ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Τομέας Επικοινωνιών, Ηλεκτρονικής & Συστημάτων Πληροφορικής
Εργαστήριο Δικτύων Υπολογιστών

Ασφαλής Πρόσβαση και Διαχείριση σε Ασύρματα Τοπικά Δίκτυα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Νικόλαος Δ. Λιαμπότης

Επιβλέπων: Ευστάθιος Δ. Συκάς
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2005

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
Τομέας Επικοινωνιών, Ηλεκτρονικής & Συστημάτων Πληροφορικής
Εργαστήριο Δικτύων Υπολογιστών

Ασφαλής Πρόσβαση και Διαχείριση σε Ασύρματα Τοπικά Δίκτυα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Νικόλαος Δ. Λιαμπότης

Επιβλέπων: Ευστάθιος Δ. Συκάς
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 19η Οκτωβρίου 2005.

.....	.....	.....
Ευστάθιος Συκάς Καθηγητής Ε.Μ.Π.	Μιχαήλ Θεολόγου Καθηγητής Ε.Μ.Π.	Γεώργιος Στασινόπουλος Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2005

.....
Νικόλαος Δ. Λιαμπότης
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

. . . στη Χριστίνα.

Περίληψη

Η τεχνολογία των Ασύρματων Τοπικών Δικτύων (Wireless Local Area Networks – WLAN) είναι ευρέως γνωστή και κερδίζει συνεχώς έδαφος στο χώρο των ασύρματων επικοινωνιών. Όσο όμως εξαπλώνεται η χρήση του WLAN, τόσο εντείνεται και η ανάγκη των χρηστών του για ένα ασφαλές περιβάλλον ανταλλαγής πληροφοριών. Το αρχικό σχήμα ασφαλείας που προδιαγράφεται από το πρωτόκολλο Απορρήτου Ισοδύναμου Ενσυρμάτου (Wired Equivalent Privacy – WEP), θεωρείται πλέον παρωχημένο, καθώς δεν ικανοποιεί τις ολοένα αυξανόμενες απαιτήσεις σε ασφάλεια.

Το πρωτόκολλο IEEE 802.11i από την άλλη πλευρά, αποτελεί ένα καινούργιο και πολλά υποσχόμενο πρότυπο ασφαλείας, το οποίο ορίζει ένα νέο τύπο ασύρματου δικτύου που λέγεται Δίκτυο Εύρωστης Ασφαλείας (Robust Security Network – RSN). Ωστόσο, επειδή οι απαιτούμενες κρυπτογραφικές λειτουργίες δεν υποστηρίζονται από το υπάρχον υλισμικό (hardware), το IEEE 802.11i ορίζει ένα Δίκτυο Μεταβατικής Ασφαλείας (Transitional Security Network – TSN), το οποίο καλύπτει όλες τις αδυναμίες του WEP, ενώ ταυτόχρονα εξασφαλίζει τη διαλειτουργικότητα (interoperability) μεταξύ συμβατικού και νεότερου εξοπλισμού ασύρματης δικτύωσης.

Σκοπός της παρούσας διπλωματικής εργασίας είναι η αξιολόγηση των νέων προσεγγίσεων για ασφάλεια στα Ασύρματα Τοπικά Δίκτυα και η εφαρμογή τους για την επίτευξη του μέγιστου επιπέδου ασφαλείας. Για το σκοπό αυτό, υλοποιήθηκε ένα Ασύρματο Τοπικό Δίκτυο που εφαρμόζει τις αρχές του Δικτύου Μεταβατικής Ασφάλειας και ελέγχθηκε η σωστή λειτουργία του, όσον αφορά τους μηχανισμούς πρόσβασης και διαχείρισής του. Επιπλέον, πραγματοποιήθηκαν δοκιμές, τόσο σε Ασύρματα Τοπικά Δίκτυα WEP, όσο και στην υλοποίησή μας, για την εξαγωγή συμπερασμάτων σε ό,τι αφορά την παρεχόμενη ασφάλεια.

Λέξεις Κλειδιά

Ασφάλεια, Ασύρματα Τοπικά Δίκτυα, IEEE 802.11, WEP, IEEE 802.11i, WPA, TSN, WPA2, RSN, IEEE 802.1X, EAP, RADIUS, Ασύρματες Επικοινωνίες

Abstract

The Wireless Local Area Networks (WLAN) technology is widely known and continuously gains appreciation in the field of wireless communications. However, as the usage of WLAN expands, its users' need for a secure environment for exchanging information becomes a necessity. The original security context specified in the Wired Equivalent Privacy protocol (WEP) is obsolete, as it fails to meet its growing demand for security.

On the other hand, the IEEE 802.11i protocol constitutes a new and promising security standard that defines a new type of wireless network called a Robust Security Network (RSN). However, as the required cryptographic functions are not supported by the existing hardware, the IEEE 802.11i defines a Transitional Security Network (TSN) that covers all weaknesses of WEP, while, at the same time, ensures interoperability between legacy and newer wireless networking equipment.

The scope of this thesis is the evaluation of the new Wireless Local Area Networks security approaches and their appliance to achieve the maximum level of security. For this purpose, a Wireless Local Area Network based on the Transitional Security Network was deployed and tested regarding its access and management mechanisms. Moreover, a series of experiments was done on WEP enabled Wireless Local Area Networks, as well as, our deployment, in order to extract conclusions related to the provided security.

Keywords

Security, Wireless Local Area Networks, IEEE 802.11, WEP, IEEE 802.11i, WPA, TSN, WPA2, RSN, IEEE 802.1X, EAP, RADIUS, Wireless Communications

Ευχαριστίες

Η πραγματοποίηση της παρούσας διπλωματικής εργασίας δε θα ήταν εφικτή χωρίς τη βοήθεια κάποιων ανθρώπων οι οποίοι συνέβαλαν με την καθοδήγησή τους, τόσο στο σχεδιασμό όσο και στην υλοποίηση της εφαρμογής που αναπτύχθηκε, βοήθησαν στη διόρθωση σφαλμάτων που ανέκυψαν και με την τεχνογνωσία τους διευκόλυναν τη διεξαγωγή των πειραμάτων και των μετρήσεων για την αξιολόγησή της.

Όλους αυτούς δε θα μπορούσα να παραλείψω να τους ευχαριστήσω, ξεκινώντας από τον επιβλέποντα της διπλωματικής μου εργασίας, Καθηγητή Ε.Μ.Π. κ. Ευστάθιο Συκά για τη δυνατότητα που μου προσέφερε να ασχοληθώ με αυτό το ενδιαφέρον και σύγχρονο θέμα, αλλά και για την αμέριστη βοήθεια που μου παρείχε κατά την πορεία της έρευνας και της ανάπτυξης της παρούσας εργασίας.

Τέλος, θα ήθελα να ευχαριστήσω όλα τα μέλη του Εργαστηρίου Δικτύων Υπολογιστών για τις συμβουλές τους και τις εύστοχες παρατηρήσεις τους, τη μαμά μου, Μαρία, το μπαμπά μου, Δημήτρη και τον αδερφό μου, Γιάννη.

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Κεφάλαιο 1

Εισαγωγή

Το εισαγωγικό αυτό κεφάλαιο πραγματεύεται το αντικείμενο της παρούσας διπλωματικής εργασίας και παρουσιάζει εν συντομία τα κεφάλαια που θα ακολουθήσουν.

1.1 Αντικείμενο της διπλωματικής εργασίας

Η ανάγκη για μια ασύρματη μέθοδο επικοινωνίας των υπολογιστικών συστημάτων οδήγησε στην ανάπτυξη των Ασύρματων Τοπικών Δικτύων, που προδιαγράφονται από το πρωτόκολλο IEEE 802.11. Αποτελούν μια καινοτόμο τεχνολογία, η οποία συγκεντρώνει μεγάλο ερευνητικό ενδιαφέρον και εξελίσσεται ραγδαία. Η ταχύτατη πρόοδος της τεχνολογίας αυτής, σε συνδυασμό με τη μείωση του οικονομικού της κόστους, την καθιστά σήμερα ως το πιο δημοφιλές μέσο ασύρματης δικτύωσης, τόσο στο εταιρικό, όσο και στο οικιακό περιβάλλον. Η δυνατότητα που προσφέρουν τα Ασύρματα Τοπικά Δίκτυα για επικοινωνία μεγάλων σχετικά αποστάσεων και υψηλών ταχυτήτων, επιπέδου ενσύρματων Τοπικών Δικτύων, είναι ο λόγος που έχει επικρατήσει έναντι άλλων τεχνολογιών ασύρματης επικοινωνίας, όπως είναι το Bluetooth.

Το αντικείμενο της παρούσας διπλωματικής εργασίας είναι η ασφάλεια στα Ασύρματα Τοπικά Δίκτυα, ένα ζήτημα ιδιαίτερης σημασίας στο χώρο των δικτύων δεδομένων γενικότερα. Στο πλαίσιο αυτό θα παρουσιάσουμε τις αδυναμίες της νέας αυτής μεθόδου ασύρματης επικοινωνίας σε θέματα ασφαλείας. Στη συνέχεια θα μελετήσουμε το πρωτόκολλο Απορρήτου Ισοδύναμου Ενσυρμάτου, το οποίο αποτελεί το αρχικό σχήμα κρυπτογραφικής ασφάλειας που υιοθετήθηκε από το IEEE 802.11. Αναλύονται οι δυνατότητες του προτύπου όσον αφορά το απόρρητο, την επαλήθευση ταυτότητας και τον έλεγχο πρόσβασης και επισημαίνονται οι αδυναμίες του.

Στη συνέχεια θα μελετήσουμε το πρωτόκολλο IEEE 802.11i, το νέο πρότυπο ασφαλείας στα Ασύρματα Τοπικά Δίκτυα. Θα συγκρίνουμε τις δύο προσεγγίσεις που αυτό προτείνει: 1) πρωτόκολλο Ασύρματης Προστατευμένης Πρόσβασης, γνωστό ακόμη με την ονομασία Δίκτυο Μεταβατικής Ασφάλειας, και 2) το Δίκτυο Εύρωστης Ασφαλείας. Σκοπός είναι να γίνει πλήρως κατανοητό ότι η πρώτη προσέγγιση αποτελεί τη δόκιμη λύση ασφαλείας σήμερα. Για την ενίσχυση της άποψης αυτής και προκειμένου να απορριφθούν οι άλλες τεχνικές, θα πραγματοποιήσουμε επιθέσεις σε δίκτυα στα οποία έχουν εγκαταταθεί τα διάφορα σχήματα ασφαλείας και θα παρουσιάσουμε τα αποτελέσματα.

Έπειτα θα προχωρήσουμε στο σχεδιασμό και την υλοποίηση ενός Δικτύου Μεταβατικής Ασφάλειας, κατάλληλου για την παροχή υπηρεσιών σε διαφορετικές κατηγορίες χρηστών: 1) εξουσιοδοτημένα μέλη και 2) επισκέπτες. Για το σκοπό αυτό, θα χρησιμοποιήσουμε το

μηχανισμό ελέγχου πρόσβασης που προτείνει το IEEE 802.11i για επαλήθευση ταυτότητας βάσει θύρας (IEEE 802.1X), σε συνδυασμό με τα πρωτόκολλα Remote Access Dial-In User Service και Lightweight Directory Access Protocol για κεντρικοποιημένη διαχείριση των πιστοποιητικών ταυτότητας.

1.2 Οργάνωση του κειμένου

Ακολουθεί μια σύντομη περιγραφή της διάρθρωσης της παρούσας διπλωματικής εργασίας και μια ακριβής αναφορά στα θέματα που καλύπτονται σε κάθε κεφάλαιο.

