα βασικά χαρακτηριστικά των πρωτοκόλλων ασφάλειας WEP και TKIP είναι σκόπιμο να γίνει κατανοητή η διαφορά ανάμεσα σε δύο έννοιες που συχνά συγχέονται: το MAC Protocol Data Unit (MPDU) και το MAC Service Data Unit (MSDU), ώστε να γίνει κατανοητός ο τρόπος λειτουργίας των δύο πρωτοκόλλων. Τόσο το MPDU όσο και το MSDU αναφέρονται σε ένα πακέτο δεδομένων, το οποίο περιέχει διεύθυνση αποστολέα και παραλήπτη. Το MSDU είναι το πακέτο δεδομένων που μεταφέρεται από το λογισμικό που τρέχει στον υπολογιστή του χρήστη στο επίπεδο Medium Access Control (MAC) του ασύρματου δικτύου. Το MPDU είναι το πακέτο δεδομένων που μεταφέρεται από το επίπεδο MAC στην κεραία της ασύρματης συσκευής. Κατά τη μετάδοση ενός μηνύματος στέλνονται διαδοχικά MSDU από το λειτουργικό σύστημα προς το MAC επίπεδο και μετατρέπονται σε MPDU έτοιμα για αποστολή από το ασύρματο δίκτυο μέσω του αέρα. Κατά την παραλαβή ενός μηνύματος, τα MPDU φτάνουν μέσω της κεραίας και ενώνονται ώστε να σχηματίσουν τα αρχικά MSDU πριν παραδοθούν στο λειτουργικό σύστημα για επεξεργασία, ανάλογα με την εφαρμογή του χρήστη. Ένα MSDU μπορεί να χωριστεί σε αρκετά MPDU, μέσω της διαδικασίας που ονομάζεται τεμαχισμός (fragmentation). Τα διάφορα MPDU ενώνονται στη συνέχεια στο άλλο άκρο, ώστε να σχηματίσουν το MSDU. Αυτό γίνεται ώστε αν χαθεί η ασύρματη σύνδεση ανάμεσα στα δύο άκρα λόγω θορύβου, μόνο τα MPDU που χάθηκαν να πρέπει να σταλούν εκ νέου. Όσον αφορά τη διαδικασία της κρυπτογράφησης, έχει σημασία αν αυτή συμβαίνει στο επίπεδο του MPDU ή του MSDU, αφού επηρεάζεται η πολυπλοκότητα και η συνολική απόδοση του πρωτόκολλου ασφάλειας.

2 Τα συστήματα ασφάλειας WEP και TKIP

2.1 Περιγραφή του WEP

Η μέθοδος ασφάλειας που προβλέπεται από το πρότυπο 802.11 είναι το πρωτόκολλο WEP. Το WEP είναι ένα πρωτόκολλο ασφάλειας του MAC επιπέδου, το οποίο βασίζεται στη χρήση του αλγόριθμου κρυπτογράφησης RC4 με κάποιο μυστικό κλειδί. Το μυστικό κλειδί είναι κοινό ανάμεσα στους χρήστες που επικοινωνούν και τα AP. Σκοπός του WEP είναι να παρέχει επικύρωση για κάθε χρήστη, προστασία και ακεραιότητα των δεδομένων.

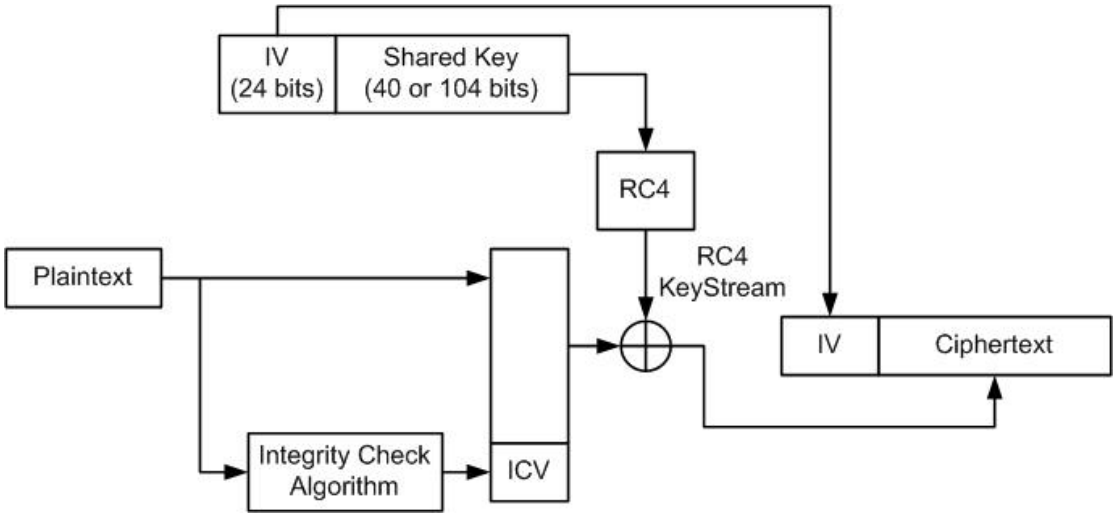
Το πρότυπο IEEE 802.11 ορίζει δύο μηχανισμούς ασφάλειας όσον αφορά τον έλεγχο της πρόσβασης στο ασύρματο δίκτυο: την ανοικτή ασφάλεια (open security) και την ασφάλεια κοινού κλειδιού (shared key security). Ο πρώτος μηχανισμός συνεπάγεται μηδενική ασφάλεια, αφού ο χρήστης πρέπει να δώσει απλά το σωστό Service Set Identity (SSID), που αντιστοιχεί στο όνομα του ασύρματου δικτύου που επιθυμεί να συνδεθεί. Ο δεύτερος βασίζεται στη χρήση ενός κοινού μυστικού κλειδιού και αποτελεί ουσιαστικά το σύστημα WEP. Σύμφωνα με αυτό το μηχανισμό το AP στέλνει ένα μήνυμα στο χρήστη που επιθυμεί να εισέλθει στο δίκτυο (challenge). Ο χρήστης ανταποκρίνεται κρυπτογραφώντας το μήνυμα αυτό με τη βοήθεια του μυστικού κλειδιού. Το AP κρυπτογραφεί το ίδιο μήνυμα, με το δικό του μυστικό κλειδί και συγκρίνει το αποτέλεσμα με το κρυπτογραφημένο μήνυμα του χρήστη. Αν ταιριάζουν, τότε ο χρήστης θεωρείται ότι είναι νόμιμος και η αίτησή του για είσοδο γίνεται αποδεκτή.

Το WEP χειρίζεται ταυτόχρονα την προστασία και την ακεραιότητα των δεδομένων. Το WEP επεξεργάζεται το πακέτο δεδομένων με ένα αλγόριθμο για τον έλεγχο του υπόλοιπου[1] (checksum) και δημιουργεί την τιμή ελέγχου ακεραιότητας (Integrity Check Value - ICV). Το ICV τοποθετείται στο τέλος του πακέτου και στη συνέχεια κρυπτογραφούνται χρησιμοποιώντας τον αλγόριθμο RC4. Ο αλγόριθμος RC4 παίρνει ως είσοδο ένα διάνυσμα αρχικοποίη-

[1] Για το σκοπό αυτό το WEP χρησιμοποιεί ένα συνηθισμένο κυκλικό κώδικα υπολοίπου (Cyclic Redundancy Code - CRC).

σης (Initialization Vector - IV), που παρέχει ο χρήστης καθώς και το μυστικό κλειδί. Παράγει μία ψευδοτυχαία ακολουθία από bits, τα οποία χρησιμοποιούνται για την κρυπτογράφηση του πακέτου δεδομένων και του ICV. Για να είναι εφικτή η αποκρυπτογράφηση από τον παραλήπτη, πρέπει το IV να σταλεί μαζί με το κρυπτογραφημένο πακέτο. Μόλις ο παραλήπτης αποκρυπτογραφήσει το πακέτο υπολογίζει ξανά το ICV από τα δεδομένα και συγκρίνει την τιμή με αυτή που περιείχε το πακέτο που παρέλαβε. Αν οι δύο τιμές ταυτίζονται, τότε θεωρείται ότι το πακέτο δεν έχει υποστεί πλαστογράφηση και είναι έγκυρο. Στη συνέχεια προωθείται στα ανώτερα στρώματα για επεξεργασία [4].

Για την προστασία των δεδομένων του χρήστη το WEP ορίζει τη χρήση ενός στατικού κλειδιού μήκους 40 bits και ενός IV μήκους 24 bits. Νεότερες εκδόσεις του WEP υποστηρίζουν μήκος κλειδιού 104 bits και μήκος IV 24 bits. Το κλειδί και το IV ενώνονται για να σχηματίσουν το κλειδί μήκους 64 bits, ή 128 bits αντίστοιχα που χρησιμοποιείται ως είσοδος για τον αλγόριθμο RC4. Ο αλγόριθμος RC4 παράγει μία ψευδοτυχαία ακολουθία από bits (KeyStream), που χρησιμοποιείται για την κρυπτογράφηση των δεδομένων και του ICV μέσω της πράξης XOR. Το IV αποστέλλεται μαζί με το κρυπτογραφημένο πακέτο. Η διαδικασία φαίνεται στο Σχήμα 2.1. Ο παραλήπτης παράγει την ίδια ψευδοτυχαία ακολουθία με τη βοήθεια του IV και αποκρυπτογραφεί τα δεδομένα μέσω της πράξης XOR.



Σχήμα 2.1 Κρυπτογράφηση με βάση το WEP.

Η υλοποίηση το RC4 είναι το κρίσιμο σημείο για την αποδοτικότητα του WEP, όσον αφορά την εμπιστευτικότητα των δεδομένων, αφού ο RC4 είναι στην

ουσία η μηχανή κρυπτογράφησης. Επειδή το μυστικό κλειδί είναι στατικό, το IV επεκτείνει το χρόνο ζωής του κλειδιού. Κάθε νέο IV έχει ως αποτέλεσμα τη δημιουργία διαφορετικής ψευδοτυχαίας ακολουθίας. Επομένως, ο αλγόριθμος RC4 εξαρτάται μόνο από το IV.

2.2 Αδυναμίες του WEP

Το WEP έχει σημαντικά εν γένει προβλήματα. Δεν ικανοποιεί σε καμία περίπτωση τους θεμελιώδεις στόχους της εμπιστευτικότητας σε επίπεδο αντίστοιχο με αυτό των ενσύρματων δικτύων. Επίσης, αποτυγχάνει να εξασφαλίσει την επικύρωση και την ακεραιότητα των δεδομένων.

Το WEP έχει δύο γενικούς περιορισμούς [5]. Πρώτον, η ίδια η χρήση του WEP είναι προαιρετική και πολλές πραγματικές εγκαταστάσεις δικτύων δεν το χρησιμοποιούν. Δεύτερον, το WEP χρησιμοποιεί ένα μοναδικό κοινό κλειδί, που είναι γνωστό σε όλους τους χρήστες του δικτύου και αυτό το κλειδί είναι συχνά αποθηκευμένο στη μνήμη των συσκευών που προσπελαύνεται εύκολα μέσω λογισμικού. Αν κάποια συσκευή χαθεί, ή κλαπεί τότε το μόνο μέτρο που μπορεί να ληφθεί είναι η αλλαγή του κοινού κλειδιού σε όλες τις συσκευές. Επειδή το WEP δεν περιλαμβάνει ένα πρωτόκολλο για τη διαχείριση των κλειδιών η διανομή του νέου κλειδιού στους χρήστες είναι εξαιρετικά χρονοβόρα διαδικασία. Όσο καλός και αν είναι ο αλγόριθμος κρυπτογράφησης που χρησιμοποιείται, τα μειονεκτήματα αυτά καθιστούν δύσκολη την επίτευξη των στόχων του WEP.

Στην πράξη το σημαντικότερο πρόβλημα του WEP είναι ότι το κλειδί της κρυπτογράφησης μπορεί να αποκαλυφθεί με μεθόδους κρυπτανάλυσης. Το WEP χρησιμοποιεί έναν κοινό αλγόριθμο κρυπτογράφησης ροής (stream cipher), τον RC4, αλλά με ειδικό τρόπο λειτουργίας. Ενώνει, όπως είδαμε στην §2.1, το κλειδί κρυπτογράφησης με το IV και χρησιμοποιεί τον αλγόριθμο RC4 για να εξάγει την ψευδοτυχαία ακολουθία που χρησιμοποιείται για την κρυπτογράφηση των δεδομένων και του ICV. Τον Αύγουστο του 2001 οι Fluhrer, Mantin και Shamir περιέγραψαν τον τρόπο για να επιτεθεί κάποιος σε αυτό το σύστημα ασφάλειας [6]. Απέδειξαν ότι κάποιος που απλά παρακολουθεί την ανταλλαγή μηνυμάτων στο δίκτυο και μπορεί να συλλέξει μερικά εκατομμύρια

κρυπτογραφημένα μηνύματα μπορεί να εξάγει το κλειδί της κρυπτογράφησης. Για να το επιτύχει αυτό θα πρέπει να είναι γνωστό το πρώτο byte των δεδομένων κάθε πακέτου, πριν την κρυπτογράφηση. Η αδυναμία έγκειται στις ιδιότητες της ψευδοτυχαίας ακολουθίας bits, που παράγει ο RC4, για συγκεκριμένες τιμές του IV.

Πολύ σύντομα οι Stubblefield, Ioannidis και Rubin υλοποίησαν πειραματικά την επίθεση και έδειξαν τον τρόπο με τον οποίο μπορεί κανείς να παραβιάσει πραγματικά συστήματα μέσα σε λίγες ώρες [7]. Από τότε αρκετοί άλλοι υλοποίησαν την επίθεση Fluhrer-Mantin-Shamir (FMS) και δημοσιοποίησαν εργαλεία για την αυτοματοποίηση της παραβίασης δικτύων που προστατεύονται από το WEP, ακόμα και αν βρίσκεται κανείς σε απόσταση μεγαλύτερη του ενός χιλιομέτρου από το στόχο.

Η επίθεση FMS έχει καταστροφικά αποτελέσματα για το WEP. Όταν αποκαλυφθεί το κλειδί χάνεται κάθε έννοια ασφάλειας. Αυτός που πραγματοποιεί την επίθεση μπορεί να αποκρυπτογραφήσει τα πακέτα που διακινούνται και να διαβάσει το περιεχόμενό τους, καταρρίπτοντας την εμπιστευτικότητα που παρέχει το WEP. Επιπλέον, μπορεί να πλαστογραφήσει καινούργια κρυπτογραφημένα πακέτα, τα οποία γίνονται αποδεκτά από τα AP και τους τελικούς παραλήπτες καταρρίπτοντας την επικύρωση και ακεραιότητα που υπόσχεται το WEP.

Η ομάδα εργασίας του προτύπου IEEE 802.11 είχε αρκετές ενδείξεις για τις αδυναμίες του WEP πολλούς μήνες πριν την πραγματοποίηση της επίθεσης FMS. Το 2000 παρατηρήθηκε ότι το μικρό μέγεθος του IV αυξάνει τον κίνδυνο να δημιουργηθεί ξανά η ίδια ψευδοτυχαία ακολουθία, λόγω της επανάληψης της τιμής του IV. Αφού το μήκος του IV είναι μόνο 24 bits υπάρχουν μόνο 2^{24} διαφορετικά IV, τα οποία εξαντλούνται μέσα σε λίγες ώρες αν αναλογιστεί κανείς την κίνηση των πακέτων στα ασύρματα δίκτυα και το γεγονός ότι υπάρχουν πολλοί χρήστες, που παράγουν ταυτόχρονα κάποια τιμή του IV. Με τον τρόπο αυτό κάποιος επιτήδειος μπορεί να διαβάσει το περιεχόμενο των κρυπτογραφημένων μηνυμάτων, χωρίς καν να γνωρίζει το κλειδί [8]. Το 2001 οι Borisov, Goldberg και Wagner αποκάλυψαν και άλλες επιθέσεις. Μεταξύ αυτών είναι το γεγονός ότι τα κρυπτογραφημένα μηνύματα μπορούν να τροπο-

ποιηθούν από κάποιο επιτήδειο χωρίς να υπάρχει ο κίνδυνος να γίνει αντιληπτός, καθώς επίσης ότι το πρωτόκολλο επικύρωσης των χρηστών παρακάμπτεται πολύ εύκολα [9]. Σύντομα η αδυναμία αυτή μετατράπηκε σε πρακτική επίθεση, με την οποία κάθε πακέτο μπορούσε να αποκρυπτογραφηθεί μέσα σε λίγες ώρες.

Συνοπτικά τα προβλήματα με το σχεδιασμό του WEP, τα οποία μεταφράζονται σε μειωμένα επίπεδα ασφάλειας, είναι τα παρακάτω:

1. Το μήκος του IV είναι μικρό, αφού τα 24 bits είναι λίγα για να εξασφαλιστεί η εμπιστευτικότητα των δεδομένων.

2. Η τιμή ελέγχου ακεραιότητας (ICV), που χρησιμοποιείται από το WEP, για την προστασία της ακεραιότητας των δεδομένων δεν παρέχει την απαιτούμενη ασφάλεια και δεν αποτρέπει την τροποποίηση των μηνυμάτων από κάποιο επιτήδειο.

3. Το WEP συνδυάζει το κλειδί της κρυπτογράφησης με το IV, με τέτοιο τρόπο ώστε να είναι δυνατές οι επιθέσεις που βασίζονται στην κρυπτανάλυση. Ως αποτέλεσμα, κάποιος παθητικός παρατηρητής του δικτύου μπορεί να αποκτήσει το κλειδί της κρυπτογράφησης χρησιμοποιώντας μερικά εκατομμύρια κρυπτογραφημένα πακέτα.

4. Δεν παρέχεται προστασία της ακεραιότητας των διευθύνσεων του αποστολέα και του παραλήπτη.

Σε κάθε περίπτωση οι σχεδιαστές του WEP χρησιμοποίησαν ένα πολύ ανθεκτικό αλγόριθμο κρυπτογράφησης, όπως είναι ο RC4, με λάθος τρόπο. Δημιούργησαν με αυτό τον τρόπο ένα πρωτόκολλο ασφάλειας, το οποίο δεν ικανοποιεί κανένα από τους σκοπούς για τους οποίους αναπτύχθηκε.

2.3 Περιγραφή του TKIP

Τα περισσότερα ασύρματα συστήματα IEEE 802.11 που υπάρχουν σήμερα υλοποιούν το WEP ως πρωτόκολλο ασφάλειας σε υλικό. Για την αντιμετώπιση των αδυναμιών του WEP, που αναλύθηκαν στην §2.2, η ομάδα εργασίας TGi, όρισε το πρωτόκολλο ασφάλειας TKIP. Η εγκατάστασή του προβλέπει

την αναβάθμιση του firmware και του λογισμικού της συσκευής. Το TKIP δημιουργήθηκε μόνο ως προσωρινή λύση και συμπεριλήφθηκε στο πρότυπο IEEE 802.11i. Η ανάγκη για την εγκατάσταση του πρωτόκολλου TKIP στα υπάρχοντα συστήματα επιβάλλει τους ακόλουθους περιορισμούς, όσον αφορά τη λειτουργία του:

- Η αναβάθμιση των συστημάτων πρέπει να γίνει μόνο μέσω του λογισμικού ή του firmware, ώστε να μη χρειάζεται η αντικατάσταση της ασύρματης συσκευής.
- Η υλοποίηση του WEP σε υλικό πρέπει να παραμείνει αμετάβλητη.
- Η μείωση της απόδοσης που επιφέρει το πρωτόκολλο TKIP, ώστε να παρέχει αυξημένα επίπεδα ασφάλειας, πρέπει να είναι η μικρότερη δυνατή.

Το TKIP αποτελείται από ένα σύνολο αλγορίθμων, οι οποίοι χρησιμοποιούν ως βάση το WEP, αλλά αντιμετωπίζουν τις αδυναμίες του και ταυτόχρονα ικανοποιούν τους προηγούμενους περιορισμούς. Το TKIP προσθέτει στο μηχανισμό ασφάλειας του WEP τα παρακάτω νέα χαρακτηριστικά:

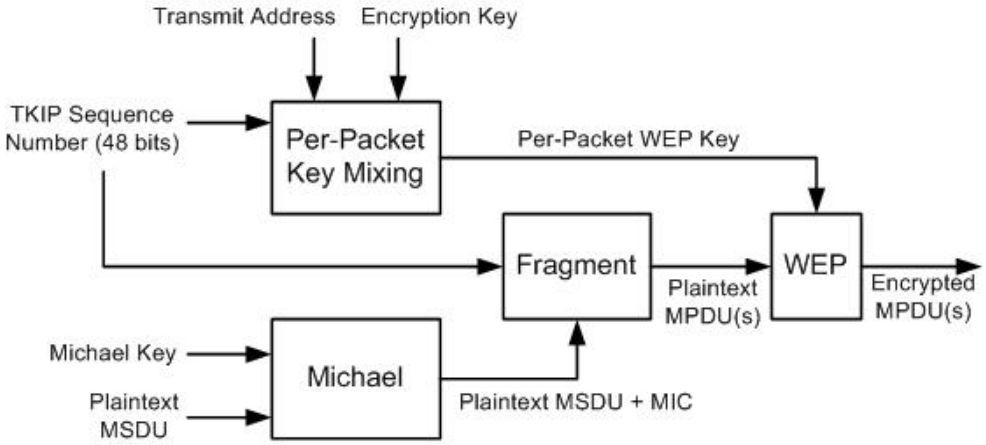
- Ένα κώδικα ακεραιότητας του μηνύματος (Message Integrity Code - MIC), που ονομάζεται Michael, για την αντιμετώπιση των πλαστογραφημένων μηνυμάτων.
- Μία μεθοδολογία αρίθμησης των πακέτων, για την αντιμετώπιση των επιθέσεων που βασίζονται στην επανάληψη πακέτων (replay attack).
- Μία συνάρτηση, η οποία για κάθε πακέτο δεδομένων επιδρά στο κλειδί της κρυπτογράφησης για την αντιμετώπιση των επιθέσεων FMS.

Το TKIP επιλύει το πρόβλημα της επαναχρησιμοποίησης ενός κλειδιού, επιβάλλοντας τη χρήση νέων ασυσχέτιστων μεταξύ τους κλειδιών κρυπτογράφησης. Τα κλειδιά αυτά παρέχονται μέσω του προτύπου διαχείρισης κλειδιών IEEE 802.1X.

Στο Σχήμα 2.2 φαίνεται η λειτουργία του TKIP, ως μία προκαταρκτική διαδικασία (front-end) για το WEP. Το TKIP εφαρμόζει τη συνάρτηση Key Mixing και στη συνέχεια κρυπτογραφεί βάση του WEP τα MPDU, που προκύπτουν

από τον τεμαχισμό των MSDU. Ο αλγόριθμος Michael για τον υπολογισμό του MIC εφαρμόζεται στα πακέτα MSDU.

Η διασύνδεση των διαφόρων τμημάτων του TKIP ενισχύει τελικά τη συνολική ασφάλεια. Το TKIP χρησιμοποιεί τον αλγόριθμο RC4 για την κρυπτογράφηση του MIC, γεγονός που μειώνει την πληροφορία σχετικά με το κλειδί του Michael, που αποκαλύπτεται σε ένα πιθανό εχθρό. Μία επίθεση που αλλάζει την τιμή του μετρητή για το πακέτο, αλλάζει επίσης το τελικό κλειδί της κρυπτογράφησης για το συγκεκριμένο πακέτο, αυξάνοντας την πιθανότητα το WEP ICV και το MIC να προκαλέσουν σφάλμα κατά την αποκρυπτογράφηση. Ο αλγόριθμος Michael καθιστά τις επιθέσεις που τροποποιούν τα κρυπτογραφημένα δεδομένα υπολογιστικά ασύμφορες. Εφόσον το MPDU προστατεύεται από τυχαία σφάλματα σε μεμονωμένα bits από το IEEE 802.11 FCS[2] και το WEP ICV, έγκυρη τιμή των FCS και ICV, αλλά μη έγκυρη τιμή του MIC σημαίνει ότι πιθανότατα το πακέτο είναι πλαστογραφημένο. Τέλος, εφόσον το MIC προστατεύει τις διευθύνσεις του αποστολέα και του παραλήπτη, τα πακέτα δεν μπορούν πλέον να κατευθυνθούν σε μη εξουσιοδοτημένους χρήστες ή να αλλάξει η διεύθυνση του αποστολέα.



Σχήμα 2.2 Ροή των δεδομένων κατά την επεξεργασία με το TKIP.

2.3.1 Ο αλγόριθμος Michael

Ένας αλγόριθμος υπολογισμού του MIC χρησιμοποιεί μία συνάρτηση του κλειδιού κρυπτογράφησης και των δεδομένων του πακέτου για την εξαγωγή

[2] Το πεδίο FCS (Frame Check Sequence) έχει μήκος 4 bytes και αποτελεί ένα κυκλικό κώδικα για την ανίχνευση λαθών (Cyclic Redundancy Code – CRC).

μίας τιμής που αποστέλλεται ως ετικέτα μαζί με τα δεδομένα στον παραλήπτη. Ο παραλήπτης υπολογίζει ξανά την τιμή του MIC και τη συγκρίνει με την τιμή που περιέχει η ετικέτα. Αν οι δύο τιμές ταυτίζονται τότε αποδέχεται το πακέτο ως αυθεντικό. Σε διαφορετική περίπτωση πρόκειται για πλαστογραφημένο πακέτο το οποίο απορρίπτεται.

Παραδείγματα αλγορίθμων για τον υπολογισμό του MIC, αποτελούν ο HMAC-SHA1, ο οποίος χρησιμοποιείται στο IPsec και ο DES-CBC-MAC, ο οποίος χρησιμοποιείται ευρύτατα σε οικονομικές εφαρμογές. Οι αλγόριθμοι αυτοί απαιτούν πολλούς υπολογιστικούς πόρους και η εκτέλεσή τους στα υπάρχοντα συστήματα δεν μπορεί να γίνει χωρίς σημαντική μείωση της απόδοσης. Για το λόγο αυτό η ομάδα εργασίας TGi δημιούργησε ένα νέο αλγόριθμο για τον υπολογισμό του MIC, ο οποίος υιοθετήθηκε τελικά από το πρωτόκολλο TKIP [10].

Ο αλγόριθμος Michael χρησιμοποιεί κλειδί μήκους 64 bits και χωρίζει τα πακέτα δεδομένων σε μπλοκ των 32 bits. Στη συνέχεια εφαρμόζει ολισθήσεις, προσθέσεις και την πράξη XOR για την επεξεργασία κάθε μπλοκ. Κατά την επεξεργασία χρησιμοποιούνται δύο καταχωρητές των 32 bits, οι οποίοι αναπαριστούν την έξοδο του αλγόριθμου μήκους 64 bits. Ο Michael περιορίζει το σύνολο εντολών (instruction set) που χρησιμοποιεί ώστε να μη μειωθεί σημαντικά η απόδοση του συστήματος. Το κόστος για την εκτέλεση του αλγόριθμου είναι περίπου 3.5 κύκλοι/byte σε ένα επεξεργαστή ARM7 και περίπου 5.5 κύκλοι/byte σε ένα επεξεργαστή i486. Είναι προφανές ότι αναμένεται μείωση της απόδοσης του συστήματος, αλλά δεν υπάρχουν εναλλακτικές λύσεις που να ικανοποιούν παράλληλα τις απαιτήσεις σε ασφάλεια.

Το επίπεδο ασφάλειας ενός αλγορίθμου υπολογισμού του MIC μετριέται σε bits. Αν το επίπεδο ασφάλειας είναι *s* bits, τότε εξ ορισμού κάποιος επιτήδειος μπορεί να παράγει ένα πλαστογραφημένο μήνυμα μετά από 2^{-s+1} πακέτα. Προκειμένου να ικανοποιηθούν οι στόχοι σχετικά με την απόδοση, ο Michael σχεδιάστηκε ώστε να παρέχει ασφάλεια περίπου 20 bits. Ο Michael είναι αρκετά αδύναμος, οπότε το TKIP λαμβάνει πρόσθετα μέτρα. Όποτε εντοπίζεται ένα σφάλμα κατά τον έλεγχο του MIC, το πρωτόκολλο επιβάλλει την αλλαγή του κλειδιού κρυπτογράφησης. Η αλλαγές κλειδιού περιορίζονται σε μία κάθε

λεπτό. Με αυτές τις προϋποθέσεις ο μέγιστος αναμενόμενος αριθμός πλαστογραφημένων μηνυμάτων, τα οποία γίνονται αποδεκτά ως αυθεντικά, περιορίζεται σε περίπου ένα κάθε χρόνο.

2.3.2 Αρίθμηση πακέτων

Τα πακέτα που επαναλαμβάνονται από κάποιο επιτήδειο, κατά τη διάρκεια μίας επίθεσης, δεν μπορούν εύκολα να εντοπιστούν με χρήση μόνο του αλγόριθμου MIC. Η συνηθισμένη μέθοδος για την αντιμετώπιση της επανάληψης πακέτων περιλαμβάνει την αποστολή της τιμής ενός μετρητή ακολουθίας (sequence counter) μαζί με κάθε πακέτο που περιέχει MIC. Ο μετρητής αρχικοποιείται κάθε φορά που αλλάζει το κλειδί της κρυπτογράφησης.

Το TKIP επεκτείνει τη δομή του WEP με τη χρησιμοποίηση μίας τιμής μετρητή μήκους 48 bits. Λόγω των περιορισμών που επιβάλλουν οι υπάρχουσες υλοποιήσεις, η τιμή αυτή συσχετίζεται με το κλειδί της κρυπτογράφησης, αντί με το κλειδί του Michael. Το πρωτόκολλο TKIP «ανακατεύει» την τιμή του μετρητή με το κλειδί της κρυπτογράφησης και κρυπτογραφεί το MIC και το ICV που υπολογίζει το WEP. Ο σχεδιασμός αυτός αντιστοιχεί τις επιθέσεις που βασίζονται στην επανάληψη πακέτων σε σφάλματα που παρουσιάζονται κατά τον υπολογισμό των τιμών του MIC και του ICV στον παραλήπτη.

2.3.3 Προεπεξεργασία του κλειδιού κρυπτογράφησης

Η ένωση του κλειδιού κρυπτογράφησης με το WEP IV μήκους 24 bits, δίνει τη δυνατότητα σε κάποιο επιτήδειο να ανακαλύψει το κλειδί της κρυπτογράφησης μέσω της επίθεσης FMS. Για την αντιμετώπιση του προβλήματος το TKIP εισάγει την προεπεξεργασία του κλειδιού κρυπτογράφησης, βάση μίας συνάρτησης. Η συνάρτηση αυτή παίρνει το κλειδί, τη MAC διεύθυνση του αποστολέα και την τιμή του μετρητή για το πακέτο ως εισόδους και παράγει στην έξοδο ένα νέο κλειδί κρυπτογράφησης του WEP για κάθε πακέτο (Per-Packet WEP Key). Για την ελαχιστοποίηση των υπολογιστικών απαιτήσεων, η συνάρτηση χωρίζεται σε δύο φάσεις.

Η πρώτη φάση χρησιμοποιεί ένα μη γραμμικό πίνακα αντικατάστασης (S-Box) και συνδυάζει το κλειδί, τη MAC διεύθυνση του αποστολέα και τα τέσσερα πιο σημαντικά bytes της τιμής του μετρητή. Στην έξοδο παράγεται μία ενδιάμεση τιμή. Η τιμή αυτή μπορεί να αποθηκευτεί προσωρινά και χρησιμοποιηθεί μέχρι και για 2^{16} πακέτα. Εφόσον λαμβάνεται υπόψη η διεύθυνση του αποστολέα, η συνάρτηση παράγει διαφορετική ενδιάμεση τιμή για κάθε συσκευή, ακόμα και αν χρησιμοποιείται το ίδιο κλειδί κρυπτογράφησης από όλες τις συσκευές.

Η δεύτερη φάση «ανακατεύει» την ενδιάμεση τιμή με τα δύο λιγότερο σημαντικά bytes της τιμής του μετρητή για την εξαγωγή του τελικού κλειδιού κρυπτογράφησης. Χρησιμοποιεί ένα μικρό αλγόριθμο κρυπτογράφησης για να κατανείμει ομοιόμορφα τα bytes της ενδιάμεσης τιμής και της τιμής του μετρητή στο κλειδί που προκύπτει. Η δεύτερη φάση απαλείφει τη συσχέτιση ανάμεσα στην τιμή του μετρητή και το νέο κλειδί, καθιστώντας άχρηστες τις επιθέσεις FMS. Το κόστος αυτής της προεπεξεργασίας είναι περίπου 150 κύκλοι/byte.

Δεν υπάρχει ποσοτική ανάλυση της ασφάλειας που προσφέρει αυτή η συνάρτηση. Παρόλα αυτά όλες οι κρυπτογραφικές μελέτες επιβεβαιώνουν ότι επιτυγχάνει τους στόχους της.

2.3.4 Τα κλειδιά του TKIP

Το TKIP απαιτεί δύο διακριτά κλειδιά: ένα κλειδί μήκους 128 bits, που χρησιμοποιείται για την εξαγωγή του κλειδιού κρυπτογράφησης κάθε πακέτου και ένα κλειδί μήκους 64 bits, για τον αλγόριθμο Michael. Το TKIP υποθέτει ότι αυτά τα κλειδιά είναι καινούργια για κάθε μετάδοση. Η ομάδα εργασίας TGi υιοθέτησε το πρότυπο IEEE 802.1X τόσο για την επικύρωση του χρήστη σε υψηλό επίπεδο, όσο και για τη διαχείριση κλειδιών. Μετά την αίτηση για εισαγωγή στο ασύρματο δίκτυο, το πρότυπο IEEE 802.1X επικυρώνει το χρήστη και δημιουργεί ένα καινούργιο κλειδί. Το κλειδί αυτό διανέμεται στη συνέχεια για χρήση από το πρωτόκολλο ασφάλειας. Η ασύρματη συσκευή και το AP χρησιμοποιούν το κλειδί αυτό για την εξαγωγή του ζεύγους κλειδιών που απαιτούνται από το TKIP. Οι λεπτομέρειες του μηχανισμού εξαγωγής των κλειδιών είναι πέρα από τους σκοπούς της παρούσας διπλωματικής.

3 Ο αλγόριθμος κρυπτογράφησης AES

3.1 Ιστορική ανασκόπηση

Στις 15 Μαΐου 1973, το Εθνικό Γραφείο Προτύπων[3] (National Bureau of Standards) έκανε μία εισήγηση για τα συστήματα κρυπτογράφησης στο Ομοσπονδιακό Αρχείο (Federal Register). Το γεγονός αυτό οδήγησε σταδιακά στην υιοθέτηση του Προτύπου Κρυπτογράφησης Δεδομένων (Data Encryption Standard - DES), το οποίο εξελίχθηκε στο πιο διαδεδομένο σύστημα κρυπτογράφησης παγκοσμίως. Το πρότυπο DES αναπτύχθηκε από την IBM και προέκυψε από την τροποποίηση ενός παλιότερου συστήματος γνωστό ως Lucifer. Το DES δημοσιεύτηκε για πρώτη φορά στο Ομοσπονδιακό Αρχείο στις 17 Μαρτίου 1975. Μετά από ένα σημαντικό χρονικό διάστημα δημόσιας συζήτησης, το DES υιοθετήθηκε ως πρότυπο για «μη απόρρητες» εφαρμογές στις 15 Ιανουαρίου 1977. Η αρχική εκτίμηση ήταν ότι το DES θα παρέμενε σε χρήση για 10-15 χρόνια. Παρόλα αυτά αποδείχτηκε πιο ανθεκτικό στο χρόνο. Το DES ανανεωνόταν και βελτιωνόταν περίπου κάθε πέντε χρόνια μετά την υιοθέτηση του, καθώς είχε αρχίσει η ανάπτυξη ενός νέου προτύπου για την αντικατάστασή του.

Η τελευταία ανανέωση του DES έγινε το 1999, αλλά ήδη από το 1997 το NIST είχε ξεκινήσει τη διαδικασία επιλογής ενός αντικαταστάτη του DES. Το πρότυπο που αντικατέστησε το DES ονομάστηκε Προχωρημένο Πρότυπο Κρυπτογράφησης (Advanced Encryption Standard - AES). Το επίσημο κάλεσμα για την κατάθεση των υποψήφιων αλγόριθμων έγινε στις 12 Σεπτεμβρίου 1997. Ήταν απαραίτητο το AES να έχει μέγεθος μπλοκ 128 bits και να υποστηρίζει μέγεθος κλειδιού 128, 192 και 256 bits. Ήταν επίσης αναγκαίο το AES να είναι διαθέσιμο παγκοσμίως.

Η τελική ημερομηνία κατάθεσης των υποψήφιων αλγορίθμων ήταν στις 15 Ιουνίου 1998. Από τα 21 συστήματα κρυπτογράφησης που κατατέθηκαν, τα 15 πληρούσαν όλες τις απαραίτητες προϋποθέσεις και έγιναν δεκτά ως υπο-

[3] Πλέον έχει μετονομαστεί σε Εθνικό Ινστιτούτο Προτύπων και Τεχνολογίας (National Institute of Standards and Technology - NIST).

ψήφια για την τελική επιλογή του AES. Το NIST ανακοίνωσε τους 15 υποψήφιους αλγόριθμους στο «First AES Candidate Conference» στις 20 Αυγούστου 1998. Το Μάρτιο του 1999 πραγματοποιήθηκε το «Second AES Candidate Conference» και τελικά τον Αύγουστο του 1999 πέντε υποψήφιοι αλγόριθμοι επιλέχθηκαν από το NIST για την τελική επιλογή. Οι αλγόριθμοι αυτοί ήταν οι: MARS, RC6, Rijndael, Serpent και Twofish.

Το «Third AES Candidate Conference» πραγματοποιήθηκε τον Απρίλιο του 2000. Στις 2 Οκτωβρίου 2000 ο Rijndael επιλέχθηκε τελικά ως το AES. Το Φεβρουάριο του 2001 το NIST ανακοίνωσε ότι ένα προσχέδιο του Ομοσπονδιακού Προτύπου Επεξεργασίας Πληροφορίας (Federal Information Processing Standard - FIPS) για το AES ήταν έτοιμο. Το AES υιοθετήθηκε ως πρότυπο στις 26 Νοεμβρίου 2001 και δημοσιεύτηκε ως FIPS-197 στο Ομοσπονδιακό Αρχείο στις 4 Δεκεμβρίου 2001 [11].