Στη συνέχεια της εργασίας, στο **κεφάλαιο 2**, παρουσιάζονται οι τεχνολογίες και αρχιτεκτονικές που χρησιμοποιήθηκαν στην παρούσα διπλωματική εργασία, ενώ επεξηγούνται έννοιες και ορισμοί απαραίτητοι για την κατανόησή τους. Αρχικά παρουσιάζεται το πρωτόκολλο IEEE 802.11, δηλαδή, η τεχνολογία των Ασύρματων Τοπικών Δικτύων και οι δυνατότητες που αυτή προσφέρει στους χρήστες της. Αναφέρονται ακόμη οι αδυναμίες της συγκεκριμένης τεχνολογίας, όσον αφορά την ασφάλεια, και επιπλέον περιγράφονται οι τύποι των επιθέσεων στις οποίες είναι ευάλωτη.

Στο **κεφάλαιο 3**, παρουσιάζεται το πρωτόκολλο Απορρήτου Ισοδύναμου Ενσυρμάτου, που αποτέλεσε την πρώτη προσπάθεια ανάπτυξης ενός σχήματος ασφαλείας για τα Ασύρματα Τοπικά Δίκτυα. Πιο συγκεκριμένα περιγράφονται οι διάφοροι μηχανισμοί του πρωτοκόλλου, ενώ αναλύονται σε βάθος οι αδυναμίες του.

Έπειτα, το **κεφάλαιο 4**, πραγματεύεται το πρωτόκολλο IEEE 802.11i, το νέο πρότυπο ασφαλείας, όσον αφορά τα Ασύρματα Τοπικά Δίκτυα. Παρουσιάζονται οι δύο προσεγγίσεις που αυτό προτείνει, δηλαδή το πρωτόκολλο Ασύρματης Προστατευμένης Πρόσβασης και το Δίκτυο Εύρωστης Ασφαλείας.

Το **κεφάλαιο 5**, παρουσιάζει τους διάφορους μηχανισμούς ελέγχου πρόσβασης που προτείνονται στα πλαίσια του προτύπου ασφαλείας IEEE 802.11i. Πιο συγκεκριμένα, περιγράφονται τα πρωτόκολλα IEEE 802.1X, Πρωτόκολλο Επεκτάσιμης Επαλήθευσης Ταυτότητας και Remote Access Dial-In User Service, ενώ παρουσιάζεται η αλληλεπίδρασή τους για την παροχή ελέγχου πρόσβασης στα Ασύρματα Τοπικά Δίκτυα.

Το **κεφάλαιο 6**, αφορά εξολοκλήρου στο σχεδιασμό και την υλοποίηση ενός Δικτύου Μεταβατικής Ασφάλειας. Αρχικά, αναφέρονται οι λόγοι για τους οποίους καθίσταται επιτακτική η ανάγκη μετάβασης στο νέο σχήμα ασφαλείας. Στο πλαίσιο αυτό, παρουσιάζονται παραδείγματα επιθέσεων στο Απόρρητο Ισοδύναμο Ενσυρμάτου, ενώ ύστερα από λήψη μετρήσεων συγκρίνεται η συμπεριφορά του πρωτοκόλλου Ασύρματης Προστατευμένης Πρόσβασης σε αντίστοιχες επιθέσεις. Εξάλλου, στο μεγαλύτερο μέρος του κεφαλαίου, περιγράφεται ο τρόπος σχεδιασμού του Δικτύου Μεταβατικής Ασφάλειας και τα τεχνικά χαρακτηριστικά της πλατφόρμας που υλοποιήθηκε.

Στο **κεφάλαιο 7**, παρατίθεται μια σύνοψη των αποτελεσμάτων της παρούσας διπλωματικής εργασίας και των συμπερασμάτων τα οποία εξήχθησαν. Τέλος, πραγματοποιούνται προτάσεις για περαιτέρω έρευνα πιθανών μελλοντικών επεκτάσεων, η διερεύνηση των οποίων, παρουσιάζει σημαντικό ενδιαφέρον.

Η εργασία κλείνει με την παρουσίαση της βιβλιογραφίας που χρησιμοποιήθηκε.

Κεφάλαιο 2

Ασύρματα Τοπικά Δίκτυα και ασφάλεια

Στο κεφάλαιο αυτό, παρατίθεται αρχικά μια επισκόπηση των ασύρματων επικοινωνιών και στη συνέχεια παρουσιάζεται το πρωτόκολλο IEEE 802.11, δηλαδή, η τεχνολογία των Ασύρματων Τοπικών Δικτύων και οι δυνατότητες που αυτή προσφέρει στους χρήστες της. Αναφέρονται ακόμη οι αδυναμίες της συγκεκριμένης τεχνολογίας, όσον αφορά την ασφάλεια, και επιπλέον περιγράφονται οι τύποι των επιθέσεων στις οποίες είναι ευάλωτη.

2.1 Επισκόπηση ασύρματων επικοινωνίων

Τα ασύρματα δίκτυα, όπως και τα αντίστοιχα ενσύρματα, εκμεταλλεύονται την ηλεκτρική τάση, ώστε να είναι δυνατή η επικοινωνία μεταξύ των συσκευών. Μεταβολές στην ισχύ του σήματος από μηδέν μέχρι μια μέγιστη τιμή (πλάτος) και ο ρυθμός των μεταβολών αυτών (συχνότητα), χρησιμοποιούνται κατάλληλα για την κωδικοποίηση και την αποκωδικοποίηση της πληροφορίας. Όταν δύο συσκευές κατανοούν τις μεθόδους που χρησιμοποιούνται για την κωδικοποίηση και αποωδικοποίηση της πληροφορίας που περιέχεται στις μεταβολές των ηλεκτρικών ιδιοτήτων του μέσου επικοινωνίας (κανάλι), τότε είναι σε θέση να επικοινωνούν μεταξύ τους.

Η προφάνης και ειδοποιός διαφορά μεταξύ των ενσύρματων και ασύρματων δικτύων είναι ότι τα δεύτερα χρησιμοποιούν σήματα ραδιοσυχνότητας (Radio Frequency – RF), που δημιουργούνται εφαρμόζοντας εναλλασσόμενο ρεύμα σε μια κεραία για την παραγωγή ενός ηλεκτρομαγνητικού πεδίου (Electromagnetic – EM). Το πεδίο RF που προκύπτει χρησιμοποιείται από τις συσκευές για μετάδοση και λήψη. Στην περίπτωση των ασύρματων δικτύων, το μέσο επικοινωνίας είναι το πεδίο EM, δηλαδή, η περιοχή του χώρου που επηρεάζεται από την ηλεκτρομαγνητική ακτινοβολία. Όπως συμβαίνει και στα ενσύρματα δίκτυα, το πλάτος μειώνεται με την απόσταση, με αποτέλεσμα την υποβάθμιση της ισχύος του σήματος και τελικά της δυνατότητας επικοινωνίας.

Τα ασύρματα δίκτυα λειτουργούν στο Φυσικό (Physical) επίπεδο και το επίπεδο Ζεύξης Δεδομένων (Data Link), σύμφωνα με το πρότυπο OSI [1].

2.2 Πρωτόκολλο IEEE 802.11

Τα Ασύρματα Τοπικά Δίκτυα (Wireless Local Area Networks – WLANs) καλύπτονται από το πρωτόκολλο IEEE 802.11, στόχος του οποίου είναι, η προδιαγραφή ενός ασύρματου ισοδύναμου των δικτύων που βασίζονται στο IEEE 802.3. Το πρότυπο IEEE 802.3, ορίζει τη μέθοδο Ανιχνευσης Φέροντος Μέσου Πολλαπλής Πρόσβασης με Εντοπισμό Σύγκρουσης (Carrier Sense Multiple Access with Collision Detect – CSMA/CD) για την αντιμετώπιση των συγκρούσεων, διάφορες ταχύτητες λειτουργίας (10, 100, 1000 Mbps) και τύπους καλωδίων (Κατηγορία 5 συνεστραμμένου ζεύγους και οπτικές ίνες). Το πρότυπο εξασφαλίζει τη διαλειτουργικότητα (interoperablility) των διαφόρων συσκευών με διαφορετικές ταχύτητες και τύπους καλωδίωσης.

Αντίστοιχα, το πρότυπο IEEE 802.11 ορίζει μεθόδους για την αντιμετώπιση συγκρούσεων και διαφορετικών ταχυτήτων λειτουργίας. Ωστόσο, εξαιτίας των διαφορών όσον αφορά το μέσο μετάδοσης, τις συσκευές που χρησιμοποιούνται, την ενδεχόμενη κινητικότητα (mobility) των συνδεδεμένων στο δίκτυο χρηστών, και των εναλλακτικών τοπολογιών ασύρματων δικτύων, το IEEE 802.11 διαφέρει σημαντικά από το IEEE 802.3. Τα Ασύρματα Τοπικά Δίκτυα χρησιμοποιούν το πρωτόκολλο Ανίχνευσης Φέροντος Μέσου Πολλαπλής Πρόσβασης με Αποφυγή Σύγκρουσης (Carrier Sense Multiple Access with Collision Avoidance – CSMA/CA) για την αντιμετώπιση ενδεχόμενων συγκρούσεων, σε αντίθεση με το CSMA/CD του IEEE 802.3, καθώς όλοι οι σταθμοί σε ένα ασύρματο δίκτυο δεν είναι σε θέση να ακούσουν τις συγκρούσεις που μπορεί να συμβούν στο δίκτυο.

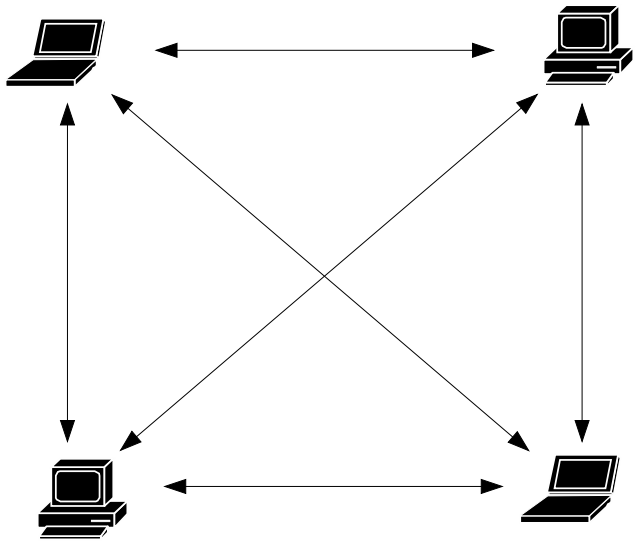
Εκτός από την επίλυση των προβλημάτων που σχετίζονται με τις συγκρούσεις που συμβαίνουν σε ένα ασύρματο δίκτυο, το πρότυπο IEEE 802.11 καλείται να αντιμετωπίσει επιπλέον θέματα που αφορούν ειδικά τη φύση των ασύρματων συσκευών και επικοινωνιών γενικότερα. Για παράδειγμα οι ασύρματες συσκευές πρέπει να είναι σε θέση να εντοπίζουν άλλες ασύρματες συσκευές και να επικοινωνούν με αυτές. Επιπλέον, λόγω της κινητικότητας τους, οι χρήστες ασύρματων συσκευών πρέπει να έχουν τη δυνατότητα να μετακινούνται από τη μια ασύρματη ζώνη στην άλλη. Τέλος, απαιτείται η παροχή ασφάλειας, όσον αφορά την ασύρματη επικοινωνία, που είναι και το θέμα της παρούσας διπλωματικής εργασίας.

2.2.1 Αρχιτεκτονική

Το πρότυπο IEEE 802.11 παρέχει δύο διαφορετικές αρχιτεκτονικές για την επικοινωνία μεταξύ ασύρματων συσκευών: λειτουργία **επί τούτω** (ad-hoc mode) και λειτουργία **υποδομής** (infrastructure mode). Στη συνέχεια παρουσιάζουμε κάθε τρόπο λειτουργίας ξεχωριστά.

Λειτουργία επί τούτω

Η λειτουργία Ασύρματων Τοπικών Δικτύων επί τούτω, επιτυγχάνει τη διασύνδεση ασύρματων συσκευών που είναι σε θέση να επικοινωνούν απ΄ ευθείας μεταξύ τους. Επομένως, στην αρχιτεκτονική αυτή, οι σταθμοί ομαδοποιούνται σε μια περιορισμένη γεωγραφική περιοχή. Η λειτουργία επί τούτω είναι παρόμοια με τα δίκτυα ομοτίμων οντοτήτων (peer-to-peer networks), όπου κανένας κόμβος δεν απαιτείται να παίζει το ρόλο του εξυπηρετητή. Οι διασυνδεδεμένες συσκευές στη λειτουργία επί τούτω, αναφέρονται και με τον όρο **Ανεξάρτητο Συνόλο Βασικών Υπηρεσιών** (Independent Basic Service Set – IBSS). Η τοπολογία επί τούτω φαίνεται στο Σχ. 2.1.



Σχήμα 2.1: Λειτουργία Ασύρματου Τοπικού Δικτύου επί τούτω

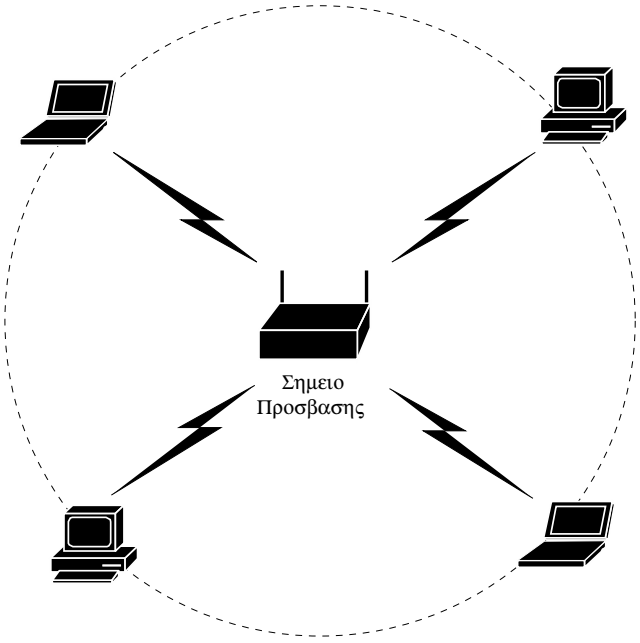
Λειτουργία υποδομής

Η λειτουργία Ασύρματων Τοπικών Δικτύων υποδομής βασίζεται σε σταθερά **σημεία πρόσβασης** (access points), με τη βοήθεια των οποίων, καθίσταται δυνατή η επικοινωνία των ασύρματων κόμβων. Το δίκτυο υποδομής επιτυγχάνει την επέκταση του εύρους του ενσύρματου Τοπικού Δικτύου σε ασύρματες κυψέλες. Μια ασύρματη συσκευή είναι σε θέση να μετακινείται από κυψέλη σε κυψέλη, δηλαδή από σημείο πρόσβασης σε σημείο πρόσβασης, διατηρώντας την πρόσβασή της στους πόρους του Τοπικού Δικτύου. Μια κυψέλη είναι η περιοχή που καλύπτεται από ένα σημείο πρόσβασης και ονομάζεται **Σύνολο Βασικών Υπηρεσιών** (Basic Service Set – BSS). Το σύνολο όλων των κυψελών ενός δικτύου υποδομής ονομάζεται **Επεκτεταμένο Σύνολο Υπηρεσιών** (Extended Service Set – ESS). Η τοπολογιά υποδομής φαίνεται στο Σχ. 2.2.

2.2.2 Τεχνικά χαρακτηριστικά

Οι λειτουργίες που προδιαγράφονται από το πρότυπο IEEE 802.11 ανήκουν στο Φυσικό επίπεδο και το επίπεδο Ζεύξης Δεδομένων, ενώ τα δεδομένα των ανώτερων επιπέδων θεωρούνται ωφέλιμο φορτίο (payload). Τα πλαίσια (frames) που ορίζονται από το πρωτόκολλο διαφέρουν σημαντικά από τα αντίστοιχα του IEEE 802.3 και διακρίνονται σε τρεις τύπους: **διαχείρισης** (management), **ελέγχου** (control) και **δεδομένων** (data). Γενικά, κάθε τύπος πλαισίου παρέχει μεθόδους προκειμένου οι ασύρματες συσκευές να εντοπίσουν, να συσχετιστούν (associate), να αποσυσχετιστούν (disassociate), και να επαληθεύσουν την ταυτότητα τους μεταξύ τους. Επιπρόσθετα, ορίζονται λειτουργίες ώστε να μεταβάλλεται ο ρυθμός μετάδοσης ανάλογα με το επίπεδο ισχύος του σήματος, καθώς επίσης, και λειτουργίες για την εξοικονόμηση ενέργειας.

Από το σύνολο των πληροφοριών που μεταδίδονται μέσω των πλαισίων ελέγχου του πρωτοκόλλου IEEE 802.11, αξίζει να γίνει αναφορά στο **Αναγνωριστικό Συνόλου Υπηρεσιών** (Service Set Identifier – SSID). Πρόκειται για το αναγνωριστικό που διαθέτει κάθε ασύρματο δίκτυο (ή υποσύνολό του) και διακρίνεται σε IBSSID, BSSID και ESSID, ανάλογα με τον τύπο λειτουργίας που αναφέρεται, IBSS, BSS και ESS, αντίστοιχα. Όταν λειτουργούν



Σχήμα 2.2: Λειτουργία Ασύρματου Τοπικού Δικτύου υποδομής

περισσότερα από ένα Ασύρματα Τοπικά Δίκτυα στον ίδιο χώρο, το SSID χρησιμοποιείται για την επιλογή του δικτύου με το οποίο θέλει να συνδεθεί μια ασύρματη συσκευή.

Το πρωτόκολλο IEEE 802.11 είναι στην πραγματικότητα ένα *σύνολο* προτύπων που προδιαγράφουν τη μετάδοση δεδομένων πάνω από Ασύρματα Τοπικά Δίκτυα. Τα πρότυπα αυτά είναι τα εξής:

IEEE 802.11a: Περιγράφει ένα Ασύρματο Τοπικό Δίκτυο που λειτουργεί στη ζώνη των 5 GHz και παρέχει μέγιστο ρυθμό μετάδοσης δεδομένων που φτάνει τα 54 Mbps. Χρησιμοποιεί την τεχνολογία *Πολυπλεξίας Ορθογώνιας Διαίρεσης Συχνότητας* (Orthogonal Frequency Division Multiplexing – OFDM) για τη διαμόρφωση (modulation). Η ζώνη λειτουργίας των 5 GHz είναι λιγότερο συνωστισμένη από άλλου τύπου συσκευές, σε σχέση με τη ζώνη των 2,4 GHz, και επομένως ο κίνδυνος παρεμβολών είναι πιο περιορισμένος.

IEEE 802.11b: Πρόκειται για την πιο διαδεδομένη τεχνολογία Ασύρματων Τοπικών Δικτύων. Περιγράφει ένα Ασύρματο Τοπικό Δίκτυο που λειτουργεί στη ζώνη των 2,4 GHz και παρέχει μέγιστο ρυθμό μετάδοσης δεδομένων που φτάνει τα 11 Mbps. Χρησιμοποιεί τη μέθοδο της *Φασματικής Εξάπλωσης Άμεσης Ακολουθίας Στοιχείων* (Direct Sequence Spread Spectrum – DSSS) για τη διαμόρφωση. Το πρότυπο αυτό, ήταν επίσης γνωστό ως τεχνολογία **Ασύρματης Αξιοπιστίας** (Wireless Fidelity – Wi-Fi), ωστόσο, σήμερα, το Wi-Fi περιλαμβάνει *όλα* τα πρότυπα Ασύρματων Τοπικών Δικτύων. Η ζώνη λειτουργίας των 2,4 GHz είναι ιδιαίτερα συνωστισμένη (φούρνοι μικροκυμάτων, κινητά τηλέφωνα, Bluetooth) και επομένως ο κίνδυνος από παρεμβολές είναι σημαντικός.

IEEE 802.11g: Περιγράφει ένα Ασύρματο Τοπικό Δίκτυο που λειτουργεί στη ζώνη των 2,4 GHz και παρέχει μέγιστο ρυθμό μετάδοσης δεδομένων που φτάνει τα 54 Mbps (για μικρές αποστάσεις). Χρησιμοποιεί την τεχνολογία *Πολυπλεξίας Ορθογώνιας Διαίρεσης*

Συχνότητας (Orthogonal Frequency Division Multiplexing – OFDM) για τη διαμόρφωση. Προφανώς, όσον αφορά τις παρεμβολές στη ζώνη των 2,4 GHz ισχύει ό,τι και στο πρότυπο IEEE 802.11b.

Τα τεχνικά χαρακτηριστικά των παραπάνω προτύπων, συνοψίζονται στον Πίνακα 2.1.

Πρότυπο	IEEE 802.11a	IEEE 802.11b	IEEE 802.11g
Ζώνη Συχνοτήτων	5 GHz	2,4 GHz	2,4 GHz
Μέγιστος Ρυθμός Μετάδοσης	54 Mbps	11 Mbps	54 Mbps
Διαμόρφωση	OFDM	DSSS	OFDM

Πίνακας 2.1: Τεχνικά χαρακτηριστικά προτύπων IEEE 802.11

2.3 Ασφάλεια και τύποι επιθέσεων

Η ενότητα αυτή, παρουσιάζει τους διαφορετικούς τύπους επιθέσεων που μπορεί να αντιμετωπίσει ένα Ασύρματο Τοπικό Δίκτυο. Οι επιθέσεις αυτές μπορούν να ταξινομηθούν σε τέσσερις βασικές κατηγορίες:

1. Λήψη Πληροφοριών (Snooping/Footprinting)
2. Τροποποιήση Δεδομένων
3. Μεταμφίεση (Masquerading)
4. Άρνηση Υπηρεσιών (Denial of Service – DoS)

Στην πράξη, μια επίθεση ενδέχεται να εφαρμόσει συνδυασμό των παραπάνω προσεγγίσεων, ενώ το σύνολο των επιθέσεων ξεκινά με τη λήψη πληροφοριών. Στις υποενότητες που ακολουθούν, παρουσιάζεται κάθε κατηγορία ξεχωριστά.

2.3.1 Λήψη πληροφοριών

Η λήψη πληροφοριών αναφέρεται στην πρόσβαση σε ευαίσθητα και απόρρητα δεδομένα ενός δικτύου από μη εξουσιοδοτημένους χρήστες. Η μέθοδος της κρυπτογράφησης μπορεί να χρησιμοποιηθεί, προκειμένου να αντιμετωπιστούν τέτοιου είδους επιθέσεις. Στην περίπτωση αυτή, είναι απαραίτητη στον επίδοξο εισβολέα, είτε η γνώση του μυστικού κλειδιού κρυπτογράφησης, είτε κάποια ευφυής τεχνική για την ανάκτηση των πληροφοριών από τα κρυπτογραφημένα δεδομένα.

2.3.2 Τροποποιήση δεδομένων

Οι μέθοδοι τροποποιήσης δεδομένων διαφοροποιούνται σημαντικά. Υπάρχουν λιγότερο και περισσότερο προφανείς περιπτώσεις, όπως για παράδειγμα σε μια ηλεκτρονική τραπεζική συναλλαγή, η αντικατάσταση του λογαριασμού τράπεζας που προορίζεται ένα ποσό ή η τροποποίηση του ίδιου του ποσού μου μεταφέρεται, προς όφελος ενός κακόβουλου χρήστη. Τροποποιήσεις υψηλού επιπέδου, όπως αυτές που αναφέρθηκαν, αν και εφικτές, είναι αρκετά περιορισμένες στην πράξη λόγω του μεγάλου βαθμού δυσκολίας που ενέχουν. Ωστόσο, υπάρχουν εκλεπτυσμένες μέθοδοι τροποποίησης που είναι πιο κοντά στην πραγματικότητα. Για παράδειγμα, η

αλλαγή της διεύθυνσης IP του παραλήπτη στην επικεφαλίδα ενός μηνύματος, θα μπορούσε να το προωθήσει εκτός της κρυπτογραφημένες ασύρματης ζεύξης σε κάποιο κακόβουλο χρήστη, αντί για τον αρχικό προορισμό. Έτσι, ενώ η τροποποίηση του ίδιου του περιεχομένου ενός μηνύματος δεν είναι πάντοτε εφικτή, το ίδιο δεν ισχύει για παραμέτρους του μηνύματος που έχουν σαφώς περισσότερο περιορισμένο σύνολο δυνατών τιμών, όπως για παράδειγμα η επικεφαλίδα IP που αναφέρθηκε παραπάνω.