Η διαδικασία επιλογής για το AES ήταν αξιοσημείωτη για το παγκόσμιο ενδιαφέρον που προκάλεσε. Τα τρία συνέδρια για τα υποψήφια κρυπτογραφικά συστήματα, καθώς επίσης και οι επίσημες προσκλήσεις για σχολιασμό και παρατηρήσεις, έδωσαν την ευκαιρία για δημόσιο διάλογο και ανάλυση των υποψήφιων αλγορίθμων. Η διαδικασία αντιμετωπίστηκε θετικά από όλους όσοι ασχολήθηκαν με την επιλογή του πιο κατάλληλου αλγόριθμου για το AES. Το παγκόσμιο ενδιαφέρον αποδεικνύεται από το γεγονός ότι οι διαφορετικοί αλγόριθμοι που προτάθηκαν ανήκουν σε συγγραφείς από διάφορες χώρες όπως η Αυστραλία, το Βέλγιο, ο Καναδάς, η Γαλλία, η Γερμανία, η Αγγλία, οι ΗΠΑ, η Ιαπωνία, η Κορέα, το Ισραήλ, η Κόστα Ρίκα και η Νορβηγία. Ο αλγόριθμος Rijndael, που επιλέχθηκε τελικά για το AES, εφευρέθηκε από δύο Βέλγους ερευνητές, τους Daemen και Rijmen. Σημαντικό είναι επίσης το γεγονός ότι, σε αντίθεση με τις πρακτικές του παρελθόντος, το δεύτερο συνέδριο για τους υποψήφιους αλγόριθμους πραγματοποιήθηκε εκτός ΗΠΑ, στη Ρώμη.

Οι υποψήφιοι αλγόριθμοι για το AES αξιολογήθηκαν με βάση την αποδοτικότητά τους σύμφωνα με τρία κυρίως κριτήρια:

- ασφάλεια
- κόστος

- χαρακτηριστικά του αλγόριθμου και της υλοποίησης

Η ασφάλεια που θα παρείχε ο προτεινόμενος αλγόριθμος ήταν απολύτως αναγκαία και οποιοσδήποτε αλγόριθμος δεν παρείχε την απαιτούμενη ασφάλεια εξαιρέθηκε εξ αρχής από τη διαδικασία επιλογής. Το κόστος αναφέρεται στην υπολογιστική ισχύ, απαιτήσεις σε ταχύτητα και μνήμη, που απαιτούν οι διάφορες υλοποιήσεις κάθε αλγόριθμου, συμπεριλαμβανομένου υλοποιήσεων σε λογισμικό, υλικό και έξυπνες κάρτες (smart cards). Τα χαρακτηριστικά του αλγόριθμου και της υλοποίησης περιλαμβάνουν, μεταξύ άλλων, την απλότητα και την προσαρμοστικότητα του προτεινόμενου αλγόριθμου.

Τελικά, οι πέντε υποψήφιοι αλγόριθμοι παρείχαν όλοι το ίδιο επίπεδο ασφάλειας. Ο Rijndael επιλέχθηκε μετά το πέρας της διαδικασίας καθώς ο συνδυασμός ασφάλειας, απόδοσης, εύκολης υλοποίησης και προσαρμοστικότητας που παρείχε κρίθηκε ότι ήταν ανώτερος από αυτό των υπολοίπων.

3.2 Συμβολισμοί

3.2.1 Είσοδος και έξοδος

Η **είσοδος** και η **έξοδος** για τον αλγόριθμο AES αποτελείται από μία ακολουθία 128 bits. Αυτές οι ακολουθίες αναφέρονται μερικές φορές ως **μπλοκ** και ο αριθμός των bits που περιλαμβάνουν αναφέρεται ως το μήκος τους. Το **κλειδί κρυπτογράφησης** (Cipher Key) για τον αλγόριθμο AES είναι μία **ακολουθία από 128, 192 ή 256 bits**. Διαφορετικά μήκη εισόδου, εξόδου και κλειδιού κρυπτογράφησης δεν επιτρέπονται από το πρότυπο FIPS-197 [11].

Τα bits σε κάθε τέτοια ακολουθία αριθμούνται ξεκινώντας από το μηδέν και τελειώνοντας σε ένα λιγότερο από το μήκος της ακολουθίας (μήκος μπλοκ ή μήκος κλειδιού). Ο αριθμός i δίπλα σε κάθε bit αποτελεί το δείκτη της θέσης του και παίρνει τιμές στα διαστήματα $0 \le i < 128$, $0 \le i < 192$ ή $0 \le i < 256$, ανάλογα με το μήκος του μπλοκ και του κλειδιού.

3.2.2 Βασική μονάδα επεξεργασίας

Η βασική μονάδα για την επεξεργασία των δεδομένων στον αλγόριθμο AES είναι το **byte**, μία ακολουθία από οκτώ bits. Οι ακολουθίες από bits της εισόδου, της εξόδου και του κλειδιού κρυπτογράφησης, όπως περιγράφηκαν στην §3.2.1 αντιμετωπίζονται ως πίνακες από bytes (βλέπε §3.2.3). Για μία ακολουθία εισόδου, εξόδου ή κλειδιού κρυπτογράφησης που συμβολίζεται με *α*, τα bytes στον πίνακα που προκύπτει θα αναφέρονται χρησιμοποιώντας ένα από τους δύο συμβολισμούς α_n ή *α[n]*, όπου το *n* θα παίρνει τιμές από τα ακόλουθα διαστήματα:

Μήκος κλειδιού = 128 bits, $0 \le n < 16$ Μήκος μπλοκ = 128 bits, $0 \le n < 16$

Μήκος κλειδιού = 192 bits, $0 \le n < 24$

Μήκος κλειδιού = 256 bits, $0 \le n < 32$

Όλες οι τιμές των bytes στον αλγόριθμο AES αναπαριστώνται ως η ένωση των τιμών των bits σε αγκύλες με τη σειρά $\{b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0\}$. Αυτά τα bytes ερμηνεύονται ως στοιχεία ενός πεπερασμένου πεδίου χρησιμοποιώντας την πολυωνυμική αναπαράσταση:

$$b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0 = \sum_{i=0}^{7} b_i x_i \qquad (3.1)$$

Για παράδειγμα η ακολουθία *{01100011}* ορίζει το συγκεκριμένο στοιχείο πεπερασμένου πεδίου $x^6 + x^5 + x + 1$.

Είναι επίσης βολικό να δηλώνονται οι τιμές των bytes χρησιμοποιώντας τη δεκαεξαδική αναπαράσταση, όπου κάθε ομάδα από τέσσερα bits αποτελεί ένα χαρακτήρα όπως φαίνεται παρακάτω.

0000	0
0001	1
0010	2
0011	3

0100	4
0101	5
0110	6
0111	7

1000	8
1001	9
1010	a
1011	b

1100	c
1101	d
1110	e
1111	f

Επομένως το στοιχείο *{01100011}* μπορεί να αναπαρασταθεί ως *{63}*. Μερικές πράξεις πεπερασμένου πεδίου απαιτούν ένα επιπλέον bit (b_8) στα αριστερά του byte. Όπου αυτό το bit είναι απαραίτητο θα εμφανίζεται ως *'{01}'*

ακριβώς μπροστά από το byte. Για παράδειγμα μία ακολουθία από εννέα bits θα συμβολίζεται ως *{01}{1b}*.

3.2.3 Πίνακες από Bytes

Οι πίνακες από bytes αναπαριστώνται με τον ακόλουθο τρόπο:

$$\alpha_0\, \alpha_1\, \alpha_2 \ldots\, \alpha_{15}$$

Η διάταξη των bytes και των bits μέσα στα bytes προκύπτει από την ακολουθία εισόδου των 128 bits

$$input_0\, input_1\, input_2 \ldots\, input_{126}\, input_{127}$$

ως εξής:

$$\begin{aligned} \alpha_0 &= \{input_0, input_1, \ldots, input_7\} \\ \alpha_1 &= \{input_8, input_9, \ldots, input_{15}\} \\ &\vdots \\ \alpha_{15} &= \{input_{120}, input_{121}, \ldots, input_{127}\} \end{aligned}$$

Ο συμβολισμός αυτός μπορεί να επεκταθεί ώστε να περιλαμβάνει ακολουθίες μεγαλύτερου μήκους, δηλαδή 192 και 256 bits, έτσι ώστε γενικά

$$\alpha_n = \{input_{8n}, input_{8n+1}, \ldots, input_{8n+7}\} \qquad (3.2)$$

Συνοψίζοντας τις παραγράφους 3.2.2 και 3.2.3, στον πίνακα που ακολουθεί φαίνεται πως αριθμούνται τα bits μέσα σε κάθε byte.

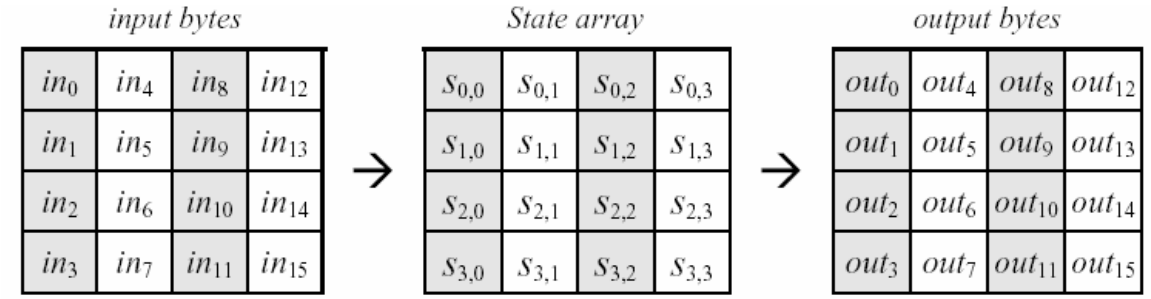
Bit εισόδου	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	…
Αριθμός byte	0								7								…
Αριθμός bit στο byte	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	…

3.2.4 Η Κατάσταση (State)

Εσωτερικά οι λειτουργίες του αλγορίθμου AES πραγματοποιούνται σε ένα πίνακα δύο διαστάσεων από bytes, που ονομάζεται **State**. Το State αποτελείται από τέσσερις σειρές από bytes, που κάθε μία περιέχει **Nb** bytes, όπου **Nb** είναι το μήκος του μπλοκ διαιρεμένο με 32. Στον πίνακα State, που συμβολίζεται με s, κάθε byte έχει δύο δείκτες. Ο δείκτης γραμμής r παίρνει τιμές στο διάστημα $0 \le r < 4$ και ο δείκτης στήλης c παίρνει τιμές στο διάστημα $0 \le c <$

Nb. Με τον τρόπο αυτό η αναφορά σε κάθε byte του πίνακα State γίνεται με το συμβολισμό $s_{r,c}$ ή *s[r,c]*. Για το πρότυπο αυτό ισχύει **Nb** = 4, δηλαδή $0 \le c < 4$.

Αρχικά, η είσοδος, δηλαδή ο πίνακας από bytes $in_0, in_1, \ldots, in_{15}$, αντιγράφεται στον πίνακα State, όπως φαίνεται στο Σχήμα 3.1. Οι μετασχηματισμοί που προβλέπει ο αλγόριθμος εφαρμόζονται στον πίνακα State, η τελική τιμή του οποίου αντιγράφεται στην έξοδο, δηλαδή τον πίνακα από bytes $out_0, out_1, \ldots, out_{15}$.



Σχήμα 3.1 Η χρήση του πίνακα State σε σχέση με την είσοδο και την έξοδο.

Επομένως, με την εκκίνηση του αλγόριθμου, είτε για την κρυπτογράφηση είτε για την αποκρυπτογράφηση, ο πίνακας εισόδου *in* αντιγράφεται στον πίνακα State σύμφωνα με τη σχέση

$$s[r,c] = in[r+4c],\ 0 \le r < 4 \text{ και } 0 \le c < Nb \qquad (3.3)$$

και κατά τον τερματισμό του αλγόριθμου ο πίνακας State αντιγράφεται στον πίνακα εξόδου *out* σύμφωνα με τη σχέση

$$out[r+4c] = s[r,c],\ 0 \le r < 4 \text{ και } 0 \le c < Nb \qquad (3.4)$$

3.2.5 Το State ως ένας πίνακας στηλών

Τα τέσσερα bytes σε κάθε στήλη του πίνακα State σχηματίζουν λέξεις των 32 bits, όπου ο αριθμός της γραμμής r είναι ο δείκτης για τα τέσσερα bytes μέσα στη λέξη. Με τον τρόπο αυτό ο πίνακας State μπορεί να αντιμετωπιστεί ως ένας πίνακας μίας διάστασης από λέξεις των 32 bits (στήλες), w_0, w_1, w_2, w_3, όπου ο αριθμός στήλης c αποτελεί το δείκτη μέσα σε αυτό τον πίνακα. Επομένως, για το παράδειγμα στο Σχήμα 3.1, ο πίνακας State θεωρείται ένας πίνακας τεσσάρων λέξεων με τον εξής τρόπο

$$w_0 = s_{0,0}\ s_{1,0}\ s_{2,0}\ s_{3,0} \qquad w_2 = s_{0,2}\ s_{1,2}\ s_{2,2}\ s_{3,2}$$
$$w_1 = s_{0,1}\ s_{1,1}\ s_{2,1}\ s_{3,1} \qquad w_3 = s_{0,3}\ s_{1,3}\ s_{2,3}\ s_{3,3} \tag{3.5}$$

3.3 Μαθηματικό υπόβαθρο

Όλα τα bytes στον αλγόριθμο AES αντιμετωπίζονται ως στοιχεία πεπερασμένου πεδίου σύμφωνα με την αναπαράσταση που εισάγαμε στην §3.2.2. Τα στοιχεία ενός πεπερασμένου πεδίου μπορούν να προστεθούν και να πολλαπλασιαστούν, αλλά οι πράξεις αυτές διαφέρουν από τις πράξεις που χρησιμοποιούνται για τους πραγματικούς αριθμούς. Στην παράγραφο αυτή εισάγουμε τις βασικές μαθηματικές έννοιες, που είναι αναγκαίες για τον ορισμό του αλγόριθμου AES.

3.3.1 Πρόσθεση

Η πρόσθεση δύο στοιχείων πεπερασμένου πεδίου επιτυγχάνεται «προσθέτοντας» τους συντελεστές για τις αντίστοιχες δυνάμεις στα πολυώνυμα των δύο στοιχείων. Η πρόσθεση επιτυγχάνεται με την πράξη XOR (συμβολίζεται με $\oplus$), δηλαδή modulo 2, έτσι ώστε $0 \oplus 0 = 0$, $1 \oplus 0 = 1$ και $1 \oplus 1 = 0$. Επομένως, η αφαίρεση των πολυωνύμων είναι ίδια ακριβώς με την πρόσθεση πολυωνύμων.

Εναλλακτικά, η πρόσθεση στοιχείων πεπερασμένου πεδίου μπορεί να περιγραφεί ως η modulo 2 πρόσθεση των αντίστοιχων bits, που αποτελούν το byte. Για δύο bytes $\{a_7a_6a_5a_4a_3a_2a_1a_0\}$ και $\{b_7b_6b_5b_4b_3b_2b_1b_0\}$ το άθροισμα είναι $\{c_7c_6c_5c_4c_3c_2c_1c_0\}$, όπου κάθε $c_i = a_i \oplus b_i$.

Για παράδειγμα οι ακόλουθες παραστάσεις είναι ισοδύναμες μεταξύ τους:

$(x^6 + x^4 + x^2 + x + 1) + (x^7 + x + 1) = x^7 + x^6 + x^4 + x^2$ Πολυωνυμική αναπαράσταση

$\{01010111\} \oplus \{10000011\} = \{11010100\}$ Δυαδική αναπαράσταση

$\{57\} \oplus \{83\} = \{d4\}$ Δεκαεξαδική αναπαράσταση

3.3.2 Πολλαπλασιασμός

Στην πολυωνυμική αναπαράσταση, ο πολλαπλασιασμός στο πεδίο $GF(2^8)$, που συμβολίζεται με $\bullet$, αντιστοιχεί στον πολλαπλασιασμό των πολυωνύμων modulo ένα πολυώνυμο βαθμού οκτώ, του οποίου οι μοναδικοί διαιρέτες είναι ο εαυτός του και το ένα (irreducible polynomial). Για τον αλγόριθμο AES αυτό το πολυώνυμο είναι το

$$m(x) = x^8 + x^4 + x^3 + x + 1 \qquad (3.6)$$

δηλαδή το {01}{1b} σε δεκαεξαδική αναπαράσταση.

Για παράδειγμα, {57} $\bullet$ {83} = {c1} καθώς

$$\begin{aligned}(x^6 + x^4 + x^2 + x + 1)(x^7 + x + 1) &= x^{13} + x^{11} + x^9 + x^8 + x^7 + \\ &\quad x^7 + x^5 + x^3 + x^2 + x + \\ &\quad x^6 + x^4 + x^2 + x + 1 \\ &= x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1\end{aligned}$$

και ισχύει

$$\begin{aligned}x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1 \text{ modulo } (x^8 + x^4 + x^3 + x + 1) \\ = x^7 + x^6 + 1\end{aligned}$$

Η modulo μείωση με το πολυώνυμο m(x) εξασφαλίζει ότι το αποτέλεσμα θα είναι ένα δυαδικό πολυώνυμο βαθμού μικρότερου από οκτώ και επομένως μπορεί να αναπαρασταθεί από ένα byte. Αντίθετα από την πρόσθεση, δεν υπάρχει απλή πράξη στο επίπεδο του byte που να αντιστοιχεί σε αυτό τον πολλαπλασιασμό.

Ο πολλαπλασιασμός που ορίστηκε παραπάνω διαθέτει την επιμεριστική ιδιότητα και το στοιχείο {01} είναι το ταυτοτικό στοιχείο. Για οποιοδήποτε μη μηδενικό δυαδικό πολυώνυμο b(x) βαθμού μικρότερου από οκτώ, το πολλαπλασιαστικό αντίστροφο του b(x), το οποίο συμβολίζεται με $b^{-1}(x)$, μπορεί να βρεθεί χρησιμοποιώντας τον αλγόριθμο του Ευκλείδη για τον υπολογισμό των πολυωνύμων a(x) και c(x) έτσι ώστε

$$b(x)a(x) + m(x)c(x) = 1 \qquad (3.7)$$

Επομένως, $a(x) \bullet b(x) \bmod m(x) = 1$, που σημαίνει ότι

$$b^{-1}(x) = a(x) \bmod m(x) \qquad (3.8)$$

Επιπλέον, για οποιαδήποτε πολυώνυμα a(x), b(x) και c(x) στο πεδίο ισχύει ότι

$$a(x) \bullet (b(x) + c(x)) = a(x) \bullet b(x) + a(x) \bullet c(x)$$

Προκύπτει ότι το σύνολο των 256 δυνατών τιμών των bytes, με τη χρήση του XOR για την πρόσθεση και τον πολλαπλασιασμό όπως ορίστηκε παραπάνω, έχει τη δομή ενός πεπερασμένου πεδίου $GF(2^8)$.

3.3.3 Πολλαπλασιασμός με το x

Πολλαπλασιάζοντας το πολυώνυμο της εξίσωσης (3.1) με το πολυώνυμο x προκύπτει

$$b_7x^8 + b_6x^7 + b_5x^6 + b_4x^5 + b_3x^4 + b_2x^3 + b_1x^2 + b_0x \qquad (3.9)$$

Το αποτέλεσμα $x \bullet b(x)$ προκύπτει μειώνοντας το παραπάνω πολυώνυμο modulo m(x), όπως αυτό ορίστηκε στην εξίσωση (3.6). Αν ισχύει $b_7 = 0$, τότε το αποτέλεσμα είναι ήδη μειωμένο, δηλαδή ο βαθμός του πολυωνύμου είναι μικρότερος του οκτώ. Αν ισχύει $b_7 = 1$, η μείωση επιτυγχάνεται με την αφαίρεση του πολυωνύμου m(x), δηλαδή παίρνοντας το αποτέλεσμα της πράξης XOR για τους αντίστοιχους συντελεστές. Επομένως, ο πολλαπλασιασμός με το x, δηλαδή το {00000010} ή το {02} σε δυαδική και δεκαεξαδική αναπαράσταση αντίστοιχα, υλοποιείται σε επίπεδο byte ως μία αριστερή ολίσθηση και μία υπό συνθήκη πράξη XOR ανάμεσα στα αντίστοιχα bits με το {1b}. Αυτή η πράξη σε επίπεδο byte ονομάζεται xtime( ). Ο πολλαπλασιασμός με μεγαλύτερες δυνάμεις του x υλοποιείται με την επαναληπτική εφαρμογή της xtime( ). Προσθέτοντας τα ενδιάμεσα αποτελέσματα, μπορούμε να επιτύχουμε τον πολλαπλασιασμό με οποιαδήποτε σταθερά.

Για παράδειγμα, $\{57\} \bullet \{13\} = \{fe\}$ επειδή

$$\{57\} \bullet \{02\} = \text{xtime}(\{57\}) = \{ae\}$$

$$\{57\} \bullet \{04\} = \text{xtime}(\{ae\}) = \{47\}$$

$$\{57\} \bullet \{08\} = \text{xtime}(\{47\}) = \{8e\}$$

$$\{57\} \bullet \{10\} = \text{xtime}(\{8e\}) = \{07\}$$

και επομένως

$$\{57\} \bullet \{13\} = \{57\} \bullet (\{01\} \oplus \{02\} \oplus \{10\}) = \{57\} \oplus \{ae\} \oplus \{07\} = \{fe\}$$

3.3.4 Πολυώνυμα με συντελεστές στο GF(2^8)

Τα πολυώνυμα που έχουν τέσσερις όρους, με συντελεστές που είναι στοιχεία πεπερασμένου πεδίου, ορίζονται ως εξής:

$$a(x) = a_3x^3 + a_2x^2 + a_1x + a_0 \qquad (3.10)$$

το οποίο θα συμβολίζεται ως μία λέξη με τη μορφή [a_0, a_1, a_2, a_3]. Παρατηρείστε ότι τα πολυώνυμα σε αυτή την παράγραφο συμπεριφέρονται με διαφορετικό τρόπο από τα πολυώνυμα που χρησιμοποιήθηκαν στον ορισμό των στοιχείων πεπερασμένου πεδίου, παρόλο που και οι δύο τύποι πολυωνύμων χρησιμοποιούν την ίδια μεταβλητή x. Οι συντελεστές σε αυτή την παράγραφο είναι οι ίδιοι στοιχεία πεπερασμένου πεδίου, δηλαδή bytes αντί bits. Επιπλέον, ο πολλαπλασιασμός των πολυωνύμων με τέσσερις όρους χρησιμοποιεί ένα διαφορετικό πολυώνυμο για τη μείωση, το οποίο ορίζεται στη συνέχεια της παραγράφου.

Ακολουθεί ένα παράδειγμα για να γίνει κατανοητός ο τρόπος με τον οποίο πραγματοποιείται η πρόσθεση και ο πολλαπλασιασμός. Έστω,

$$b(x) = b_3x^3 + b_2x^2 + b_1x + b_0 \qquad (3.11)$$

ένα δεύτερο πολυώνυμο με τέσσερις όρους. Η πρόσθεση υλοποιείται προσθέτοντας τους συντελεστές πεπερασμένου πεδίου των όμοιων δυνάμεων του x. Η πρόσθεση αυτή αντιστοιχεί σε πράξη XOR ανάμεσα στα αντίστοιχα bytes σε κάθε μία λέξη.

Επομένως, χρησιμοποιώντας τις εξισώσεις (3.10) και (3.11)

$$a(x) + b(x) = (a_3 \oplus b_3)x^3 + (a_2 \oplus b_2)x^2 + (a_1 \oplus b_1)x + (a_0 \oplus b_0) \qquad (3.12)$$

Ο πολλαπλασιασμός επιτυγχάνεται σε δύο βήματα. Στο πρώτο βήμα, το πολυωνυμικό γινόμενο $c(x) = a(x) \bullet b(x)$ αναπτύσσεται αλγεβρικά και οι όμοιες δυνάμεις συγκεντρώνονται ώστε να προκύψει

$$c(x) = c_6x^6 + c_5x^5 + c_4x^4 + c_3x^3 + c_2x^2 + c_1x + c_0 \qquad (3.13)$$

όπου

$$\begin{aligned}
c_0 &= a_0 \bullet b_0 & c_4 &= a_3 \bullet b_1 \oplus a_2 \bullet b_2 \oplus a_1 \bullet b_3 \\
c_1 &= a_1 \bullet b_0 \oplus a_0 \bullet b_1 & c_5 &= a_3 \bullet b_2 \oplus a_2 \bullet b_3 \\
c_2 &= a_2 \bullet b_0 \oplus a_1 \bullet b_1 \oplus a_0 \bullet b_2 & c_6 &= a_3 \bullet b_3 \\
c_3 &= a_3 \bullet b_0 \oplus a_2 \bullet b_1 \oplus a_1 \bullet b_2 \oplus a_0 \bullet b_3
\end{aligned} \qquad (3.14)$$

Το πολυώνυμο c(x) δεν αναπαριστά μία λέξη τεσσάρων bytes. Επομένως, το δεύτερο βήμα του πολλαπλασιασμού είναι η μείωση του c(x) modulo ένα πολυώνυμο βαθμού τέσσερα. Με τον τρόπο αυτό, το c(x) μειώνεται σε ένα πολυώνυμο με βαθμό μικρότερο του τέσσερα. Για τον αλγόριθμο AES αυτό επιτυγχάνεται με το πολυώνυμο $x^4 + 1$, έτσι ώστε

$$x^i \bmod (x^4 + 1) = x^{i \bmod 4} \qquad (3.15)$$

Το modular γινόμενο των πολυωνύμων a(x) και b(x), που συμβολίζεται ως $a(x) \otimes b(x)$ δίνεται από το πολυώνυμο τεσσάρων όρων d(x), το οποίο ορίζεται ως εξής:

$$d(x) = d_3x^3 + d_2x^2 + d_1x + d_0 \qquad (3.16)$$

όπου

$$d_0 = (a_0 \bullet b_0) \oplus (a_3 \bullet b_1) \oplus (a_2 \bullet b_2) \oplus (a_1 \bullet b_3)$$

$$d_1 = (a_1 \bullet b_0) \oplus (a_0 \bullet b_1) \oplus (a_3 \bullet b_2) \oplus (a_2 \bullet b_3)$$

$$d_2 = (a_2 \bullet b_0) \oplus (a_1 \bullet b_1) \oplus (a_0 \bullet b_2) \oplus (a_3 \bullet b_3) \qquad (3.17)$$

$$d_3 = (a_3 \bullet b_0) \oplus (a_2 \bullet b_1) \oplus (a_1 \bullet b_2) \oplus (a_0 \bullet b_3)$$

Όταν το a(x) είναι σταθερό πολυώνυμο η πράξη που ορίζεται από την εξίσωση (3.16) μπορεί να γραφεί με μορφή πινάκων με τον ακόλουθο τρόπο:

$$\begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} a_0 & a_3 & a_2 & a_1 \\ a_1 & a_0 & a_3 & a_2 \\ a_2 & a_1 & a_0 & a_3 \\ a_3 & a_2 & a_1 & a_0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} \qquad (3.18)$$

Επειδή το πολυώνυμο $x^4 + 1$ δεν είναι reducible πολυώνυμο στο GF(2^8), ο πολλαπλασιασμός με ένα σταθερό πολυώνυμο τεσσάρων όρων δεν είναι απαραίτητα αντιστρέψιμη πράξη. Παρόλα αυτά ο αλγόριθμος AES καθορίζει ένα σταθερό πολυώνυμο τεσσάρων όρων το οποίο έχει αντίστροφο (βλέπε §3.5.3). Το πολυώνυμο αυτό καθώς και το αντίστροφό του είναι τα ακόλουθα:

$$a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\} \qquad (3.19)$$

$$a^{-1}(x) = \{0b\}x^3 + \{0d\}x^2 + \{09\}x + \{0e\} \qquad (3.20)$$

Ένα άλλο πολυώνυμο που χρησιμοποιείται στον αλγόριθμο AES (βλέπε τη συνάρτηση RotWord( ) στην §3.6) έχει $a_0 = a_1 = a_2 = \{00\}$ και $a_3 = \{01\}$, που αντιστοιχεί στο πολυώνυμο x^3. Εξετάζοντας την εξίσωση (3.18) παρατηρούμε ότι το αποτέλεσμά της είναι ο σχηματισμός της λέξης εξόδου από την περιστροφή των bytes της λέξης εισόδου. Αυτό σημαίνει ότι η λέξη [b_0, b_1, b_2, b_3] μετασχηματίζεται στη λέξη [b_1, b_2, b_3, b_0].

3.4 Προδιαγραφές του αλγόριθμου AES

Για τον αλγόριθμο AES, το μήκος του μπλοκ εισόδου, του μπλοκ εξόδου και του State είναι 128 bits. Αυτό αναπαρίσταται από την τιμή Nb = 4, η οποία αντιστοιχεί στον αριθμό των λέξεων 32 bits, δηλαδή στον αριθμό των στηλών, του πίνακα State.

Για τον αλγόριθμο AES, το μήκος του κλειδιού κρυπτογράφησης K, είναι 128, 192 ή 256 bits. Το μήκος του κλειδιού αναπαρίσταται από την τιμή Nk = 4, 6 ή 8, η οποία αντιστοιχεί στον αριθμό των λέξεων 32 bits, δηλαδή στον αριθμό των στηλών, του κλειδιού.

Για τον αλγόριθμο AES, ο αριθμός των επαναλήψεων που πραγματοποιούνται κατά την εκτέλεση του αλγόριθμου εξαρτάται από το μέγεθος του κλειδιού. Ο αριθμός των επαναλήψεων αναπαρίσταται από την τιμή Nr = 10, 12 ή 14, αντίστοιχα για κάθε τιμή του Nk.

Οι μόνοι συνδυασμοί τιμών για το κλειδί κρυπτογράφησης, το μέγεθος του μπλοκ επεξεργασίας και του αριθμού των επαναλήψεων του AES δίνονται στον πίνακα που ακολουθεί.

	Μήκος κλειδιού (Nk λέξεις)	**Μέγεθος μπλοκ (Nb λέξεις)**	**Αριθμός επαναλήψεων (Nr)**
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

Τόσο για την κρυπτογράφηση, όσο και για την αποκρυπτογράφηση, ο αλγόριθμος AES χρησιμοποιεί σε κάθε επανάληψη μία συνάρτηση που αποτελείται από τέσσερις διαφορετικούς μετασχηματισμούς που εφαρμόζονται στο μπλοκ εισόδου σε επίπεδο byte. Οι μετασχηματισμοί αυτοί είναι οι παρακάτω:

1. Αντικατάσταση κάθε byte χρησιμοποιώντας ένα πίνακα αντικατάστασης, που ονομάζεται S-box.
2. Ολίσθηση των γραμμών του πίνακα State, κατά διαφορετικό αριθμό θέσεων ανάλογα με τη γραμμή.
3. Ανάμειξη των δεδομένων μέσα σε κάθε στήλη του πίνακα State.

4. Πρόσθεση ενός κλειδιού, το οποίο είναι διαφορετικό για κάθε επανάληψη, με τον πίνακα State.

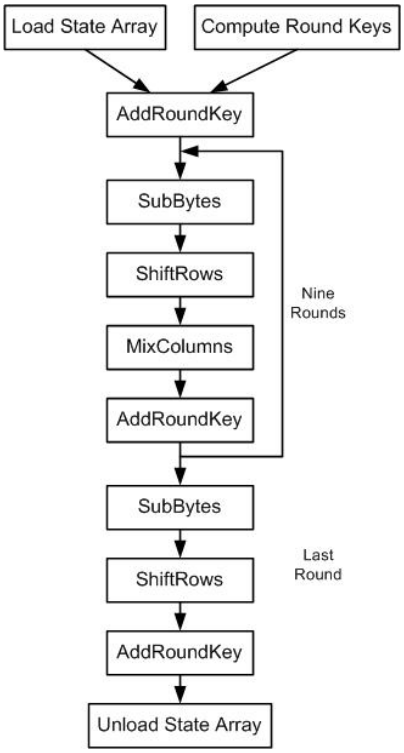
Αυτοί οι μετασχηματισμοί περιγράφονται αναλυτικά στις §3.5.1 – 3.5.4. Για τους σκοπούς της συγκεκριμένης εργασίας επικεντρώνουμε το ενδιαφέρον μας μόνο στην ανάλυση των μετασχηματισμών κατά την κρυπτογράφηση και όχι των αντίστροφών τους, που εφαρμόζονται κατά την αποκρυπτογράφηση. Ο λόγος που συμβαίνει αυτό θα γίνει κατανοητός κατά την ανάλυση του πρωτοκόλλου CCMP, για το οποίο τόσο για την κρυπτογράφηση, όσο και για την αποκρυπτογράφηση απαιτούνται μόνο οι ευθείς μετασχηματισμοί.

Ο προγραμματισμός των κλειδιών, δηλαδή η εξαγωγή του κατάλληλου κλειδιού για κάθε επανάληψη, περιγράφεται αναλυτικά στην §3.6.

3.5 Κρυπτογράφηση

Στην αρχή της διαδικασίας, το μπλοκ εισόδου αντιγράφεται στον πίνακα State με τον τρόπο που αναλύθηκε στην §3.2.4. Μετά την αρχική πρόσθεση ενός κλειδιού, ο πίνακας State μετασχηματίζεται υλοποιώντας τη συνάρτηση που αναλύθηκε παραπάνω για 10, 12 ή 14 επαναλήψεις, ανάλογα με το μήκος του κλειδιού. Η τελευταία επανάληψη διαφέρει ελαφρώς από τις Nr – 1 προηγούμενες επαναλήψεις. Ο τελικός πίνακας State που προκύπτει αντιγράφεται στην έξοδο με τον τρόπο που αναλύθηκε στην §3.2.4. Η διαδικασία, στην περίπτωση του RSN όπου το κλειδί κρυπτογράφησης έχει μήκος 128 bits (Nr = 10), συνοψίζεται στο διάγραμμα που φαίνεται στο Σχήμα 3.2.

Η συνάρτηση που υλοποιεί την κρυπτογράφηση δέχεται ως παράμετρο το αντίστοιχο κλειδί για κάθε επανάληψη, το οποίο αποτελείται από ένα μονοδιάστατο πίνακα που περιέχει λέξεις των τεσσάρων bytes. Τα κλειδιά εξάγονται χρησιμοποιώντας τη ρουτίνα KeyExpansion, που περιγράφεται στην §3.6.



Σχήμα 3.2 Κρυπτογράφηση με τον αλγόριθμο AES για δίκτυα RSN.

Η κρυπτογράφηση, ανεξάρτητα από το μήκος του κλειδιού κρυπτογράφησης, περιγράφεται σε μορφή ψευδοκώδικα παρακάτω. Οι επιμέρους μετασχηματισμοί, SubBytes( ), ShiftRows( ), MixColumns( ) και AddRoundKey( ) εφαρμόζονται στον πίνακα State και περιγράφονται στις επόμενες υποπαραγράφους. Ο πίνακας w[ ] περιέχει όλα τα απαραίτητα κλειδιά.

```
Cipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
begin
    byte state[4,Nb]

    state = in

    AddRoundKey(state, w[0, Nb-1])

    for round = 1 step 1 to Nr–1
        SubBytes(state)
        ShiftRows(state)
        MixColumns(state)
        AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
    end for

    SubBytes(state)
    ShiftRows(state)
    AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])

    out = state
end
```

3.5.1 Ο μετασχηματισμός SubBytes( )

Ο μετασχηματισμός SubBytes( ) είναι μία μη γραμμική αντικατάσταση, η οποία εφαρμόζεται ανεξάρτητα σε κάθε byte του πίνακα State χρησιμοποιώντας ένα πίνακα αντικατάστασης (S-box). Αυτό το S-box, που φαίνεται στο Σχήμα 3.4, είναι αντιστρέψιμο και κατασκευάζεται από το συνδυασμό δύο μετασχηματισμών:

1. Παίρνουμε το πολλαπλασιαστικό αντίστροφο στο πεπερασμένο πεδίο GF(2^8), όπως περιγράφηκε στην §3.3.2. Το αντίστροφο του στοιχείου {00} είναι ο εαυτός του
2. Εφαρμόζουμε τον ακόλουθο affine μετασχηματισμό στο πεδίο GF(2):

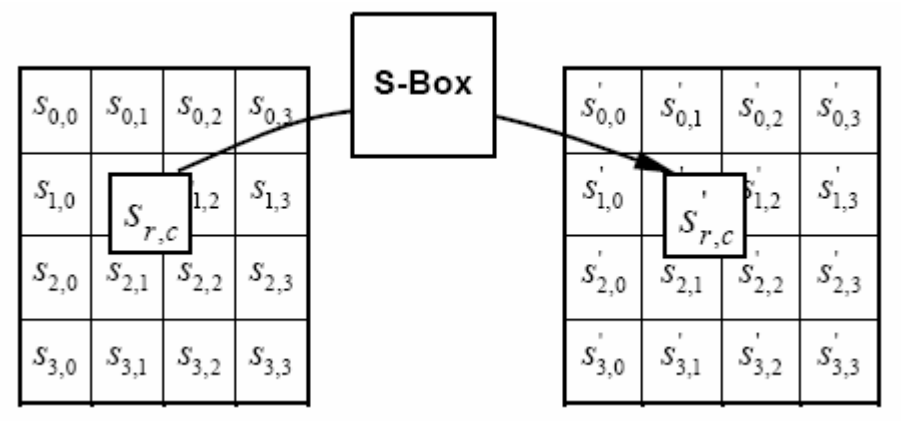
$$b_i` = b_i \oplus b_{(i+4)\bmod 8} \oplus b_{(i+5)\bmod 8} \oplus b_{(i+6)\bmod 8} \oplus b_{(i+7)\bmod 8} \oplus c_i \qquad (3.21)$$

για $0 \le i < 8$, όπου b_i είναι το i-στο bit του byte και c_i είναι το i-στο bit του byte c, που έχει τιμή {63} ή {01100011}. Από εδώ και στο εξής ο τόνος πάνω σε μία μεταβλητή, για παράδειγμα b`, δείχνει ότι η μεταβλητή θα ενημερωθεί με την τιμή που βρίσκεται στο δεξί μέλος της παράστασης.

Σε μορφή πινάκων ο affine μετασχηματισμός μπορεί να εκφραστεί με τον παρακάτω τρόπο:

$$\begin{bmatrix} b_0` \\ b_1` \\ b_2` \\ b_3` \\ b_4` \\ b_5` \\ b_6` \\ b_7` \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \qquad (3.22)$$

Στο Σχήμα 3.3 φαίνεται το αποτέλεσμα του μετασχηματισμού SubBytes( ) στον πίνακα State.



Σχήμα 3.3 Το S-box εφαρμόζεται σε κάθε byte του State.

Το S-box που χρησιμοποιείται στο μετασχηματισμό SubBytes( ) παρουσιάζεται σε δεκαεξαδική μορφή στο Σχήμα 3.4. Για παράδειγμα, αν $s_{1,1}$ = {53}, τότε η τιμή που θα την αντικαταστήσει προκύπτει από την τομή της γραμμής με δείκτη '5' και της στήλης με δείκτη '3'. Αυτό έχει ως αποτέλεσμα το $s'_{1,1}$ = {ed}.

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Σχήμα 3.4 S-box: οι τιμές αντικατάστασης για το byte xy.

3.5.2 Ο μετασχηματισμός ShiftRows( )

Κατά το μετασχηματισμό ShiftRows( ), τα bytes στις τρεις τελευταίες γραμμές του πίνακα State ολισθαίνουν κυκλικά κατά διαφορετικό αριθμό θέσεων. Η πρώτη γραμμή r = 0 δεν ολισθαίνει.