2.3.3 Μεταμφίεση

Ο όρος μεταμφίεση αναφέρεται στην περίπτωση κατά την οποία η δικτυακή συσκευή ενος κακόβουλου χρήστη παριστάνει μια άλλη έγκυρη συσκευή. Είναι η ιδανική προσέγγιση, εφόσον ο επίδοξος εισβολέας επιθυμεί να παραμείνει απαρατήρητος. Από τη στιγμή που η συσκευή κατορθώνει να αναγνωριστεί ως νόμιμη, ο κακόβουλος χρήστης αποκτά όλα τα δικαιώματα που παρέχει το δίκτυο στα εξουσιοδοτημένα μέλη του.

2.3.4 Άρνηση υπηρεσιών

Η επίθεση αυτή διαφέρει από τις τρεις πρώτες κατηγορίες, τόσο όσον αφορά την τεχνική, όσο και τους στόχους της. Έτσι, ενώ στις άλλες ο κακόβουλος χρήστης *αποκτά* επιπλέον δικαιώματα, κατά την επίθεση άρνησης υπηρεσιών, *αφαιρούνται* τα δικαιώματα από όλους τους χρήστες, νόμιμους και μη. Το αντικείμενο μιας επίθεσης DoS είναι η διαταραχή της ομαλής λειτουργίας του δικτύου που αποτελεί στόχος. Για παράδειγμα, στέλνοντας ένα μεγάλο αριθμό αιτήσεων σε έναν εξυπηρετητή ιστού, εξαντλούνται οι πόροι του τελευταίου, με αποτέλεσμα να μη μπορούν να εξυπηρετηθούν οι έγκυρες αιτήσεις. Στις περισσότερες περιπτώσεις, κίνητρο πίσω από τέτοιου είδους επιθέσεις, αποτελεί η επίδειξη δύναμης και η ικανοποίηση του εγωισμού κάποιου κακόβουλου χρήστη, παρά κάποιο οικονομικό όφελος. Ειδικά στην περίπτωση των ασύρματων τοπικών δικτύων, οι επιθέσεις DoS υλοποιούνται σχετικά εύκολα και είναι σχεδόν αδύνατο να αποτραπούν.

Κεφάλαιο 3

Απόρρητο Ισοδύναμο Ενσυρμάτου

Το κεφάλαιο αυτό, είναι αφιερωμένο στο Απόρρητο Ισοδύναμο Ενσυρμάτου (Wired Equivalent Privacy – WEP), που αποτελεί το αρχικό σχήμα ασφαλείας του προτύπου IEEE 802.11. Στις επόμενες ενότητες παρουσιάζονται οι μηχανισμοί του WEP και ο τρόπος υλοποίησης τους, ενώ παρατίθεται εκτενής ανάλυση των αδυναμιών του.

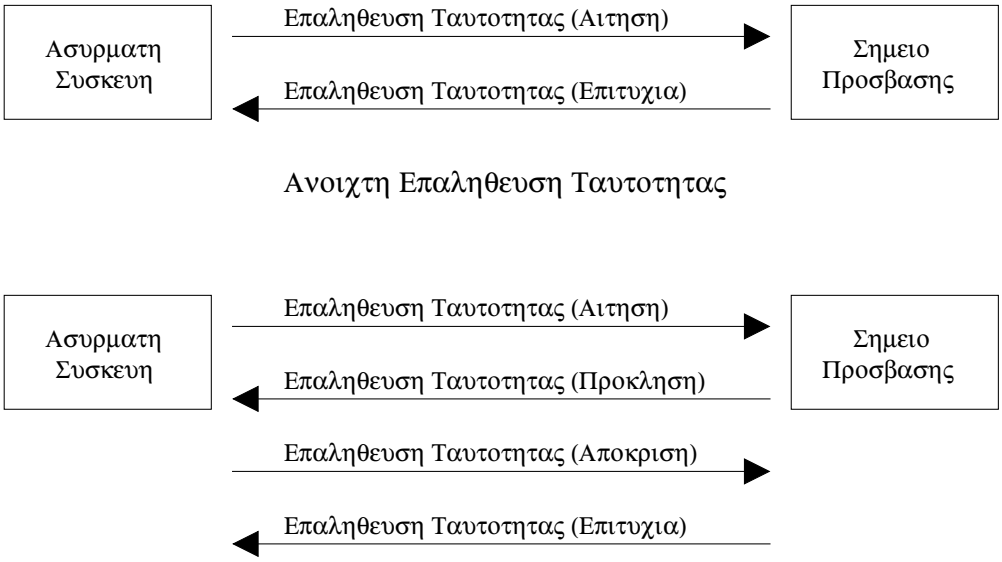
3.1 Επαλήθευση ταυτότητας

Η επαλήθευση ταυτότητας αποτελεί τη διαδικασία κατά την οποία μια δικτυακή οντότητα πιστοποιεί την ταυτότητά της. Στα πλαίσια των Ασύρματων Τοπικών Δικτύων, μια κινητή συσκευή προκειμένου να συνδεθεί στο δίκτυο μέσω κάποιου σημείου πρόσβασης, απαιτείται να αποδείξει πρώτα την ταυτότητά της. Ιδανικά, θα έπρεπε και το σημείο πρόσβασης να πράξει όμοια.

Τα αρχικό πρότυπο IEEE 802.11 προδιαγράφει δύο διαδικασίες επαλήθευσης ταυτότητας: α) την **ανοιχτή** (open) επαλήθευση ταυτότητας και β) την επαλήθευση ταυτότητας **WEP**. Τα μηνύματα που ανταλλάσσονται μεταξύ μιας ασύρματης συσκευής και του σημείου πρόσβασης σε κάθε περίπτωση, φαίνονται στο Σχ. 3.1. Όσον αφορά την ανοιχτή επαλήθευση ταυτότητας, παρατηρούμε ότι η ασύρματη συσκευή στέλνει αρχικά ένα μήνυμα αίτησης επαλήθευσης ταυτότητας και το σημείο πρόσβασης αποκρίνεται με μήνυμα επιτυχίας. Αντίστοιχα, στην περίπτωση του WEP, απαιτούνται τέσσερα μηνύματα. Κατ΄ αρχήν, η ασύρματη συσκευή ζητά επαλήθευση ταυτότητας και το σημείο πρόσβασης απαντά με μήνυμα πρόκλησης (challenge). Η ασύρματη συσκευή αποκρίνεται στην πρόκληση προκειμένου να αποδείξει ότι γνωρίζει το μυστικό κλειδί και, αν η απόδειξη γίνει αποδεκτή, το σημείο πρόσβασης στέλνει μήνυμα επιτυχίας.

Επομένως, όταν το σημείο πρόσβασης λειτουργεί με ανοιχτό τρόπο, αποδέχεται το σύνολο των αιτήσεων επαλήθευσης ταυτότητας και αποκρίνεται με μήνυμα επιτυχίας. Εν τούτοις, πολλά συστήματα χρησιμοποιούν ιδιοκτησιακές μεθόδους επαλήθευσης ταυτότητας, η πιο δημοφιλής των οποίων, είναι η τήρηση λίστας διευθύνσεων MAC (Media Access Control). Πιο συγκεκριμένα, το σημείο πρόσβασης διαθέτει μια λίστα των διευθύνσεων MAC των οποίων τη σύνδεση θα επιτρέψει στο δίκτυο. Η λίστα διαμορφώνεται από το διαχειριστή, ενώ η επαλήθευση ταυτότητας επιτυγχάνει, μόνο εφόσον η διεύθυνση MAC της κινητής συσκευής περιλαμβάνεται στη λίστα αυτή. Η μέθοδος αυτή δεν παρέχει προστασία από ψευδεπίγραφες διευθύνσεις MAC, ωστόσο αποτελεί μια βασική μορφή ασφάλειας από απλές επιθέσεις.

Σε αντίθεση με τον ανοιχτό τρόπο λειτουργίας, ο σκοπός της επαλήθευσης ταυτότητας WEP είναι να αποδείξει η ασύρματη συσκευή στο σημείο πρόσβασης ότι γνωρίζει το μυστικό



Σχήμα 3.1: Επαλήθευση Ταυτότητας στο αρχικό πρότυπο IEEE 802.11

κλειδί κρυπτογράφησης. Όταν υποβάλλεται η αίτηση επαλήθευσης ταυτότητας από την ασύρματη συσκευή, το σημείο πρόσβασης στέλνει έναν αριθμό που ονομάζεται κείμενο πρόκλησης. Πρόκειται για ένα τυχαίο αριθμό μήκους 128 bit. Η κινητή συσκευή κρυπτογραφεί αυτόν τον αριθμό χρησιμοποιώντας το μυστικό κλειδί με τη βοήθεια του WEP και επιστρέφει το αποτέλεσμα στο σημείο πρόσβασης. Αυτό είναι σε θέση να ελέγξει αν η κρυπτογράφηση έγινε με το σωστό κλειδί, ωστόσο η όλη διαδικασία δεν εξασφαλίζει στην ασύρματη συσκευή ότι το σημείο πρόσβασης όντως γνωρίζει το μυστικό κλειδί. Ένα ακόμη μεγαλύτερο πρόβλημα της επαλήθευσης ταυτότητας WEP είναι ότι προσφέρει σε ένα κακόβουλο χρήστη που παρακολουθεί την επικοινωνία, πληροφορία, τόσο στην κρυπτογραφημένη, όσο και στην αρχική της μορφή· η πρόκληση περιέχει το αρχικό κείμενο, ενώ η απόκριση το κρυπτογραφημένο.

Το μόνο πλεονέκτημα από τη διαδικασία επαλήθευσης ταυτότητας WEP είναι ότι δεν επιτρέπει σε σταθμούς που δε γνωρίζουν το μυστικό κλειδί να στέλνουν πλαίσια, καθώς αυτοί δε συσχετίζονται με το σημείο πρόσβασης. Αντίθετα, στον ανοιχτό τρόπο λειτουργίας, η ασύρματη συσκευή γίνεται αποδεκτή από το σημείο πρόσβασης, ωστόσο κάθε πλαίσιο που μεταδίδει απορρίπτεται εφόσον δεν κρυπτογραφείται με το σωστό κλειδί. Αυτό, βέβαια, αποτελεί πλεονέκτημα περισσότερο διαχειριστικού τύπου, παρά ασφάλειας.

3.2 Απόρρητο

Στην ενότητα αυτή, παρουσιάζεται το απόρρητο στα πλαίσια του WEP, το οποίο έχει ιδιαίτερη σημασία όσον αφορά την ασφάλεια στα Ασύρματα Τοπικά Δίκτυα.