Πιο συγκεκριμένα, ο μετασχηματισμός ShiftRows( ) επιδρά στον πίνακα State με τον ακόλουθο τρόπο:

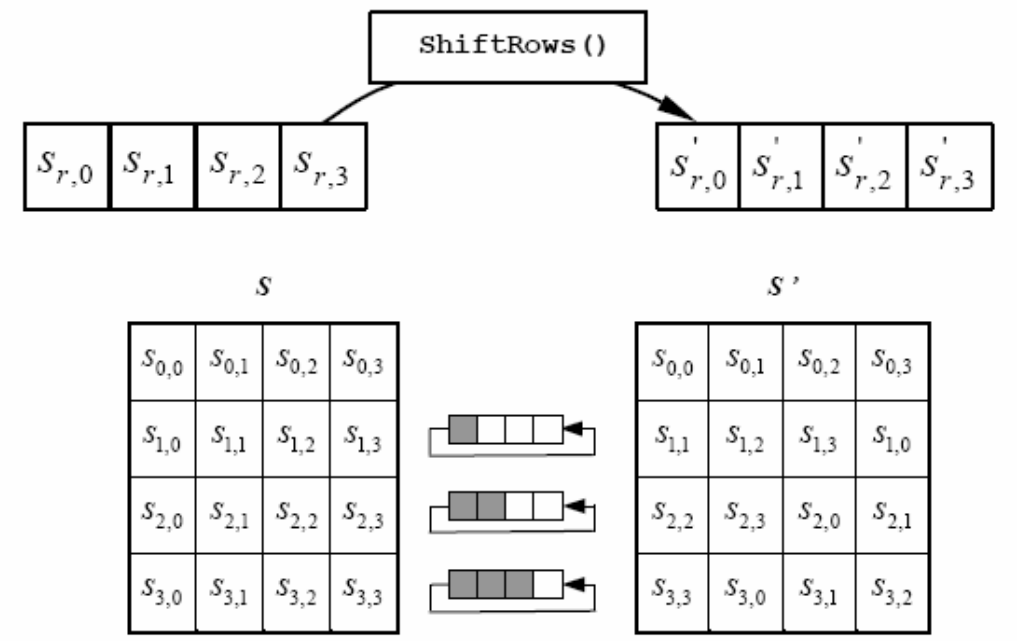
$$s`_{r,c} = s_{r,(c+shift(r,Nb))modNb} \quad (3.23)$$

για 0 < r < 4 και 0 ≤ c < Nb, όπου η τιμή ολίσθησης shift(r,Nb) εξαρτάται από τον αριθμό της γραμμής r. Θυμίζουμε ότι Nb = 4.

shift(1,4) = 1 shift(2,4) = 2 shift(3,4) = 3 (3.24)

Αυτό έχει ως αποτέλεσμα τη μετακίνηση των bytes σε «χαμηλότερες» θέσεις μέσα στη γραμμή, δηλαδή μικρότερες τιμές του c σε μία συγκεκριμένη γραμμή. Αντίθετα, τα «χαμηλότερα» bytes μετακινούνται προς τις «υψηλότερες» θέσεις της γραμμής, δηλαδή μεγαλύτερες τιμές του c σε μία συγκεκριμένη γραμμή.

Στο Σχήμα 3.5 φαίνεται ο μετασχηματισμός ShiftRows( ).



Σχήμα 3.5 Ο μετασχηματισμός ShiftRows( ).

3.5.3 Ο μετασχηματισμός MixColumns( )

Ο μετασχηματισμός MixColumns( ) εφαρμόζεται στον πίνακα State στήλη προς στήλη, αντιμετωπίζοντας κάθε στήλη ως ένα πολυώνυμο με τέσσερις

όρους, όπως περιγράφηκε στην §3.3.4. Οι στήλες θεωρούνται πολυώνυμα στο πεδίο GF(2^8) και πολλαπλασιάζονται modulo $x^4 + 1$ με ένα πολυώνυμο a(x), που δίνεται από τη σχέση

$$a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\} \tag{3.25}$$

Όπως είδαμε στην §3.3.4 αυτό μπορεί να γραφεί ως πολλαπλασιασμός πινάκων. Έστω $s`(x) = a(x) \otimes s(x)$. Θα ισχύει τότε:

$$\begin{bmatrix} s`_{0,c} \\ s`_{1,c} \\ s`_{2,c} \\ s`_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \tag{3.26}$$

για $0 \le c < Nb$. Ο πολλαπλασιασμός αυτός έχει ως αποτέλεσμα τα τέσσερα bytes σε μία στήλη να αντικατασταθούν από τα ακόλουθα:

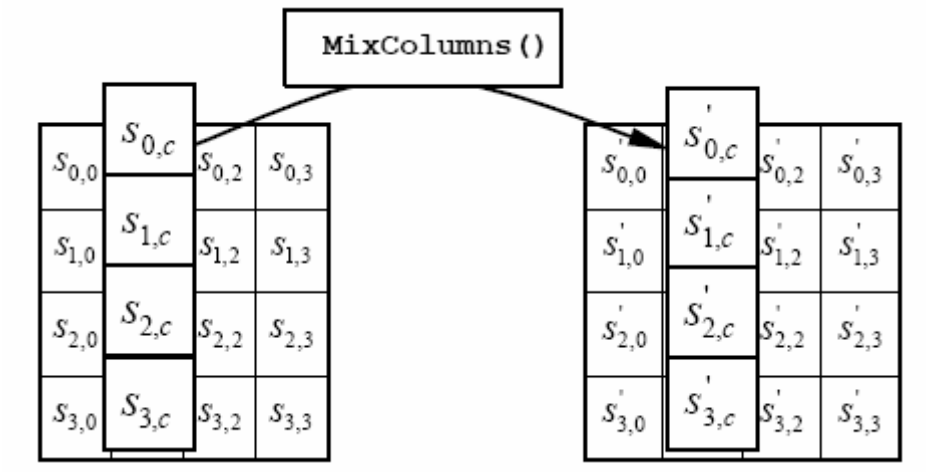
$$s`_{0,c} = (\{02\} \bullet s_{0,c}) \oplus (\{03\} \bullet s_{1,c}) \oplus s_{2,c} \oplus s_{3,c}$$

$$s`_{1,c} = s_{0,c} \oplus (\{02\} \bullet s_{1,c}) \oplus (\{03\} \bullet s_{2,c}) \oplus s_{3,c}$$

$$s`_{2,c} = s_{0,c} \oplus s_{1,c} \oplus (\{02\} \bullet s_{2,c}) \oplus (\{03\} \bullet s_{3,c})$$

$$s`_{3,c} = (\{03\} \bullet s_{0,c}) \oplus s_{1,c} \oplus s_{2,c} \oplus (\{02\} \bullet s_{3,c})$$

Στο Σχήμα 3.6 φαίνεται ο μετασχηματισμός MixColumns( ).



Σχήμα 3.6 Ο μετασχηματισμός MixColumns( ).

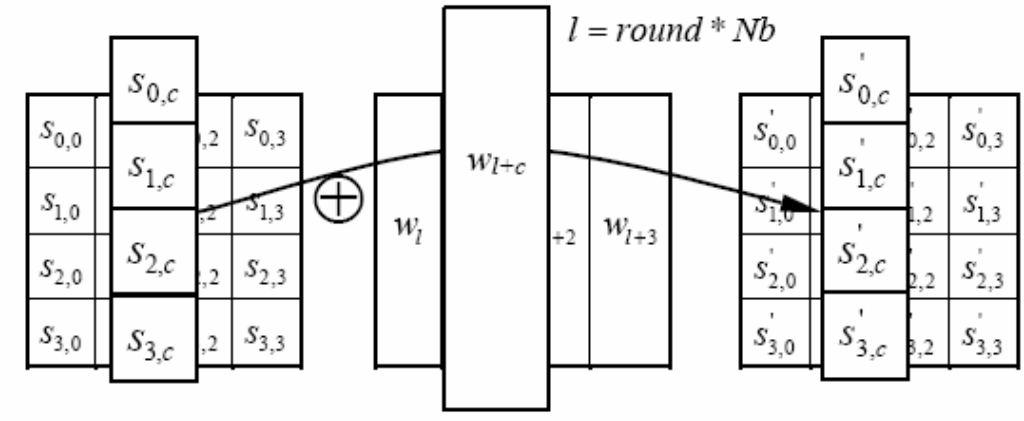
3.5.4 Ο μετασχηματισμός AddRoundKey( )

Κατά το μετασχηματισμό AddRoundKey( ) το κλειδί για κάθε επανάληψη του AES προστίθεται στο State με μία απλή πράξη XOR σε επίπεδο bit. Κάθε τέτοιο κλειδί αποτελείται από Nb λέξεις από το «πρόγραμμα» κλειδιών, δηλαδή τον πίνακα που περιλαμβάνει όλα τα απαιτούμενα κλειδιά το ένα μετά το άλλο. Η περιγραφή του τρόπου με τον οποίο προκύπτει το «πρόγραμμα» κλειδιών γίνεται στην §3.6. Αυτές οι Nb λέξεις προστίθενται κάθε μία στις στήλες του πίνακα State, έτσι ώστε

$$\left[s`_{0,c}, s`_{1,c}, s`_{2,c}, s`_{3,c}\right] = \left[s_{0,c}, s_{1,c} s_{2,c}, s_{3,c}\right] \oplus \left[w_{round*Nb+c}\right] \qquad (3.27)$$

για $0 \le c < Nb$, όπου $[w_i]$ είναι οι λέξεις από το «πρόγραμμα» κλειδιών, που περιγράφεται στην §3.6 και round είναι μία τιμή στο διάστημα $0 \le round \le r$. Κατά την κρυπτογράφηση η αρχική πρόσθεση του κλειδιού συμβαίνει όταν round = 0, πριν την πρώτη εφαρμογή της συνάρτησης κρυπτογράφησης (βλέπε Σχήμα 3.2). Η εφαρμογή του μετασχηματισμού AddRoundKey( ) στις Nr επαναλήψεις που απαιτούνται για να ολοκληρωθεί η κρυπτογράφηση, γίνεται όταν $1 \le round \le r$.

Το αποτέλεσμα αυτού του μετασχηματισμού φαίνεται στο Σχήμα 3.7, όπου l = round * Nb.



Σχήμα 3.7 Ο μετασχηματισμός AddRoundKey( ).

3.6 Επέκταση κλειδιών

Ο αλγόριθμος AES παίρνει το κλειδί κρυπτογράφησης K και εκτελεί τη ρουτίνα KeyExpansion( ) για την επέκταση των κλειδιών, ώστε να δημιουργήσει το «πρόγραμμα» κλειδιών. Η ρουτίνα αυτή παράγει συνολικά Nb(Nr + 1) λέξεις. Ο αλγόριθμος απαιτεί ένα αρχικό σύνολο από Nb λέξεις και κάθε μία από τις Nr επαναλήψεις απαιτεί Nb λέξεις για το κλειδί. Το «πρόγραμμα» κλειδιών που προκύπτει αποτελείται από ένα γραμμικό πίνακα με λέξεις των τεσσάρων bytes, που συμβολίζεται ως $[w_i]$, όπου το i παίρνει τιμές στο διάστημα $0 \leq i < Nb(Nr + 1)$.

Η επέκταση του αρχικού κλειδιού στο «πρόγραμμα» κλειδιών εκτελείται σύμφωνα με τον ψευδοκώδικα που ακολουθεί.

```
KeyExpansion(byte key[4*Nk], word w[Nb*(Nr+1)], Nk)
begin
    word temp

    i = 0

    while (i < Nk)
        w[i] = word(key[4*i], key[4*i+1], key[4*i+2], key[4*i+3])
        i = i+1
    end while

    i = Nk

    while (i < Nb * (Nr+1)]
    temp = w[i-1]
    if (i mod Nk = 0)
        temp = SubWord(RotWord(temp)) xor Rcon[i/Nk]
     else if (Nk > 6 and i mod Nk = 4)
        temp = SubWord(temp)
    end if
    w[i] = w[i-Nk] xor temp
    i = i + 1
    end while
end
```

Η SubWord( ) είναι μία συνάρτηση που παίρνει ως είσοδο μία λέξη των τεσσάρων bytes και εφαρμόζει το S-box (βλέπε §3.5.1, Σχήμα 3.4) σε κάθε ένα από τα τέσσερα bytes, ώστε να επιστρέψει τη νέα λέξη στην έξοδο. Η συνάρτηση RotWord( ) παίρνει ως είσοδο μία λέξη $[a_0, a_1, a_2, a_3]$, εκτελεί μία κυκλική εναλλαγή μεταξύ των στηλών και επιστρέφει τη λέξη $[a_1, a_2, a_3, a_0]$. Ο

πίνακας με τις σταθερές λέξεις κάθε επανάληψης Rcon[i], περιέχει τις τιμές που δίνονται από το [x^{i-1}, {00}, {00}, {00}] με το x^{i-1} να δηλώνει δυνάμεις του x στο πεδίο GF(2^8), όπως είδαμε στην §3.3.3. Σημειώστε ότι το x συμβολίζεται ως {02} και ότι το i ξεκινάει από την τιμή 1 και όχι 0.

Από τον ψευδοκώδικα φαίνεται ότι οι πρώτες Nk λέξεις του «προγράμματος» κλειδιών συμπληρώνονται με το κλειδί κρυπτογράφησης. Κάθε επόμενη λέξη w[i] προκύπτει από το αποτέλεσμα της πράξης XOR ανάμεσα στην προηγούμενη λέξη w[i – 1] και τη λέξη Nk θέσεις πριν w[i – Nk]. Για τις λέξεις που βρίσκονται σε θέσεις οι οποίες είναι πολλαπλάσια του Nk, εφαρμόζεται ένας μετασχηματισμός στη λέξη w[i – 1] πριν την εκτέλεση της πράξης XOR και ακολουθεί μία πράξη XOR με την κατάλληλη σταθερά κάθε επανάληψης Rcon[i]. Ο μετασχηματισμός αυτός αποτελείται από μία κυκλική ολίσθηση των bytes μιας λέξης, με τη συνάρτηση RotWord( ), που ακολουθείται από την εφαρμογή του S-box στα τέσσερα bytes της λέξης, με τη συνάρτηση SubWord( ).

Είναι σημαντικό να παρατηρήσουμε ότι η ρουτίνα για την επέκταση του κλειδιού, στην περίπτωση που το κλειδί κρυπτογράφησης έχει μήκος 256 bits (Nk = 8), διαφέρει ελαφρά από τις περιπτώσεις που το μήκος του κλειδιού είναι 128 ή 192 bits. Αν Nk = 8 και το i – 4 είναι πολλαπλάσιο του Nk, τότε η συνάρτηση SubWord( ) εφαρμόζεται στη λέξη w[i – 1] πριν την εκτέλεση της πράξης XOR.

3.7 Καταστάσεις λειτουργίας του AES

Ο αλγόριθμος AES μπορεί να χρησιμοποιηθεί για την κρυπτογράφηση και αποκρυπτογράφηση ενός μπλοκ δεδομένων σταθερού μεγέθους. Παρόλα αυτά, στην πράξη τα πραγματικά μηνύματα σπάνια έχουν τη μορφή μπλοκ σταθερού μεγέθους. Για παράδειγμα, τα δεδομένα σε ένα ασύρματο δίκτυο μεταδίδονται σε πλαίσια με διάφορα μήκη, που ποικίλουν από 512 ως 12000 bits σε κάθε πλαίσιο. Επομένως, για να χρησιμοποιήσουμε ένα μπλοκ αλγόριθμο κρυπτογράφησης, όπως ο AES, πρέπει να καθορίσουμε μία μέθοδο ώστε τα μηνύματα με τυχαίο μήκος να μετατρέπονται σε μία ακολουθία από μπλοκ σταθερού μήκους πριν την κρυπτογράφηση. Επιπλέον, η μέθοδος

αυτή θα πρέπει να επιτρέπει την επανένωση των μπλοκ, ώστε να προκύψει το αρχικό μήνυμα, κατά τη διάρκεια της αποκρυπτογράφησης. Η μέθοδος που χρησιμοποιείται για τη μετατροπή από τα μηνύματα στα μπλοκ και το αντίστροφο ονομάζεται κατάσταση λειτουργίας του μπλοκ αλγόριθμου.

Υπάρχουν αρκετές καταστάσεις λειτουργίας που μπορούν να χρησιμοποιηθούν σε συνδυασμό με τον AES [12]. Το NIST έχει καθορίσει μία λίστα με δεκαέξι διαφορετικές προσεγγίσεις και είναι ανοικτό σε νέες προτάσεις. Η επιλογή της κατάστασης λειτουργίας είναι πολύ σημαντικό θέμα, καθώς έχει επιπτώσεις τόσο στην πολυπλοκότητα της υλοποίησης, όσο και στην ασφάλεια του τελικού πρωτοκόλλου ασφαλείας. Κακές καταστάσεις λειτουργίας μπορούν να δημιουργήσουν προβλήματα στην ασφάλεια του δικτύου, παρόλο που ο αλγόριθμος AES είναι πολύ ισχυρός.

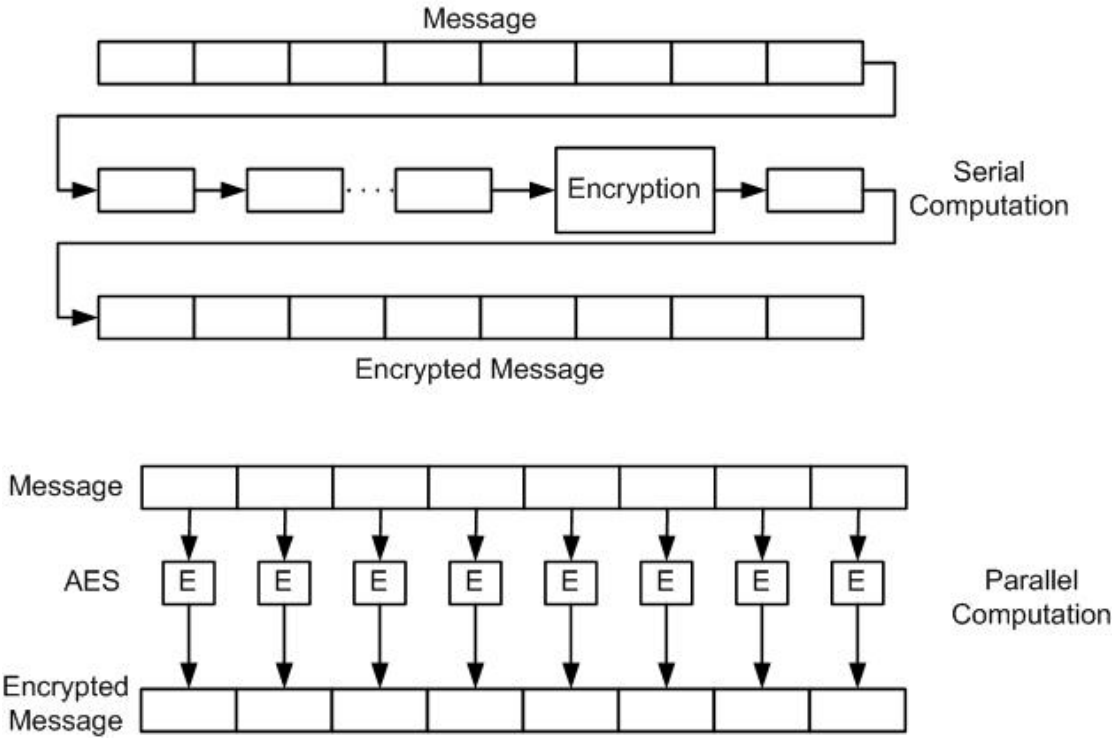
Για την κατανόηση των καταστάσεων λειτουργίας εξετάζουμε αρχικά μία από τις πιο απλές και κατανοητές καταστάσεις, το Ηλεκτρονικό Βιβλίο Κωδικών (Electronic Code Book - ECB). Γενικά, η κατάσταση λειτουργίας τοποθετείται ως ένδειξη δίπλα από τη λέξη AES. Επομένως, ένα σύστημα που χρησιμοποιεί το ECB, χαρακτηρίζεται ως AES/ECB.

3.7.1 Κατάσταση λειτουργίας ECB

Η κατάσταση λειτουργίας ECB [3] απλά παίρνει ένα τμήμα του μηνύματος, ένα μπλοκ κάθε φορά και κρυπτογραφεί κάθε μπλοκ ακολουθιακά χρησιμοποιώντας το ίδιο κλειδί μέχρι το τελευταίο τμήμα του μηνύματος. Η διαδικασία αυτή φαίνεται στο Σχήμα 3.8, που δείχνει τον υπολογισμό τόσο για τη σειριακή, δηλαδή ένα μπλοκ κάθε φορά, όσο και για την παράλληλη κρυπτογράφηση.

Η προσέγγιση αυτή φαίνεται απλή, αλλά παρουσιάζει ορισμένα προβλήματα. Το πιο εμφανές πρόβλημα είναι ότι το μήνυμα μπορεί να μην έχει μέγεθος ακριβώς πολλαπλάσιο του μεγέθους του μπλοκ και πρέπει να συμπληρώσουμε με μηδέν (padding) το τελευταίο μπλοκ και να θυμόμαστε το πραγματικό μήκος. Υπάρχει όμως και ένα πρόβλημα στην ασφάλεια. Αν δύο μπλοκ περιέχουν τα ίδια δεδομένα, το αποτέλεσμα της κρυπτογράφησης θα περιέχει

επίσης δύο ίδια μπλοκ, αποκαλύπτοντας πληροφορίες σε κάποιο εξωτερικό παρατηρητή του δικτύου.

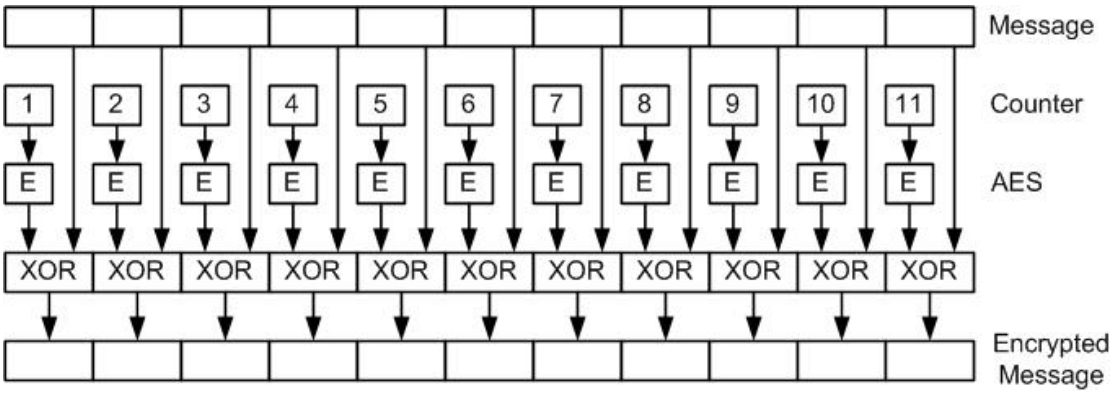


Σχήμα 3.8 Η κατάσταση λειτουργίας ECB.

Θεωρείστε ένα μήνυμα που αποτελείται από μία γραμματοσειρά, όπου το ίδιο γράμμα επαναλαμβάνεται 64 φορές, για παράδειγμα “AAAAAA…”. Αν το μήκος μπλοκ του AES είναι 128 bits, δηλαδή 16 bytes, τότε χρησιμοποιώντας το ECB το μήνυμα θα σπάσει σε τέσσερα μπλοκ καθένα από τα οποία περιέχει 16 γράμματα A. Μετά την κρυπτογράφηση τα τέσσερα μπλοκ θα μετατραπούν σε τέσσερα πανομοιότυπα κρυπτογραφημένα μπλοκ. Με τον τρόπο αυτό, ο εξωτερικός παρατηρητής πληροφορείται ότι αυτό το μήνυμα περιέχει στοιχεία που επαναλαμβάνονται. Εξαιτίας αυτής της αδυναμίας, καθώς και άλλων, τα συστήματα ασφαλείας στην πράξη δε χρησιμοποιούν την κατάσταση λειτουργίας ECB. Δε βρίσκεται καν στη λίστα με τις προτεινόμενες καταστάσεις λειτουργίας του NIST. Ακόμα και ο πιο ισχυρός μπλοκ αλγόριθμος κρυπτογράφησης δεν μπορεί να παρέχει την επιθυμητή ασφάλεια, αν υπάρχουν αδύνατα σημεία στην κατάσταση λειτουργίας.

3.7.2 Κατάσταση λειτουργίας μετρητή (CTR)

Η κατάσταση λειτουργίας μετρητή είναι πιο πολύπλοκη από την ECB και λειτουργεί με αρκετά διαφορετικό τρόπο [3]. Δε χρησιμοποιεί τον αλγόριθμο AES απευθείας για την κρυπτογράφηση των δεδομένων. Αντίθετα, κρυπτογραφεί την τυχαία τιμή ενός μετρητή και στη συνέχεια εφαρμόζει την πράξη XOR ανάμεσα στο αποτέλεσμα της κρυπτογράφησης και τα δεδομένα. Με τον τρόπο αυτό προκύπτει το κρυπτογραφημένο κείμενο, όπως φαίνεται στο Σχήμα 3.9. Ο μετρητής γενικά αυξάνεται κατά 1 με κάθε διαδοχικό μπλοκ που κρυπτογραφείται.



Σχήμα 3.9 Η κατάσταση λειτουργίας μετρητή.

Το μήνυμα χωρίζεται σε μπλοκ και το κρυπτογραφημένο κείμενο είναι το αποτέλεσμα της πράξης XOR ανάμεσα στο μπλοκ δεδομένων και την κρυπτογραφημένη τιμή του μετρητή από τον AES. Στο Σχήμα 3.9 ο μετρητής ξεκινάει από την τιμή 1 και αυξάνεται μέχρι την τιμή 11. Στην πράξη ο μετρητής μπορεί να ξεκινάει από μία τυχαία τιμή και να αυξάνεται κατά μία άλλη τιμή. Το σημαντικό είναι ο παραλήπτης του μηνύματος, που επιθυμεί να αποκρυπτογραφήσει το κείμενο, να γνωρίζει την αρχική τιμή του μετρητή καθώς και το βήμα της αύξησης.

Επειδή ο μετρητής αλλάζει τιμή για κάθε μπλοκ αποφεύγεται το πρόβλημα που εμφανίζεται στην κατάσταση λειτουργίας ECB με τα επαναλαμβανόμενα μπλοκ. Ακόμα και αν δύο μπλοκ δεδομένων είναι πανομοιότυπα, θα συνδυαστούν με διαφορετική τιμή του μετρητή και θα προκύψει διαφορετικό αποτέλεσμα για κάθε μπλοκ. Παρόλα αυτά, με τον τρόπο που παρουσιάζεται, η μέθοδος αυτή έχει ως αποτέλεσμα δύο ίδια, αλλά ξεχωριστά μηνύματα, να παρουσιάζουν το ίδιο αποτέλεσμα μετά την κρυπτογράφηση. Για το λόγο αυτό, στην

πράξη, ο μετρητής δεν ξεκινάει από την τιμή 1. Τυπικά, αρχικοποιείται με μία τιμή η οποία αλλάζει για κάθε διαδοχικό μήνυμα (nonce).

Η κατάσταση λειτουργίας μετρητή παρουσιάζει μερικές ενδιαφέρουσες ιδιότητες. Η αποκρυπτογράφηση είναι ακριβώς η ίδια διαδικασία με την κρυπτογράφηση, καθώς η εφαρμογή της πράξης XOR δύο φορές στα ίδια δεδομένα επιστρέφει τα αρχικά δεδομένα[4]. Αυτό σημαίνει ότι οι διάφορες υλοποιήσεις πρέπει να υλοποιήσουν μόνο τον αλγόριθμο κρυπτογράφησης και όχι τον αλγόριθμο αποκρυπτογράφησης. Ένα άλλο πολύ χρήσιμο χαρακτηριστικό, για ορισμένες εφαρμογές, είναι ότι η κρυπτογράφηση μπορεί να πραγματοποιηθεί παράλληλα. Εφόσον όλες οι τιμές του μετρητή είναι γνωστές εκ των προτέρων, μπορούμε να έχουμε μία τράπεζα από μονάδες κρυπτογράφησης AES ώστε να κρυπτογραφήσουμε ολόκληρο το μήνυμα σε ένα μόνο πέρασμα. Αυτό δεν ισχύει για πολλές από τις υπόλοιπες καταστάσεις λειτουργίας. Η τελευταία χρήσιμη ιδιότητα είναι ότι δεν υπάρχει πρόβλημα αν το μήνυμα δεν αποτελείται από ακριβή αριθμό μπλοκ. Απλά παίρνουμε το τελευταίο (μη συμπληρωμένο) μπλοκ και εφαρμόζουμε την πράξη XOR με την κρυπτογραφημένη τιμή του μετρητή, χρησιμοποιώντας μόνο τον αριθμό από bits που είναι αναγκαίος. Επομένως, το μήκος του κρυπτογραφημένου μηνύματος μπορεί να είναι το ίδιο ακριβώς με το αρχικό μήνυμα. Επειδή κάθε πράξη σε μπλοκ εξαρτάται από την κατάσταση του μετρητή από το προηγούμενο μπλοκ, η κατάσταση λειτουργίας μετρητή αποτελεί ουσιαστικά, κρυπτογράφηση ροής (stream cipher).

Η κατάσταση λειτουργίας μετρητή έχει χρησιμοποιηθεί για περισσότερο από είκοσι χρόνια, είναι ευρύτατα γνωστός και χαίρει της εμπιστοσύνης της κοινότητας των κρυπτογράφων. Η απλότητά της και η ωριμότητά της, καθιστούν αυτή την κατάσταση λειτουργίας ελκυστική επιλογή για το RSN. Παρόλα αυτά η κατάσταση λειτουργίας μετρητή δεν προστατεύει καθόλου την ακεραιότητα των μηνυμάτων. Επομένως, για χρήση στο RSN, πρέπει να προστεθούν επιπλέον χαρακτηριστικά.

[4] Αυτό είναι ένα παράδειγμα όπου ο αλγόριθμος κρυπτογράφησης δε χρειάζεται να έχει αντίστροφο.

3.7.3 Κατάσταση λειτουργίας CBC

Η κατάσταση λειτουργίας Cipher Block Chaining (CBC) χρησιμοποιείται για να παράγει έναν κώδικα ακεραιότητας του μηνύματος (Message Integrity Code - MIC). Το MIC ονομάζεται επίσης κώδικας επικύρωσης του μηνύματος (Message Authentication Code - MAC) από την κρυπτογραφική κοινότητα, οπότε προκύπτει το όνομα CBC-MAC[5].

Το CBC-MAC είναι μία τεχνική που χρησιμοποιείται για πολλά χρόνια και έχει γίνει διεθνές πρότυπο. Η λειτουργία του είναι αρκετά απλή [12]:

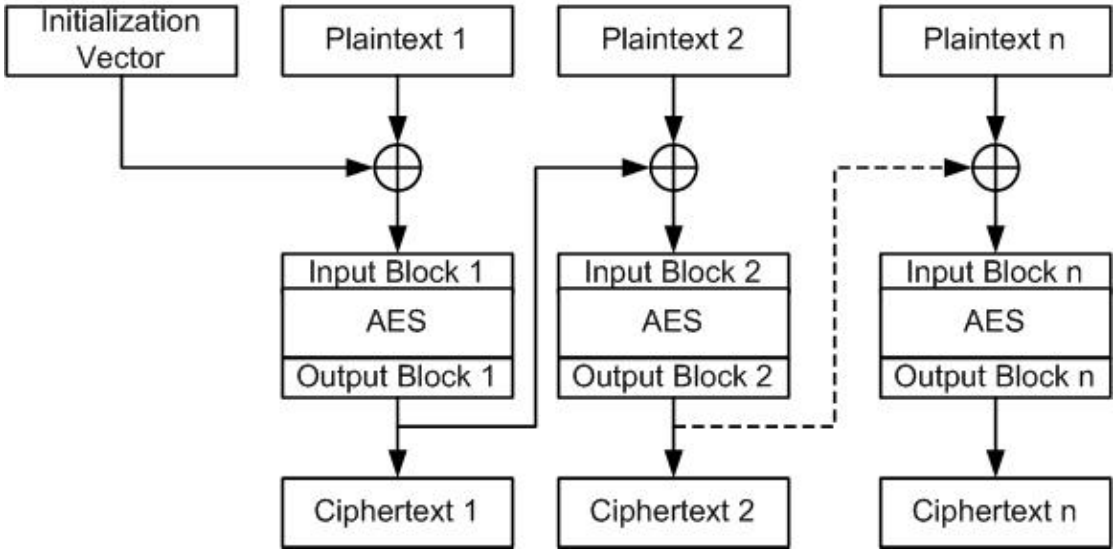
1. Γίνεται η κρυπτογράφηση του πρώτου μπλοκ στο μήνυμα χρησιμοποιώντας τον AES, ή οποιονδήποτε άλλο μπλοκ αλγόριθμο
2. Παίρνουμε το αποτέλεσμα της πράξης XOR ανάμεσα στο (1) και το δεύτερο μπλοκ του μηνύματος και στη συνέχεια κρυπτογραφούμε το νέο αποτέλεσμα
3. Παίρνουμε το αποτέλεσμα της πράξης XOR ανάμεσα στο (2) και το επόμενο μπλοκ του μηνύματος και στη συνέχεια κρυπτογραφούμε το νέο αποτέλεσμα κοκ

Τα παραπάνω βήματα φαίνονται στο Σχήμα 3.10, όπου σημειώνεται επίσης η χρήση ενός διανύσματος αρχικοποίησης (Initialization Vector - IV), για το πρώτο μπλοκ δεδομένων. Το τελικό αποτέλεσμα είναι ένα μόνο μπλοκ, των 128 bits στην προκειμένη περίπτωση, που συνδυάζει όλα τα δεδομένα του μηνύματος. Αν ένα ή περισσότερα bits αλλάξουν στο αρχικό μήνυμα, το αποτέλεσμα θα είναι εντελώς διαφορετικό. Για την ακρίβεια υπάρχει πιθανότητα μόλις 2^{-128} να προκύψει το ίδιο αποτέλεσμα. Το CBC-MAC είναι απλό, αλλά η λειτουργία του δεν μπορεί να γίνει παράλληλα. Η διαδικασία της κρυπτογράφησης πρέπει να γίνει ακολουθιακά. Επιπλέον, πρέπει να σημειώσουμε ότι το CBC-MAC μπορεί να χρησιμοποιηθεί μόνο για την επεξεργασία μηνυμάτων που περιλαμβάνουν ακριβή αριθμό από μπλοκ[6].

[5] Πλέον χρησιμοποιείται μόνο ο όρος MIC για να αποφεύγεται η σύγχυση με το επίπεδο Medium Access Control (MAC).

[6] Το πρωτόκολλο CCMP παρέχει λύση στο πρόβλημα αυτό με χρήση padding, η οποία όμως είναι μία μέθοδος που έχει εγείρει ανησυχίες μεταξύ ορισμένων κρυπτογράφων.

Ένα άλλο θέμα που προκύπτει από τη χρήση της κατάστασης λειτουργίας CBC είναι αυτό της υλοποίησης του αλγόριθμου AES σε υλικό. Συγκεκριμένα στη βιβλιογραφία υπάρχουν δημοσιευμένες αρκετές υλοποιήσεις του AES σε υλικό, που χαρακτηρίζονται από υψηλό ρυθμό πράξεων (throughput). Οι υλοποιήσεις αυτές βασίζονται κυρίως στην τεχνική «ξεδιπλώματος» του αλγόριθμου (loop unrolling) και στην εισαγωγή καταχωρητών (pipeline registers) ανάμεσα στις επαναλήψεις του για την αύξηση του throughput στην τάξη των Gbit/s. Είναι προφανές ότι αν επιλεγεί το CBC ως κατάσταση λειτουργίας του AES, δεν είναι δυνατή η χρήση τέτοιων υλοποιήσεων. Αυτό συμβαίνει επειδή η επεξεργασία ενός μπλοκ δεδομένων εξαρτάται από το αποτέλεσμα της επεξεργασίας του προηγούμενου μπλοκ.



Σχήμα 3.10 Η κατάσταση λειτουργίας CBC.

3.7.4 Κατάσταση λειτουργίας CCM

Η κατάσταση λειτουργίας CCM δημιουργήθηκε αποκλειστικά για χρήση στο IEEE 802.11i RSN, αλλά μπορεί να εφαρμοστεί και σε άλλα συστήματα. Έχει προταθεί στο NIST και έγινε αποδεκτή ως γενική κατάσταση λειτουργίας για χρήση με τον αλγόριθμο AES. Έχει γίνει επίσης αποδεκτή από τον IETF για χρήση στο πρωτόκολλο IPSec. Το CCM δημιουργήθηκε από τρεις κρυπτογράφους που συμμετέχουν στην ομάδα για το πρότυπο 802.11i, τους Doug Whiting, Russ Housley και Niels Ferguson [13].

Το CCM χρησιμοποιεί την κατάσταση λειτουργίας μετρητή για την εμπιστευτικότητα σε συνδυασμό με την κατάσταση λειτουργίας CBC για την επικύρωση και την ακεραιότητα των μηνυμάτων.

Η κατάσταση λειτουργίας CCM προσθέτει ορισμένα χρήσιμα χαρακτηριστικά για συγκεκριμένες εφαρμογές, όπως το RSN. Τα επιπλέον χαρακτηριστικά είναι τα παρακάτω:

- Παρέχει τις προδιαγραφές για μία τιμή αρχικοποίησης της διαδικασίας (nonce), έτσι ώστε διαδοχικά μηνύματα να διαχωρίζονται κρυπτογραφικά
- Συνδέει την κρυπτογράφηση και την επικύρωση με ένα μόνο κλειδί
- Επεκτείνει την επικύρωση ώστε να καλύψει δεδομένα της επικεφαλίδας του μηνύματος, τα οποία δεν κρυπτογραφούνται

Το τελευταίο χαρακτηριστικό είναι πολύ σημαντικό για την υλοποίηση του RSN. Στις περισσότερες υπάρχουσες μεθόδους, οι οποίες πραγματοποιούν τόσο κρυπτογράφηση, όσο και επικύρωση γίνεται η υπόθεση ότι ολόκληρο το μήνυμα θα κρυπτογραφηθεί. Παρόλα αυτά, στο 802.11, μόνο ένα μέρος του μηνύματος χρειάζεται να κρυπτογραφηθεί. Η επικεφαλίδα του μηνύματος περιέχει τις MAC διευθύνσεις που χρησιμοποιούνται ώστε να παραδοθεί το μήνυμα στο σωστό παραλήπτη, καθώς και πληροφορίες σχετικές με τη λειτουργία του ασύρματου δικτύου. Τα πεδία της επικεφαλίδας πρέπει να σταλούν χωρίς να κρυπτογραφηθούν έτσι ώστε οι υπόλοιπες ασύρματες συσκευές να μπορούν να λειτουργήσουν. Επομένως, μόνο το τμήμα του μηνύματος που περιέχει τα δεδομένα κρυπτογραφείται. Παρόλο που η επικεφαλίδα δεν κρυπτογραφείται, ο παραλήπτης θέλει τη διαβεβαίωση ότι αυτή δεν έχει τροποποιηθεί. Για παράδειγμα δεν επιθυμούμε σε καμία περίπτωση κάποιος εχθρός να αλλάξει τη διεύθυνση προέλευσης, ώστε να απαντήσουμε κατά λάθος σε αυτόν αντί για το σωστό αποστολέα του μηνύματος. Για να επιτευχθεί αυτό η κατάσταση λειτουργίας CCM επιτρέπει την κρυπτογράφηση ενός τμήματος του μηνύματος, το οποίο επικυρώνεται με τη βοήθεια του CBC-MAC.

Ως γενικός κανόνας ισχύει ότι δεν είναι σωστό να χρησιμοποιείται το ίδιο κλειδί για δύο ξεχωριστές κρυπτογραφικές λειτουργίες. Στην κατάσταση λει-

τουργίας CCM αυτός ο κανόνας φαίνεται να παραβιάζεται, καθώς χρησιμοποιείται το ίδιο κλειδί τόσο για την κρυπτογράφηση, όσο και για το επικύρωση. Παρόλο που χρησιμοποιείται το ίδιο κλειδί, αυτό συνδυάζεται σε κάθε περίπτωση με ένα διαφορετικό IV. Η κατασκευή του IV είναι διαφορετική για τις καταστάσεις λειτουργίας μετρητή και CBC-MAC αντίστοιχα, οδηγώντας έτσι ουσιαστικά σε δύο ξεχωριστά κλειδιά. Η αποδοτικότητα αυτού του διαχωρισμού έχει αποδειχθεί από τους κρυπτογράφους [14].

4 Το πρωτόκολλο CCMP

4.1 Σύνοψη του πρωτοκόλλου CCMP

Το πρωτόκολλο CCMP βελτιώνει σημαντικά το επίπεδο της ασφάλειας στα ασύρματα δίκτυα 802.11. Παρέχει εμπιστευτικότητα, επικύρωση, ακεραιότητα, προστασία από την επανάληψη πακέτων και η υλοποίησή του είναι υποχρεωτική ώστε η συσκευή να είναι συμβατή με το πρότυπο RSN [2].

Το CCMP βασίζεται στην κατάσταση λειτουργίας CCM του αλγορίθμου κρυπτογράφησης AES. Η κατάσταση λειτουργίας CCM συνδυάζει την κατάσταση λειτουργίας μετρητή (CTR) για εμπιστευτικότητα και CBC-MAC για επικύρωση και ακεραιότητα. Το CCM προστατεύει το ακεραιότητα τόσο των δεδομένων του MPDU, όσο και μερικών τμημάτων της IEEE 802.11 επικεφαλίδας του MPDU.

Ο αλγόριθμος AES ορίζεται στο FIPS-197 [11]. Η επεξεργασία των δεδομένων σύμφωνα με το πρωτόκολλο CCMP χρησιμοποιεί τον AES με μήκος κλειδιού 128 bits και μέγεθος μπλοκ 128 bits.

Η κατάσταση λειτουργίας CCM ορίζεται στο IETF RFC-360 [39]. Όπως είδαμε το CCM είναι μία γενικής χρήσης κατάσταση λειτουργίας, η οποία μπορεί να χρησιμοποιηθεί με οποιοδήποτε μπλοκ αλγόριθμο κρυπτογράφησης. Το CCM έχει δύο παραμέτρους, M και L και το πρωτόκολλο CCMP χρησιμοποιεί τις παρακάτω τιμές για αυτές τις παραμέτρους:

- M = 8, που ορίζει ότι το μέγεθος του MIC είναι 8 bytes
- L = 2, που ορίζει ότι το μέγεθος του πεδίου μήκους, το οποίο δείχνει το μήκος του MPDU σε bytes, είναι 2 bytes

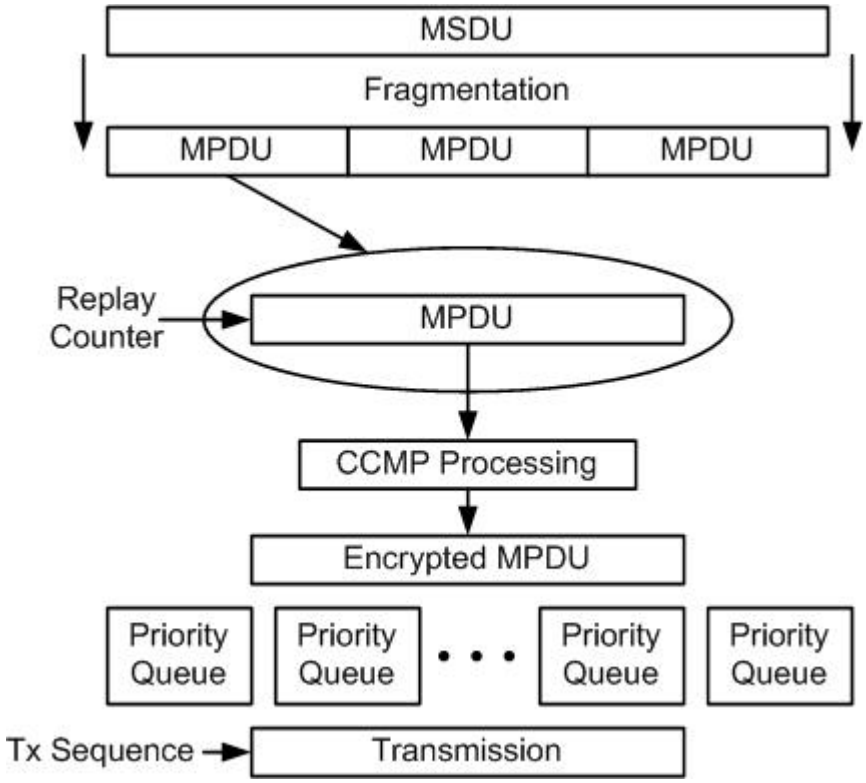
Η τιμή του L είναι αρκετά μεγάλη ώστε να αναπαραστήσει το μήκος του μεγαλύτερου δυνατού IEEE 802.11 πακέτου, σε bytes.

Το CCM απαιτεί ένα καινούργιο προσωρινό κλειδί για κάθε σύνδεση (session). Το CCM χρειάζεται επίσης μία μοναδική τιμή αρχικοποίησης (nonce) για κάθε πλαίσιο (frame) που προστατεύεται από ένα συγκεκριμένο προσωρινό

κλειδί. Για το σκοπό αυτό το πρωτόκολλο CCMP χρησιμοποιεί έναν αριθμό πακέτου (Packet Number - PN). Η χρήση για δεύτερη φορά ενός PN με το ίδιο προσωρινό κλειδί αναιρεί όλες τις εγγυήσεις για ασφάλεια που παρέχει το πρωτόκολλο.

4.2 Κρυπτογράφηση δεδομένων

Το CCMP κρυπτογραφεί τα δεδομένα σε επίπεδο MPDU. Τα MPDU, είναι το αποτέλεσμα του τεμαχισμού των MSDU, δηλαδή μεγαλύτερων πακέτων που έρχονται από υψηλότερο επίπεδο. Στο Σχήμα 4.1 φαίνεται η ροή των δεδομένων από το επίπεδο MSDU στο MPDU και τελικά την κεραία για την μετάδοση.



Σχήμα 4.1 Η ροή των δεδομένων κατά την επεξεργασία με το CCMP.

Τα δεδομένα φτάνουν ως MSDU και τεμαχίζονται σε MPDU. Σε κάθε MPDU ανατίθεται η δική του επικεφαλίδα IEEE 802.11 MAC, η οποία περιέχει τις διευθύνσεις αποστολέα, παραλήπτη καθώς και άλλες πληροφορίες απαραίτητες για τη σωστή λειτουργία των δικτυακών συσκευών. Σε αυτό το σημείο, ο αλγόριθμος CCMP επεξεργάζεται κάθε MPDU ώστε να προκύψει το νέο κρυ-

πτογραφημένο MPDU. Μόνο το τμήμα που περιέχει δεδομένα κρυπτογραφείται και όχι η επικεφαλίδα. Παρόλα αυτά, το CCMP κάνει περισσότερες λειτουργίες από να κρυπτογραφεί απλά τμήματα του MPDU. Προσθέτει επίσης επιπλέον πεδία με αποτέλεσμα το τελικό κρυπτογραφημένο MPDU να είναι 16 bytes μεγαλύτερο σε μήκος από το αρχικό.

Τα βήματα που ακολουθούνται για την κρυπτογράφηση ενός MPDU φαίνονται συνοπτικά στο Σχήμα 4.2 και περιγράφονται παρακάτω:

1. Η διαδικασία ξεκινάει με ένα μη κρυπτογραφημένο MPDU, συμπληρωμένο με την επικεφαλίδα IEEE 802.11 MAC. Η επικεφαλίδα περιέχει τις διευθύνσεις αποστολέα και παραλήπτη, αλλά οι τιμές ορισμένων πεδίων δεν είναι γνωστές ακόμα οπότε τίθενται στην τιμή 0 προς το παρόν.

2. Η MAC επικεφαλίδα ξεχωρίζεται από το MPDU, ώστε να προστεθεί αυτούσια στο τελικό MPDU. Εξάγεται πληροφορία από την επικεφαλίδα και χρησιμοποιείται κατά τον υπολογισμό της τιμής MIC, η οποία έχει μήκος 8 bytes.

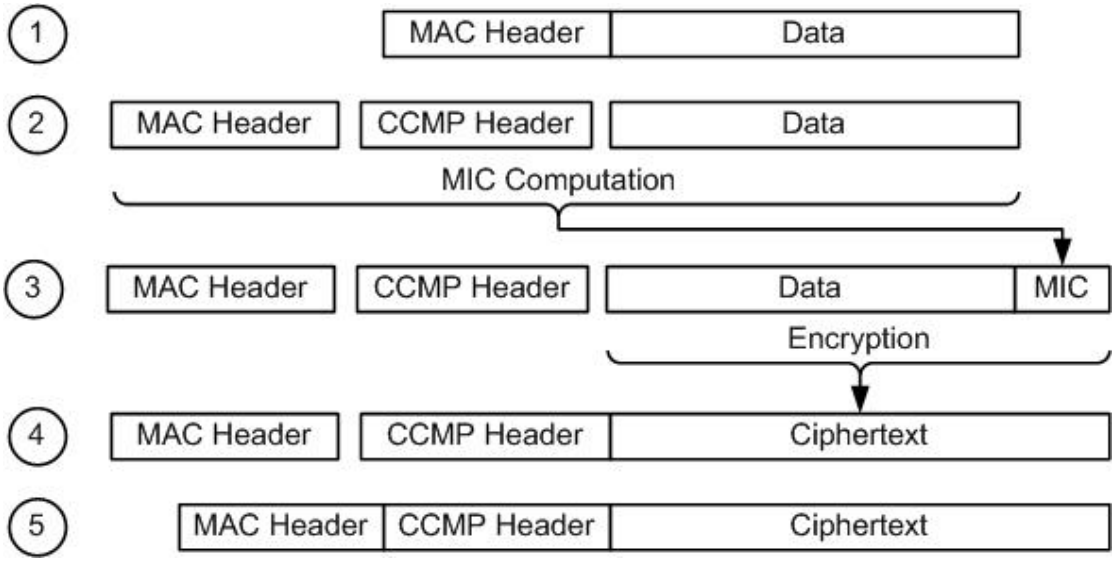
3. Υπολογίζεται η τιμή του MIC, ώστε να προστατευθούν τα δεδομένα και τμήματα της επικεφαλίδας IEEE 802.11 MAC. Η τιμή MIC προσαρτάται στα δεδομένα.

4. Ο συνδυασμός δεδομένων και MIC κρυπτογραφείται και στο τέλος προστίθεται στην αρχή του πακέτου η CCMP επικεφαλίδα.

5. Η MAC επικεφαλίδα προστίθεται αυτούσια στην αρχή του νέου MPDU, το οποίο είναι έτοιμο να μπει στην ουρά μετάδοσης. Η λογική που αναλαμβάνει την αναμετάδοση δε χρειάζεται να γνωρίζει τίποτα για τη CCMP επικεφαλίδα. Από εδώ και πέρα, μέχρι τη μετάδοση του MPDU, μόνο η MAC επικεφαλίδα ενημερώνεται.

Τα κρυπτογραφημένα MPDU τοποθετούνται σε ουρά προτεραιότητας για τη μετάδοση. Είναι πιθανό να υπάρχουν διάφορες ουρές προτεραιότητας, που περιμένουν να εξυπηρετηθούν, σύμφωνα με κάποια πολιτική. Αυτό το γεγονός επιτρέπει τη μελλοντική επέκταση του πρωτοκόλλου, ώστε να μπορεί να καλύψει διάφορες κλάσεις προτεραιότητας σύμφωνα με το πρότυπο IEEE

802.11e. Ακριβώς πριν τη μετάδοση, ορισμένα πεδία της IEEE 802.11 επικεφαλίδας ανανεώνονται ώστε να ικανοποιούν τους κανόνες μετάδοσης. Αυτά τα πεδία που η τιμή τους αλλάζει κατά τη μετάδοση ή την αναμετάδοσή τους από δρομολογητές, αποκλείονται από τον υπολογισμό της τιμής του MIC.



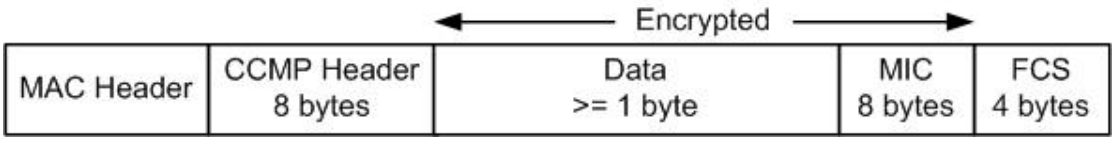
Σχήμα 4.2 Τα βήματα της κρυπτογράφησης ενός MPDU.

4.3 Τα πεδία του MPDU

Η επεξεργασία με το πρωτόκολλο CCMP επεκτείνει το αρχικό μέγεθος του MPDU κατά 16 bytes, 8 bytes για το πεδίο της επικεφαλίδας του CCMP και 8 bytes για το πεδίο MIC. Στο Σχήμα 4.3 φαίνονται τα πεδία του MPDU, μετά την επεξεργασία με το CCMP. Το πεδίο της επικεφαλίδας MAC έχει μήκος ως 32 bytes, ανάλογα με την παρουσία ορισμένων υποπεδίων που αναλύονται στην §4.3.1. Το πεδίο των δεδομένων έχει μήκος από 1 – 2296 bytes[7]. Πριν τη μετάδοση του MPDU στο ασύρματο δίκτυο, μεταδίδεται μία ειδική ακολουθία από bits η οποία ονομάζεται preamble. Η ακολουθία αυτή αναγνωρίζεται από όλους τους ασύρματους δέκτες ως σήμα κατατεθέν του 802.11. Η μετάδοση του preamble διαρκεί μερικά μόνο μsec και μετά το τέλος του όλοι οι δέκτες που βρίσκονται σε μικρή ακτίνα έχουν κλειδωθεί και ρυθμιστεί ώστε να λάβουν και να επεξεργαστούν τα δεδομένα που ακολουθούν. Μετά το

[7] Το μέγιστο επιτρεπτό μέγεθος δεδομένων σύμφωνα με το πρότυπο 802.11 είναι 2312 bytes. Επομένως, υπάρχει περιθώριο για 1 – 2296 bytes δεδομένων, καθώς 2296 = 2312 – 8 MIC bytes – 8 CCMP Header bytes.

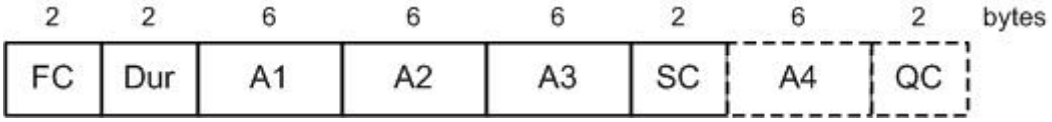
preamble ακολουθεί μία ακόμη επικεφαλίδα, η οποία ονομάζεται PLCP (Physical Layer Convergence Protocol) επικεφαλίδα και μεταδίδεται πριν το MPDU. Η επικεφαλίδα αυτή δεν είναι σημαντική για τη λειτουργία του πρωτοκόλλου CCMP. Περιέχει απλά πληροφορία σχετικά με τη λογική του δέκτη, όπως ο ρυθμός μετάδοσης δεδομένων και το μήκος του MPDU.



Σχήμα 4.3 Τα πεδία του κρυπτογραφημένου MPDU.

4.3.1 Η επικεφαλίδα MAC

Η επικεφαλίδα MAC παρουσιάζεται με τρεις διαφορετικές δομές ανάλογα με το είδος της πληροφορίας που μεταδίδεται στο ασύρματο δίκτυο: πληροφορία ελέγχου, δεδομένων ή διαχείρισης. Θα αναφερθούμε συνοπτικά στα πεδία της επικεφαλίδας MAC στην περίπτωση που αποστέλλονται δεδομένα. Τα πεδία αυτά φαίνονται στο Σχήμα 4.4.



Σχήμα 4.4 Τα πεδία της επικεφαλίδας MAC.

Τα πιο σημαντικά πεδία της επικεφαλίδας MAC είναι τα πεδία διευθυνσιοδότησης. Τα πεδία των διευθύνσεων είναι αριθμημένα επειδή χρησιμοποιούνται για διαφορετικούς σκοπούς ανάλογα με το είδος του πακέτου που μεταδίδεται. Σε γενικές γραμμές η διεύθυνση A1 χρησιμοποιείται για τον παραλήπτη (receiver) και η διεύθυνση A2 χρησιμοποιείται για τον αποστολέα (transmitter). Όπως ορίζει το πρότυπο της IEEE για τα τοπικά δίκτυα (LANs), οι διευθύνσεις αυτές έχουν μήκος 6 bytes και κάθε συσκευή έχει μοναδική διεύθυνση, η οποία ανατίθεται κατά τη κατασκευή της. Η διεύθυνση προορισμού μπορεί να δείχνει ένα μοναδικό παραλήπτη, οπότε το πακέτο παραδίδεται μόνο στη συσκευή που ταιριάζει η διεύθυνση της (unicast). Μπορεί επίσης να δείχνει πολλαπλούς παραλήπτες οπότε το πακέτο παραδίδεται σε πολλές συσκευές, ίσως και σε όλες τις συσκευές που

βρίσκονται στην εμβέλεια του αποστολέα. Επομένως, η διεύθυνση προορισμού ανήκει σε μία από τις παρακάτω κατηγορίες:

1. Unicast: Το πακέτο παραδίδεται σε μία συσκευή.
2. Multicast: Το πακέτο παραδίδεται σε πολλές συσκευές.
3. Broadcast: Το πακέτο παραδίδεται σε όλες τις συσκευές εντός εμβέλειας (ειδική περίπτωση της κατηγορίας multicast).

Η επικεφαλίδα MAC του IEEE 802.11 είναι πιο πολύπλοκη σε σχέση με άλλες επικεφαλίδες τοπικών δικτύων, όπως πχ του IEEE 802.3 (Ethernet), αφού περιέχει αρκετά πεδία για το συντονισμό των 802.11 δικτύων. Η επικεφαλίδα MAC του IEEE 802.11 MPDU μπορεί να έχει από δύο ως τέσσερα πεδία διευθύνσεων ανάλογα με την περίπτωση.

1. Διεύθυνση αποστολέα (Transmitter Address – TA): Η συσκευή που μεταδίδει το πακέτο.
2. Διεύθυνση παραλήπτη (Receiver Address – RA): Η συσκευή που παραλαμβάνει το πακέτο.
3. Διεύθυνση πηγής (Source Address – SA): Η συσκευή που δημιούργησε το αρχικό πακέτο.
4. Διεύθυνση προορισμού (Destination Address – DA): Η συσκευή που παραλαμβάνει τελικά και επεξεργάζεται το πακέτο.

Όλα αυτά τα πεδία διευθύνσεων χρειάζονται επειδή μπορεί να υπάρξουν διάφορα σενάρια κατά την μετάδοση ενός πακέτου σε ασύρματο δίκτυο 802.11. Σε ένα ad-hoc δίκτυο, όπου δε χρησιμοποιούνται AP, οι συσκευές στέλνουν μηνύματα απευθείας η μία στην άλλη. Σε αυτή την περίπτωση η συσκευή που δημιουργεί το μήνυμα είναι ταυτόχρονα αποστολέας και αντίστοιχα η συσκευή που παραλαμβάνει το μήνυμα είναι αυτή που το επεξεργάζεται. Επομένως, σε αυτή την περίπτωση μόνο δύο διευθύνσεις περιέχονται στη MAC επικεφαλίδα.

Σε ένα δομημένο δίκτυο (infrastructure network), όπου λειτουργεί ένα AP, όλες οι συσκευές στέλνουν τα πακέτα στο AP, το οποίο τα προωθεί στη συνέχεια στο σωστό προορισμό. Σε αυτή την περίπτωση, η ασύρματη συσκευή δη-

μιουργεί και στέλνει το μήνυμα. Το AP το παραλαμβάνει, αλλά δεν είναι ο τελικός παραλήπτης. Επομένως χρειάζονται τρεις διευθύνσεις:

- Η διεύθυνση της ασύρματης συσκευής (SA = TA)
- Η διεύθυνση του AP (RA)
- Η διεύθυνση της συσκευής που παραλαμβάνει τελικά το πακέτο (DA)

Όταν τα μηνύματα κινούνται από το AP προς την ασύρματη συσκευή οι τρεις διευθύνσεις είναι οι ακόλουθες:

- Η διεύθυνση της συσκευής που έστειλε αρχικά το μήνυμα (SA)
- Η διεύθυνση του AP, που προωθεί το μήνυμα (TA)
- Η διεύθυνση της συσκευής που παραλαμβάνει τελικά το πακέτο (RA = DA)

Γενικά, οι τέσσερις διευθύνσεις χρησιμοποιούνται ταυτόχρονα όταν ένα AP επικοινωνεί ασύρματα με κάποιο άλλο AP[8]. Παρόλα αυτά, αυτή η κατάσταση λειτουργίας δεν είναι πλήρως καθορισμένη από το πρότυπο IEEE 802.11 και υπάρχουν λίγες υλοποιήσεις που το εκμεταλλεύονται, οι οποίες εξαρτώνται από τον κάθε κατασκευαστή.

Οι MAC διευθύνσεις αποτελούν σημαντικό κομμάτι της ασφάλειας καθώς, αν και κάθε συσκευή έχει μοναδική διεύθυνση, είναι εύκολο για έναν εχθρό να προσποιηθεί ότι είναι κάποιος νόμιμος χρήστης του ασύρματου δικτύου, αντιγράφοντας τη MAC διεύθυνση του δεύτερου. Αυτή είναι η κλασσική μεθοδολογία επίθεσης, όπου επιτρέπεται σε ένα νόμιμο χρήστη να αποκτήσει πρόσβαση στο δίκτυο και μετά λαμβάνεται ο έλεγχος της σύνδεσης για μη επιτρεπτούς σκοπούς χωρίς να γίνει κάτι αντιληπτό (hijack attack).

Το πρωτόκολλο CCMP εξασφαλίζει την ακεραιότητα μερικών τμημάτων της MAC επικεφαλίδας του MPDU, αφού πρώτα θέσει στην τιμή 0 ορισμένα bits των πεδίων τα οποία ενδεχομένως να αλλάξουν στη συνέχεια τιμή λόγω της λειτουργίας του IEEE 802.11. Τα τμήματα αυτά της επικεφαλίδας που χρησι-

[8] Η διαδικασία αυτή ονομάζεται ασύρματη γεφύρωση (wireless bridging).

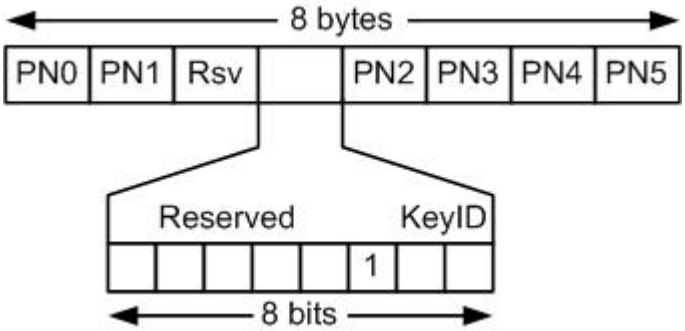
μοποιούνται κατά τον υπολογισμό της τιμής MIC ονομάζονται Additional Authentication Data – AAD.

4.3.2 Η επικεφαλίδα του πρωτοκόλλου CCMP

Η επικεφαλίδα του CCMP πρέπει να προστεθεί μπροστά από τα κρυπτογραφημένα δεδομένα και να μεταδοθεί «καθαρή», δηλαδή χωρίς να υποστεί κρυπτογράφηση. Η επικεφαλίδα του CCMP παρέχει ουσιαστικά τον αριθμό πακέτου (PN) μήκους 48 bits, ο οποίος εξυπηρετεί δύο σκοπούς:

1. Εξασφαλίζει προστασία κατά της επανάληψης πακέτων.
2. Δίνει τη δυνατότητα στον παραλήπτη να δημιουργήσει την τιμή αρχικοποίησης (nonce), που χρησιμοποιήθηκε κατά την κρυπτογράφηση.

Η δομή της επικεφαλίδας αυτής είναι αρκετά όμοια με την επικεφαλίδα που χρησιμοποιείται στην περίπτωση του TKIP. Αυτό έγινε ενσυνείδητα με σκοπό να απλοποιήσει την υλοποίηση των AP, τα οποία πρέπει να λαμβάνουν δεδομένα από ασύρματες συσκευές που βασίζονται είτε στο TKIP, είτε στο CCMP. Η δομή της επικεφαλίδας του CCMP φαίνεται στο Σχήμα 4.5.



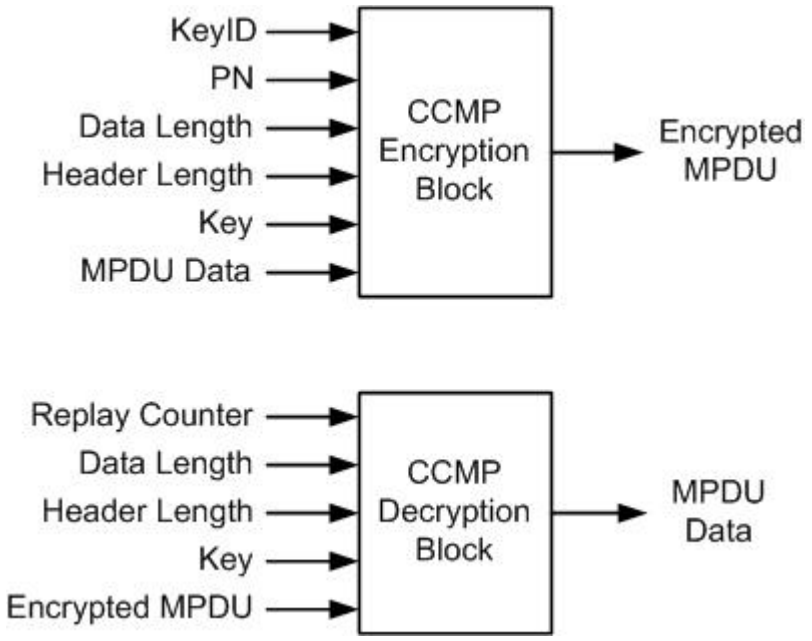
Σχήμα 4.5 Η επικεφαλίδα του CCMP.

Η επικεφαλίδα αποτελείται από τα πεδία PN, ExtIV και KeyID. Το πεδίο PN αναπαρίσταται ως ένας γραμμικός πίνακας από 6 bytes. Το PN5 είναι το πιο σημαντικό byte και το PN0 είναι το λιγότερο σημαντικό byte. Το ExtIV bit έχει πάντα την τιμή ένα στην περίπτωση που χρησιμοποιείται το CCMP ως πρωτόκολλο ασφάλειας. Η τιμή ένα δείχνει ότι η επικεφαλίδα του CCMP επεκτείνει το μήκος του MPDU κατά 8 bytes, αντί για 4 bytes όταν χρησιμοποιείται το

WEP. Το πεδίο KeyID υποδεικνύει στον παραλήπτη πιο κλειδί κρυπτογράφησης έχει χρησιμοποιηθεί.

4.4 Η λειτουργία του πρωτοκόλλου CCMP

Η υλοποίηση του CCMP, για την κρυπτογράφηση και την αποκρυπτογράφηση, μπορεί να γίνει εύκολα κατανοητή, αν αντιμετωπιστεί ως black box με εισόδους και εξόδους όπως φαίνεται στο Σχήμα 4.6.



Σχήμα 4.6 Κρυπτογράφηση και αποκρυπτογράφηση με το CCMP.

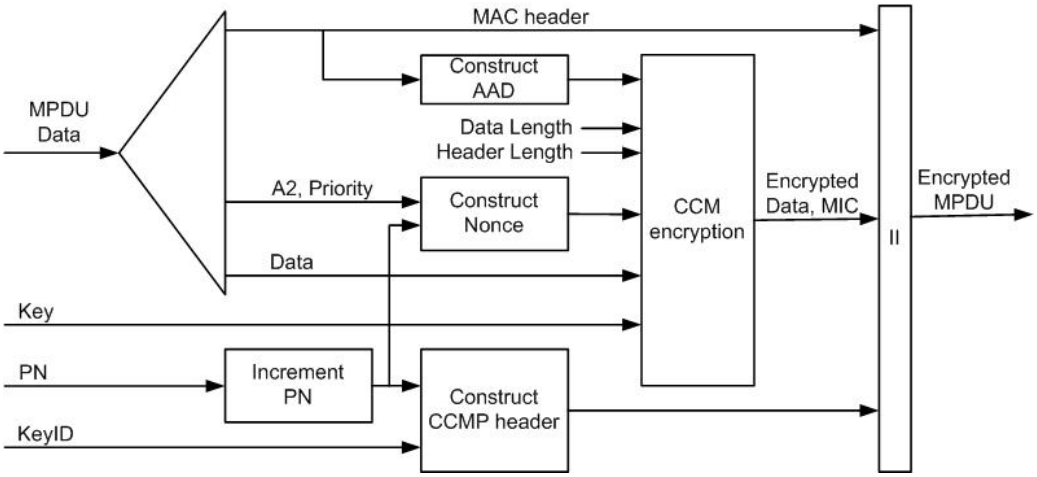
Η υλοποίηση του CCMP πρέπει να διατηρεί ένα μετρητή ακολουθίας, που ονομάζεται αριθμός πακέτου (PN), ο οποίος αυξάνεται πριν την κρυπτογράφηση του κάθε πακέτου. Αυτό εμποδίζει κάποιο εχθρό από το να χρησιμοποιήσει ξανά ένα πακέτο που έχει σταλεί με επιτυχία προηγουμένως. Ο PN έχει μήκος 48 bits, που είναι αρκετό ώστε να μην υπάρξει ποτέ υπερχείλιση (overflow). Δύο πακέτα δεν πρέπει να σταλούν ποτέ με το ίδιο PN, αν δεν έχει αλλάξει πρώτα το κλειδί της κρυπτογράφησης. Φυσικά, αν η συσκευή αναγκαστεί να επανεκκινήσει τη λειτουργία της, ο PN θα τεθεί και πάλι στην τιμή μηδέν, αλλά πλέον το κλειδί θα είναι διαφορετικό και δεν υπάρχει λόγος ανησυχίας.

Η φάση της αποκρυπτογράφησης έχει περίπου τις ίδιες εισόδους με τη φάση της κρυπτογράφησης. Τα PN και KeyID, που χρειάζονται κατά την αποκρυ-

πτογράφηση, είναι διαθέσιμα μέσω της επικεφαλίδας του CCMP. Ο Replay Counter είναι ένας τοπικός μετρητής που διατηρεί η συσκευή για την αποκρυπτογράφηση, ο οποίος αρχικοποιείται στην τιμή μηδέν κάθε φορά που η συσκευή ξεκινάει τη λειτουργία της ή αλλάζει την τιμή του κλειδιού κρυπτογράφησης. Στη συνέχεια ο Replay Counter ενημερώνεται με την τιμή PN, μετά την επιτυχημένη παραλαβή ενός MPDU.

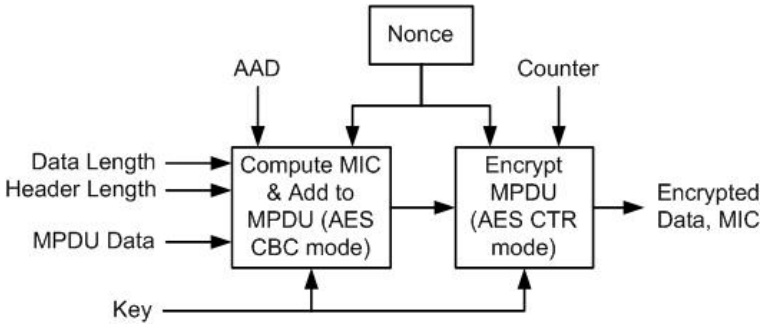
4.4.1 Επεξεργασία του MPDU κατά την κρυπτογράφηση

Το πρωτόκολλο CCMP κατά την κρυπτογράφηση των δεδομένων του MPDU προσθέτει στο αρχικό πακέτο την επικεφαλίδα του CCMP και το MIC. Ολόκληρη η διαδικασία της επεξεργασίας, που φαίνεται στο Σχήμα 4.7, ονομάζεται encapsulation.



Σχήμα 4.7 Το διάγραμμα της διαδικασίας encapsulation.

Το encapsulation αποτελείται από δύο κύρια στάδια. Πρώτα υπολογίζεται το MIC και προσαρτάται στο τέλος του MPDU. Στη συνέχεια ολόκληρο το MPDU μαζί με το MIC κρυπτογραφείται ώστε να προκύψει το τελικό αποτέλεσμα, όπως φαίνεται στο Σχήμα 4.8.



Σχήμα 4.8 Το μπλοκ διάγραμμα της κρυπτογράφησης.

Αναλυτικά τα βήματα του encapsulation είναι τα ακόλουθα:

1. Αυξάνεται ο μετρητής PN, ώστε να προκύψει μία νέα τιμή για κάθε MPDU. Με τον τρόπο αυτό δεν επαναλαμβάνεται ποτέ η ίδια τιμή του PN για το ίδιο κλειδί κρυπτογράφησης. Τα MPDU που αναμεταδίδονται δεν τροποποιούνται κατά την αναμετάδοση.

2. Χρησιμοποιούνται τα πεδία της επικεφαλίδας του MPDU για τη δημιουργία των δεδομένων AAD, που χρειάζεται ο CCM αλγόριθμος. Ο αλγόριθμος CCM διασφαλίζει την ακεραιότητα των πεδίων που περιλαμβάνονται στα δεδομένα AAD. Τα πεδία, ή ορισμένα μόνο bits, που μπορεί να αλλάξουν τιμή κατά την αναμετάδοση των MPDU, τίθενται στην τιμή μηδέν κατά τη δημιουργία των δεδομένων AAD.

3. Κατασκευάζεται η τιμή αρχικοποίησης (nonce) μήκους 104 bits για το CCM.

4. Δημιουργείται η επικεφαλίδα του CCMP από το PN και το KeyID, όπως είδαμε στη §4.3.2.

5. Χρησιμοποιείται το κλειδί κρυπτογράφησης, τα δεδομένα AAD, το nonce και τα δεδομένα του MPDU για τον υπολογισμό του MIC. Στη συνέχεια τα δεδομένα του MPDU και το MIC κρυπτογραφούνται.

6. Προστίθεται η αρχική επικεφαλίδα του MPDU, καθώς και η επικεφαλίδα του CCMP. Το κρυπτογραφημένο MPDU είναι πλέον έτοιμο να μπει στις ουρές μετάδοσης.

Παρόλο που το μήκος του MIC είναι το μισό του μπλοκ δεδομένων για τον αλγόριθμο AES, είναι αρκετό ώστε να μειώσει την πιθανότητα επιτυχημένης πλαστογράφησης του MIC στο 10^{-19} [3].

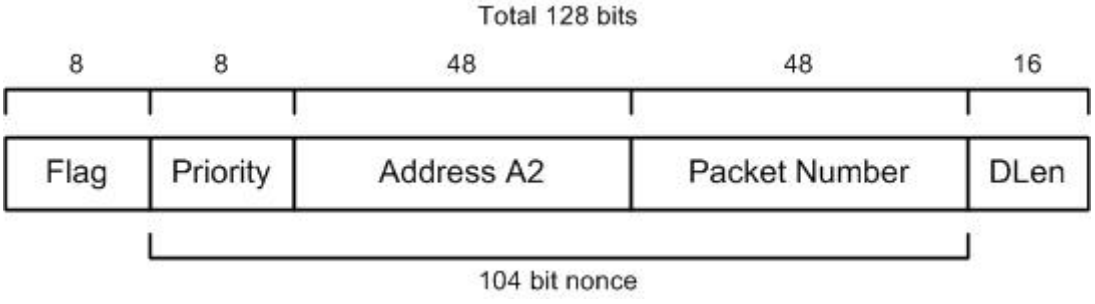
4.4.2 Υπολογισμός του MIC

Ο υπολογισμός του MIC γίνεται χρησιμοποιώντας την κατάσταση λειτουργίας CBC για τον αλγόριθμο AES. Για την εκκίνηση της διαδικασίας κρυπτογραφείται ένα μπλοκ αρχικοποίησης, το MIC_IV και στη συνέχεια κρυπτογραφείται

το αποτέλεσμα της πράξης XOR ανάμεσα στο κρυπτογραφημένο μπλοκ και το επόμενο μπλοκ. Η διαδικασία συνεχίζεται μέχρι το τέλος του MPDU.

Το μπλοκ MIC_IV έχει μήκος 128 bits και δεν προέρχεται από το MPDU, αλλά προκύπτει με ειδικό τρόπο χρησιμοποιώντας την τιμή αρχικοποίησης (nonce). Η δομή αυτού του μπλοκ φαίνεται στο Σχήμα 4.9.

Η τιμή αρχικοποίησης κατασκευάζεται από τα πεδία τα PN, A2 (Source Address) και το πεδίο προτεραιότητας (Priority). Σκοπός της είναι να εξασφαλίσει ότι η εκκίνηση κάθε κρυπτογράφησης πραγματοποιείται με δεδομένα που δεν έχουν χρησιμοποιηθεί πιο πριν, για ένα συγκεκριμένο κλειδί. Κάποιος μπορεί να υποθέσει λανθασμένα ότι για το nonce μπορούμε απλά να χρησιμοποιήσουμε την τιμή PN, αφού αυτή αυξάνει για κάθε MPDU και δεν επαναλαμβάνεται. Παρόλα αυτά, πρέπει να έχουμε υπόψη ότι το κλειδί κρυπτογράφησης είναι ίδιο για τους δύο ή περισσότερους χρήστες που επικοινωνούν μεταξύ τους και μπορεί κατά τη διάρκεια της επικοινωνίας κάποιος χρήστης να χρησιμοποιήσει μία τιμή του PN που έχει επαναληφθεί από άλλο χρήστη. Με τον τρόπο αυτό παραβιάζεται ο κανόνας της «μοναδικής χρήσης για συγκεκριμένο κλειδί», που είναι απαραίτητος ώστε να διατηρηθεί το υψηλό επίπεδο ασφάλειας που εγγυάται το πρωτόκολλο CCMP. Για να αποφευχθεί αυτό το πρόβλημα η τιμή αρχικοποίησης προκύπτει από το συνδυασμό του PN με τη MAC διεύθυνση του αποστολέα. Το τρίτο πεδίο που περιλαμβάνεται στην τιμή αρχικοποίησης είναι το πεδίο προτεραιότητας. Το πεδίο αυτό είναι δεσμευμένο για μελλοντική χρήση και προς το παρόν τίθεται στην τιμή μηδέν. Μελλοντικά θα λαμβάνει διάφορες τιμές ανάλογα με τα χαρακτηριστικά της ροής των δεδομένων που αποστέλλονται στο ασύρματο δίκτυο, όπως ήχος, βίντεο κτλ. Σε αυτές τις περιπτώσεις θα είναι χρήσιμο να υπάρχουν διαφορετικές τιμές του πεδίου προτεραιότητας για κάθε τύπο δεδομένων.