3.2.1 Κρυπτογράφηση

Όταν ενεργοποιείται το WEP, τα μηνύματα δεδομένων κρυπτογραφούνται προκειμένου να διαφυλάσσεται το περιεχόμενό τους από ένα κακόβουλο χρήστη που παρακολουθεί την ασύρ-

ματη επικοινωνία. Για την αποκρυπτογράφηση του μηνύματος, απαιτείται γνώση του μυστικού κλειδιού. Το αρχικό πρότυπο IEEE 802.11 προδιαγράφει μια προσέγγιση δύο φάσεων: α) επαλήθευση ταυτότητας και β) κρυπτογράφηση. Ωστόσο, όπως διαπιστώθηκε στην ενότητα 3.1, η διαδικασία της επαλήθευσης ταυτότητας καθίσται επικίνδυνη, καθώς παρέχει χρήσιμες πληροφορίες για επιθέση από ένα επίδοξο εισβολέα. Ως εκ τούτου, τα περισσότερα συστήματα Ασύρματων Τοπικών Δικτύων, κάνουν χρήση ανοιχτής επαλήθευσης ταυτότητας και προχωρούν στην κρυπτογράφηση μετά το συσχετισμό της ασύρματης συσκευής με το σημείο πρόσβασης. Η παράλειψη της επαλήθευσης ταυτότητας δεν παρέχει κάποιο πλεονέκτημα σε ένα επίδοξο εισβολέα, καθώς, παρόλο που μπορεί να συνδεθεί στο δίκτυο ελεύθερα, δε μπορεί να στείλει ή να λάβει δεδομένα, εφόσον δε γνωρίζει τα κλειδιά WEP για κρυπτογράφηση. Μέχρι το τέλος της ενότητας αυτής, υποθέτουμε ότι η ασύρματη συσκευή έχει συσχετιστεί με κάποιον τρόπο (ανοιχτό ή WEP).

Το WEP χρησιμοποιεί τον αλγόριθμο **RC4** για την κρυπτογράφηση των δεδομένων [2]. Πρόκειται για ένα *κρυπτογραφικό αλγόριθμό ροής* (stream cipher), ο οποίος λαμβάνει ως είσοδο ένα byte κάθε φορά της ροής των αρχικών δεδομένων και παράγει το αντίστοιχο byte σε κρυπτογραφημένη μορφή. Η αποκρυπτογράφηση χρησιμοποιεί τα ίδια κλειδιά με την κρυπτογράφηση και, ως εκ τούτου, ο RC4 ανήκει στην κατηγορία των *συμμετρικών αλγορίθμων*.

Ένα από τα πλεονεκτήματα του RC4 είναι η ευκολία υλοποίησης, όπου αποφεύγονται περίπλοκες και χρονοβόρες λειτουργίες, όπως είναι ο πολλαπλασιασμός. Ο RC4 περιλαμβάνει δύο φάσεις. Πρώτη είναι η φάση της αρχικοποίησης, κατά την οποία κατασκευάζονται ορισμένοι πίνακες εσωτερικών δεδομένων, βάσει της τιμής κλειδιού που έχει οριστεί. Στη δεύτερη φάση, γίνεται η ίδια η κρυπτογράφηση των δεδομένων.

Στην περίπτωση του WEP, τόσο η φάση αρχικοποίησης, όσο και της κρυπτογράφησης, λαμβάνουν χώρα για κάθε πακέτο, με άλλα λόγια, κάθε πακέτο ξεχωριστά θεωρείται ως μια νέα ροή δεδομένων. Με αυτόν τον τρόπο εξασφαλίζεται ότι, στην περίπτωση που ένα πακέτο χαθεί, η αποκρυπτογράφηση του επόμενου είναι πάλι δυνατή. Ωστόσο, όπως θα δούμε στην ενότητα 3.4, το συγκεκριμένο χαρακτηριστικό αποτελεί αδυναμία, όσον αφορά την ασφάλεια. Περισσότερες πληροφορίες για τον αλγόριθμο κρυπτογράφησης RC4 και τον τρόπο υλοποίησής του στο WEP, υπάρχουν στην ενότητα 3.3.4.

3.2.2 Κλειδιά WEP

Τα κλειδιά κρυπτογράφησης στο WEP έχουν τα ακόλουθα χαρακτηριστικά:

- **Σταθερό μήκος**: Συνήθως 40 ή 104 bit.
- **Στατικά**: Δε μεταβάλλεται η τιμή του κλειδιού, εφόσον δεν αλλάξουν οι ρυθμίσεις.
- **Μεριζόμενα (shared)**: Τόσο το σημείο πρόσβασης όσο και η κινητή συσκευή διαθέτουν αντίγραφο των ίδιων κλειδιών.
- **Συμμετρικά**: Χρήση του ίδιου κλειδιού για κρυπτογράφηση και αποκρυπτογράφηση των πληροφοριών.

Σύμφωνα με το πρότυπο IEEE 802.11, η διάθεση των κλειδιών στα σημεία πρόσβασης και τις ασύρματες συσκευές πρέπει να γίνεται με ασφαλείς μεθόδους ανεξάρτητες του πρωτοκόλλου.

3.2.3 Διάνυσμα Αρχικοποίησης

Τα κλειδιά κρυπτογράφησης στο WEP έχουν μήκη των 40 ή 104 bit, ωστόσο, συχνά οι κατασκευαστές αναφέρονται σε αυτά ως 64 ή 128 bit αντίστοιχα. Ο λόγος είναι ότι τα επιπλέον 24 bit χρησιμοποιούνται από το διάνυσμα αρχικοποίησης (Initialization Vector – IV).

Η λειτουργία του IV στο WEP έχει ως εξής: συνδυάζεται το μυστικό κλειδί με έναν αριθμό μήκους 24 bit που αλλάζει για κάθε πακέτο, και με το αποτέλεσμα κρυπτογραφούνται τα δεδομένα. Ο αριθμός αυτός αποτελεί το IV και έχει ως στόχο την αποφυγή χρήσης ενός στατικού κλειδιού για την κρυπτογράφηση του συνόλου των πακέτων. Ως εκ τούτου, ακόμη και στην περίπτωση που τα αρχικά δεδομένα είναι ίδια, η κρυπτογραφημένη μορφή τους είναι πάντα διαφορετική.

Πρέπει να τονιστεί ότι το IV δεν είναι μυστικό. Μάλιστα, η τιμή του στέλνεται σε μη κρυπτογραφημένη μορφή σε κάθε μετάδοση ώστε ο παραλήπτης να είναι σε θέση να αποκρυπτογραφήσει την πληροφορία χρησιμοποιώντας την αντίστοιχη τιμή του IV.

3.3 Η υλοποίηση του WEP

Στην ενότητα αυτή, θα παρουσιάσουμε την υλοποίηση των μηχανισμών του WEP, ώστε να γίνουν καλύτερα κατανοητές οι αδυναμίες του προτύπου.

3.3.1 Θραυσματισμός

Στην περίπτωση που ένα δίκτυο περιλαμβάνει μια ασύρματη ζεύξη τοπικού δικτύου, τα δεδομένα από το λειτουργικό σύστημα πρέπει να περάσουν στο στρώμα υπηρεσιών MAC IEEE 802.11. Με άλλα λόγια, ένα πακέτο δεδομένων καταφθάνει στο WLAN με κατάλληλες οδηγίες ώστε να γίνει η αποστολή του. Αυτό το πακέτο δεδομένων καλείται MSDU (MAC Service Data Unit). Αν δεν υπάρξει λάθος, η MSDU θα εμφανιστεί στο στρώμα υπηρεσιών MAC της συσκευής προορισμού προκειμένου να περάσει στο λειτουργικό σύστημα για παράδοση στην κατάλληλη εφαρμογή. Ωστόσο, πριν από τη μετάδοση, αυτή η MSDU μπορεί να χωριστεί σε αρκετά μικρότερα τμήματα, μια διαδικασία γνωστή ως **θραυσματισμός** (fragmentation). Κάθε θραύσμα υφίσταται επεξεργασία για κρυπτογράφηση WEP. Μια επικεφαλίδα MAC προστίθεται μπροστά και μια τιμή ελέγχου στο τέλος.

Ως εκ τούτου, παρατηρούμε ότι αυτή η αρχική MSDU μπορεί τώρα να χωριστεί σε αρκετά μικρότερα μηνύματα και να προστεθούν περισσότερα byte, εκτός από το γεγονός ότι είναι πλέον κρυπτογραφημένη. Κάθε ένα από τα μικρότερα μηνύματα καλείται MPDU (MAC Protocol Data Unit). Στη συνέχεια εξετάζονται τα τελευταία στάδια, κατά τα οποία μια MPDU υφίσταται τη διαδικασία κρυπτογράφησης.

Η διαδικασία αντιμετωπίζει τα δεδομένα ως μια ακολουθία από byte, το μέγεθος της οποίας, εξαρτάται από τα αρχικά περιεχόμενα της MSDU και τις ρυθμίσεις θραυσματισμού. Συνήθως, κυμαίνεται από 10 έως 1500 byte. Το πρώτο βήμα στην κρυπτογράφηση είναι η προσθήκη μερικών byte που αποτελούν την τιμή ελέγχου ακεραιότητας.

3.3.2 Τιμή Ελέγχου Ακεραιότητας

Ο σκοπός της Τιμής Ελέγχου Ακεραιότητας (Integrity Check Value – ICV) είναι η προστασία από την τροποποίηση του μηνύματος κατά τη μεταφορά. Τόσο στα κρυπτογραφημένα, όσο και στα μη κρυπτογραφημένα μηνύματα, γίνεται έλεγχος προκειμένου να εντοπιστεί αν η

τιμή οποιουδήποτε από τα bit έχει μεταβληθεί κατά τη μετάδοση. Όλα τα byte του μηνύματος συνδυάζονται στον έλεγχο κυκλικού πλεονασμού (Cyclic Redundancy Check – CRC). Αυτή η τιμή μήκους τεσσάρων byte προστίθεται στο τέλος του πλαισίου ακριβώς πριν από την επεξεργασία για μετάδοση. Ακόμη κι αν ένα μόνο bit του μηνύματος μεταβληθεί, η λαμβάνουσα συσκευή θα υπολογίσει τιμή CRC διαφορετική από αυτή που μετέφερε το πλαίσιο και θα απορρίψει το μήνυμα. Παρόλο που ο έλεγχος αυτός εντοπίζει τυχαία λάθη, εν τούτοις, δεν παρέχει προστασία από σκόπιμα λάθη, καθώς ένα κακόβουλος χρήστης μπορεί απλά να υπολογίσει τη νέα τιμή CRC που προκύπτει από την τροποποίηση του μηνύματος και να αντικαταστήσει την αρχική.

Το ICV λειτουργεί όμοια με το CRC, με τη διαφορά ότι υπολογίζεται και προστίθεται *πριν* την κρυπτογράφηση. Το συμβατικό CRC εξακολουθεί να προστίθεται μετά την κρυπτογράφηση. Θεωρητικά, επειδή το ICV κρυπτογραφείται, ένας επίδοξος εισβολέας δεν είναι σε θέση να το υπολογίσει ξανά όταν τροποποιεί το μήνυμα. Ωστόσο, στην ενότητα 3.4, θα δούμε ότι κάτι τέτοιο δεν ισχύει στην πράξη.

Συνεπώς, το ICV υπολογίζεται συνδυάζοντας όλα τα δεδομένα και προκύπτει ως μία τιμή μήκους τεσσάρων byte, η οποία προστίθεται στο τέλος, όπως φαίνεται στο Σχ. 3.2.