Σχήμα 4.9 Η δομή του μπλοκ MIC_IV.

Το δεύτερο πεδίο που χρησιμοποιείται για το σχηματισμό του MIC_IV είναι το πεδίο σηματοδότησης (Flag), που έχει μήκος ένα byte. Το πεδίο αυτό έχει σταθερή τιμή για τη λειτουργία του πρωτοκόλλου CCMP, στο περιβάλλον ενός δικτύου RSN, ίση με 01011001 ή {59} σε δεκαεξαδική αναπαράσταση. Η τιμή αυτή κωδικοποιεί τις παραμέτρους M και L, που απαιτούνται για την κατάσταση λειτουργίας CCM, με τον τρόπο που φαίνεται στον παρακάτω πίνακα.

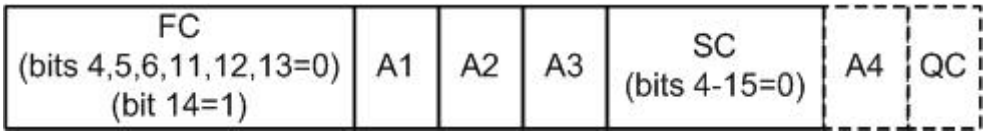
Παράμετρος	Περιγραφή	Μήκος	Κωδικοποίηση
M	Μέγεθος του MIC σε bytes	3 bits	(M – 2)/2
L	Μέγεθος του DLen σε bytes	3 bits	L – 1

Παρακάτω φαίνεται η δομή του πεδίου Flag, όπου το πιο σημαντικό bit έχει την τιμή 0 και το Adata bit έχει την τιμή 1, αφού χρησιμοποιούνται δεδομένα AAD. Εφόσον το πρωτόκολλο CCMP ορίζει ότι το μέγεθος του MIC είναι 64 bits (M = 8) και το μέγεθος του DLen είναι 16 bits (L = 2), προκύπτει βάση της κωδικοποίησης η παραπάνω τιμή για το Flag.

Bit no:	7	6	5	4	3	2	1	0
Content:	Rsvd	Adata	M			L		

Τέλος, το πεδίο DLen δείχνει το μήκος των δεδομένων του MPDU σε bytes.

Το δεύτερο και τρίτο μπλοκ μήκους 128 bits που χρησιμοποιούνται στην κατάσταση λειτουργίας CBC, μετά το MIC_IV, ονομάζονται MIC_HDR1 και MIC_HDR2 αντίστοιχα και αποτελούν στην ουσία τα δεδομένα AAD. Τα δεδομένα AAD προκύπτουν από τη MAC επικεφαλίδα θέτοντας στην τιμή μηδέν ορισμένα bits του πεδίου FC και SC. Οι τροποποιήσεις που γίνονται στη MAC επικεφαλίδα φαίνονται στο Σχήμα 4.10 και περιγράφονται συνοπτικά παρακάτω.



Σχήμα 4.10 Η κατασκευή των δεδομένων AAD.

1. Πεδίο Duration (Dur): Δεν περιλαμβάνεται
2. Πεδίο Frame Control (FC):
    i. Τίθενται στην τιμή 0 τα Subtype bits (bits 4, 5, 6)

ii. Τίθεται στην τιμή 0 το Retry bit (bit 11)

iii. Τίθεται στην τιμή 0 το PwrMgt bit (bit 12)

iv. Τίθεται στην τιμή 0 το MoreData bit (bit 13)

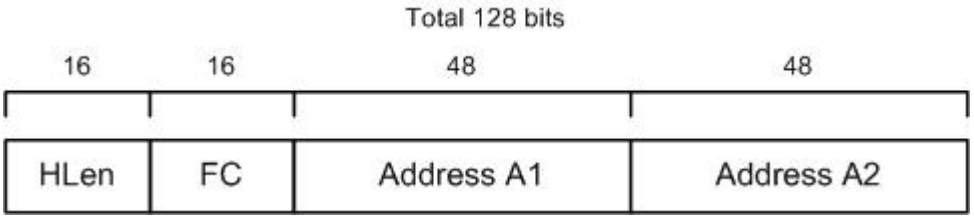
v. Τίθεται στην τιμή 1 το Protected Frame bit (bit 14)

3. Πεδίο Sequence Control (SC):

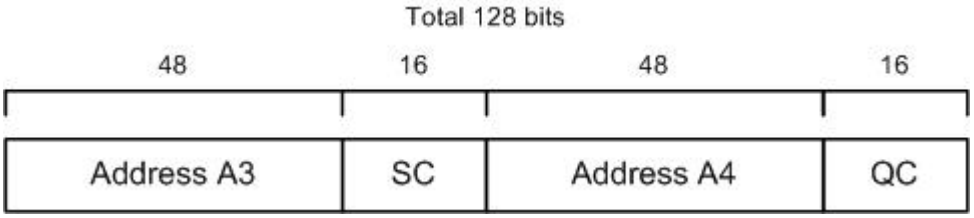
i. Τίθενται στην τιμή 0 τα Sequence Number bits (bits 4 – 15)

ii. Δεν τροποποιούνται τα Fragment Number bits (bits 0 – 3)

Η δομή των μπλοκ MIC_HDR1 και MIC_HDR2 φαίνεται στο Σχήμα 4.11 και 4.12 αντίστοιχα. Το πεδίο HLen δείχνει το μήκος της MAC επικεφαλίδας σε bytes, χωρίς να λαμβάνεται υπόψη το πεδίο Duration.



Σχήμα 4.11 Η δομή του μπλοκ MIC_HDR1.

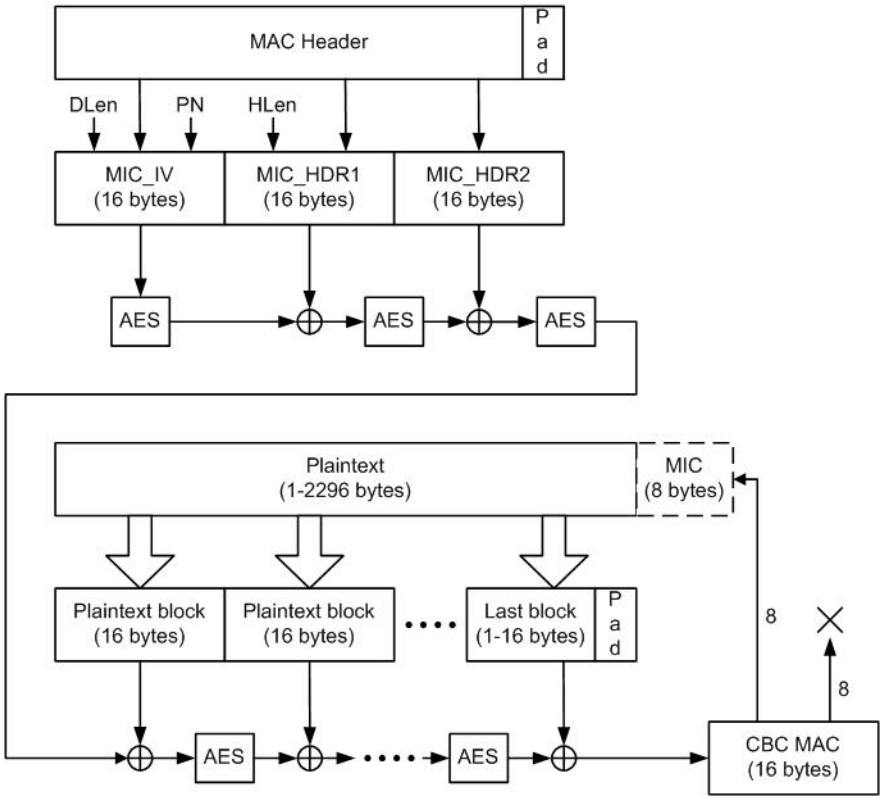


Σχήμα 4.12 Η δομή του μπλοκ MIC_HDR2.

Στη συνέχεια ακολουθούν τα μπλοκ των δεδομένων μήκους 128 bits, ώσπου να τελειώσει το MPDU. Προκύπτει με αυτό τον τρόπο ένα μπλοκ μήκους 128 bits, από τα οποία τα 64 πιο σημαντικά bits αποτελούν το MIC. Η διαδικασία υπολογισμού του MIC φαίνεται συνολικά στο Σχήμα 4.13.

Σημειώνουμε ότι ένα χαρακτηριστικό της κατάστασης λειτουργίας CBC είναι ότι απαιτείται το μήκος των δεδομένων να είναι πολλαπλάσιο του μπλοκ. Επομένως, το πρωτόκολλο CCMP απαιτεί τόσο τα δεδομένα AAD, όσο και τα δεδομένα του πακέτου να διαιρούνται ακριβώς σε μπλοκ. Αν αυτό δεν ισχύει θα πρέπει να συμπληρωθούν τα κενά μπλοκ, δηλαδή το MIC_HDR2 και το

τελευταίο μπλοκ δεδομένων του MPDU, με μηδενικά bytes (zero padding). Τα bytes αυτά εισάγονται μόνο για τον υπολογισμό του MIC και δεν εισάγονται στο τελικό MPDU.



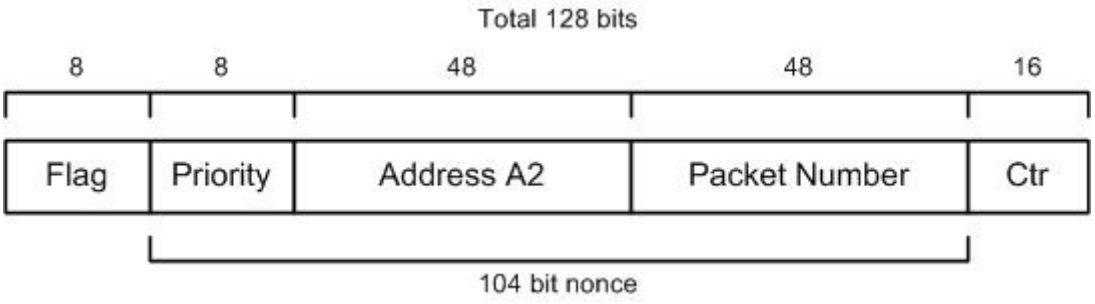
Σχήμα 4.13 Υπολογισμός του MIC.

4.4.3 Κρυπτογράφηση του MPDU

Μετά τον υπολογισμό του MIC και την εισαγωγή του στο τέλος των δεδομένων, ακολουθεί η διαδικασία της κρυπτογράφησης του MPDU. Η κρυπτογράφηση πραγματοποιείται χρησιμοποιώντας την κατάσταση λειτουργίας CTR του αλγόριθμου AES. Εξαιτίας του zero padding όπου ήταν απαραίτητο, είναι σίγουρο ότι τα μπλοκ που κρυπτογραφούνται είναι στοιχισμένα με τα μπλοκ που χρησιμοποιήθηκαν για τον υπολογισμό του MIC. Τα κρυπτογραφημένα δεδομένα αντικαθιστούν τα αρχικά δεδομένα, τόσο για το κομμάτι των δεδομένων, όσο και για το MIC. Προκύπτει με αυτό τον τρόπο το τελικό κρυπτογραφημένο MPDU, έτοιμο να εισέλθει στις ουρές μετάδοσης. Παρατηρείστε ότι δεν είναι απαραίτητο να χρησιμοποιηθεί zero padding για τη φάση της κρυπτογράφησης, καθώς η κατάσταση λειτουργίας CTR επιτρέπει την απόρριψη των επιπλέον bits της τελευταίας τιμής του μετρητή. Διατηρούνται δηλαδή

μόνο τα απαραίτητα bits για την κρυπτογράφηση του τελευταίου μπλοκ δεδομένων, το οποίο μπορεί να μην είναι συμπληρωμένο.

Ένα απαραίτητο στοιχείο για την κατάσταση λειτουργίας CTR είναι η αρχικοποίηση της τιμής του μπλοκ μετρητή (Counter) μήκους 128 bits, με τέτοιο τρόπο ώστε να μη χρησιμοποιηθεί ξανά η ίδια τιμή στη συνέχεια. Επομένως, ο μπλοκ μετρητής κατασκευάζεται βάση μίας τιμής αρχικοποίησης (nonce) με σχεδόν ίδιο τρόπο, όπως στην περίπτωση του MIC_IV. Στην πραγματικότητα η τιμή του nonce είναι η ίδια με πριν και αποτελείται από τα PN, Source Address και το πεδίο Priority. Το nonce στη συνέχεια συνδυάζεται με δύο άλλα πεδία: το byte σηματοδότησης (Flag), το οποίο έχει την τιμή {01} σε δεκαεξαδική αναπαράσταση και την τιμή ενός μετρητή μήκους 16 bits (Ctr), όπως φαίνεται στο Σχήμα 4.14.

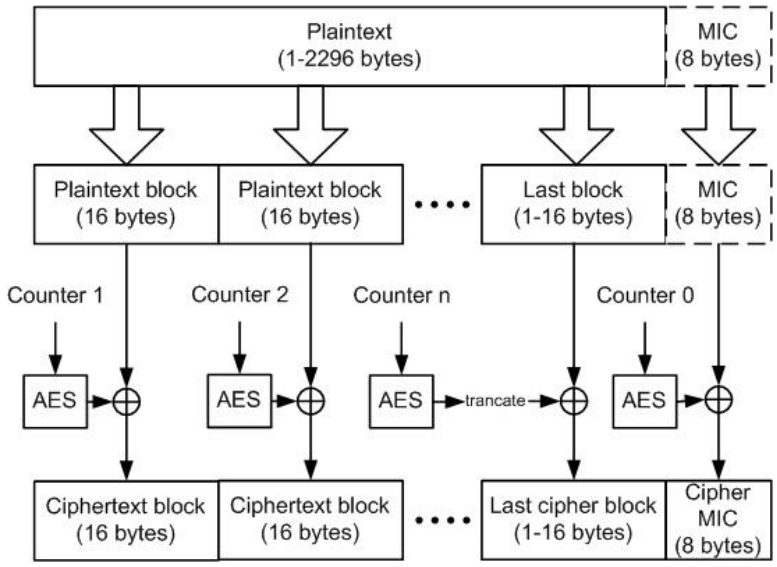


Σχήμα 4.14 Η δομή του μπλοκ μετρητή για την κατάσταση λειτουργίας CTR.

Η τιμή του μετρητή Ctr αρχίζει από το 1 και αυξάνει με κάθε μπλοκ δεδομένων καθώς προχωράει η διαδικασία της κρυπτογράφησης. Επειδή η τιμή του nonce είναι μοναδική και το πεδίο του μετρητή Ctr έχει μήκος 16 bits, είναι βέβαιο πως θα υπάρχουν διαθέσιμες μοναδικές τιμές για το μπλοκ μετρητή για οποιοδήποτε μήνυμα με λιγότερο από 65536 μπλοκ. Αυτή η τιμή καλύπτει ακόμα και το μεγαλύτερο μέγεθος MPDU που επιτρέπει το πρωτόκολλο 802.11.

Όταν αρχικοποιηθεί ο μετρητής, η κρυπτογράφηση προχωράει με τον τρόπο που περιγράφηκε στη §3.7.2. Κάθε διαδοχική τιμή του μετρητή κρυπτογραφείται χρησιμοποιώντας το κλειδί κρυπτογράφησης. Το κρυπτογραφημένο MPDU προκύπτει από το αποτέλεσμα της πράξης XOR ανάμεσα στην κρυπτογραφημένη τιμή του μετρητή και το μπλοκ των δεδομένων. Η τιμή του με-

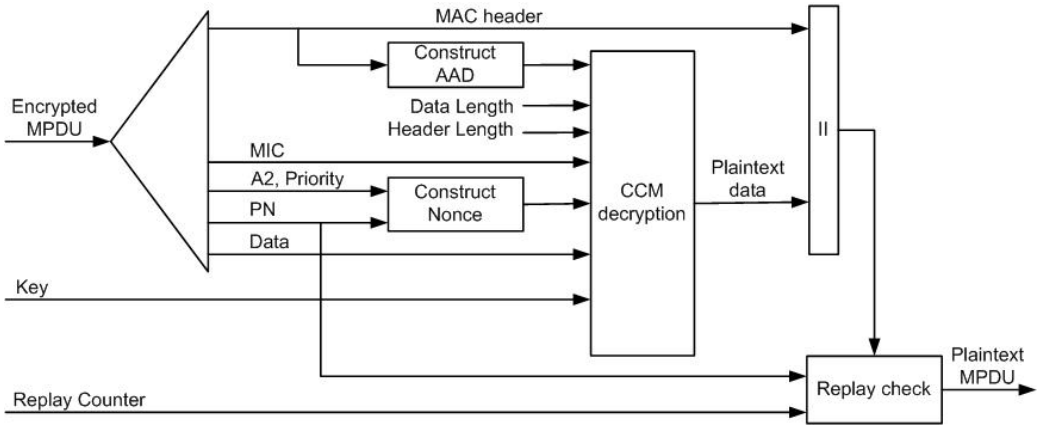
τρητή για Ctr = 0 χρησιμοποιείται για την κρυπτογράφηση του MIC. Η διαδικασία της κρυπτογράφησης φαίνεται συνολικά στο Σχήμα 4.15.



Σχήμα 4.15 Κρυπτογράφηση του MPDU.

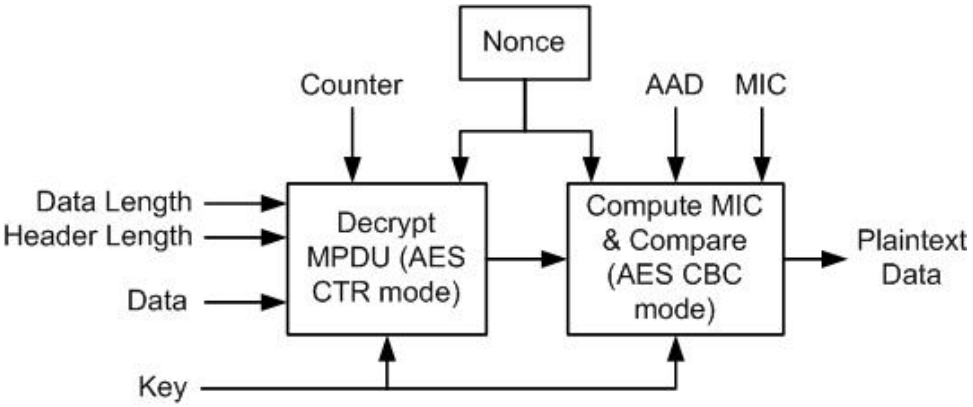
4.4.4 Επεξεργασία του MPDU κατά την αποκρυπτογράφηση

Όταν το κρυπτογραφημένο MPDU παραδίδεται στον αποδέκτη, πρέπει να βρεθεί το σωστό κλειδί για την αποκρυπτογράφηση. Το κατάλληλο κλειδί επιλέγεται ανάλογα με τη MAC διεύθυνση του αποστολέα στη MAC επικεφαλίδα. Υπάρχει μία σειρά από βήματα που οφείλει να ακολουθήσει ο παραλήπτης ώστε να εξάγει και να ελέγξει τη γνησιότητα των δεδομένων που παρέλαβε. Η αποκρυπτογράφηση είναι ένα βήμα αυτής της διαδικασίας, η οποία ονομάζεται decapsulation. Στο Σχήμα 4.16 φαίνεται το μπλοκ διάγραμμα της διαδικασίας decapsulation.



Σχήμα 4.16 Το μπλοκ διάγραμμα της διαδικασίας decapsulation.

Η αποκρυπτογράφηση αποτελείται από δύο κύρια στάδια. Πρώτα αποκρυπτογραφείται ολόκληρο το MPDU μαζί με το MIC χρησιμοποιώντας την κατάσταση λειτουργίας CTR του αλγόριθμου AES. Στη συνέχεια υπολογίζεται ξανά το MIC, όπως στην περίπτωση της κρυπτογράφησης και συγκρίνεται με το αποκρυπτογραφημένο MIC για έλεγχο της ακεραιότητας, όπως φαίνεται στο Σχήμα 4.17. Αν ταιριάζουν, τότε τα δεδομένα θεωρούνται έγκυρα και προωθούνται στα ανώτερα επίπεδα για επεξεργασία.



Σχήμα 4.17 Το μπλοκ διάγραμμα της αποκρυπτογράφησης.

Αναλυτικά τα βήματα του decapsulation είναι τα ακόλουθα:

1. Η επανάληψη MPDU, που έχουν ήδη σταλεί, αποτρέπεται επιβεβαιώνοντας ότι η τιμή PN, που περιέχεται χωρίς κρυπτογράφηση στη CCMP επικεφαλίδα του MPDU, είναι μεγαλύτερη από την τιμή του Replay Counter, που διατηρείται για τη συγκεκριμένη μετάδοση. Αν η τιμή PN είναι μικρότερη ή ίση με την τιμή του Replay Counter τότε τα δεδομένα απορρίπτονται και η διαδικασία για το συγκεκριμένο MPDU διακόπτεται[9].

2. Δημιουργούνται τα δεδομένα AAD, που χρειάζεται ο CCM αλγόριθμος, από τα πεδία της επικεφαλίδας του MPDU, η οποία αποστέλλεται χωρίς κρυπτογράφηση. Κατά τη δημιουργία των δεδομένων AAD τίθενται στην τιμή μηδέν τα ίδια πεδία, ή bits, που τέθηκαν στην τιμή μηδέν κατά τη διαδικασία του encapsulation.

[9] Στο πρότυπο IEEE 802.11i το βήμα αυτό εμφανίζεται τελευταίο κατά τη διαδικασία του decapsulation. Το ορθό είναι να γίνει ο έλεγχος της τιμής του PN από την αρχή ώστε να αποφευχθεί η επεξεργασία, αφού πρόκειται ουσιαστικά για επανάληψη παλαιότερου MPDU.

3. Σχηματίζεται η τιμή αρχικοποίησης (nonce) από τα πεδία A2, PN και Priority, ώστε να είναι η ίδια με την τιμή που χρησιμοποιήθηκε κατά τη διαδικασία του encapsulation.

4. Ο παραλήπτης χρησιμοποιεί το κλειδί κρυπτογράφησης, τα δεδομένα AAD και την τιμή αρχικοποίησης, όπως στην περίπτωση της κρυπτογράφησης, καθώς και τα κρυπτογραφημένα δεδομένα και MIC ώστε να προκύψουν τα αποκρυπτογραφημένα δεδομένα και MIC.

5. Υπολογίζεται το MIC από την αρχή και συγκρίνεται με την αποκρυπτογραφημένη τιμή για τον έλεγχο της ακεραιότητας.

6. Αν ήταν επιτυχημένος ο έλεγχος της ακεραιότητας, η MAC επικεφαλίδα του MPDU που παραλήφθηκε ενώνεται με τα αποκρυπτογραφημένα δεδομένα ώστε να προκύψει το αρχικό MPDU.

Παρατηρείστε ότι για την αποκρυπτογράφηση του MPDU απαιτείται ο υπολογισμός της τιμής αρχικοποίησης για το μετρητή που χρησιμοποιείται στην κατάσταση λειτουργίας CTR. Η τιμή αυτή πρέπει να είναι ίδια με τη τιμή που είχε προκύψει κατά την κρυπτογράφηση των δεδομένων. Όλη η απαραίτητη πληροφορία περιέχεται, χωρίς να είναι κρυπτογραφημένη στο MPDU που παραλήφθηκε. Η τιμή PN συνδυάζεται με τη διεύθυνση του αποστολέα (A2) και το πεδίο προτεραιότητας για τη δημιουργία του nonce. Το nonce στη συνέχεια συνδυάζεται με την ίδια τιμή Flag και την ίδια ακριβώς τιμή του 16 bit μετρητή Ctr, δηλαδή την τιμή ένα, ώστε να προκύψει το αρχικό μπλοκ μετρητή μήκους 128 bit. Δεν υπάρχει κανένα απολύτως μυστικό στη διαδικασία αυτή. Οποιοσδήποτε εχθρός μπορεί να υπολογίσει τον αρχικό μπλοκ μετρητή για την κατάσταση λειτουργίας CTR. Παρόλα αυτά, του είναι άχρηστος, εκτός αν βρει τρόπο να αποκτήσει το κλειδί της κρυπτογράφησης. Στη συνέχεια η αποκρυπτογράφηση προχωράει όπως ακριβώς η κρυπτογράφηση. Διαδοχικές τιμές του μπλοκ μετρητή κρυπτογραφούνται και χρησιμοποιούνται μαζί με τα δεδομένα που παραλήφθηκαν για να προκύψουν τα αρχικά δεδομένα και το MIC, μέσω της πράξης XOR.

Στη συνέχεια υπολογίζεται από την αρχή η τιμή του MIC. Είναι προφανές ότι αν τα δεδομένα δεν τροποποιήθηκαν από τη στιγμή της αποστολής τους και

χρησιμοποιηθεί το σωστό κλειδί, θα προκύψει η ίδια τιμή MIC, οπότε το MPDU γίνεται αποδεκτό. Σε διαφορετική περίπτωση το πιο πιθανό είναι ότι κάποιος προσπάθησε να αλλάξει τα αρχικά δεδομένα και το MPDU πρέπει να απορριφθεί. Μετά την απομάκρυνση της MAC επικεφαλίδας, της CCMP επικεφαλίδας και του MIC, το αποκρυπτογραφημένο MPDU ενώνεται με τα προηγούμενα ώστε να σχηματιστεί τελικά το MSDU, που χρησιμοποιείται σε επίπεδο εφαρμογής.

Το πιο ενδιαφέρον χαρακτηριστικό της διαδικασίας αποκρυπτογράφησης είναι ότι είναι σχεδόν ίδια με τη διαδικασία της κρυπτογράφησης. Αυτό έχει ως αποτέλεσμα να απλοποιείται σημαντικά η υλοποίηση του πρωτοκόλλου CCMP. Δε χρειάζεται να υλοποιηθεί ο αντίστροφος αλγόριθμος κρυπτογράφησης (Inverse Cipher) και μειώνονται με τον τρόπο αυτό οι απαιτήσεις σε επιφάνεια, στην περίπτωση που το πρωτόκολλο υλοποιείται εξ ολοκλήρου σε υλικό.

5 Υλοποίηση του αλγορίθμου AES

5.1 Σύνοψη υλοποιήσεων

Το κύκλωμα που υλοποιεί το πρωτόκολλο MAC στις Wi-Fi συσκευές αποτελείται από ένα μικροεπεξεργαστή μαζί με το απαραίτητο firmware, μία μνήμη τυχαίας προσπέλασης (Random Access Memory - RAM) και μία μονάδα υλικού (Hardware Assist), η οποία υλοποιεί τον αλγόριθμο κρυπτογράφησης με σκοπό να αποφορτίσει το μικροεπεξεργαστή και να αυξήσει το συνολικό ρυθμό επεξεργασίας δεδομένων (throughput) της συσκευής. Η μονάδα υλικού υλοποιείται με τη μορφή σχεδιασμού ASIC ή FPGA. Τα FPGA είναι πλέον κατάλληλα για εφαρμογές όπου απαιτείται η κρυπτογράφηση δεδομένων σε πολύ υψηλές ταχύτητες. Μπορούν να παρέχουν με τη σημερινή τεχνολογία την απαιτούμενη απόδοση, χωρίς το υψηλό κόστος της διαδικασίας ενός ASIC σχεδιασμού. Επιπλέον, όταν απαιτείται περισσότερα από ένα πρωτόκολλα να υλοποιηθούν κάτω από την ίδια πλατφόρμα, τα FPGA έχουν το πλεονέκτημα ότι μπορούν να αλλάξουν τη λειτουργία τους (reconfiguration) σε σύντομο χρονικό διάστημα, ακόμα και κατά τη διάρκεια της λειτουργίας του συστήματος.

Από το 2001 που ο αλγόριθμος AES έγινε πρότυπο από το NIST, έχουν παρουσιαστεί στη βιβλιογραφία πάρα πολλές υλοποιήσεις του αλγορίθμου σε τεχνολογία FPGA, ενώ παράλληλα υπάρχουν διαθέσιμοι στην αγορά πολλοί «πυρήνες» του AES (IP cores) που χρησιμοποιούνται στη διαδικασία σχεδιασμού προϊόντων, όταν απαιτείται η χρήση μίας μονάδας κρυπτογράφησης που βασίζεται στον AES.

Υπάρχουν διάφορες κατηγοριοποιήσεις των υλοποιήσεων. Η πιο σημαντική από αυτές βασίζεται στο συμβιβασμό (trade-off) ανάμεσα στην απόδοση, ανάλογα με το throughput που επιτυγχάνεται και τους πόρους, ανάλογα με τον αριθμό των CLB και της συνολικής επιφάνειας του FPGA μετά τη δρομολόγηση του σχεδιασμού (routing).

Από τη μία πλευρά, υπάρχουν οι υλοποιήσεις που λειτουργούν σε πολύ υψηλές συχνότητες και επιτυγχάνουν ταχύτητες επεξεργασίας δεδομένων της τά-

ξης των 20 Gbps ([15, 16, 17, 18, 19, 33]). Οι υλοποιήσεις αυτές είναι εξαιρετικά απαιτητικές όσον αφορά την τελική επιφάνεια που καταλαμβάνουν και μόνο οι πολύ μεγάλες συσκευές FPGA μπορούν να τις χωρέσουν. Για να επιτύχουν τους υψηλούς ρυθμούς επεξεργασίας, οι υλοποιήσεις αυτές χρησιμοποιούν κυρίως τρεις αρχιτεκτονικές βελτιστοποιήσεις: το «ξεδίπλωμα» των επαναλήψεων του αλγόριθμου AES (loop unrolling), τη διοχέτευση δεδομένων ανάμεσα σε κάθε επανάληψη με τη βοήθεια ενδιάμεσων καταχωρητών (pipeline registers) και τη διοχέτευση δεδομένων ανάμεσα στα στάδια επεξεργασίας της ίδιας επανάληψης με τη βοήθεια ενδιάμεσων καταχωρητών (subpipeline registers). Οι περισσότερες από αυτές υλοποιούν το μετασχηματισμό SubBytes με χρήση συνδυαστικής λογικής, δηλαδή πολύπλοκες πράξεις στο πεδίο GF(2^8) όπως η αντιστροφή ([15, 16, 17, 20, 21, 22, 23]).

Ενδιάμεσα υπάρχουν οι υλοποιήσεις, οι οποίες παρέχουν υψηλούς ρυθμούς επεξεργασίας μέχρι 2 Gbps, ενώ παράλληλα απαιτούν σημαντικά λιγότερους πόρους σε σχέση με την προηγούμενη κατηγορία. Για αυτό το λόγο είναι κατάλληλες για υλοποίηση σε μεσαίου μεγέθους FPGA. Βασίζονται κυρίως στην υλοποίηση μίας επανάληψης του αλγορίθμου AES στο FPGA και την επαναληπτική ανατροφοδότηση των δεδομένων μέχρι την ολοκλήρωση της λειτουργίας του [19, 23, 29]. Για να μειωθεί ακόμα περισσότερο η επιφάνεια που απαιτείται, οι σχεδιασμοί αυτής της κατηγορίας χρησιμοποιούν look-up table (LUT) για την εκτέλεση του μετασχηματισμού SubBytes. Η υλοποίηση των LUTs γίνεται κυρίως με χρήση της ενσωματωμένης μνήμης του FPGA ([24, 25, 26, 27, 28]). Σε αυτές τις αρχιτεκτονικές του AES παρατηρείται ότι η καθυστέρηση των LUT, για το μετασχηματισμό SubBytes, είναι μεγαλύτερη από την καθυστέρηση των υπόλοιπων μετασχηματισμών σε κάθε επανάληψη [15]. Το χαρακτηριστικό αυτό δεν επιτρέπει τον διαχωρισμό κάθε επανάληψης σε περισσότερα από δύο στάδια για να επιτευχθεί μεγαλύτερη αύξηση της ταχύτητας του σχεδιασμού, στα επίπεδα της προηγούμενης κατηγορίας. Οι υλοποιήσεις αυτής της κατηγορίας είναι κατάλληλες όταν ο αλγόριθμος AES χρησιμοποιείται σε κατάσταση λειτουργίας που απαιτείται ανατροφοδότηση των κρυπτογραφημένων δεδομένων (feedback mode), όπως είναι η κατάσταση λειτουργίας CBC που χρησιμοποιείται στο πρωτόκολλο CCMP. Είναι προφανές ότι οι ταχύτατες υλοποιήσεις της προηγούμενης κατηγορίας δεν

είναι κατάλληλες για αυτές τις καταστάσεις λειτουργίας, αφού παραμένουν κενοί οι καταχωρητές διοχέτευσης ανάμεσα στις επαναλήψεις του αλγορίθμου.

Στην άλλη πλευρά υπάρχουν οι υλοποιήσεις που παρέχουν ρυθμούς επεξεργασίας της τάξης των 200 Mbps και ταυτόχρονα χρειάζονται ελάχιστους πόρους ([30, 31]). Η κύρια τεχνική με την οποία επιτυγχάνεται η δραστική μείωση της επιφάνειας και της ενσωματωμένης μνήμης του FPGA ονομάζεται «δίπλωμα» του αλγορίθμου (folding). Με την τεχνική αυτή η επανάληψη του αλγορίθμου εκτελείται συνήθως σε τέσσερις κύκλους, όπου σε κάθε κύκλο γίνεται η επεξεργασία των αντίστοιχων 32 bits του μπλοκ δεδομένων, που έχει μήκος 128 bits. Τέτοιες υλοποιήσεις είναι κατάλληλες για χρήση κυρίως σε smart card.

Η δεύτερη κατηγοριοποίηση των υλοποιήσεων βασίζεται στον υπολογισμό και χειρισμό των προσωρινών κλειδιών[10] που χρησιμοποιούνται σε κάθε επανάληψη του αλγορίθμου AES. Θυμίζουμε ότι τα προσωρινά κλειδιά προκύπτουν από το κλειδί της κρυπτογράφησης μέσω της διαδικασίας επέκτασης κλειδιών (βλέπε §3.6, ρουτίνα KeyExpansion).

Η πρώτη κατηγορία βασίζεται στο δυναμικό υπολογισμό κάθε νέου προσωρινού κλειδιού σε κάθε επανάληψη του αλγόρίθμου, χρησιμοποιώντας το προηγούμενο προσωρινό κλειδί. Η προσέγγιση αυτή ονομάζεται on-the-fly ή online υπολογισμός του κλειδιού κρυπτογράφησης για κάθε επανάληψη [15, 16, 29, 32].

Η δεύτερη κατηγορία βασίζεται στον υπολογισμό όλων των απαραίτητων προσωρινών κλειδιών, πριν την έναρξη της διαδικασίας κρυπτογράφησης (offline). Τα κλειδιά υπολογίζονται μόλις είναι διαθέσιμο το αρχικό κλειδί της κρυπτογράφησης και αποθηκεύονται σε μία μικρή μνήμη [29, 31, 33]. Αυτή η υλοποίηση έχει το πλεονέκτημα ότι διαχωρίζει τον υπολογισμό του κλειδιού κάθε επανάληψης από την εκτέλεση της επανάληψης του αλγορίθμου. Με τον τρόπο αυτό μειώνεται σημαντικά το κρίσιμο μονοπάτι του σχεδιασμού. Επιπλέον, μετά τον υπολογισμό των προσωρινών κλειδιών αυτά μπορούν να χρησιμοποιηθούν για την κρυπτογράφηση δεδομένων με μέγεθος μεγαλύτερο

[10] Στο Κεφάλαιο 5 ο όρος «προσωρινά κλειδιά» υπονοεί τα δέκα κλειδιά που χρησιμοποιεί ο αλγόριθμος AES για την ολοκλήρωση των δέκα επαναλήψεων και δεν πρέπει να συγχέεται με την έννοια που χρησιμοποιήθηκε στη §1.6, ως μέρος της ιεραρχίας κλειδιών.

από 128 bits, όπως αρχεία ή πακέτα εφόσον δεν αλλάζει το αρχικό κλειδί. Αντίθετα, ο on-the-fly υπολογισμός των προσωρινών κλειδιών σπαταλάει πόρους και ισχύ άσκοπα στην περίπτωση που το αρχικό κλειδί κρυπτογράφησης παραμένει το ίδιο για μεγάλο αριθμό από μπλοκ δεδομένων, όπως συμβαίνει συνήθως στις ασύρματες επικοινωνίες. Παρόλα αυτά, η offline προσέγγιση έχει το κόστος της μεγαλύτερης καθυστέρησης (latency) για την παραγωγή του πρώτου κρυπτογραφημένου μπλοκ δεδομένων. Το κόστος αυτό δεν είναι σημαντικό, αλλά για μερικές εφαρμογές, οι οποίες απαιτούν συχνή αλλαγή του αρχικού κλειδιού, μπορεί να είναι ανεπίτρεπτο.

Υπάρχει τέλος μία κατηγορία υλοποιήσεων η οποία είναι κατάλληλη για συστήματα με περιορισμένους πόρους που δεν έχουν απαιτήσεις για υψηλή απόδοση. Σε αυτές τα περιπτώσεις, τα προσωρινά κλειδιά υπολογίζονται από μία εξωτερική πηγή, όπως μία γεννήτρια κλειδιών ή ένα λογισμικό που εκτελείται σε επεξεργαστή και φορτώνονται στο κύκλωμα κρυπτογράφησης ακολουθιακά [34]. Η ισχύς που εξοικονομείται από την απουσία του κυκλώματος για τον υπολογισμό των προσωρινών κλειδιών αντισταθμίζεται από την αυξημένη δραστηριότητα πάνω στην αρτηρία δεδομένων (bus), για τη μεταφορά των κλειδιών. Στην περίπτωση που η αρτηρία αυτή είναι εξωτερική, δηλαδή ο πυρήνας της κρυπτογράφησης δεν αποτελεί μέρος ενός ευρύτερου σχεδιασμού μέσα στο ίδιο ολοκληρωμένο κύκλωμα, τότε η κατανάλωση ισχύος αυξάνεται σημαντικά.

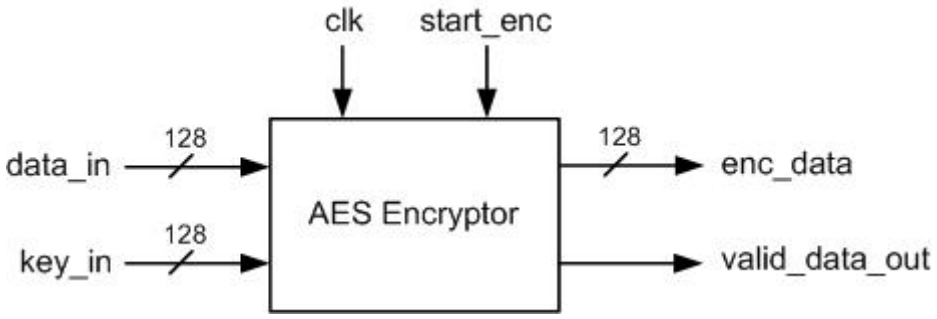
Στα πλαίσια της υλοποίησης του πρωτοκόλλου ασφαλείας CCMP, υλοποιήθηκαν διάφορες αρχιτεκτονικές για τον αλγόριθμο AES όσον αφορά τον υπολογισμό των προσωρινών κλειδιών (on-the-fly ή offline). Πιο συγκεκριμένα, υλοποιήθηκαν τρεις αρχιτεκτονικές, οι οποίες μελετήθηκαν και αξιολογήθηκαν μέσα από τη συνολική λειτουργία του πρωτοκόλλου CCMP:

1. Υπολογισμός των προσωρινών κλειδιών on-the-fly
2. Υπολογισμός των προσωρινών κλειδιών offline
3. Υπολογισμός των προσωρινών κλειδιών offline, με χρησιμοποίηση κοινών πόρων (resource sharing)

Υλοποιήσεις που χρησιμοποιούν το «ξεδίπλωμα» του αλγόριθμου δε μελετήθηκαν καθώς δεν έχουν εφαρμογή στην περίπτωση του πρωτοκόλλου CCMP. Οι υπόλοιπες σχεδιαστικές επιλογές, που έγιναν κατά την υλοποίηση του αλγορίθμου AES, αναλύονται σε σχέση με τα επιμέρους τμήματα του σχεδιασμού, αφού υπάρχουν αρκετές επιλογές για την υλοποίηση των μετασχηματισμών που απαρτίζουν τον AES.

5.2 Αρχιτεκτονική υπολογισμού κλειδιών on-the-fly

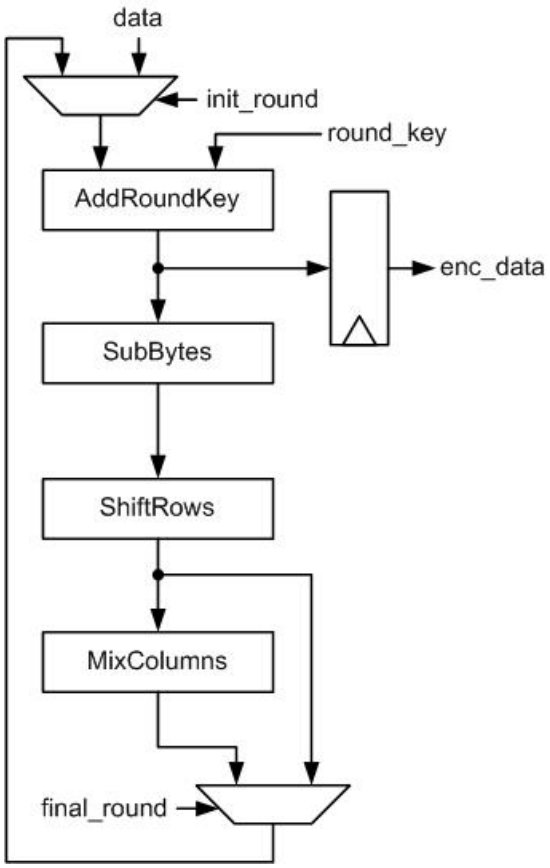
Η αρχιτεκτονική αυτή βασίζεται στον on-the-fly υπολογισμό των προσωρινών κλειδιών, που απαιτούνται για την ολοκλήρωση όλων των επαναλήψεων του αλγορίθμου AES. Αποτελείται από το κύκλωμα κρυπτογράφησης, το κύκλωμα υπολογισμού κλειδιών και τη μονάδα ελέγχου, που αναλαμβάνει να συγχρονίσει τα δύο προηγούμενα κυκλώματα για την ομαλή λειτουργία του συνολικού σχεδιασμού. Η μονάδα ελέγχου υλοποιείται με ένα μετρητή, η τιμή του οποίου αντιστοιχεί στην τρέχουσα επανάληψη του αλγόριθμου. Ανάλογα με την τιμή του μετρητή ενεργοποιούνται τα κατάλληλα σήματα συγχρονισμού του σχεδιασμού. Στο Σχήμα 5.1 φαίνονται οι είσοδοι και έξοδοι του αλγορίθμου AES. Το σήμα start_enc σηματοδοτεί την έναρξη της διαδικασίας κρυπτογράφησης, οπότε το κύκλωμα θεωρεί ότι στις δύο εισόδους εύρους 128 bits data_in και key_in βρίσκονται οι επιθυμητές τιμές του μπλοκ δεδομένων και του κλειδιού κρυπτογράφησης αντίστοιχα για τουλάχιστον ένα κύκλο ρολογιού. Μετά την ολοκλήρωση όλων των επαναλήψεων το κρυπτογραφημένο μπλοκ εμφανίζεται στην έξοδο enc_data και διατηρείται σταθερό για ένα κύκλο ρολογιού. Κατά τη διάρκεια αυτού του κύκλου το σήμα valid_data_out είναι ενεργοποιημένο και σηματοδοτεί την ύπαρξη έγκυρων δεδομένων στην έξοδο του συστήματος. Σημειώνουμε ότι η ύπαρξη δύο ξεχωριστών μονοπατιών δεδομένων (data paths) μήκους 128 bits για το μπλοκ δεδομένων και το κλειδί κρυπτογράφησης οφείλεται στο ότι το κύκλωμα αποτελεί στην ουσία υποσύστημα του κυκλώματος που υλοποιεί το πρωτόκολλο CCMP. Επομένως, οι δύο είσοδοι θα προέρχονται από καταχωρητές, αφού γίνει η εισαγωγή των τιμών τους στο σύστημα μέσω ενός κατάλληλου interface.



Σχήμα 5.1 Είσοδοι και έξοδοι του κυκλώματος του αλγορίθμου AES.

5.2.1 Κύκλωμα κρυπτογράφησης

Το κύκλωμα κρυπτογράφησης υλοποιεί τους απαραίτητους μετασχηματισμούς μίας επανάληψης του αλγορίθμου AES. Η ροή των δεδομένων, μέσα από τα στάδια του αλγορίθμου, φαίνεται στο Σχήμα 5.2. Ο σχεδιασμός αυτός βασίζεται στο μπλοκ διάγραμμα του αλγορίθμου (βλέπε Σχήμα 3.2), με σκοπό τη χρήση πολυπλεκτών και την αποφυγή της επανάληψης επιμέρους σταδίων. Ο πρώτος πολυπλέκτης επιλέγει ανάμεσα στο μπλοκ δεδομένων εισόδου και την ανατροφοδότηση για την εκτέλεση της επόμενης επανάληψης, ανάλογα με την τιμή του σήματος init_round. Ο δεύτερος πολυπλέκτης επιλέγει ανάμεσα στην έξοδο της μονάδας ShiftRows και MixColumns αντίστοιχα, ανάλογα με την τιμή του σήματος final_round. Χρησιμοποιείται για την εκτέλεση της τελευταίας επανάληψης του AES, όπου δεν πραγματοποιείται ο μετασχηματισμός MixColumns. Σημαντικό χαρακτηριστικό του σχεδιασμού αποτελεί το γεγονός ότι η τιμή του 128 bit καταχωρητή στην έξοδο του κυκλώματος δε μεταβάλλεται μετά από κάθε επανάληψη του αλγόριθμου, αλλά μόνο στο τέλος της διαδικασίας που τα κρυπτογραφημένα δεδομένα είναι έγκυρα. Με τον τρόπο αυτό δεν καταναλώνεται άσκοπα ισχύς από τη μεταβολή της κατάστασης του καταχωρητή.

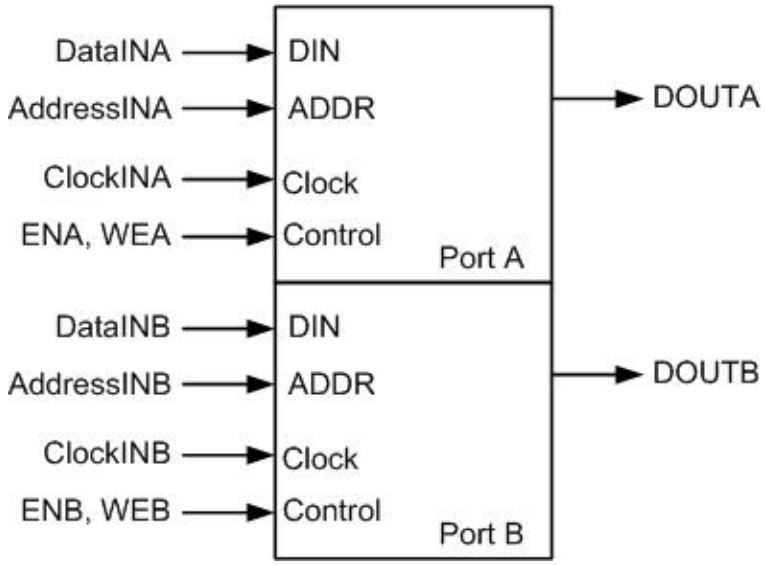


Σχήμα 5.2 Το μπλοκ διάγραμμα του κυκλώματος κρυπτογράφησης.

Η μονάδα AddRoundKey αναλαμβάνει να προσθέσει το προσωρινό κλειδί κάθε επανάληψης, μέσω της εισόδου round_key, στα ενδιάμεσα δεδομένα. Αποτελείται από ένα δίκτυο 128 πυλών XOR ενός επιπέδου, που αναλαμβάνει την εκτέλεση της πράξης ανάμεσα στα αντίστοιχα bits των ενδιάμεσων δεδομένων και του προσωρινού κλειδιού.

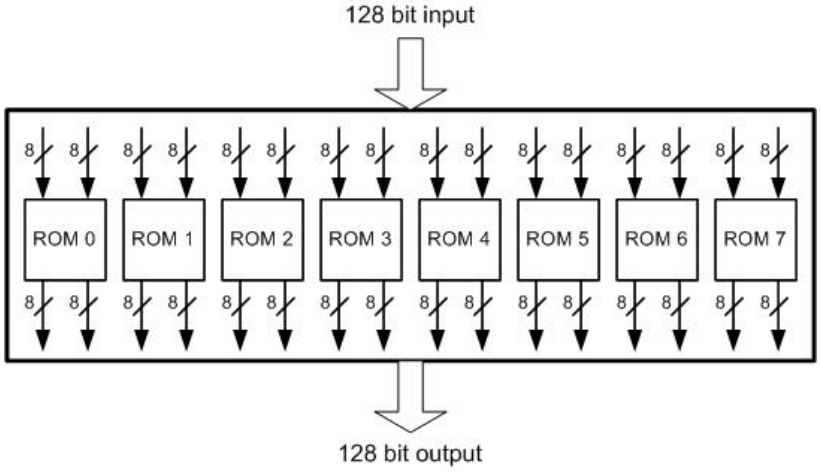
Η μονάδα SubBytes υλοποιεί τη μη γραμμική συνάρτηση αντικατάστασης της τιμής ενός byte στην είσοδο με μία άλλη τιμή στο πεδίο GF(2^8). Μπορεί να υλοποιηθεί είτε με συνδυαστική λογική, είτε με μία μνήμη ROM η οποία περιέχει όλες τις δυνατές 256 τιμές. Η μνήμη ROM παίρνει ως διεύθυνση το byte εισόδου και στην έξοδο εμφανίζεται η τιμή αντικατάστασης. Η πιο κατάλληλη υλοποίηση, όσον αφορά το λόγο επιφάνεια προς απόδοση σε τεχνολογία FPGA, είναι η χρήση της μνήμης ROM. Οι υλοποιήσεις που βασίζονται σε συνδυαστική λογική είναι κατάλληλες για αρχιτεκτονικές όπου στόχος είναι η υψηλή ταχύτητα λειτουργίας. Στην περίπτωση που το κύκλωμα υλοποιεί μόνο μία επανάληψη του αλγορίθμου τότε το κόστος στη χρήση πόρων του FPGA είναι σημαντικά μεγαλύτερο, χωρίς αντίστοιχη αύξηση της απόδοσης. Η εται-

ρία Xilinx παρέχει γρήγορη ενσωματωμένη μνήμη πάνω στο FPGA, η οποία ονομάζεται BlockRAM και είναι ιδανική για την υλοποίηση της ROM για τη μονάδα SubBytes. Η είσοδος στη μονάδα SubBytes αποτελείται από ένα μπλοκ μήκους 128 bits ή 16 bytes. Εφόσον ο μετασχηματισμός πρέπει να εφαρμοστεί ταυτόχρονα και στα 16 bytes χρειάζονται συνολικά 16 ROMs. Οι BlockRAM μπορούν να προγραμματιστούν ώστε να λειτουργούν ως dual port rom. Η βασική δομή της BlockRAM φαίνεται στο Σχήμα 5.3.



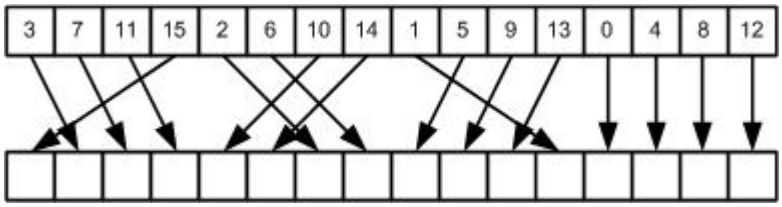
Σχήμα 5.3 Βασική δομή της BlockRAM.

Επομένως, ο συνολικός αριθμός από μνήμες που απαιτούνται μειώνεται στις 8. Στο Σχήμα 5.4 φαίνεται η συνολική οργάνωση των μνημών. Σημειώνουμε ότι οι BlockRAM είναι σύγχρονες μνήμες και η εμφάνιση των αποτελεσμάτων στην έξοδο απαιτεί ένα κύκλο ρολογιού.



Σχήμα 5.4 Η μονάδα SubBytes οργανωμένη σε 8 256x8 dual port ROMs.

Η μονάδα ShiftRows αλλάζει τη διάταξη των bytes μέσα στο μπλοκ των ενδιάμεσων δεδομένων. Θυμίζουμε ότι ο αλγόριθμος AES αντιμετωπίζει το μπλοκ των δεδομένων ως ένα δισδιάστατο πίνακα 16 στοιχείων (State). Σύμφωνα με τις προδιαγραφές του AES τα bytes του πίνακα ολισθαίνουν κατά συγκεκριμένο αριθμό θέσεων ανάλογα με τη γραμμή στην οποία βρίσκονται (βλέπε §3.5.2). Η μονάδα ShiftRows δεν περιέχει καθόλου συνδυαστική λογική, μόνο δρομολόγηση (routing). Η τελική αναδιάταξη των 16 bytes μέσα στο μπλοκ φαίνεται στο Σχήμα 5.5.



Σχήμα 5.5 Η μονάδα ShiftRows.

Για τη μονάδα MixColumns έχουν προταθεί στη βιβλιογραφία αρκετές υλοποιήσεις ([15], [16], [35]). Η λύση που ακολουθείται είναι αυτή που προτείνεται στο [15]. Σύμφωνα με αυτή, οι εξισώσεις που δίνουν τα νέα στοιχεία κάθε στήλης μετά την εφαρμογή του μετασχηματισμού MixColumns, μπορούν να γραφτούν με τον ακόλουθο τρόπο:

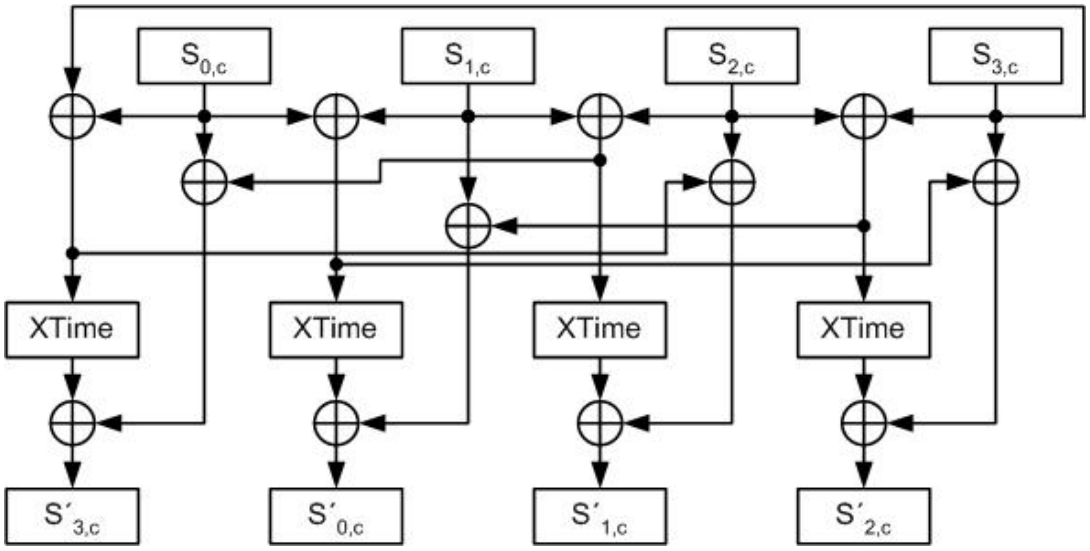
$$s'_{0,c} = \{02\} \bullet (s_{0,c} \oplus s_{1,c}) \oplus (s_{2,c} \oplus s_{3,c}) \oplus s_{1,c}$$

$$s'_{1,c} = \{02\} \bullet (s_{1,c} \oplus s_{2,c}) \oplus (s_{3,c} \oplus s_{0,c}) \oplus s_{2,c}$$

$$s'_{2,c} = \{02\} \bullet (s_{2,c} \oplus s_{3,c}) \oplus (s_{0,c} \oplus s_{1,c}) \oplus s_{3,c}$$

$$s'_{3,c} = \{02\} \bullet (s_{3,c} \oplus s_{0,c}) \oplus (s_{1,c} \oplus s_{2,c}) \oplus s_{0,c}$$

Επομένως, ο μετασχηματισμός μπορεί να υλοποιηθεί από την αρχιτεκτονική που φαίνεται στο Σχήμα 5.6.



Σχήμα 5.6 Υλοποίηση του μετασχηματισμού MixColumns.

Η λειτουργία της μονάδας XTime είναι ο πολλαπλασιασμός με το x, δηλαδή το στοιχείο {02} σε δεκαεξαδική αναπαράσταση. Σύμφωνα με τις προδιαγραφές του AES (βλέπε §3.3.3), ο πολλαπλασιασμός αυτός μπορεί να υλοποιηθεί ως μία αριστερή ολίσθηση του byte στην είσοδο και μία υπό συνθήκη πράξη XOR, ανάλογα με την τιμή του πιο σημαντικού bit, ανάμεσα στα αντίστοιχα bits με το {1b}. Η απευθείας υλοποίηση απαιτεί την ύπαρξη μίας πύλης XOR και ενός πολυπλέκτη στο κρίσιμο μονοπάτι της μονάδας XTime. Στο [16] προτείνεται μία διαφορετική υλοποίηση της μονάδας XTime η οποία απαιτεί συνολικά 4 πύλες XOR, με μία πύλη στο κρίσιμο μονοπάτι. Στο [15] προτείνεται μία παρόμοια υλοποίηση, η οποία είναι πιο αποδοτική.

Ένα στοιχείο του πεδίου GF(2^8) γράφεται σε πολυωνυμική μορφή ως

$$B = b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0$$

όπου τα $b_0, b_1, ..., b_7 \in GF(2)$. Ο πολλαπλασιασμός με το x έχει ως αποτέλεσμα

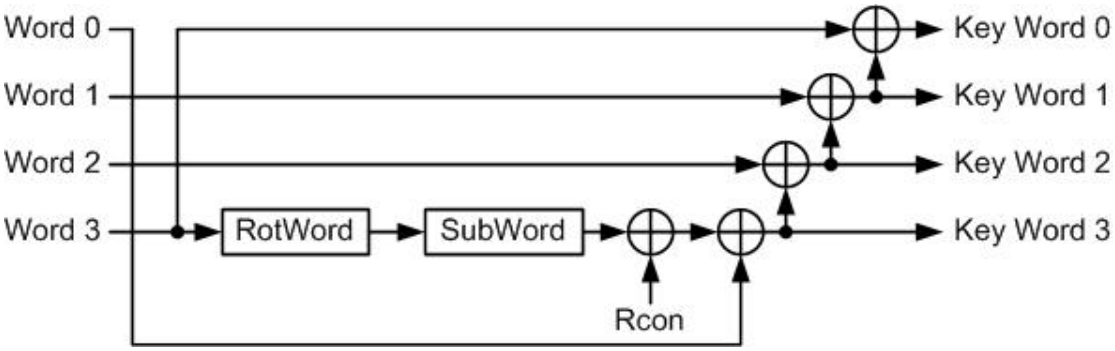
$$\begin{aligned} xB &= b_7x^8 + b_6x^7 + b_5x^6 + b_4x^5 + b_3x^4 + b_2x^3 + b_1x^2 + b_0x \bmod m(x) \\ &= b_6x^7 + b_5x^6 + b_4x^5 + (b_3 + b_7)x^4 + (b_2 + b_7)x^3 + b_1x^2 + (b_0 + b_7)x + b_7 \end{aligned}$$

Η υλοποίηση αυτή απαιτεί 3 πύλες XOR συνολικά με μία μόνο πύλη στο κρίσιμο μονοπάτι.

Η μονάδα MixColumns, όπως φαίνεται στο Σχήμα 5.6, απαιτεί συνολικά 108 πύλες XOR για τον υπολογισμό μίας στήλης του πίνακα State και το κρίσιμο μονοπάτι είναι 3 πύλες. Συνολικά το κύκλωμα κρυπτογράφησης περιλαμβάνει 4 μονάδες MixColumns, μία για κάθε στήλη του πίνακα State.

5.2.2 Κύκλωμα υπολογισμού κλειδιών

Σκοπός του κυκλώματος υπολογισμού κλειδιών είναι να παρέχει στο κύκλωμα κρυπτογράφησης το κατάλληλο προσωρινό κλειδί ανάλογα με την επανάληψη του αλγορίθμου AES. Το κύκλωμα χειρίζεται τα δεδομένα εισόδου, δηλαδή το κλειδί της κρυπτογράφησης μήκους 128 bits, ως λέξεις των 32 bits και παράγει ένα προσωρινό κλειδί για κάθε επανάληψη του AES χρησιμοποιώντας το προηγούμενο κλειδί. Σύμφωνα με τις προδιαγραφές του AES (βλέπε §3.6), για την παραγωγή ενός νέου προσωρινού κλειδιού πρέπει να πραγματοποιηθούν δύο μετασχηματισμοί πάνω στο προηγούμενο κλειδί: ο RotWord και ο SubWord. Στο Σχήμα 5.7 φαίνονται οι δύο μετασχηματισμοί καθώς και η συνολική λειτουργία του κυκλώματος, το οποίο χρησιμοποιεί πύλες XOR για τις υπόλοιπες πράξεις.



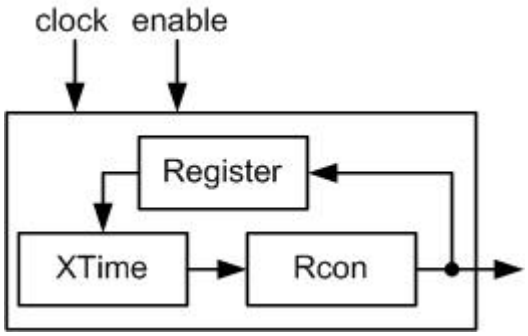
Σχήμα 5.7 Μπλοκ διάγραμμα του κυκλώματος υπολογισμού κλειδιών.

Η μονάδα που υλοποιεί το μετασχηματισμό RotWord απλά ολισθαίνει τα 4 λιγότερο σημαντικά bytes του κλειδιού, δηλαδή την πρώτη λέξη των 32 bits, κατά μία θέση προς τα αριστερά. Δεν περιλαμβάνει καθόλου συνδυαστική λογική, αλλά μόνο δρομολόγηση (routing).

Η μονάδα SubWord υλοποιεί την αντικατάσταση των ολισθημένων bytes της πρώτης λέξης σύμφωνα με τον πίνακα αντικατάστασης του αλγορίθμου AES (S-Box). Το S-Box υλοποιείται ως μνήμη ROM με τη βοήθεια της BlockRAM

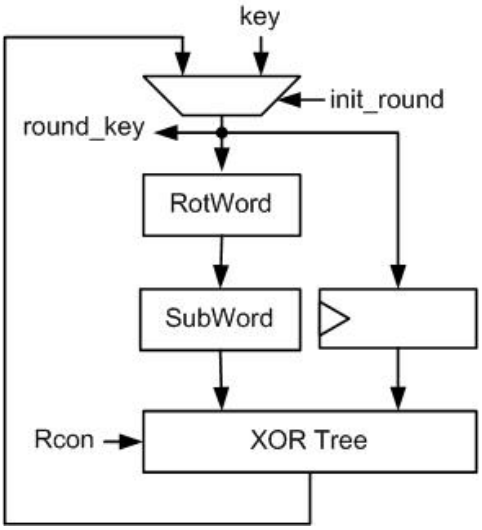
του FPGA. Ο μετασχηματισμός εφαρμόζεται ταυτόχρονα στα 4 bytes της λέξης, οπότε χρειάζονται συνολικά 4 S-Box, δηλαδή 2 BlockRAM προγραμματισμένες να λειτουργούν ως dual port rom.

Η σταθερή λέξη Rcon, που προστίθεται σε κάθε επανάληψη ξεκινάει από την τιμή [{01}, {00}, {00}, {00}] και ανανεώνεται για κάθε επανάληψη του αλγόριθμου υπολογισμού των κλειδιών. Το πρώτο byte της λέξης δηλώνει την αντίστοιχη δύναμη του x (βλέπε §3.6) και είναι αυτό που ανανεώνεται καθώς τα υπόλοιπα bytes έχουν μηδενική τιμή. Η μονάδα που αναλαμβάνει την ανανέωση της λέξης Rcon αποτελείται από το κύκλωμα XTime και ένα καταχωρητή, που διατηρεί την προηγούμενη τιμή του πρώτου byte, όπως φαίνεται στο Σχήμα 5.8. Το σήμα enable προέρχεται από τη μονάδα ελέγχου και σηματοδοτεί τον κύκλο στον οποίο αρχίζει η λειτουργία της μονάδας.



Σχήμα 5.8 Κύκλωμα ανανέωσης της τιμής Rcon.

Στο Σχήμα 5.9 φαίνεται η αρχιτεκτονική του κυκλώματος υπολογισμού των προσωρινών κλειδιών. Ο πολυπλέκτης επιλέγει ανάμεσα στο αρχικό κλειδί της κρυπτογράφησης και την ανατροφοδότηση του προηγούμενου προσωρινού κλειδιού. Η επιλογή γίνεται βάση του σήματος init_round, όπως στην περίπτωση του κυκλώματος κρυπτογράφησης, το οποίο προέρχεται από τη μονάδα ελέγχου. Σκοπός του ενδιάμεσου καταχωρητή των 128 bits είναι η καθυστέρηση του προσωρινού κλειδιού ώστε να συμβαδίζουν τα δεδομένα με την έξοδο της μονάδας SubWord. Θυμίζουμε ότι τα δεδομένα εμφανίζονται στην έξοδο της μονάδας SubWord μετά από ένα κύκλο ρολογιού, εφόσον η μνήμη που χρησιμοποιείται είναι σύγχρονη.



Σχήμα 5.9 Αρχιτεκτονική του κυκλώματος υπολογισμού κλειδιών.

Η υλοποίηση του αλγορίθμου AES, με την τεχνική του υπολογισμού των προσωρινών κλειδιών on-the-fly, χρειάζεται δώδεκα (12) κύκλους ρολογιού για την κρυπτογράφηση ενός μπλοκ δεδομένων μήκους 128 bits. Ο τελευταίος κύκλος είναι αυτός κατά τον οποίο τα έγκυρα δεδομένα διατηρούνται σταθερά στην έξοδο. Το κρίσιμο μονοπάτι του σχεδιασμού βρίσκεται στο κύκλωμα κρυπτογράφησης.

5.3 Αρχιτεκτονικές υπολογισμού κλειδιών offline

Στις αρχιτεκτονικές του AES, που βασίζονται στον υπολογισμό των προσωρινών κλειδιών offline τα κλειδιά για όλες τις επαναλήψεις του αλγόριθμου υπολογίζονται πριν την κρυπτογράφηση των δεδομένων και αποθηκεύονται σε μία μικρή μνήμη. Τα κυκλώματα κρυπτογράφησης και υπολογισμού των κλειδιών δεν αλληλεπιδρούν. Πρώτα το κύκλωμα υπολογισμού των κλειδιών παράγει τα απαραίτητα κλειδιά, ξεκινώντας από το αρχικό κλειδί κρυπτογράφησης και τα αποθηκεύει σε διαδοχικές θέσεις της μνήμης. Στη συνέχεια το κύκλωμα κρυπτογράφησης διευθυνσιοδοτεί τη μνήμη και χρησιμοποιεί το κατάλληλο προσωρινό κλειδί, ανάλογα με την επανάληψη του αλγορίθμου. Η διαδικασία ελέγχεται από ένα εξωτερικό κύκλωμα ελέγχου. Τέτοιες αρχιτεκτονικές είναι κατάλληλες για εφαρμογές όπου το ίδιο κλειδί κρυπτογράφησης χρησιμοποιείται για την εξασφάλιση της εμπιστευτικότητας δεδομένων με μήκος μεγαλύτερο από 128 bits. Με τον τρόπο αυτό όλα τα απαραίτητα προσω-

ρινά κλειδιά υπολογίζονται μία φορά, όποτε αλλάζει το κλειδί της κρυπτογράφησης και δεν καταναλώνεται άσκοπα ισχύς. Υπάρχει βέβαια το κόστος της καθυστέρησης (latency) για την παραγωγή του πρώτου κρυπτογραφημένου μπλοκ δεδομένων, αλλά αυτό είναι αμελητέο στις περιπτώσεις που το κλειδί κρυπτογράφησης δεν αλλάζει με μεγάλη συχνότητα.

Στα πλαίσια της διπλωματικής υλοποιήθηκαν δύο παρόμοιες αρχιτεκτονικές του AES που βασίζονται στην τεχνική του υπολογισμού των κλειδιών offline. Η βασική διαφορά τους είναι ότι στη μία περίπτωση γίνεται χρήση κοινών πόρων από τα κυκλώματα κρυπτογράφησης και υπολογισμού των κλειδιών. Οι κοινοί πόροι που χρησιμοποιούνται είναι οι πίνακες αντικατάστασης (S-Box) του μετασχηματισμού SubBytes. Εφόσον τα δύο κυκλώματα δε λειτουργούν παράλληλα είναι δυνατός ο διαμοιρασμός αυτών των πόρων, που αντιστοιχούν σε BlockRAM του FPGA.

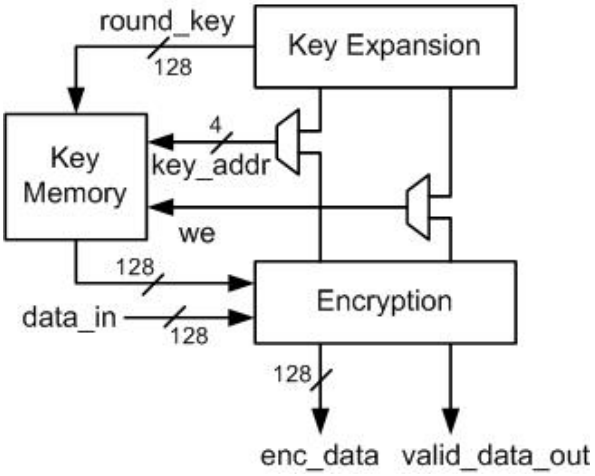
Η μνήμη στην οποία αποθηκεύονται τα προσωρινά κλειδιά (Key Memory) μπορεί να υλοποιηθεί με τη βοήθεια BlockRAM. Η μνήμη BlockRAM προγραμματίζεται ως single port ram, με δυνατότητα σύγχρονης ανάγνωσης και εγγραφής των κλειδιών. Έχει τέτοιο μέγεθος ώστε να χωράει το αρχικό κλειδί κρυπτογράφησης και τα 10 προσωρινά κλειδιά. Επειδή το εύρος των κλειδιών είναι μεγάλο η Key Memory αντιστοιχίζεται σε δύο BlockRAM. Εναλλακτικά, αν δεν είναι διαθέσιμη επιπλέον BlockRAM, είναι δυνατή η χρήση της κατανεμημένης μνήμης (distributed RAM) η, οποία χρησιμοποιεί τα LUTs του FPGA. Με τον τρόπο αυτό η Key Memory οργανώνεται ως ένα αρχείο καταχωρητών (register file) 11×128 bit, το οποίο καταλαμβάνει μόνο 138 CLB Slices επιπλέον. Η κατανεμημένη μνήμη μπορεί επίσης να προγραμματιστεί ώστε να λειτουργεί ως single port ram. Για τις αρχιτεκτονικές επιλέχθηκε η λύση της κατανεμημένης μνήμης.

Το κύκλωμα κρυπτογράφησης (Encryption) βασίζεται στις ίδιες αρχές, για το σχεδιασμό των επιμέρους μετασχηματισμών, που αναλύθηκαν στη §5.2. Η μόνη διαφορά είναι ότι πλέον τα σήματα ελέγχου των πολυπλεκτών δεν παράγονται από τη μονάδα ελέγχου, αλλά εσωτερικά από το κύκλωμα κρυπτογράφησης. Το κύκλωμα παράγει την κατάλληλη διεύθυνση, με τη βοήθεια ενός μετρητή, ανάλογα με την επανάληψη του αλγορίθμου AES. Στη συνέχεια

διαβάζει το προσωρινό κλειδί από την έξοδο της μνήμης κλειδιών. Για να γίνει η ανάγνωση των κλειδιών το κύκλωμα θέτει το σήμα we, προς τη μνήμη κλειδιών, στην τιμή 0.

Το κύκλωμα υπολογισμού κλειδιών (Key Expansion) υπολογίζει όλα τα απαραίτητα προσωρινά κλειδιά, ξεκινώντας από το αρχικό κλειδί κρυπτογράφησης και τα αποθηκεύει στη μνήμη κλειδιών. Η εγγραφή των κλειδιών στη μνήμη πραγματοποιείται όταν το κύκλωμα θέσει το σήμα we στην τιμή 1. Το κύκλωμα ακολουθεί την αρχιτεκτονική που περιγράφηκε στη §5.2.2. Εφόσον ο υπολογισμός των κλειδιών γίνεται ανεξάρτητα από τη λειτουργία της κρυπτογράφησης, το κύκλωμα υπολογισμού κλειδιών περιλαμβάνει τη μονάδα ανανέωσης της τιμής Rcon και ελέγχει τη λειτουργία της. Το ίδιο συμβαίνει με το σήμα ελέγχου του πολυπλέκτη (βλέπε Σχήμα 5.9), το οποίο παράγεται εσωτερικά.

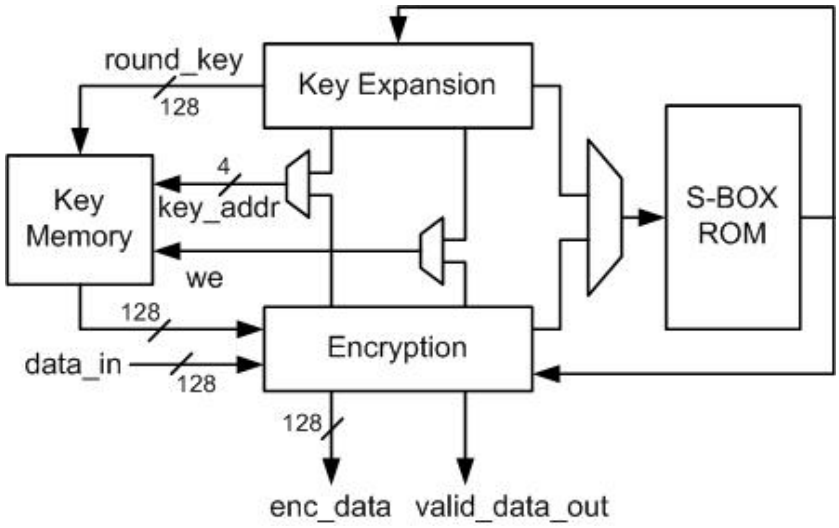
Ένα εξωτερικό κύκλωμα ελέγχου ενεργοποιεί πρώτα το κύκλωμα υπολογισμού κλειδιών και στη συνέχεια το κύκλωμα κρυπτογράφησης για την ολοκλήρωση της διαδικασίας. Ταυτόχρονα, θέτει κατάλληλα τα σήματα ελέγχου των πολυπλεκτών. Στο Σχήμα 5.10 φαίνεται η αλληλεπίδραση του κυκλώματος υπολογισμού κλειδιών και κρυπτογράφησης με τη μνήμη κλειδιών. Το κρίσιμο μονοπάτι παραμένει το ίδιο με την αρχιτεκτονική που αναλύθηκε στη §5.2.



Σχήμα 5.10 Αρχιτεκτονική του AES με offline υπολογισμό κλειδιών.

Υπάρχει η δυνατότητα να μειωθεί ο αριθμός των απαιτούμενων BlockRAM με την τεχνική του διαμοιρασμού πόρων (resource sharing). Ο διαμοιρασμός

αφορά τους πίνακες αντικατάστασης (S-Box) της μονάδας SubBytes, του κυκλώματος κρυπτογράφησης και της μονάδας SubWord, του κυκλώματος υπολογισμού κλειδιών. Εφόσον τα δύο κυκλώματα ενεργοποιούνται διαδοχικά το ένα μετά το άλλο, είναι εφικτή η κοινή χρήση τεσσάρων S-Box, που αντιστοιχούν σε δύο BlockRAM. Η αρχιτεκτονική που εκμεταλλεύεται αυτό το χαρακτηριστικό φαίνεται στο Σχήμα 5.11. Το latency παραμένει το ίδιο με πριν, αλλά αυξάνεται λίγο η συνολική επιφάνεια του σχεδιασμού λόγω της δρομολόγησης και των πολυπλεκτών στις εισόδους διεύθυνσης των κοινών μνημών. Επίσης, αναμένεται αύξηση της καθυστέρησης, εφόσον τοποθετούνται πολυπλέκτες πάνω στο κρίσιμο μονοπάτι για την κοινή χρήση των S-Box.



Σχήμα 5.11 Αρχιτεκτονική του AES με διαμοιρασμό πόρων.