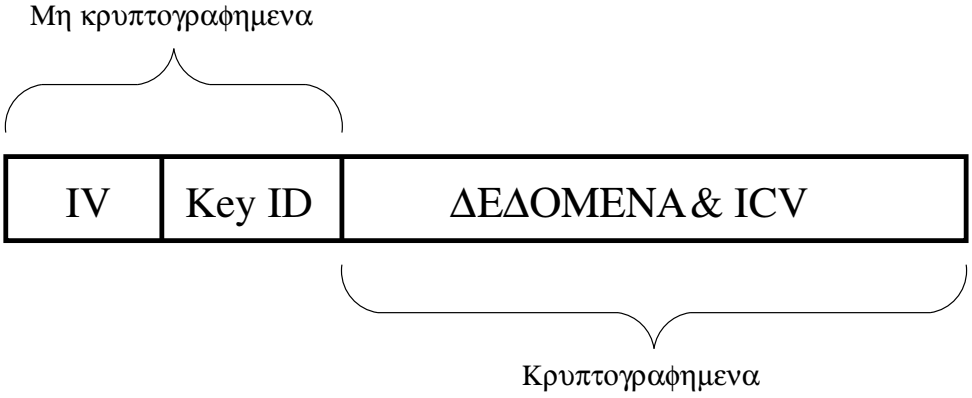
Σχήμα 3.2: Προσθήκη του ICV

3.3.3 Μετάδοση πλαισίων

Μετά την προσθήκη του ICV, το πλαίσιο είναι έτοιμο για κρυπτογράφηση. Αρχικά, το σύστημα πρέπει να επιλέξει μια τιμή IV και να την προσθέσει στο μυστικό κλειδί WEP. Στη συνέχεια, αρχικοποιεί τον αλγόριθμο κρυπτογράφησης RC4. Τελικά, διοχετεύει κάθε byte από τα συδυασμένα δεδομένα με το τμήμα ICV στο μηχανισμό κρυπτογράφησης. Για κάθε byte εισόδου, προκύπτει ένα κρυπτογραφημένο byte, μέχρι να ολοκληρωθεί η επεξεργασία όλων των byte. Αυτό αποτελεί τη ροή κρυπτογραφήματος.

Προκειμένου ο παραλήπτης να είναι σε θέση να αποκρυπτογραφήσει το μήνυμα, ο αριθμός κλειδιού και η τιμή IV πρέπει να τοποθετηθούν μπροστά από το μήνυμα. Τέσσερα byte προστίθενται για το σκοπό αυτό. Τα πρώτα τρία byte περιέχουν την τιμή IV μήκους 24 bit και το τελευταίο byte το Key ID (αναγνωριστικό κλειδιού) που είναι ένας από τους αριθμούς 0, 1, 2 ή 3. Η μορφή του πλαισίου φαίνεται στο Σχ. 3.3.

Τέλος, προστίθεται η επικεφαλίδα της διεύθυνσης MAC και τοποθετείται η τιμή CRC στο τέλος για τον εντοπισμό λαθών στη μετάδοση. Ένα bit στην επικεφαλίδα MAC υποδεικνύει



Σχήμα 3.3: Προσθήκη των bit IV και Key ID

στον παραλήπτη ότι το πλαίσιο έχει κρυπτογραφηθεί με WEP, ώστε να είναι σε θέση να το χειριστεί κατάλληλα.

Η διαδικασία λήψης έχει ως εξής: ο παραλήπτης παρατηρεί ότι το bit ένδειξης WEP έχει τεθεί και επομένως διαβάζει και αποθηκεύει την τιμή IV. Στη συνέχεια διαβάζει το Key ID ώστε να επιλέξει το σωστό κλειδί WEP, προσθέτει σε αυτό την τιμή του IV, και αρχικοποιεί τον αλγόριθμο κρυπτογράφησης RC4. Ας σημειωθεί ότι στον RC4 δεν υπάρχει καμία διαφορά μεταξύ της διαδικασίας κρυπτογράφησης και αποκρυπτογράφησης. Αν περάσουν τα δεδομένα δύο φορές μέσα από τη διαδικασία κρυπτογράφησης θα προκύψουν πάλι τα αρχικά δεδομένα. Ως εκ τούτου, απαιτείται ένας μόνο μηχανισμός τόσο για την κρυπτογράφηση όσο και για την αποκρυπτογράφηση. Μετά την αρχικοποίηση του μηχανισμού κρυπτογράφησης, γίνεται επεξεργασία ενός byte κάθε φορά και προκύπτει το αρχικό μήνυμα. Το τελικό βήμα, είναι ο υπολογισμός του ICV ώστε να συγκριθεί με την τιμή που μετέφερε το μήνυμα. Αν δεν εντοπιστεί λάθος, τότε το τμήμα δεδομένων προχωράει για περαιτέρω επεξεργασία.

3.3.4 Αλγόριθμος κρυπτογράφησης RC4

Ο αλγόριθμος κρυπτογράφησης που χρησιμοποιείται από το WEP είναι ο RC4[1]. Ανήκει στην οικογένεια των αλγορίθμων κρυπτογράφησης που η διαδικασία της αποκρυπτογράφησης δε διαφέρει από εκείνη της κρυπτογράφησης. Η αξία ενός αλγορίθμου κρυπτογράφησης έγκειται, κατά ένα μέρος, στην ισχύ του και, κατά ένα άλλο, στην ευκολία υλοποίησής του. Η ισχύς ενός αλγορίθμου εξαρτάται από τη δυσκολία αποκωδικοποίησης του κρυπτογραφήματος. Ενώ σίγουρα υπάρχουν ισχυρότερες μέθοδοι, εν τούτοις, ο RC4 είναι αξιοσημείωτα απλός στην υλοποίηση του και θεωρείται πολύ ισχυρός, *με την προϋπόθεση ότι χρησιμοποιείται σωστά*. Αυτό είναι ιδιαίτερα σημαντικό, καθώς, όπως φαίνεται στην ενότητα 3.4, οι αδυναμίες του WEP δεν οφείλονται σε σφάλματα του RC4, αλλά στον τρόπο χρήσης του στο WEP.

Η βασική ιδέα πίσω από τον RC4 είναι η παραγωγή μια ψευδοτυχαίας ακολουθίας από byte, που καλείται **ροή κλειδιού** (key stream), η οποία συνδυάζεται με τα δεδομένα με λογική πράξη αποκλειστικού Η (XOR). Μια σημαντική ιδιότητα της XOR είναι η εξής:

$$\text{Αν } A \oplus B = C, \text{ τότε } C \oplus B = A$$

[1] Τα αρχικά RC4 προκύπτουν από το όνομα του σχεδιαστή του αλγορίθμου Ron Rivest. Μάλιστα πρόκειται για τον τέταρτο αλγόριθμο κρυπτογράφησης που ανέπτυξε ο ίδιος.

Την παραπάνω ιδιότητα εκμεταλλεύεται ο RC4 ως εξής:

Κρυπτογράφηση: Μη κρυπτογραφημένο κείμενο ⊕ Τυχαία ακολουθία = Κρυπτογράφημα
Αποκρυπτογράφηση: Κρυπτογράφημα ⊕ Τυχαία ακολουθία = Μη κρυπτογραφημένο κείμενο

Είναι ιδιαίτερα σημαντικό, η «τυχαία ακολουθία» να φαίνεται τυχαία σε ένα επίδοξο εισβολέα, αλλά ταυτόχρονα πρέπει και τα δύο άκρα της ζεύξης να είναι σε θέση να παράγουν την ίδια «τυχαία» τιμή για κάθε byte που επεξεργάζονται. Ως εκ τούτου θεωρείται **ψευδοτυχαία** και παράγεται από τον αλγόριθμο RC4.

Η πιο σημαντική ιδιότητα μιας ψευδοτυχαίας ροής κλειδιού είναι η δυνατότητα υπολογισμού του επόμενου byte, *μόνο* εφόσον είναι γνωστό το κλειδί που χρησιμοποιήθηκε για την παραγωγή της ροής. Ας σημειωθεί ότι η πράξη XOR αποκρύπτει πλήρως τις τιμές του μη κρυπτογραφημένου κειμένου. Ακόμη κι αν το μη κρυπτογραφημένο κείμενο είναι απλά μια μεγάλη ακολουθία από μηδενικές τιμές, το κρυπτογράφημα φαίνεται τυχαίο.

Η πράξη XOR υλοποιείται πολύ εύκολα από ένα υπολογιστικό σύστημα, επομένως, πρόκληση αποτελεί ο ίδιος ο υπολογισμός μιας καλής ψευδοτυχαίας ροής αριθμών. Απαιτείται ένα ψευδοτυχαίο byte για κάθε byte του μηνύματος προς κρυπτογράφηση. Ο RC4 παράγει μια ροή αυτής της μορφής.

Ο RC4 αποτελείται από δύο φάσεις: τη ρύθμιση κλειδιού και την ψευδοτυχαία παραγωγή. Η πρώτη φάση δημιουργεί ένα πίνακα μεγέθους 256 byte με μετάθεση των αριθμών 0 έως 255, ώστε όλοι να εμφανίζονται μια φορά με τυχαία διάταξη. Η μετάθεση στον πίνακα ή κουτί-S (S-box), όπως συνήθως λέγεται, επιτυγχάνεται αρχικοποιώντας τον πίνακα με τους αριθμούς 0 – 255 στη σειρά. Τα στοιχεία του κουτιού-S αναδιατάσσονται στη συνέχεια ως εξής: Αρχικά, ένας δεύτερος πίνακας μεγέθους 256 byte ή κουτί-K, γεμίζει με το κλειδί, το οποίο επαναλαμβάνεται μέχρι την πλήρωση ολόκληρου του πίνακα. Στη συνέχεια κάθε byte στο κουτί-S ανταλλάσσεται (swap) με κάποιο άλλο byte από το κουτί-S. Αρχίζοντας από το πρώτο byte, λαμβάνει χώρα ο ακόλουθος υπολογισμός:

$$j = (\text{τιμή πρώτου byte στο κουτί-S}) + (\text{τιμή πρώτου byte στο κουτί-K})$$

όπου το j είναι τιμή μήκους ενός byte, ενώ οποιοδήποτε σφάλμα υπερχείλισης προκύψει κατά την πρόσθεση, αγνοείται.

Πλέον το j χρησιμοποιείται ως δείκτης στο κουτί-S και η τιμή στη θέση αυτή ανταλλάσσεται με την τιμή στην αρχική θέση.

Αυτή η διαδικασία επαναλαμβάνεται άλλες 255 φορές μέχρι να ανταλλαχθεί η θέση κάθε byte στο κουτί-S. Τα παραπάνω έχουν σε ψευδοκώδικα ως εξής:

$$\begin{aligned}
&i = j = 0 \\
&\text{for } i = 0 \text{ to } 255 \text{ do} \\
&\quad j = (j + S_i + K_i) \bmod 256 \\
&\text{swap}(S_i,\, S_j)
\end{aligned}$$

Αμέσως μετά την αρχικοποίηση του κουτιού-S, η επόμενη φάση του RC4 είναι η παραγωγή των ψευδοτυχαίων αριθμών. Η φάση αυτή περιλαμβάνει περισσότερες ανταλλαγές μεταξύ byte του κουτιού-S και παραγωγή ενός ψευδοτυχαίου byte ανά επανάληψη (R). Οι λειτουργίες που λαμβάνουν χώρα σε κάθε επανάληψη έχουν ως εξής:

$$i = (i + 1) \bmod 256$$
$$j = (j + S_i) \bmod 256$$
$$\text{swap}(S_i, S_j)$$
$$k = (S_i + S_j) \bmod 256$$
$$R = S_k$$

Για τον υπολογισμό του κρυπτογραφήματος, κάθε byte του μηνύματος συνδυάζεται με XOR με μια τιμή του R καθώς παράγεται από τον αλγόριθμο RC4. Ας σημειωθεί ότι η όλη διαδικασία πραγματοποιείται με πράξεις (προσθέσεις, ανταλλαγές) μήκους ενός byte, οι οποίες είναι πολύ βολικές για την υπολογιστική λογική.

Θεωρητικά, ο RC4 δε θεωρείται απόλυτα ασφαλής αλγόριθμος, καθώς παράγει μια ψευδοτυχαία ροή κλειδιού, και όχι πραγματικά τυχαία byte. Εν τούτοις, είναι επαρκώς ασφαλής για εφαρμογή στα ασύρματα τοπικά δίκτυα, εφόσον χρησιμοποιείται με σωστό τρόπο.

3.4 Οι αδυναμίες του WEP

Η ενότητα αυτή, περιγράφει λεπτομερώς γνωστές επιθέσεις που αφορούν το WEP. Με τον όρο WEP εννοούμε ένα σύνολο από μηχανισμούς ασφαλείας, οι οποίοι θα εξεταστούν ξεχωριστά:

- Επαλήθευση Ταυτότητας
- Έλεγχος Πρόσβασης
- Αποφυγή Αναπαραγωγής
- Εντοπισμός Τροποποίησης Μηνυμάτων
- Απόρρητο Μηνυμάτων
- Προστασία Κλειδιού

Δυστυχώς, το WEP αποτυγχάνει σε όλους αυτούς τους τομείς.