Οι υλοποιήσεις του αλγορίθμου AES, με την τεχνική του υπολογισμού των προσωρινών κλειδιών offline, χρειάζονται συνολικά είκοσι τρεις (23) κύκλους ρολογιού για την κρυπτογράφηση ενός μπλοκ δεδομένων μήκους 128 bits, κάθε φορά που αλλάζει το κλειδί κρυπτογράφησης. Για τον υπολογισμό των κλειδιών απαιτούνται έντεκα (11) κύκλοι και για να ολοκληρωθεί η κρυπτογράφηση χρειάζονται δώδεκα (12) κύκλοι.

Στον πίνακα που ακολουθεί φαίνεται το latency των τριών αρχιτεκτονικών για τον αλγόριθμο AES και ο αριθμός BlockRAM που απαιτείται σε κάθε περίπτωση, χωρίς να λαμβάνονται υπόψη οι BlockRAM που χρησιμοποιεί η Key Memory. Δεν περιλαμβάνονται μετρήσεις σχετικά με την επιφάνεια, που καταλαμβάνει κάθε σχεδιασμός και την ελάχιστη περίοδο ρολογιού, καθώς τα κυ-

κλώματα αυτά αποτελούν μέρος του πρωτοκόλλου CCMP, που αναλύεται συνολικά στο Κεφάλαιο 6.

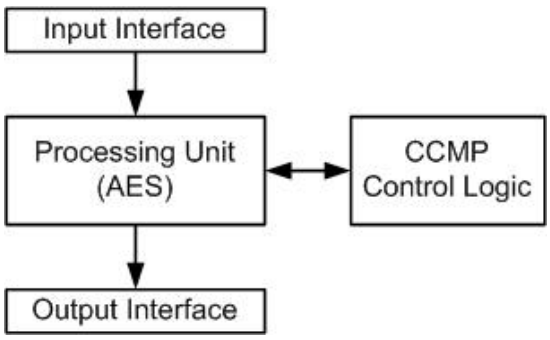
	Αρχιτεκτονική on-the-fly	Αρχιτεκτονική offline	Αρχιτεκτονική offline (resource sharing)
# of BlockRAMs	10	10	8
Latency (cycles)	12	23	23

6 Υλοποίηση του πρωτοκόλλου CCMP

6.1 Σύνοψη υλοποιήσεων

Στα πλαίσια της διπλωματικής μελετήθηκαν διάφορες αρχιτεκτονικές του πρωτοκόλλου CCMP, με στόχο το σχεδιασμό ενός κυκλώματος που θα αναλαμβάνει πλήρως την επεξεργασία των δεδομένων. Το κύκλωμα δέχεται ως είσοδο το πακέτο δεδομένων, με zero padding αν χρειάζεται για τη συμπλήρωση του τελευταίου μπλοκ της MAC επικεφαλίδας ή των δεδομένων. Στην περίπτωση του encapsulation το κύκλωμα παράγει στην έξοδο το κρυπτογραφημένο πακέτο δεδομένων και το κρυπτογραφημένο MIC, με την αρχική MAC επικεφαλίδα και τη CCMP επικεφαλίδα, έτοιμο για αποστολή. Στην περίπτωση του decapsulation το κύκλωμα, αφού κάνει έλεγχο για τυχόν επανάληψη του πακέτου, παράγει στην έξοδο το αποκρυπτογραφημένο πακέτο με την αρχική MAC επικεφαλίδα , έτοιμο για επεξεργασία από τα ανώτερα επίπεδα. Σημαντικό χαρακτηριστικό όλων των αρχιτεκτονικών αποτελεί η υποστήριξη πακέτων μεταβλητού μήκους.

Σε γενικές γραμμές όλες οι αρχιτεκτονικές που μελετήθηκαν έχουν τη δομή που φαίνεται στο Σχήμα 6.1. Αποτελούνται από την μονάδα ανάγνωσης (Input Interface), τη μονάδα επεξεργασίας (Processing Unit), τη μονάδα ελέγχου του πρωτοκόλλου (CCMP Control Logic) και τη μονάδα εγγραφής (Output Interface).



Σχήμα 6.1 Η αρχιτεκτονική υλοποίησης του πρωτοκόλλου CCMP.

Αναλυτικά, υλοποιήθηκαν τρεις αρχιτεκτονικές του CCMP, οι οποίες βασίζονται στις παραλλαγές του αλγορίθμου AES που αναλύθηκαν στο Κεφάλαιο 5.

Οι αρχιτεκτονικές αυτές πραγματοποιούν μόνο τη διαδικασία του encapsulation, δηλαδή τη διαδικασία που ακολουθείται για την αποστολή ενός πακέτου δεδομένων στο ασύρματο δίκτυο. Σκοπός ήταν η μελέτη των διαφορετικών αρχιτεκτονικών του πρωτοκόλλου CCMP, όσον αφορά τη συνολική επιφάνεια του σχεδιασμού, την ελάχιστη περίοδο ρολογιού που επιτυγχάνεται και το συνολικό ρυθμό επεξεργασίας (throughput). Οι τρεις αυτές αρχιτεκτονικές περιγράφονται αναλυτικά στη §6.3. Η πιο αποδοτική αρχιτεκτονική αποτελεί τη βάση για τους δύο επόμενους σχεδιασμούς: την παράλληλη αρχιτεκτονική για το CCMP και την αρχιτεκτονική που υλοποιεί τόσο τη διαδικασία του encapsulation, όσο και αυτή του decapsulation.

Το κύκλωμα της παράλληλης αρχιτεκτονικής υλοποιεί μόνο τη διαδικασία του encapsulation και βασίζεται στη χρήση δύο κυκλωμάτων κρυπτογράφησης. Το πρώτο κύκλωμα κρυπτογραφεί τα δεδομένα εισόδου και το δεύτερο υπολογίζει παράλληλα το ενδιάμεσο αποτέλεσμα του MIC, χρησιμοποιώντας τα ίδια δεδομένα. Με τον τρόπο αυτό μειώνεται σημαντικά ο χρόνος επεξεργασίας για κάθε πακέτο δεδομένων, αλλά αυξάνεται σημαντικά η επιφάνεια του σχεδιασμού. Αυτή η αρχιτεκτονική του CCMP είναι κατάλληλη για εφαρμογές όπου απαιτείται υψηλός ρυθμός επεξεργασίας δεδομένων, πολύ μεγαλύτερος από τις απαιτήσεις των ασύρματων δικτύων 802.11. Μειονέκτημα αυτής της αρχιτεκτονικής αποτελεί το γεγονός ότι το κύκλωμα δεν μπορεί να τροποποιηθεί ώστε να υλοποιεί και τη διαδικασία του decapsulation, με παράλληλο τρόπο. Αυτό οφείλεται στον τρόπο λειτουργίας του decapsulation, όπου πρώτα γίνεται η αποκρυπτογράφηση των δεδομένων και στη συνέχεια υπολογίζεται το MIC (βλέπε §4.4.4). Η λειτουργία του κυκλώματος για την παράλληλη αρχιτεκτονική αναλύεται στη §6.4.

Η τελευταία αρχιτεκτονική αποτελεί την ολοκληρωμένη λύση για την κρυπτογράφηση και αποκρυπτογράφηση των δεδομένων στα ασύρματα δίκτυα RSN. Το κύκλωμα αναλαμβάνει την επεξεργασία των MPDU και παρέχει εμπιστευτικότητα, επικύρωση, ακεραιότητα καθώς και προστασία από την επανάληψη πακέτων, σύμφωνα με τις προδιαγραφές του προτύπου IEEE 802.11i. Υποστηρίζει μεταβλητό μέγεθος MPDU, ενώ είναι σχεδιασμένο με τέτοιο τρόπο ώστε ο ρυθμός επεξεργασίας να διατηρείται σταθερός για τις διαδικασίες

encapsulation και decapsulation. Οι λεπτομέρειες του σχεδιασμού αναλύονται στη §6.5.

Στη §6.2 περιγράφεται η πλατφόρμα υλοποίησης και ελέγχου, καθώς και η διαδικασία για την ολοκλήρωση των σχεδιασμών, από την περιγραφή των κυκλωμάτων μέχρι την παραγωγή του bitstream για τον προγραμματισμό του FPGA. Τα αποτελέσματα σχετικά με τα χαρακτηριστικά λειτουργίας των κυκλωμάτων αποτυπώνονται και σχολιάζονται στη §6.6.

6.2 Πλατφόρμα υλοποίησης και ελέγχου

Η πλατφόρμα στην οποία έγινε η υλοποίηση και ο έλεγχος της ορθής λειτουργίας των σχεδιασμών αποτελείται από το Spartan-3 Starter Kit Board της εταιρίας Xilinx [36]. Το board περιλαμβάνει το Spartan-3 XC3S200 FPGA (XC3S200FT256) χωρητικότητας 200.000 πυλών με τεχνολογία επεξεργασίας 90 nm. Το FPGA έχει 1920 CLB Slices, που αντιστοιχούν σε 3840 LUTs. Το FPGA περιλαμβάνει ενσωματωμένη μνήμη, που αξιοποιείται για την υλοποίηση των S-Box. Συγκεκριμένα περιλαμβάνει 12 BlockRAM των 18 Kbit καθεμία, δηλαδή 216 Kbit συνολικά. Το μέγεθός του είναι αρκετό ώστε να χωράει όλους τους σχεδιασμούς, όσον αφορά τα απαιτούμενα CLB Slice, αλλά λόγω του μικρού αριθμού των διαθέσιμων BlockRAM δεν μπορεί να ικανοποιήσει τις απαιτήσεις της παράλληλης αρχιτεκτονικής. Το board περιέχει επίσης 1 Mbyte ασύγχρονη μνήμη SRAM, η οποία μπορεί να οργανωθεί σε δύο μνήμες μεγέθους 256K×16 με ανεξάρτητα σήματα ελέγχου ή σε μία μνήμη μεγέθους 256K×32. Επιλέχθηκε η πρώτη οργάνωση της μνήμης SRAM κατά τον έλεγχο της λειτουργίας των κυκλωμάτων, με τη μία μνήμη να περιέχει το κλειδί, τις παραμέτρους και τα δεδομένα του πακέτου MPDU, ενώ το αποτέλεσμα της επεξεργασίας (encapsulation ή decapsulation) του MPDU γράφεται στη δεύτερη μνήμη.

Η περιγραφή των σχεδιασμών έγινε με τη γλώσσα Verilog HDL. Στη συνέχεια έγινε η λογική εξομοίωση των σχεδιασμών με τη βοήθεια του εργαλείου ModelSim. Για τους σκοπούς της εξομοίωσης δημιουργήθηκαν λειτουργικά μοντέλα για τη μνήμη SRAM, καθώς και τη BlockRAM. Χρησιμοποιήθηκαν

διάφορα διανύσματα ελέγχου για το πρωτόκολλο CCMP [2, 37] προκειμένου να γίνει η πιστοποίηση της ορθής λειτουργίας των σχεδιασμών.

Η σύνθεση των κυκλωμάτων έγινε με το εργαλείο XST, καθώς το εργαλείο Leonardo Spectrum δεν μπορεί να αντιστοιχίσει την περιγραφή της μνήμης BlockRAM σε dual-port μνήμη. Επίσης, δεν υπήρχαν διαθέσιμες για το Leonardo Spectrum οι βιβλιοθήκες για την οικογένεια Spartan-3. Η υλοποίηση του σχεδιασμού, από το mapping μέχρι τη δημιουργία του bitsteam για τον προγραμματισμό του FPGA, έγινε με τα εργαλεία Xilinx ISE 6.3i [38].

6.3 Αρχιτεκτονικές για το encapsulation

Οι τρεις αρχιτεκτονικές που υλοποιήθηκαν διαφέρουν μόνο όσον αφορά το κύκλωμα του αλγορίθμου AES, που χρησιμοποιείται. Η υπόλοιπη λογική, που συγχρονίζει τη λειτουργία του πρωτοκόλλου για τη σωστή επεξεργασία και η διασύνδεση με τη μνήμη SRAM, για την ανάγνωση και εγγραφή των δεδομένων, παραμένει περίπου η ίδια. Οι αρχιτεκτονικές βασίζονται στη χρήση διαφόρων μονάδων, που αναλαμβάνουν την ανάγνωση από τη μνήμη, την επεξεργασία και την εγγραφή των δεδομένων στη μνήμη. Οι μονάδες αυτές είναι οι παρακάτω:

1. Μονάδα ανάγνωσης ενός μπλοκ δεδομένων
2. Μονάδα επεξεργασίας
3. Μονάδα εγγραφής ενός μπλοκ δεδομένων
4. Μονάδα ελέγχου του πρωτοκόλλου CCMP

Η μονάδα ανάγνωσης αποτελεί τη διασύνδεση εισόδου (input interface) του κυκλώματος. Αναλαμβάνει την ανάγνωση ενός μπλοκ δεδομένων από τη μνήμη μήκους 128 bits. Εφόσον το εύρος των δεδομένων της μνήμης είναι 16 bits, η μονάδα χρειάζεται οχτώ κύκλους για την ανάγνωση ολόκληρου του μπλοκ. Οι λέξεις της μνήμης ολισθαίνουν στον καταχωρητή των 128 bits της μονάδας και διατηρούνται σταθερές για ένα κύκλο ρολογιού. Με αυτό τον τρόπο τα δεδομένα που διαβάζονται, όπως το κλειδί κρυπτογράφησης και η MAC επικεφαλίδα, αποθηκεύονται στους διάφορους καταχωρητές του κυκλώ-

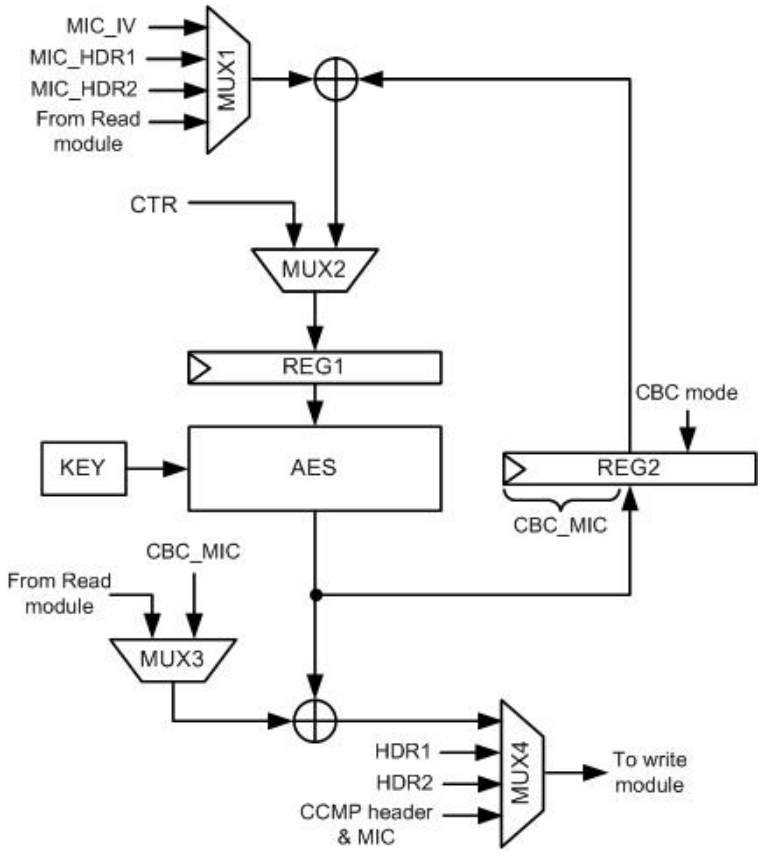
ματος για να χρησιμοποιηθούν στη συνέχεια. Η μονάδα αρχίζει την ανάγνωση από συγκεκριμένη θέση Μ της μνήμης και κάθε φορά που ενεργοποιείται συνεχίζει από τη θέση που αντιστοιχεί στο επόμενο μπλοκ, μέχρι να τελειώσουν τα δεδομένα του MPDU. Η μονάδα διευθυνσιοδοτεί κατάλληλα τη μνήμη SRAM με τη βοήθεια ενός μετρητή και ενεργοποιεί τα κατάλληλα σήματα ελέγχου για την ολοκλήρωση της ανάγνωσης.

Η μονάδα επεξεργασίας αποτελεί τον πυρήνα του πρωτοκόλλου CCMP και υλοποιεί τον αλγόριθμο κρυπτογράφησης AES. Σε αυτή τη μονάδα εντοπίζεται η διαφορά ανάμεσα στις αρχιτεκτονικές για το CCMP Encapsulation. Η αρχιτεκτονική Arch1, βασίζεται στη χρήση του AES με on-the-fly υπολογισμό των κλειδιών για τις επαναλήψεις του αλγορίθμου, όπως αναλύθηκε στη §5.2. Η αρχιτεκτονική Arch2, βασίζεται στη χρήση του AES με offline υπολογισμό των κλειδιών, όπως αναλύθηκε στη §5.3 (Σχήμα 5.10). Η αρχιτεκτονική Arch3, βασίζεται στη χρήση του AES με offline υπολογισμό των κλειδιών και διαμοιρασμό των πόρων, όπως αναλύθηκε στη §5.3 (Σχήμα 5.11). Στις αρχιτεκτονικές Arch2 και Arch3 τα κυκλώματα κρυπτογράφησης και υπολογισμού των κλειδιών, που αποτελούν την μονάδα επεξεργασίας, είναι ανεξάρτητα και την ενεργοποίησή τους αναλαμβάνει η μονάδα ελέγχου του πρωτοκόλλου CCMP.

Η μονάδα εγγραφής αποτελεί τη διασύνδεση εξόδου (output interface) του κυκλώματος. Αναλαμβάνει την εγγραφή ενός μπλοκ δεδομένων στη μνήμη μήκους 128 bits. Χρειάζεται οχτώ κύκλους για την εγγραφή ολόκληρου του μπλοκ. Παίρνει είσοδο από τους διάφορους καταχωρητές του κυκλώματος, ανάλογα με το είδος των δεδομένων (MAC επικεφαλίδα, κρυπτογραφημένα δεδομένα και MIC ή CCMP επικεφαλίδα) που εγγράφονται. Όπως και η μονάδα ανάγνωσης, η μονάδα εγγραφής διευθυνσιοδοτεί κατάλληλα τη μνήμη SRAM με τη βοήθεια ενός μετρητή και ενεργοποιεί τα κατάλληλα σήματα ελέγχου.

Η μονάδα ελέγχου του πρωτοκόλλου αναλαμβάνει το συντονισμό όλων των μονάδων για την επιτυχημένη επεξεργασία του MPDU. Υλοποιείται ως ένα FSM με αρκετές καταστάσεις. Σε κάθε κατάσταση ενεργοποιείται η κατάλληλη μονάδα, ανάλογα με το στάδιο της επεξεργασίας και καθορίζεται η τιμή των σημάτων ελέγχου για τους πολυπλέκτες του κυκλώματος.

Στο Σχήμα 6.2 φαίνεται η αρχιτεκτονική για το CCMP Encapsulation. Για λόγους ευκρίνειας δεν περιλαμβάνονται οι μονάδες ανάγνωσης, εγγραφής και ελέγχου.



Σχήμα 6.2 Αρχιτεκτονική για το CCMP encapsulation.

Ο πολυπλέκτης MUX2 επιλέγει την είσοδο για τον αλγόριθμο AES ανάλογα με τη φάση της επεξεργασίας του πρωτοκόλλου. Όταν γίνεται η κρυπτογράφηση των δεδομένων (CTR mode) επιλέγεται είσοδος από το μετρητή CTR. Κατά τον υπολογισμό του MIC (CBC mode) επιλέγεται ως είσοδος το αποτέλεσμα της πράξης XOR ανάμεσα στον καταχωρητή REG2 και την έξοδο του πολυπλέκτη MUX1.

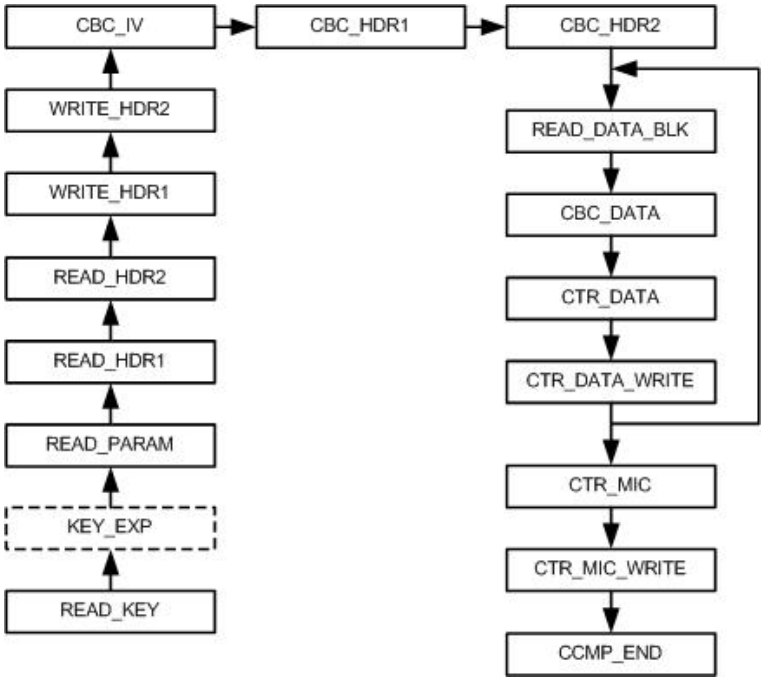
Ο πολυπλέκτης MUX1 επιλέγει ανάλογα με τη φάση της κατάστασης λειτουργίας CBC ανάμεσα στο MIC_IV, MIC_HDR1, MIC_HDR2 και την έξοδο από τη μονάδα ανάγνωσης, δηλαδή το αντίστοιχο μπλοκ δεδομένων. Ο καταχωρητής REG2 μήκους 128 bits αποθηκεύει το ενδιάμεσο αποτέλεσμα της κατάστασης λειτουργίας CBC κατά τον υπολογισμό του MIC. Μετά την ολοκλήρωση της κατάστασης CBC τα 64 πιο σημαντικά bits του καταχωρητή REG2 περιέχουν την τιμή του MIC (CBC_MIC). Η χρήση του καταχωρητή θα γίνει κατανοητή

στη συνέχεια με τη βοήθεια του διαγράμματος ροής των δεδομένων, το οποίο δείχνει τις καταστάσεις λειτουργίας της μονάδας ελέγχου.

Ο πολυπλέκτης MUX3 χρησιμοποιείται στη φάση της κρυπτογράφησης και επιλέγει ανάμεσα στο μπλοκ δεδομένων, από τη μονάδα ανάγνωσης και την τιμή του MIC. Η έξοδος του πολυπλέκτη γίνεται XOR με το αποτέλεσμα της κρυπτογράφησης του μετρητή CTR και αποτελεί το κρυπτογραφημένο μπλοκ δεδομένων που γράφεται στη μνήμη.

Ο πολυπλέκτης MUX4 επιλέγει την είσοδο για τη μονάδα εγγραφής ανάμεσα στους καταχωρητές HDR1 και HDR2, που περιέχουν το πρώτο και το δεύτερο τμήμα της MAC επικεφαλίδας αντίστοιχα, το κρυπτογραφημένο MIC και το κρυπτογραφημένο μπλοκ δεδομένων.

Η επεξεργασία του MPDU για την ολοκλήρωση του encapsulation εξελίσσεται σύμφωνα με το διάγραμμα ροής δεδομένων που φαίνεται στο Σχήμα 6.3. Στη συνέχεια περιγράφονται οι αντίστοιχες καταστάσεις της μονάδας ελέγχου.



Σχήμα 6.3 Διάγραμμα ροής δεδομένων κατά το encapsulation.

1. Γίνεται η εισαγωγή του κλειδιού της κρυπτογράφησης (READ_KEY), το οποίο αποθηκεύεται στον καταχωρητή KEY. Στην περίπτωση των δύο αρχιτεκτονικών που χρησιμοποιούν τον offline υπολογισμό κλειδιών, στη συνέχεια ακολουθεί η κατάσταση κατά την οποία υπολογί-

ζονται τα κλειδιά των επαναλήψεων και αποθηκεύονται στη μνήμη Key Memory (KEY_EXP).

2. Γίνεται η εισαγωγή των παραμέτρων για το συγκεκριμένο MPDU (READ_PARAM), δηλαδή ο αριθμός πακέτου (PN), το μήκος των δεδομένων σε bytes (DLen), το μήκος της MAC επικεφαλίδας (HLen) και ο αριθμός του κλειδιού κρυπτογράφησης (KeyID) και η αποθήκευσή τους σε αντίστοιχους καταχωρητές. Η παράμετρος DLen συγκρίνεται με έναν τοπικό μετρητή, που αυξάνεται μετά την επεξεργασία κάθε μπλοκ δεδομένων, ώστε η μονάδα ελέγχου να ελέγχει δυναμικά τις επαναλήψεις του βρόχου που περιλαμβάνει τις καταστάσεις 10 – 13.
3. Γίνεται ανάγνωση του πρώτου τμήματος της MAC επικεφαλίδας (READ_HDR1) και αποθηκεύεται στον καταχωρητή HDR1.
4. Γίνεται ανάγνωση του δεύτερου τμήματος της MAC επικεφαλίδας (READ_HDR2) και αποθηκεύεται στον καταχωρητή HDR2.
5. Γίνεται εγγραφή του πρώτου τμήματος της MAC επικεφαλίδας (WRITE_HDR1) στη μνήμη SRAM.
6. Γίνεται εγγραφή του δεύτερου τμήματος της MAC επικεφαλίδας (WRITE_HDR2) στη μνήμη SRAM.
7. Εκτελείται ο AES σε κατάσταση λειτουργίας CBC (CBC_IV) με είσοδο το μπλοκ MIC_IV.
8. Εκτελείται ο AES σε κατάσταση λειτουργίας CBC (CBC_HDR1) με είσοδο το μπλοκ MIC_HDR1.
9. Εκτελείται ο AES σε κατάσταση λειτουργίας CBC (CBC_HDR2) με είσοδο το μπλοκ MIC_ HDR2.
10. Γίνεται ανάγνωση του πρώτου μπλοκ δεδομένων μήκους 128 bit (READ_DATA_BLOCK).
11. Εκτελείται ο AES σε κατάσταση λειτουργίας CBC (CBC_DATA) με είσοδο το μπλοκ δεδομένων. Το αποτέλεσμα αποθηκεύεται στον καταχωρητή REG2.

12. Εκτελείται ο AES σε κατάσταση λειτουργίας CTR με είσοδο την τιμή 1 του μετρητή CTR. Το αποτέλεσμα της κρυπτογράφησης γίνεται XOR με το πρώτο μπλοκ δεδομένων (CTR_DATA).

13. Γίνεται εγγραφή του κρυπτογραφημένου μπλοκ δεδομένων στη μνήμη SRAM (CTR_DATA_WRITE). Αν το MPDU δεν έχει άλλα μπλοκ δεδομένων τότε η διαδικασία προχωράει στην επόμενη κατάσταση, διαφορετικά επαναλαμβάνεται η διαδικασία με τη ανάγνωση του επόμενου μπλοκ δεδομένων.

14. Εκτελείται ο AES σε κατάσταση λειτουργίας CTR με είσοδο την τιμή 0 του μετρητή CTR. Το αποτέλεσμα της κρυπτογράφησης γίνεται XOR με την τιμή του MIC που έχει υπολογιστεί (CTR_MIC).

15. Η CCMP επικεφαλίδα μήκους 8 bytes και το κρυπτογραφημένο MIC μήκους 8 bytes γράφονται στο τέλος του κρυπτογραφημένου MPDU (CTR_MIC_WRITE).

16. Το κύκλωμα μπαίνει σε αδρανή κατάσταση (CCMP_END) και περιμένει την ενεργοποίηση κάποιου σήματος για την επεξεργασία του επόμενου MPDU. Η επόμενη κατάσταση εξαρτάται από το αν υπάρχει αλλαγή του κλειδιού κρυπτογράφησης, οπότε η επόμενη κατάσταση είναι η READ_KEY. Διαφορετικά η επόμενη κατάσταση είναι η READ_PARAM.

Παρατηρώντας τη σειρά των καταστάσεων μέσα στο βρόχο γίνεται κατανοητή η χρήση του καταχωρητή REG2, για την αποθήκευση του ενδιάμεσου αποτελέσματος κατά τον υπολογισμό του MIC. Η κατάσταση λειτουργίας CBC χρησιμοποιεί το προηγούμενο αποτέλεσμα της επεξεργασίας με τον AES. Επομένως, εφόσον μετά την κατάσταση CBC_DATA ακολουθεί η κατάσταση CTR_DATA, στην επόμενη εκτέλεση του βρόχου ο αλγόριθμος AES δε θα έχει τα σωστά δεδομένα στην είσοδο για την κατάσταση CBC. Η ύπαρξη του καταχωρητή REG2 κάνει δυνατή την ανάγνωση κάθε μπλοκ δεδομένων μία μόνο φορά, ώστε στη συνέχεια να εκτελεστούν σειριακά οι καταστάσεις CBC_DATA και CTR_DATA. Σε διαφορετική περίπτωση θα έπρεπε τα μπλοκ δεδομένων να διαβαστούν δύο φορές, μία για κάθε κατάσταση λειτουργίας

του AES. Το αποτέλεσμα θα ήταν η σημαντική μείωση του ρυθμού επεξεργασίας του κυκλώματος.

Σημειώνουμε ότι στην υλοποίηση των αρχιτεκτονικών το κύκλωμα σταματά τη λειτουργία του στην κατάσταση CCMP_END μετά την επεξεργασία του MPDU που βρίσκεται αποθηκευμένο στη μνήμη. Η επέκταση της λειτουργίας του κυκλώματος για την επεξεργασία επόμενων MPDU από μία ουρά δεν υλοποιήθηκε, αλλά είναι εύκολο να γίνει στην πράξη. Επιπλέον, επιλέχθηκε η εισαγωγή των παραμέτρων να γίνεται με ανάγνωση από τη μνήμη SRAM, παρά με ξεχωριστές εισόδους στο κύκλωμα. Αυτό έγινε για να περιοριστεί ο αριθμός των απαιτούμενων I/O ακίδων του FPGA και να αποφευχθεί ο εξωτερικός έλεγχος της τιμής τους.

6.4 Παράλληλη αρχιτεκτονική για το encapsulation

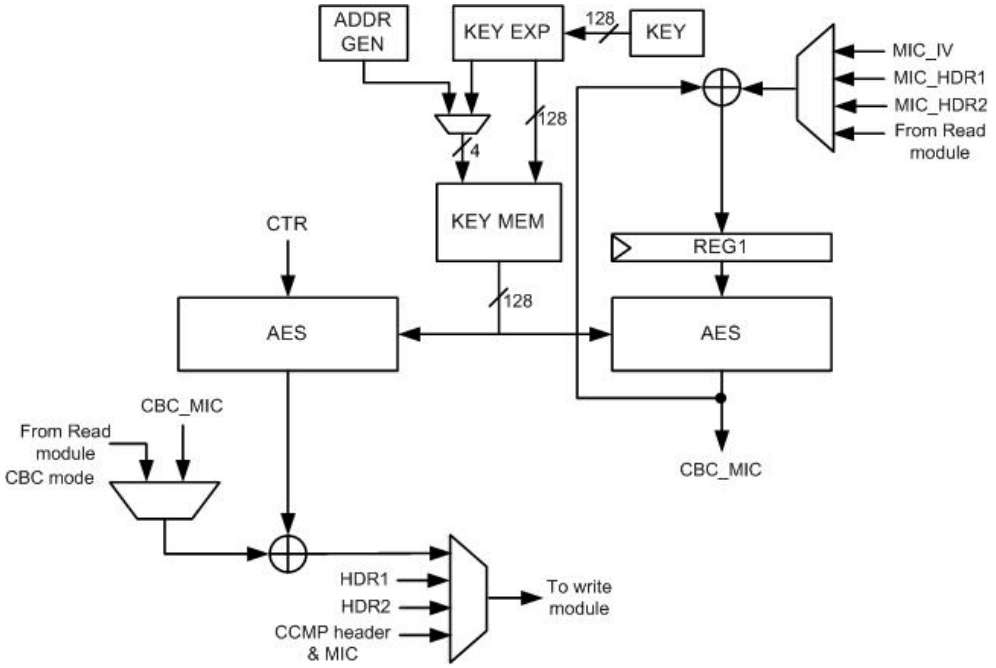
Οι προηγούμενες υλοποιήσεις του πρωτοκόλλου CCMP, όπως δείχνουν τα αποτελέσματα της §6.6, ικανοποιούν τις χαμηλές απαιτήσεις που θέτουν τα πρότυπα IEEE 802.11a/b/g, όσον αφορά το throughput. Παρόλα αυτά υπάρχουν περιπτώσεις που απαιτείται ακόμη μεγαλύτερο throughput, ενώ οι περιορισμοί σχετικά με την επιφάνεια που καταλαμβάνει ο σχεδιασμός στο FPGA δεν είναι τόσο αυστηροί. Ένας τρόπος να αυξηθεί σημαντικά το throughput είναι η παράλληλη επεξεργασία των δεδομένων. Η αρχιτεκτονική που προκύπτει βασίζεται στη χρήση δύο αντιγράφων του κυκλώματος κρυπτογράφησης για τον αλγόριθμο AES, τα οποία λειτουργούν παράλληλα (Arch4). Το πρώτο υλοποιεί την κατάσταση λειτουργίας CBC και αναλαμβάνει τον υπολογισμό του MIC, ενώ το δεύτερο υλοποιεί την κατάσταση λειτουργίας CTR και κρυπτογραφεί τα δεδομένα. Το μεγάλο μειονέκτημα της παράλληλης αρχιτεκτονικής είναι ότι το κύκλωμα δεν μπορεί να χρησιμοποιηθεί για την αντίστροφη διαδικασία (decapsulation), αφού πρέπει να γίνει πρώτα η αποκρυπτογράφηση των δεδομένων και στη συνέχεια να υπολογιστεί το MIC. Επομένως, το ένα αντίγραφο του AES παραμένει ανενεργό για όσο χρόνο λειτουργεί το άλλο.

Τα δύο κυκλώματα του AES χρησιμοποιούν τα ίδια ακριβώς προσωρινά κλειδιά για τις επαναλήψεις του αλγορίθμου. Επομένως, η χρήση της αρχιτεκτονικής του AES, που βασίζεται στον υπολογισμό των κλειδιών on-the-fly, είναι ακατάλληλη αφού αυξάνει σημαντικά την κατανάλωση ισχύος. Κατά την υλοποίηση της παράλληλης αρχιτεκτονικής, για το κύκλωμα επεξεργασίας επιλέχθηκε η λύση που εφαρμόστηκε στην Arch2. Ο συνολικός αριθμός BlockRAM που απαιτούνται, είναι δεκαοκτώ (18), ενώ η Key Memory υλοποιείται με Distributed RAM.

Η παράλληλη αρχιτεκτονική φαίνεται στο Σχήμα 6.4 και αποτελείται από τις παρακάτω μονάδες:

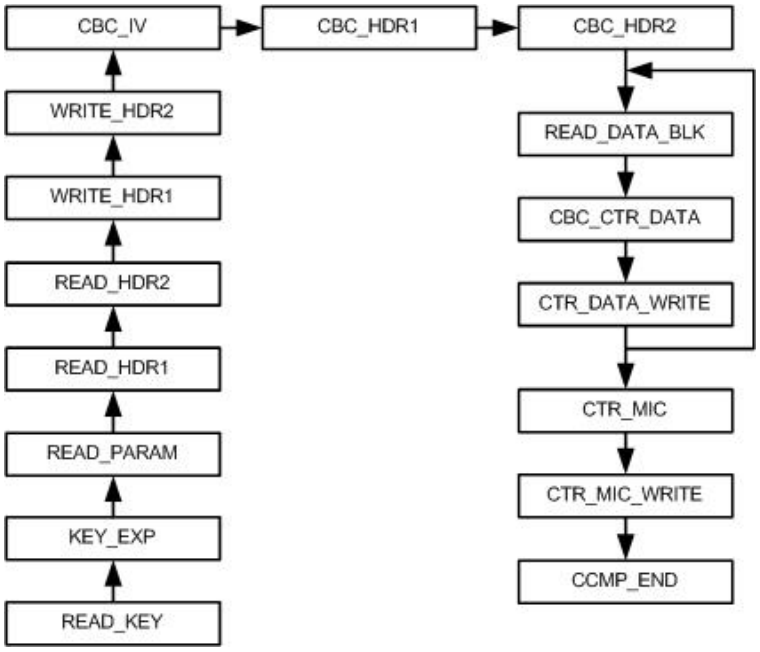
1. Μονάδα ανάγνωσης ενός μπλοκ δεδομένων
2. Μονάδα υπολογισμού των κλειδιών
3. Μονάδα AES για τον υπολογισμό του MIC (AES-CBC)
4. Μονάδα AES για την κρυπτογράφηση των δεδομένων (AES-CTR)
5. Γεννήτρια διευθύνσεων
6. Μονάδα εγγραφής ενός μπλοκ δεδομένων
7. Μονάδα ελέγχου του πρωτοκόλλου CCMP

Οι μονάδες ανάγνωσης και εγγραφής ενός μπλοκ δεδομένων στην SRAM παραμένουν οι ίδιες. Η μονάδα υπολογισμού των κλειδιών (KEY EXP) υπολογίζει τα προσωρινά κλειδιά της κρυπτογράφησης και τα αποθηκεύει στη μνήμη κλειδιών (Key Memory). Η γεννήτρια διευθύνσεων (ADDR GEN) αναλαμβάνει να παράγει την κατάλληλη διεύθυνση για τη μνήμη κλειδιών, ώστε οι δύο μονάδες AES να τροφοδοτούνται με το σωστό προσωρινό κλειδί για κάθε επανάληψη του αλγορίθμου.



Σχήμα 6.4 Παράλληλη αρχιτεκτονική για το CCMP Encapsulation.

Η μονάδα ελέγχου του CCMP αναλαμβάνει την ενεργοποίηση των μονάδων με την κατάλληλη σειρά, ώστε να ολοκληρωθεί με επιτυχία το encapsulation του MPDU. Η ροή των δεδομένων δε μεταβάλλεται και οι καταστάσεις της μονάδας ελέγχου παραμένουν σχεδόν οι ίδιες. Για τις καταστάσεις CBC_IV, CBC_HDR1 και CBC_HDR2 η μονάδα ελέγχου ενεργοποιεί μόνο τη μονάδα AES-CBC, ενώ κατά την επεξεργασία των μπλοκ δεδομένων εκτός από τη μονάδα AES-CBC ενεργοποιεί και τη μονάδα AES-CTR. Με τον τρόπο αυτό επιτυγχάνεται η παράλληλη επεξεργασία του MPDU. Το διάγραμμα ροής δεδομένων με τις καταστάσεις της μονάδας ελέγχου φαίνεται στο Σχήμα 6.5. Η κατάσταση CBC_CTR_DATA αντιστοιχεί στην παράλληλη λειτουργία των δύο μονάδων AES κατά την επεξεργασία του αντίστοιχου μπλοκ δεδομένων.



Σχήμα 6.5 Διάγραμμα ροής δεδομένων της παράλληλης αρχιτεκτονικής.