3.4.1 Επαλήθευση ταυτότητας

Η επαλήθευση ταυτότητας έχει την έννοια της απόδειξης της αυθεντικότητας της ταυτότητας μιας πλευράς προς μια άλλη. Πρέπει να είναι διαρκής διαδικασία, με άλλα λόγια δεν αρκεί να αποδείξει κανείς ότι είναι αυθεντικός μια μόνο φορά. Για να εξασφαλιστεί η ασφάλεια, πρέπει να γίνεται σε κάθε επικοινωνία. Επειδή είναι χρονοβόρα διαδικασία, η συνήθης προσέγγιση είναι να πραγματοποιείται πλήρης επαλήθευση ταυτότητας κατά την πρώτη επαφή, και στη συνέχεια να παρέχεται ένα διακριτικό ταυτότητας περιορισμένης χρονικής ζωής. Ιδανικά, το διακριτικό αυτό είναι τέτοιο ώστε να μην είναι δυνατή η μεταβίβασή του αλλού.

Στον κόσμο των ασυρμάτων επικοινωνιών, συνήθως χρειάζεται *αμοιβαία* επαλήθευση ταυτότητας. Το δίκτυο απαιτεί απόδειξη για το χρήστη, αλλά και ο χρήστης απαιτεί απόδειξη ότι το δίκτυο είναι νόμιμο. Αυτό είναι σημαντικό στα ασύρματα δίκτυα όπου είναι σχετικά εύκολο να εγκατασταθούν παραπλανητικά σημεία πρόσβασης.

Τέλος, πρέπει να τονιστεί η σπουδαιότητα της χρήσης διαφορετικών κλειδιών για την επαλήθευση ταυτότητας από αυτών που χρησιμοποιούνται για την κρυπτογράφηση. Έτσι,

συνίσταται η χρήση κλειδιών που προκύπτουν το ένα από το άλλο, γιατί τα κύρια κλειδιά δεν πρέπει να εκτίθενται ποτέ άμεσα σε επιθέσεις. Συνοψίζοντας, οι βασικές απαιτήσεις για την επαλήθευση ταυτότητας στα ασύρματα τοπικά δίκτυα είναι οι εξής:

1. Εύρωστη μέθοδος απόδειξης της ταυτότητας που δε μπορεί να παραποιηθεί
2. Μέθοδος διατήρησης της ταυτότητας κατά τη διάρκεια διαδοχικών συναλλαγών που δε μπορούν να μεταφερθούν
3. Αμοιβαία επαλήθευση ταυτότητας
4. Ανεξάρτητα κλειδιά ανεξάρτητα από τα κλειδία κρυπτογράφησης

Δυστυχώς το WEP είναι ανεπαρκές σε όλα τα παραπάνω. Όπως έχει αναφερθεί, η επαλήθευση ταυτότητας στο WEP στηρίζεται σε ένα μηχανισμός πρόκλησης – απόκρισης. Αρχικά το σημείο πρόσβασης στέλνει μια τυχαία ακολουθία αριθμών. Στη συνέχεια η κινητή συσκευή κρυπτογραφεί την ακολουθία και τη στέλνει πίσω. Ακολούθως, το σημείο πρόσβασης την αποκρυπτογραφεί και συγκρίνει με την αρχική ακολουθία. Ανάλογα με το αποτέλεσμα αποφασίζει αν θα αποδεχθεί τη συσκευή ώστε να στείλει κατάλληλο μήνυμα.

Το κλειδί που χρησιμοποιείται σε αυτή τη διαδικασία είναι το ίδιο με αυτό της κρυπτογράφησης WEP, παραβαίνοντας, έτσι, την απαίτηση 4. Ο μηχανισμός δεν επαληθεύει την ταυτότητα του σημείου πρόσβασης στην κινητή συσκευή, επιτρέποντας σε ένα μη νόμιμο σημείο πρόσβασης, που δε χρειάζεται να γνωρίζει το κλειδί, να στείλει παραπλανητικά μήνυμα αποδοχής χωρίς να έχει ελέγξει καν την ακολουθία του πελάτη. Έχουμε, δηλαδή, παράβαση του κανόνα 3.

Η απαίτηση 2 δεν τηρείται, καθώς δεν παρέχεται κάποιο τεκμήριο που να επικυρώνει διαδοχικές συναλλαγές, γεγονός που καθιστά την όλη διαδικασία επαλήθευσης ταυτότητας μάταιη.

Όσον αφορά τον κανόνα 1, αρκεί να επανεξετάσουμε τη διαδικασία της επαλήθευσης ταυτότητας, όπου το σημείο πρόσβασης στέλνει μια τυχαία ακολουθία μήκους 128 byte, την οποία λαμβάνει η κινητή συσκευή, την κρυπτογραφεί και τη στέλνει πίσω. Η κρυπτογράφηση WEP περιλαμβάνει την παραγωγή μιας ακολουθίας ψευδοτυχαίων byte, που καλείται ροή κλειδιού, και την πράξη XOR αυτής με το κείμενο προς κρυπτογράφηση. Συνεπώς, οποιοσδήποτε παρακολουθεί την συναλλαγή μπορεί να αποκτήσει την πρόκληση του μη κρυπτογραφημεου κειμένου και την κρυπτογραφημένη απόκριση. Ως εκ τούτου, με απλή πράξη XOR αυτών των δύο, ο εχθρός υπολογίζει τα τυχαία byte RC4. Ισχύει, άλλωστε, η βασική εξίσωση κρυπτογράφησης:

$$P \oplus R = C$$

όπου P είναι το μη κρυπτογραφημένο κείμενο (Plaintext), R είναι το τυχαίο κλειδί (Randombytes) και C είναι το κρυπτογραφημένο κείμενο (Ciphertext). Αντίστοιχα, για την αποκρυπτογράφηση έχουμε:

$$C \oplus R = P$$

Ομοίως, η πράξη XOR μεταξύ του κρυπτογραφημένου και του μη κρυπτογραφημένου κειμένου παράγει το τυχαίο κλειδί:

$$C \oplus P = R$$

Επομένως, ο εισβολέας γνωρίζει τώρα τη ροή κλειδιού που αντιστοιχεί σε δεδομένη τιμή του IV. Μπορεί, λοιπόν, να ζητήσει επαλήθευση ταυτότητας, να περιμένει το κείμενο της πρόκλησης, να κάνει πράξη XOR με τη ροή κλειδιού που έχει υπολογίσει προηγουμένως και να περιμένει το αποτέλεσμα με το IV που έχει ήδη καταγράψει.

Προκειμένου να ελέγξει το αποτέλεσμα, το σημείο πρόσβασης προσθέτει το IV (που έχει επιλέξει ο εισβολέας) στο μυστικό κλειδί και παράγει την τυχαία ροή κλειδιού RC4. Προφανώς, αυτή θα είναι ίδια με προηγουμένως, καθώς το κλειδί και το IV είναι ίδια. Έτσι, όταν το σημείο πρόσβασης αποκρυπτογραφήσει το μήνυμα κάνοντας πράξη XOR με τη ροή κλειδιού RC4, θα είναι ίδιο με την αρχική ακολουθία. Συνεπώς, ο εισβολέας έχει «επαληθεύσει την ταυτότητα του» χωρίς να γνωρίζει καν το μυστικό κλειδί.

Αν και ο εισβολέας μπορεί να περάσει επιτυχώς τη διαδικασία επαλήθευσης ταυτότητας με τον τρόπο αυτό, δεν είναι σε θέση, ωστόσο, να επικοινωνήσει, καθώς τα πλαίσια κρυπτογραφούνται με το WEP. Πρέπει επομένως, να σπάσει επιπλέον και την ίδια την κρυπτογράφηση WEP. Εν τούτοις, για κάποιες από τις μεθόδους επίθεσης στα κλειδιά κρυπτογράφησης, ο εχθρός χρειάζεται ένα δείγμα κειμένου και του αντίστοιχου κρυπτογραφήματος. Η πληροφορία αυτή είναι γενικά δύσκολο να βρεθεί, ωστόσο, η διαδικασία επαλήθευσης ταυτότητας στο WEP παρέχει, όπως αναφέραμε, ένα δείγμα μήκους 128 byte. Ακόμη χειρότερο είναι ότι πρόκειται για τα πρώτα 128 byte της ροής κλειδιού, που είναι τα πλέον ευάλωτα σε ενδεχόμενη επίθεση. Στα πλαίσια αυτά, η προσέγγιση του WEP όχι μόνο δεν προσφέρει ασφαλή επαλήθευση ταυτότητας, αλλά βοηθά ένα επίδοξο εισβολέα να επιτεθεί στα κλειδιά κρυπτογράφησης. Για το λόγο αυτό, τα περισσότερα συστήματα που εφαρμόζουν το WEP σήμερα, έχουν καταργήσει τη φάση επαλήθευσης ταυτότητας που παρέχει.

3.4.2 Έλεγχος πρόσβασης

Ο έλεγχος πρόσβασης αποτελεί τη διαδικασία επίτρεψης ή απαγόρευσης της επικοινωνίας μιας κινητής συσκευής με το δίκτυο. Συχνά, ο όρος συγχέεται λανθασμένα με την επαλήθευση ταυτότητας. Ωστόσο, η απόδειξη της αυθεντικότητας της ταυτότητας δεν συνεπάγεται την επίτρεψη πρόσβασης.

Γενικά, η πρόσβαση ελέγχεται διατηρώντας μια λίστα με τις επιτρεπόμενες συσκευές. Μπορεί επίσης να επιτευχθεί παρέχοντας πρόσβαση σε οποιονδήποτε κατέχει κάποιο πιστοποιητικό ή άλλο ηλεκτρονικό πάσο.

Το πρότυπο IEEE 802.11 δεν ορίζει μηχανισμούς υλοποίησης του ελέγχου πρόσβασης. Ωστόσο, οι συσκευές αναγνωρίζονται από τη διεύθυνση MAC, γεγονός που υπονοεί ότι θα μπορούσε να διατηρείται μια λίστα από αποδεκτές διευθύνσεις MAC. Την προσέγγιση αυτή υιοθετούν πολλά σημεία πρόσβασης, ακόμη κι όταν λειτουργούν χωρίς κρυπτογράφηση WEP. Εν τούτοις, δεδομένης της ευκολίας με την οποία οι διευθυνσεις MAC μπορούν ψευδεπίγραφα να αποδοθούν, αυτό δε μπορεί να θεωρηθεί σοβαρός μηχανισμός ασφαλείας.

Ως εκ τούτου, ο μοναδικός μηχανισμός ελέγχου πρόσβασης που απομένει στο WEP είναι το κλειδί κρυπτογράφησης. Αν ο κινητός σταθμός δε γνωρίζει το σωστό κλειδί WEP, τότε τα πλαίσια που στέλνει θα προκαλούν λάθος ICV όταν αποκρυπτογραφούνται. Κατά συνέπεια, τα πλαίσια θα απορρίπτονται και δε θα επιτρέπεται η πρόσβαση στη συσκευή.

3.4.3 Αποφυγή αναπαραγωγής

Ας υποθέσουμε ότι ένας κακόβουλος χρήστης καταγράφει όλα τα πλαίσια που στέλνονται μεταξύ ενός σημείου πρόσβασης και μιας κινητής συσκευής με τη βοήθεια ενός αναλυτή πρωτοκόλλων (sniffer). Παρατηρεί ότι ένας νέος χρήστης συνδέεται στο δίκτυο και, χωρίς να

είναι σε θέση να δει το πραγματικό περιεχόμενο των μηνυμάτων του, καθώς αυτά είναι κρυπτογραφημένα, μπορεί ωστόσο να υποθέσει ότι ο χρήστης στέλνει μήνυμα στον εξυπηρετητή που περιέχει το αναγνωριστικό χρήστη και τον κωδικό εισόδου.