6.5 Αρχιτεκτονική του πρωτοκόλλου CCMP

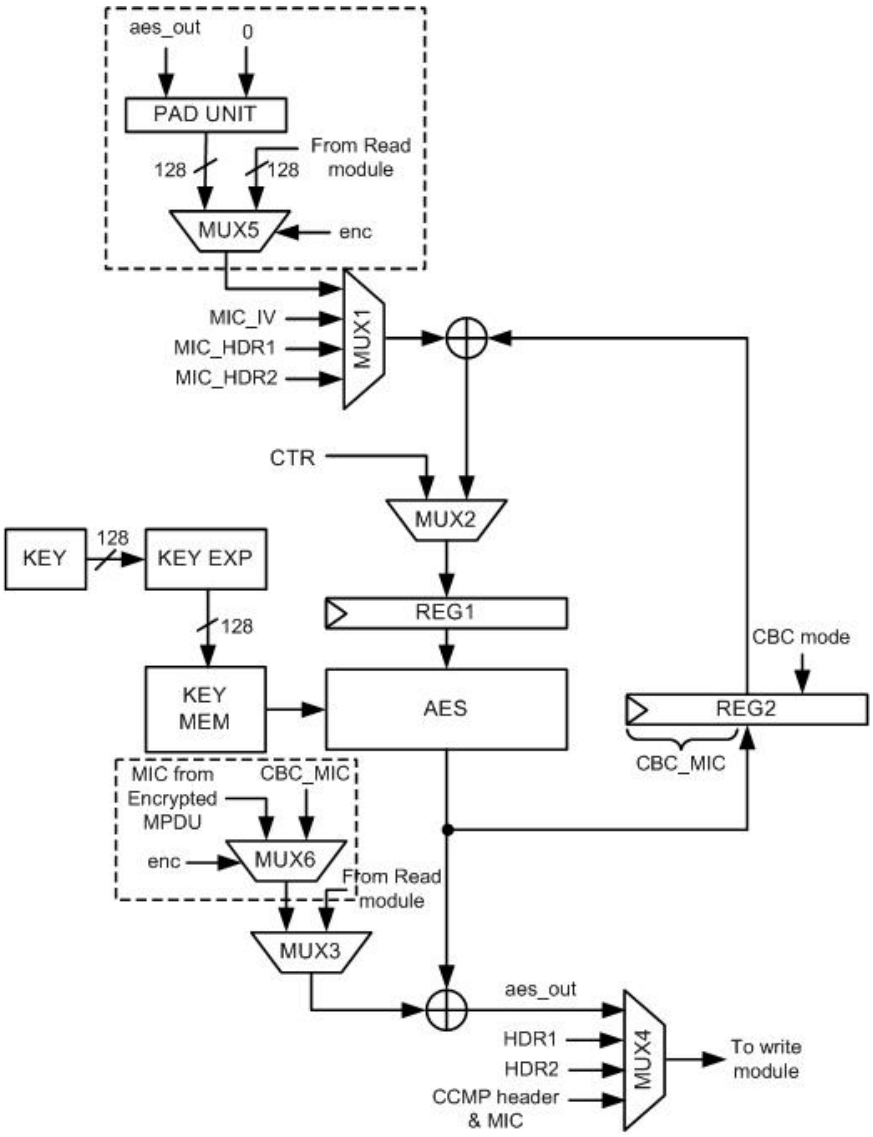
Σε αρκετές εφαρμογές είναι απαραίτητη η ολοκλήρωση ενός κυκλώματος, το οποίο αναλαμβάνει την επεξεργασία σύμφωνα με το πρωτόκολλο CCMP, τόσο για την αποστολή όσο και για τη λήψη των δεδομένων. Το κύκλωμα αυτό πρέπει να υλοποιεί με αποδοτικό τρόπο, όσον αφορά την επιφάνεια του σχεδιασμού και το ρυθμό επεξεργασίας δεδομένων, τις διαδικασίες encapsulation και decapsulation του CCMP. Η αρχιτεκτονική που περιγράφεται στη συνέχεια ικανοποιεί τις παραπάνω απαιτήσεις, προσφέροντας μία ολοκληρωμένη λύση για το σχεδιασμό δικτυακών συσκευών, που είναι πλήρως συμβατές με το πρωτόκολλο IEEE 802.11i. Επιπλέον, ο ρυθμός επεξεργασίας διατηρείται σταθερός για τις διαδικασίες encapsulation και decapsulation.

Η αρχιτεκτονική βασίζεται στη χρήση του κυκλώματος για τον offline υπολογισμό των προσωρινών κλειδιών του αλγορίθμου AES, χωρίς το διαμοιρασμό πόρων ανάμεσα στο κύκλωμα επεξεργασίας και το κύκλωμα υπολογισμού των κλειδιών. Η αρχιτεκτονική φαίνεται στο Σχήμα 6.6 και αποτελείται από τις ακόλουθες μονάδες:

1. Μονάδα ανάγνωσης ενός μπλοκ δεδομένων

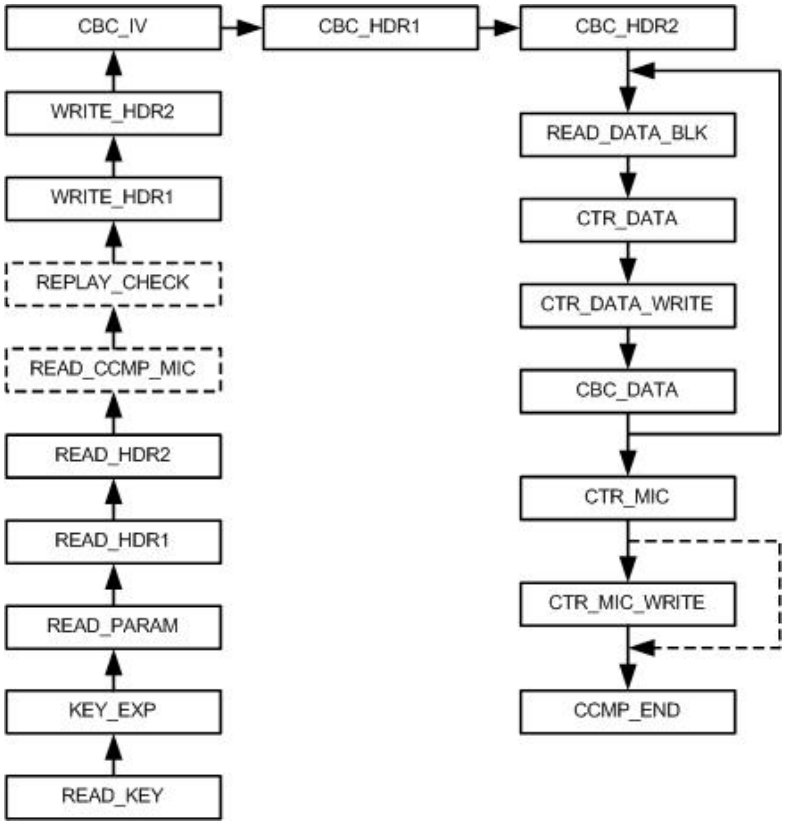
2. Μονάδα υπολογισμού των κλειδιών
3. Μονάδα AES
4. Μονάδα συμπλήρωσης με μηδενικά (Pad Unit)
5. Μονάδα εγγραφής ενός μπλοκ δεδομένων
6. Μονάδα ελέγχου του πρωτοκόλλου CCMP

Η διασύνδεση με τη μνήμη SRAM παραμένει η ίδια. Η μονάδα υπολογισμού των κλειδιών υπολογίζει τα προσωρινά κλειδιά της κρυπτογράφησης και τα αποθηκεύει στη μνήμη κλειδιών (Key Memory). Η μονάδα AES αναλαμβάνει να παράγει την κατάλληλη διεύθυνση για τη μνήμη κλειδιών, ώστε να τροφοδοτείται με το σωστό προσωρινό κλειδί για κάθε επανάληψη του αλγορίθμου.



Σχήμα 6.6 Αρχιτεκτονική του πρωτοκόλλου CCMP.

Η μονάδα ελέγχου του CCMP αναλαμβάνει την ενεργοποίηση των μονάδων με την κατάλληλη σειρά για την ολοκλήρωση των διαδικασιών encapsulation ή decapsulation. Οι καταστάσεις της μονάδας ελέγχου παραμένουν σχεδόν οι ίδιες και η ροή των δεδομένων είναι κοινή για τις δύο διαδικασίες. Ο διαχωρισμός γίνεται με ένα εξωτερικό σήμα ελέγχου (enc), το οποίο σηματοδοτεί την επιθυμητή λειτουργία του κυκλώματος και ταυτόχρονα επιλέγει την κατάλληλη είσοδο για τους πολυπλέκτες MUX5 και MUX6 (1: encapsulation, 0: decapsulation). Για τη διαδικασία του decapsulation το κύκλωμα διαβάζει από τη μνήμη τη CCMP επικεφαλίδα και το κρυπτογραφημένο MIC (READ_CCMP_MIC), τα οποία αποθηκεύονται σε ένα τοπικό καταχωρητή. Η επικεφαλίδα χρησιμοποιείται για την εξαγωγή του PN και το MIC συγκρίνεται στο τέλος της διαδικασίας με το MIC που υπολογίστηκε μετά την αποκρυπτογράφηση των δεδομένων για τον έλεγχο της ακεραιότητας. Το PN συγκρίνεται με το μετρητή επανάληψης (Replay Counter) και αν είναι μικρότερο ή ίσο τότε το πακέτο απορρίπτεται ως επανάληψη παλαιότερου πακέτου και η διαδικασία διακόπτεται (REPLAY_CHECK). Ο βρόχος επεξεργασίας των μπλοκ δεδομένων τροποποιείται ώστε να αντανακλά τη ροή των δεδομένων και για τις δύο διαδικασίες. Με τον τρόπο αυτό η κατάσταση CTR_DATA, η οποία αντιστοιχεί στην κρυπτογράφηση (αποκρυπτογράφηση) των δεδομένων και του MIC κατά το encapsulation (decapsulation), προηγείται της κατάστασης CBC_DATA, η οποία αντιστοιχεί στον υπολογισμό του MIC. Στο Σχήμα 6.7 φαίνεται το διάγραμμα ροής δεδομένων (με διακεκομμένη γραμμή για το decapsulation) με τις καταστάσεις της μονάδας ελέγχου.



Σχήμα 6.7 Διάγραμμα ροής δεδομένων του πρωτοκόλλου CCMP.

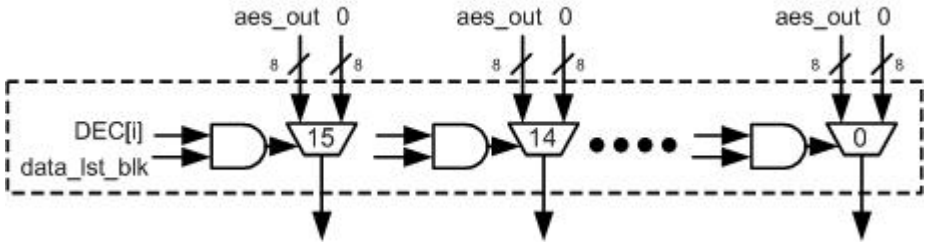
Στην πράξη ο αριθμός των κύκλων ρολογιού για την επεξεργασία ενός MPDU, τόσο κατά το encapsulation, όσο και κατά το decapsulation, παραμένει σταθερός.

Η κατάσταση REPLAY_CHECK αποτελεί τερματική κατάσταση στην περίπτωση που η τιμή της παραμέτρου PN είναι μικρότερη ή ίση με την τιμή του Replay Counter. Με τον τρόπο αυτό αποφεύγεται η άσκοπη επεξεργασία του πακέτου. Αν η τιμή του PN είναι έγκυρη τότε η επεξεργασία του MPDU συνεχίζεται κανονικά. Στην κατάσταση CCMP_END γίνεται ο έλεγχος της ακεραιότητας των δεδομένων. Η ακεραιότητα εξασφαλίζεται με τη σύγκριση ανάμεσα στην τιμή του MIC, που περιλάμβανε το κρυπτογραφημένο MPDU και την τιμή που υπολογίστηκε ξανά κατά το decapsulation. Οι τιμές πρέπει να ταυτίζονται για να γίνει αποδεκτό το MPDU, διαφορετικά απορρίπτεται ως πλαστογραφημένο. Στην πρώτη περίπτωση ο μετρητής Replay Counter ενημερώνεται με την τιμή του PN. Στο σχεδιασμό δεν υπάρχουν εξωτερικά σήματα για τον έλεγχο της ροής μετά τις καταστάσεις REPLAY_CHECK και CCMP_END, για την επεξεργασία των επόμενων MPDU. Αυτό το χαρακτηριστικό υλοποιείται

εύκολα στην περίπτωση που απαιτείται η συνεχής επεξεργασία μέσα από μία ουρά.

Κατά το decapsulation κάθε κρυπτογραφημένο μπλοκ δεδομένων αποκρυπτογραφείται, εγγράφεται στη μνήμη SRAM και στη συνέχεια χρησιμοποιείται απευθείας για τον υπολογισμό του MIC. Αυτό δημιουργεί πρόβλημα κατά την επεξεργασία του τελευταίου μπλοκ στην περίπτωση που δεν ήταν συμπληρωμένο με δεδομένα από την αρχή. Για να ολοκληρωθεί σωστά η διαδικασία πρέπει μετά την αποκρυπτογράφηση να απορριφθούν τα επιπλέον bytes και το μπλοκ να συμπληρωθεί με τον κατάλληλο αριθμό από μηδενικά. Το βήμα αυτό πραγματοποιείται με τη βοήθεια του κυκλώματος Pad Unit, που φαίνεται στο Σχήμα 6.8. Στη συνέχεια γίνεται η επεξεργασία του από τη μονάδα AES στην κατάσταση λειτουργίας CBC.

Θεωρούμε ότι το τελευταίο μπλοκ δεν είναι συμπληρωμένο. Το κύκλωμα αντιμετωπίζει κάθε byte ξεχωριστά και επιλέγει το αποκρυπτογραφημένο byte ή το μηδέν. Αποτελείται από έναν 4-σε-16 αποκωδικοποιητή (DEC) και δεκαέξι (16) πολυπλέκτες, ένα για κάθε byte. Ο αποκωδικοποιητής χρησιμοποιεί τα bits [3:0] της παραμέτρου DLen, που αντιστοιχούν στον αριθμό των bytes δεδομένων του τελευταίου μπλοκ και παράγει μία 16-bit λέξη ελέγχου για τους πολυπλέκτες. Κάθε bit μαζί με το σήμα data_lst_blk, που παράγεται από τη μονάδα ελέγχου του CCMP και σηματοδοτεί την επεξεργασία του τελευταίου μπλοκ δεδομένων, επιλέγει το byte δεδομένων ή το μηδέν από κάθε πολυπλέκτη αντίστοιχα.



Σχήμα 6.8 Κύκλωμα Pad Unit.

Η αρχιτεκτονική του πρωτοκόλλου CCMP προκύπτει εύκολα από την αρχιτεκτονική για το CCMP Encapsulation, που φαίνεται στο Σχήμα 6.2. Το μόνο overhead είναι η συνδυαστική λογική, που φαίνεται στα διακεκομμένα πλαίσια (βλέπε Σχήμα 6.6) και η μικρή αύξηση στην πολυπλοκότητα της μονάδας ελέγχου.

6.6 Αποτελέσματα

Τα κυκλώματα που υλοποιήθηκαν διατηρούν κατά βάση τα ίδια χαρακτηριστικά όσον αφορά τη ροή των δεδομένων. Οι καταστάσεις της μονάδας ελέγχου έχουν την ίδια λειτουργικότητα σε κάθε περίπτωση. Το γεγονός αυτό επιτρέπει την εύκολη επέκταση του κυκλώματος για το πρωτόκολλο CCMP, ώστε να υλοποιεί και τη διαδικασία του decapsulation, παρέχοντας μία ολοκληρωμένη λύση για την ασφάλεια των δεδομένων σε ασύρματα δίκτυα IEEE 802.11i. Το τελικό κύκλωμα αποτελεί έναν ισχυρό CCMP συνεπεξεργαστή (co-processor), που μπορεί να αποφορτίσει σημαντικά τον μικροεπεξεργαστή που χειρίζεται το MAC πρωτόκολλο. Στον πίνακα που ακολουθεί φαίνεται ο αριθμός των κύκλων ρολογιού που απαιτείται για την ολοκλήρωση των καταστάσεων Ανάγνωση, Εγγραφή, AES (CTR/CBC) και Υπολογισμός κλειδιών.

Κατάσταση	Κύκλοι ρολογιού
Ανάγνωση μπλοκ	10
Εγγραφή μπλοκ	10
AES (CTR/CBC)	13
Υπολογισμός κλειδιών	13

Όπως φαίνεται από τον πίνακα σε κάθε κατάσταση απαιτείται ένας επιπλέον κύκλος για να γίνει η ενεργοποίηση της αντίστοιχης μονάδας και ένας επιπλέον για τη σηματοδότηση του τέλους της λειτουργίας κάθε μονάδας.

Το βασικό χαρακτηριστικό των κυκλωμάτων είναι ότι υποστηρίζουν την επεξεργασία MPDU μεταβλητού μεγέθους. Με βάση την τιμή του καταχωρητή DLen, που δείχνει το μήκος των δεδομένων του MPDU σε bytes, το κύκλωμα ελέγχου αποφασίζει πόσες φορές θα εκτελεστεί ο βρόχος (loop), που φαίνεται στο Σχήμα 6.3. Για το σκοπό αυτό το κύκλωμα αυξάνει ένα μετρητή μετά από την επεξεργασία ενός μπλοκ δεδομένων. Ο αριθμός των εκτελέσεων εξαρτάται από τον αριθμό των μπλοκ δεδομένων $B = \left\lceil \frac{DLen}{16} \right\rceil$, όπου DLen ≤ 2296. Το τελευταίο μπλοκ δεδομένων αν χρειάζεται είναι αποθηκευμένο με zero padding στη μνήμη.

Για τις αρχιτεκτονικές Arch1, Arch2 και Arch3 ο συνολικός αριθμός κύκλων ρολογιού για την επεξεργασία με το πρωτόκολλο CCMP δίνεται από τη σχέση

$$N_{CCMP} = N + B(N_{READ} + 2 \times N_{AES} + N_{WRITE}) \quad (6.1)$$

όπου, $N_{READ} = N_{WRITE} = 10$, $N_{AES} = 13$ και $N = 122$ για την αρχιτεκτονική Arch1 ή $N = 135$ για τις αρχιτεκτονικές Arch2 και Arch3.

Για την παράλληλη αρχιτεκτονική Arch4 ο αριθμός των κύκλων ρολογιού για κάθε κατάσταση παραμένει αμετάβλητος. Ο συνολικός αριθμός κύκλων ρολογιού για την επεξεργασία ενός πακέτου σε αυτή την περίπτωση δίνεται από τη σχέση

$$N_{CCMP} = N + B(N_{READ} + N_{AES} + N_{WRITE}) \quad (6.2)$$

όπου $N_{READ} = N_{WRITE} = 10$, $N_{AES} = 13$ και $N = 135$.

Η αρχιτεκτονική Arch5 διατηρεί σταθερό το συνολικό αριθμό κύκλων ρολογιού, που απαιτείται για την επεξεργασία (N_{CCMP}), τόσο κατά την αποστολή, όσο και κατά τη λήψη ενός πακέτου. Το γεγονός αυτό είναι σημαντικό καθώς το throughput είναι ανεξάρτητο από την κατάσταση λειτουργίας του κυκλώματος (encapsulation ή decapsulation). Το N_{CCMP} δίνεται από τη σχέση (6.1), όπου $N = 135$.

Το πρότυπο IEEE 802.11 επιτρέπει την αποστολή πακέτων μεταβλητού μεγέθους. Υπάρχουν μικρά πακέτα, όπως τα πακέτα ACK και Beacon μήκους 14 και 72 bytes αντίστοιχα, καθώς και άλλα μικρά πακέτα διαχείρισης του προτύπου. Μετρήσεις που έγιναν για την κυκλοφορία πακέτων σε ασύρματα δίκτυα 802.11, με τη βοήθεια κατάλληλου λογισμικού (network sniffer), έδειξαν ότι τα περισσότερα πακέτα που διακινούνται έχουν μικρό μέγεθος μεταξύ 64 - 127 bytes και είναι πακέτα ελέγχου και διαχείρισης, ενώ τα πακέτα δεδομένων έχουν τυπικό μέγεθος περίπου 1024 bytes [40].

Το throughput του κυκλώματος (σε Mbps) για την επεξεργασία ενός πακέτου δεδομένων με DLen bytes δεδομένων και 32 bytes MAC επικεφαλίδα δίνεται από τη σχέση

$$\text{Throughput} = \frac{(\text{DLen} + 32) \times 8}{N_{CCMP} \times \text{clock\_cycle}} \times 1000 \quad (6.3)$$

όπου clock _ cycle είναι η περίοδος του κύκλου ρολογιού για κάθε αρχιτεκτονική.

Στον πίνακα που ακολουθεί φαίνονται τα αποτελέσματα όσον αφορά τα CLB Slices, τις BlockRAM, την ελάχιστη περίοδο ρολογιού, το throughput και το throughput/slice για μέγεθος πακέτου 1024 bytes για τις τρεις αρχιτεκτονικές. Το throughput/slice υπολογίζεται θεωρώντας ότι μία BlockRAM αντιστοιχεί σε 128 slices [18]. Σημειώνουμε ότι η υλοποίηση της παράλληλης αρχιτεκτονικής (Arch4) έγινε για FPGA της οικογένειας Spartan-3 μεγαλύτερης χωρητικότητας (xc3s1000-5ft256), το οποίο διαθέτει τον απαραίτητο αριθμό BlockRAM. Τα αποτελέσματα προέκυψαν μετά την επιτυχημένη ολοκλήρωση της διαδικασίας place 'n' route.

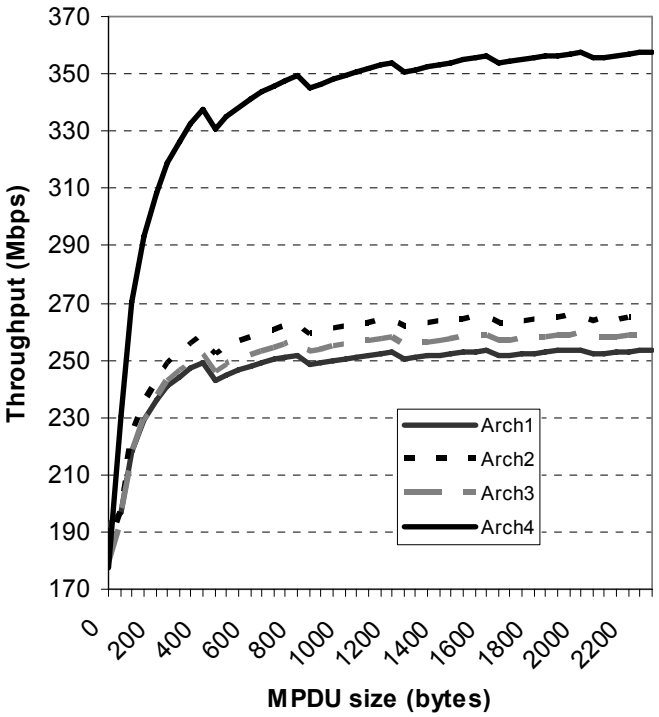
	Arch1	**Arch2**	**Arch3**	**Arch4**	**Arch5**
CLB Slices	1329	1324	1342	1638	1574
BlockRAMs	10	10	8	18	10
Clock cycle (ns)	10.909	10.407	10.665	10.671	10.558
Throughput (Mbps)	253	264	257	352	260
Throughput/Slice	0.097	0.101	0.109	0.089	0.091

Οι αρχιτεκτονικές Arch1, Arch2 και Arch3, που υλοποιούν μόνο το encapsulation σύμφωνα με το πρωτόκολλο CCMP καταλαμβάνουν περίπου την ίδια επιφάνεια πάνω στο FPGA. Η μικρή αύξηση όσον αφορά τα CLB Slices, που απαιτούνται από την Arch3, οφείλεται στην ύπαρξη των δύο πολυπλεκτών εύρους 8 bits, ώστε να είναι δυνατή η κοινή χρήση των δύο dual port ram από τη μονάδα υπολογισμού των κλειδιών και τη μονάδα κρυπτογράφησης του AES. Για αυτό το λόγο άλλωστε η Arch3 χρειάζεται οκτώ, αντί για δέκα, BlockRAMs. Αντίθετα, η παράλληλη αρχιτεκτονική Arch4 απαιτεί σημαντικά μεγαλύτερη επιφάνεια και περισσότερες BlockRAMs για να υλοποιηθεί. Αυτές οι απαιτήσεις πάντως δεν είναι πρόβλημα για τις χωρητικότητες των σημερινών συσκευών FPGA. Συγκρίνοντας την επιφάνεια που καταλαμβάνουν οι αρχιτεκτονικές Arch2 και Arch5, παρατηρούμε ότι η τροποποίηση του κυκλώματος encapsulation (Arch2), ώστε να υλοποιεί πλήρως το πρωτόκολλο CCMP (Arch5), επιτυγχάνεται με μόλις 19% αύξηση στον αριθμό των CLB Slices. Θυμίζουμε ότι και οι δύο αρχιτεκτονικές βασίζονται στη χρήση του αλγόριθμου AES με offline υπολογισμό κλειδιών. Συνολικά πρόκει-

ται για compact αρχιτεκτονικές οι οποίες καταλαμβάνουν το 69%, 69%, 70%, 21% και 81% αντίστοιχα των CLB Slices του FPGA.

Η αρχιτεκτονική Arch1 παρουσιάζει τη μεγαλύτερη καθυστέρηση για το κρίσιμο μονοπάτι, το οποίο βρίσκεται στη μονάδα επεξεργασίας του αλγορίθμου AES. Οι αρχιτεκτονικές Arch2 και Arch3 έχουν μικρότερη καθυστέρηση κατά περίπου 0.5ns και 0.25ns αντίστοιχα με το κρίσιμο μονοπάτι να βρίσκεται στη μονάδα κρυπτογράφησης. Για τις αρχιτεκτονικές Arch1 και Arch2 το κρίσιμο μονοπάτι περιλαμβάνει τα ίδια επίπεδα συνδυαστικής λογικής και η επιπλέον καθυστέρηση στην περίπτωση της Arch1 οφείλεται στη δρομολόγηση (routing) του σχεδιασμού. Στην περίπτωση της αρχιτεκτονικής Arch3, το κρίσιμο μονοπάτι περιλαμβάνει περισσότερα επίπεδα, καθώς οι πολυπλέκτες που προστίθενται στο σχεδιασμό βρίσκονται πάνω στο κρίσιμο μονοπάτι. Παρόλα αυτά η καθυστέρηση παραμένει μικρότερη σε σχέση με την Arch1, αλλά όπως αναμενόταν είναι μεγαλύτερη σε σχέση με την Arch2. Η ελάχιστη περίοδος ρολογιού για τις αρχιτεκτονικές Arch4 και Arch5 είναι περίπου η ίδια. Η τιμή της θα έπρεπε να είναι περίπου ίδια με την Arch2, αφού χρησιμοποιείται η ίδια αρχιτεκτονική για τη μονάδα επεξεργασίας του AES. Η επιπλέον καθυστέρηση ~0.3ns οφείλεται στη δρομολόγηση των σχεδιασμών.

Όσον αφορά το throughput, όλοι οι σχεδιασμοί ξεπερνούν κατά πολύ τις απαιτήσεις που θέτουν οι προδιαγραφές για τα ασύρματα δίκτυα IEEE 802.11b (11 Mbps) και IEEE 802.11a/g (54 Mbps). Είναι επομένως κατάλληλοι και για πιο απαιτητικές εφαρμογές, όπου χρησιμοποιούνται υψηλότεροι ρυθμοί μετάδοσης και οι απαιτήσεις σε ασφάλεια είναι αυξημένες. Οι αρχιτεκτονικές που υλοποιούν μόνο το encapsulation εμφανίζουν throughput που κυμαίνεται από 253 – 264 Mbps, για μέγεθος πακέτου δεδομένων 1024 bytes. Για την παράλληλη αρχιτεκτονική το throughput αυξάνεται στα 352 Mbps. Μειώνεται όμως η τιμή του throughput/slice, που λαμβάνει υπόψη τη συνολική επιφάνεια του σχεδιασμού και των απαιτούμενων BlockRAMs. Στο Σχήμα 6.9 φαίνεται το throughput για τις αρχιτεκτονικές που υλοποιούν το encapsulation.



Σχήμα 6.9Το throughput σε συνάρτηση με το μέγεθος του MPDU.

Η αρχιτεκτονική Arch5 εμφανίζει throughput αντίστοιχο με τις τρεις αρχιτεκτονικές του encapsulation. Η τιμή του throughput/slice επιβεβαιώνει ότι αυτό πραγματοποιείται χωρίς σημαντική αύξηση της επιφάνειας στο FPGA. Η αρχιτεκτονική αυτή έχει τα εξής σημαντικά χαρακτηριστικά:

1. Η πλήρης υλοποίηση του πρωτοκόλλου CCMP πραγματοποιείται διατηρώντας το throughput σε υψηλά επίπεδα.
2. Το throughput παραμένει το ίδιο για το encapsulation και το decapsulation.

Υπάρχει μόνο μία υλοποίηση του πρωτοκόλλου CCMP στη βιβλιογραφία, γνωστή στο συγγραφέα [41]. Το κύκλωμα υλοποιείται σε Virtex FPGA καταλαμβάνοντας 3750 Slices και επιτυγχάνει throughput 243 Mbps με περίοδο ρολογιού 20ns (50MHz). Όσον αφορά την απαιτούμενη επιφάνεια, δεν μπορεί να γίνει σύγκριση με την αρχιτεκτονική που προτείνεται σε αυτή τη διπλωματική, καθώς το κύκλωμα στο [41] περιλαμβάνει και τον αντίστροφο αλγόριθμο

AES για την εναλλακτική επιλογή του OCB[11], αντί για το CCMP. Επίσης, η σύγκριση αναφορικά με το throughput είναι δύσκολη για δύο λόγους. Πρώτον, χρησιμοποιείται διαφορετική τεχνολογία υλοποίησης (Virtex vs Spartan3) και δεύτερον το [41] δε λειτουργεί στη μέγιστη δυνατή συχνότητα, αλλά περιορίζεται στα 50MHz επειδή το I/O Interface είναι αρτηρία δεδομένων PCI. Επιπλέον, η τιμή του throughput στο [41] έχει υπολογιστεί χωρίς να λαμβάνονται υπόψη οι κύκλοι ανάγνωσης και εγγραφής. Παρόλα αυτά, η προτεινόμενη υλοποίηση του CCMP μπορεί να επιτύχει μεγαλύτερο throughput σε σχέση με το [41], για την ίδια τεχνολογία υλοποίησης και την ίδια συχνότητα λειτουργίας. Αυτό συμβαίνει διότι το δεύτερο υπολογίζει πρώτα την τιμή του MIC και στη συνέχεια κρυπτογραφεί τα δεδομένα. Χρειάζονται επομένως, διπλάσιοι κύκλοι για την ανάγνωση των δεδομένων, οπότε το throughput θα είναι σημαντικά μειωμένο σε σχέση με την προτεινόμενη υλοποίηση. Επιπλέον, το κύκλωμα στο [41] δεν μπορεί να επεξεργαστεί πακέτα μεταβλητού μεγέθους και επομένως δεν μπορεί να χρησιμοποιηθεί σε πραγματικά συστήματα.

[11] Η κατάσταση λειτουργίας Offset Code Book (OCB), παρέχει εμπιστευτικότητα, επικύρωση και ακεραιότητα. Αποτέλεσε την αρχική επιλογή για το IEEE 802.11i. Είναι πιο αποδοτικό σε σχέση με το CCMP, καθώς απαιτεί μόνο μία κρυπτογράφηση για τον υπολογισμό του MIC [42]. Παρόλα αυτά αμφιβολίες σχετικά με τα πνευματικά δικαιώματα, ανάγκασαν την ομάδα εργασίας TGi να προχωρήσει στην ανάπτυξη της κατάστασης λειτουργίας CCM.

Βιβλιογραφία

[1] IEEE Std. 802.11, 1999 Edition (http://ieeexplore.ieee.org/xpl/standards.jsp)
[2] IEEE Std. 802.11i-2004 (http://ieeexplore.ieee.org/xpl/standards.jsp)
[3] Real 802.11 Security: Wi-Fi Protected Access and 802.11i, Addison Wesley, 2003
[4] Matthew Gast, "802.11 Wireless Networks: The Definitive Guide", O'Reilly & Associates, Inc., Sebastopol, CA, 2002
[5] Russ Housley et al., "Security Flaws in 802.11 Data Link Protocols", Communications of the ACM, May 2003, Vol. 46, No. 5, pp. 35-39
[6] Fluhrer S., Mantin I., Shamir A., "Weaknesses in the Key Schedule Algorithm of RC4", 4th Annual Workshop on Selected Areas of Cryptography, 2001
[7] Stubblefield A., Ioannidis J., Rubin A., "Using the Fluhrer, Mantin and Shamir attack to break WEP", Network and Distributed Systems Security Symposium 2002, pp. 17-22
[8] Walker J., "Unsafe at Any Key Size: An Analysis of the WEP Encapsulation", IEEE 802.11 doc 00-362, 2000
[9] Borisov N., Goldberg I., Wagner D., "Intercepting Mobile Communications: The Insecurity of 802.11", International Conference on Mobile Computing and Networking, 2001, pp. 180-189
[10] Ferguson N., "An Improved MIC for 802.11 WEP", IEEE 802.11 doc 02-020r0, 2002
[11] FIPS-197, "Announcing the Advanced Encryption Standard (AES)", 2001
[12] Recommendation for Block Cipher Modes of Operation, NIST Special Publication 800-38A, 2001
[13] Doug Whiting, Russ Housley, Niels Ferguson, "Submission to NIST: Counter with CBC-MAC (CCM). AES Mode of Operation"
[14] Jonsson J., "On the Security of CTR & CBC-MAC", Proceedings of Selected Areas of Cryptography (SAC), 2002
[15] Zhang X., Parhi K., "High-Speed XLSI Architectures for the AES Algorithm", IEEE Transactions on VLSI Systems, Vol. 12, No. 9, 2004
[16] X. Zhang and K. K. Parhi, "Implementation approaches for the advanced encryption standard algorithm," IEEE Circuits Syst. Mag., vol. 2, no. 4, pp. 24–46, 2002.
[17] K. U. Jarvinen, M. T. Tommiska, and J. O. Skytta, "A fully pipelined memoryless 17.8 Gbps AES-128 encryptor," in Proc. Int. Symp. Field Programmable Gate Arrays (FPGA 2003), Monterey, CA, Feb. 2003, pp. 207–215.
[18] G. P. Saggese, A. Mazzeo, N. Mazocca, and A. G. M. Strollo, "An FPGA based performance analysis of the unrolling, tiling and pipelining of the AES algorithm," in Proc. FPL 2003, Portugal, Sept. 2003.
[19] F. Standaert, G. Rouvroy, J. Quisquater, and J. Legat, "Efficient implementation of Rijndael encryption in reconfigurable hardware: Improvements & design tradeoffs," in Proc. CHES 2003, Cologne, Germany, Sept. 2003.
[20] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, "A compact Rijndael hardware architecture with S-Box optimization," in Proc. ASIACRYPT 2001, Gold Coast, Australia, Dec. 2000, pp. 239–254.
[21] A. Rudra, P. K. Dubey, C. S. Jutla, V. Kumar, J. R. Rao, and P. Rohatgi, "Efficient implementation of Rijndael encryption with composite field arithmetic," in Proc. CHES 2001, Paris, France, May 2001, pp. 171–184.
[22] Christopher Caltagirone and Kasi Ananthal. High Throughput, Parallelized 128-bit AES Encryption in a Resource-Limited FPGA, in SPAA'03, June 2003.
[23] François-Xavier Standaert, Gael Rouvroy, JeanJacques Quisquater and Jean-Didier Legat. A Methodology to Implement Block Ciphers in Reconfigurable

Hardware and its Application to Fast and Compact AES RIJNDAEL, in FPGA'03, February 2003.
[24] A. J. Elbirt, W. Yip, B. Chetwynd, and C. Paar. An FPGA implementation and performance evaluation of the AES block cipher candidate algorithm finalist. presented at Proc. 3rd AES Conf. (AES3).
[25] V. Fischer and M. Drutarovsky, "Two methods of Rijndael implementation in reconfigurable hardware," in Proc. CHES 2001, Paris, France, May 2001, pp. 77–92.
[26] K. Gaj and P. Chodowiec. Comparison of the hardware performance of the AES candidates using reconfigurable hardware. presented at Proc. 3rd AES Conf. (AES3).
[27] H. Kuo and I. Verbauwhede, "Architectural optimization for a 1.82 Gbits/sec VLSI implementation of the AES Rijndael algorithm," in Proc. CHES 2001, Paris, France, May 2001, pp. 51–64.
[28] M. McLoone and J. V. McCanny, "Rijndael FPGA implementation utilizing look-up tables," in IEEE Workshop on Signal Processing Systems, 2001, pp. 349–360.
[29] N. Sklavos and O. Koufopavlou, "Architectures and VLSI Implementations of the AES-Proposal Rijndael", IEEE TRANSACTIONS ON COMPUTERS, VOL. 51, NO. 12, DECEMBER 2002
[30] K. Gaj and P. Chodowiec. Very Compact FPGA Implementation of the AES Algorithm. In the proceedings of CHES 2003, Lecture Notes in Computer Science, vol 2779, pp. 319-333, Springer-Verlag.
[31] FX. Standaert, G. Rouvoy, JJ. Quisquater, JD. Legat, "Compact and Efficient Encryption/Decryption Module for FPGA Implementation of the AES Rijndael VeryWell Suited for Small Embedded Applications", Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04)
[32] S. Mangard, M. Aigner, and S. Dominikus, "A Highly Regular and Scalable AES Hardware Architecture", IEEE TRANSACTIONS ON COMPUTERS, VOL. 52, NO. 4, APRIL 2003
[33] A. Hodjat and I. Verbauwhede, "A 21.54 Gbits/s Fully Pipelined AES Processor on FPGA", 12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM) 2004, pp. 308 - 309
[34] Helion Technology Limited, www.heliontech.com
[35] C. C. Lu and S. Y. Tseng, "Integrated design of AES (advanced encryption standard) encrypter and decrypter," in Proc. IEEE Int. Conf. Application Specific Systems, Architectures Processors, 2002, pp. 277–285.
[36] http://www.xilinx.com/bvdocs/userguides/ug130.pdf
[37] https://www.deadhat.com/wlancrypto/ccm1.2.c
[38] http://www.xilinx.com/support/sw_manuals/xilinx6/index.htm
[39] IETF RFC-3610 (http://www.ietf.org/rfc/rfc3610.txt)
[40] P. Prasithsangaree and P. Krishnamurthy, ''Analysis of Energy Consumption of RC4 and AES Algorithms in Wireless LANs'', Globecom, 2003
[41] Ho Yung Jang, "Compatible Design of CCMP and OCB AES Cipher Using Separated Encryptor and Decryptor for IEEE 802.11i", ISCAS 2004
[42] Philip Rogaway, "OCB mode", April 2001