Όταν αργότερα ο χρήστης αποσυνδεθεί από το δίκτυο, ο επίδοξος εισβολέας χρησιμοποιεί τη διεύθυνση MAC του πρώτου για να συνδεθεί στο δίκτυο. Είναι δυνατό τώρα στέλνοντας ένα αντίγραφο του παλιού μηνύματος με το αναγνωριστικό του χρήστη να αποκτήσει πρόσβαση στον εξυπηρετητή. Πρόκειται για *αναπαραγωγή* ενός παλιού μηνύματος χωρίς να είναι γνωστό το ακριβές περιεχόμενο του. Το σημείο πρόσβασης, ορθώς, θα αποκωδικοποιήσει το μήνυμα, καθώς αυτό είχε αρχικά κρυπτογραφηθεί με έγκυρο κλειδί.

Υπάρχουν πολλά άλλα παραδείγματα στα οποία μια επίθεση με αναπαραγωγή μπορεί να διαβάλει την ασφάλεια, εκτός κι αν το δίκτυο έχει σχεδιαστεί ειδικά ώστε να εντοπίζει και να απορρίπτει παλιά αντίγραφα μηνυμάτων. Ένα πρωτόκολλο ασύρματης ασφαλείας θα έπρεπε να επιτρέπει να είναι αποδεκτό μόνο ένα αντίγραφο κάθε μηνύματος.

Μπορεί να προκαλεί έκπληξη, αλλά το WEP δεν παρέχει μηχανισμούς προστασίας από την αναπαραγωγή μηνυμάτων. Υπάρχει ένα αριθμός ακολουθίας στο πλαίσιο MAC, ο οποίος πρέπει να αυξάνεται. Ωστόσο, αυτός δεν προστατεύεται από το WEP, με αποτέλεσμα να είναι εύκολο να τροποποιηθεί ο αριθμός ακολουθίας προκειμένου να φαίνεται έγκυρος, χωρίς να μεταβάλλεται το κρυπτογραφημένο τμήμα του πλαισίου.

Η προστασία, όσον αφορά την αναπαραγωγή μηνυμάτων στο WEP, δεν παρουσιάζει απλά αδυναμίες, αλλά θεωρείται ανύπαρκτη.

3.4.4 Εντοπισμός τροποποίησης μηνυμάτων

Το WEP διαθέτει μηχανισμό σχεδιασμένο για την αποφυγή της τροποποίησης των μηνυμάτων. Είναι προφανές το κέρδος ενός κακόβουλου χρήστη, αν η τροποποίηση μηνυμάτων σημαίνει την αλλαγή του περιεχομένου τους, για παράδειγμα την αντικατάσταση του λογαριασμού τράπεζας που προορίζεται μια συναλλαγή ή του ποσού μου μεταφέρεται. Ωστόσο, στην πράξη, τροποποιήσεις τέτοιας ευρείας κλίμακας δεν είναι εφικτές, καθώς προϋποθέτουν την ανάγνωση του αρχικού μηνύματος και τη δυνατότητα δημιουργίας νέων πλαστών μηνυμάτων.

Στην περίπτωση που δεν είναι δυνατή η αποκρυπτογράφηση του μηνύματος, δεν είναι προφανές το κέρδος που προκύπτει από τροποποίηση του κρυπτογραφημένου κειμένου. Ωστόσο, έχει προταθεί η μέθοδος "bit flipping" κατά την οποία μεταβάλλονται λίγα bit του κρυπτογραφημένου κειμένου κάθε φορά [3]. Στηρίζεται στην παρατήρηση ότι η θέση της επικεφαλίδας IP είναι συνήθως γνωστή μετά την κρυπτογράφηση, καθώς το WEP δεν αναδιατάσσει τις θέσεις των byte. Επειδή η επικεφαλίδα IP περιέχει έλεγχο αθροίσματος (checksum), η αλλαγή ενός bit της επικεφαλίδας οδηγεί σε αποτυχία του σχετικού ελέγχου. Αν ωστόσο, αλλάξουν *επιπλέον* και τα bit του ελέγχου αθροίσματος, τότε ο έλεγχος μπορεί να περάσει επιτυχώς. Η σχετική έρευνα έχει δείξει ότι, αλλάζοντας λίγα bit και ελέγχοντας την αποδοχή ή μη του πακέτου IP, είναι δυνατή η αποκρυπτογράφηση τμημάτων ενός πλαισίου.

Στα πλαίσια αυτά, το WEP περιλαμβάνει το πεδίο τιμής ελέγχου ακεραιότητας (Integrity Check Value – ICV). Η λογική του έχει ως εξής: υπολογίζεται μια τιμή ελέγχου ή κυκλικού ελέγχου πλεονασμού (Cyclic Redundancy Check – CRC) όλων των δεδομένων προς κρυπτογράφηση, γίνεται προσθήκη της τιμής αυτής στο τέλος των δεδομένων, και στη συνέχεια κρυπτογράφηση όλων μαζί. Αν αλλάξει κάποιο bit στο κρυπτογραφημένο κείμενο, τότε τα αποκρυπτογραφημένα δεδομένα δε θα έχουν την ίδια τιμή ελέγχου με αποτέλεσμα να εντοπιστεί η τροποποίηση. Το σκεπτικό είναι ότι, επειδή το ICV είναι κρυπτογραφημένο, δεν είναι δυνατή η τροποποίησή του ώστε να συμφωνεί με τις υπόλοιπες αλλαγές. Στοχεύει αποκλειστικά στην προστασία του κρυπτογραφημεου κειμένου. Αν ο κακόβουλος χρήστης διαθέτει

τα κλειδιά, μπορεί να τροποποιήσει τα δεδομένα και να υπολογίσει το νέο ICV που προκύπτει, προτού κρυπτογραφήσει ξανά και προωθήσει το πλαίσιο.

Ωστόσο, το ICV παρουσιάζει αδυναμίες ακόμη και στην προστασία του κρυπτογραφημένου κειμένου. Η μέθοδος CRC που χρησιμοποιείται για τον υπολογισμό του ICV είναι *γραμμική*. Ως εκ τούτου, είναι δυνατή η πρόβλεψη των bit του ICV που θα αλλάξουν με την τροποποίηση ενός μόνο bit του μηνύματος [3]. Δεν απαιτείται γνώση της πραγματικής τιμής του μη κρυπτογραφημένου κειμένου· με την αντιστροφή της τιμής για ένα συγκεκριμένο bit των δεδομένων μπορεί να διατηρηθεί η εγκυρότητα του ICV αντιστρέφοντας επίσης ένα συγκεκριμένο συνδυασμό από bit του τελευταίου. Δυστυχώς, επειδή το WEP χρησιμοποιεί τη λογική πράξη XOR για τη λήψη του κρυπτογραφημένου κειμένου, η αντιστροφή των bit δεν επηρεάζει τη διαδικασία κρυπτογράφησης. Η αντιστροφή ενός bit στο μη κρυπτογραφημένο κείμενο αντιστρέφει πάντα το ίδιο bit στο κρυπτογραφημένο, και αντίστροφα. Συνεπώς, το WEP δεν παρέχει αποτελεσματική προστασία από την τροποποίηση του κρυπτογραφημένου κειμένου.

3.4.5 Απόρρητο μηνυμάτων

Το απόρρητο αποτελεί τον περισσότερο σημαντικό μηχανισμό ασφαλείας, καθώς αφορά την ίδια τη μέθοδο κρυπτογράφησης που χρησιμοποιεί το WEP. Η επίθεση εναντίον της κρυπτογράφησης έχει δύο βασικούς στόχους: είτε την αποκωδικοποίηση ενός μηνύματος ή τη λήψη των κλειδιών. Μεγαλύτερη επιτυχία αποτελεί η λήψη των κλειδιών. Μόλις κάποιος εισβολέας αποκτήσει τα κλειδιά είναι σε θέση να αποκωδικοποιήσει οποιοδήποτε μήνυμα. Εν τούτοις, αυτό δε συνεπάγεται αυτομάτως πρόσβαση σε εμπιστευτικές πληροφορίες, καθώς υπάρχουν επιπλέον εσωτερικά στρώματα ασφαλείας, όπως είναι η ασφάλεια κωδικού εισόδου σε εξυπηρετητές και λειτουργικά συστήματα. Ωστόσο, αν ο κακόβουλος χρήστης αποκτήσει τα κλειδιά χωρίς να εντοπιστεί, τότε μπορεί να βρει το χρόνο που χρειάζεται για να συλλέξει ευαίσθητες πληροφορίες. Σε αντίθετη πρίπτωση, το κλειδί WEP μπορεί να αλλάξει, αποκλείοντας έτσι τον εισβολέα.

Η αμέσως μεγαλύτερη επιτυχία μετά τη λήψη των κλειδιών, είναι ο προσδιορισμός του μη κρυπτογραφημεου κειμένου (plaintext), γεγονός που επιτρέπει μεγάλο εύρος επιθέσεων με χρήση τροποποίησης και αναπαραγωγής μηνυμάτων. Αυτή η πληροφορία μπορεί επίσης να χρησιμοποιηθεί και για τη λήψη των κλειδιών.

Υπάρχουν τρεις αδυναμίες στον τρόπο που το RC4 χρησιμοποιείται στο WEP, κάθε μία από τις οποίες θα εξεταστεί στη συνέχεια ξεχωριστά:

Επαναχρησιμοποίηση του IV

Προκειμένου να αναδειχθεί η αδυναμία που προκύπτει από την επαναχρησιμοποίηση του IV, θα εξετάσουμε τη λειτουργία του: αντί να χρησιμοποιείται ένα σταθερό μυστικό κλειδί, μια τιμή IV μήκους 24 bit προστίθεται στο μυστικό κλειδί και ο συνδυασμός που παράγεται αποτελεί το κλειδί κρυπτογράφησης. Η τιμή του IV στέλνεται μέσα στο πλαίσιο, ώστε η λαμβάνουσα συσκευή να είναι σε θέση να κάνει την αποκρυπτογράφηση. Κατ΄ αρχήν, το IV εξασφαλίζει, έτσι, ότι δύο πανομοιότυπα μηνύματα δε θα παράγουν το ίδιο κρυπτογράφημα. Ωστόσο, υπάρχει ένας πιο σημαντικός στόχος του IV, ο οποίος σχετίζεται με τον τρόπο που το WEP χρησιμοποιεί το XOR για τη δημιουργία του κρυπτογραφήματος.

Ας υποθέσουμε ότι δεν υπάρχει το IV και ότι χρησιμοποιείται μόνο το μυστικό κλειδί για την κρυπτογράφηση. Για κάθε πλαίσιο, ο αλγόριθμος RC4 αρχικοποιείται με την τιμή κλειδιού προτού αρχίσει την παραγωγή της ψευδοτυχαίας ροής κλειδιού. Ωστόσο, αν το κλειδί είναι σταθερό, τότε ο RC4 θα αρχικοποιείται στην ίδια κατάσταση κάθε φορά. Ως εκ τούτου, η

παραγόμενη ροή κλειδιού θα είναι η ίδια ακολουθία byte για κάθε πλαίσιο. Κάτι τέτοιο, θα ήταν φυσικά καταστροφικό, δεδομένου ότι, αν κάποιος εισβολέας μπορούσε να προσδιορίσει τη ροή κλειδιού, τότε θα ήταν σε θέση να αποκρυπτογραφήσει κάθε πλαίσιο χρησιμοποιώντας την πράξη XOR σε συνδυασμό με τη γνωστή ακολουθία. Δε χρειάζεται καν η γνώση του κλειδιού.

Προσθέτοντας κάθε φορά την τιμή του IV στο κλειδί, ο RC4 αρχικοποιείται σε διαφορετική κατάσταση για κάθε πλαίσιο και επομένως η ροή κλειδιού είναι διαφορετική για κάθε κρυπτογράφηση. Αν, ωστόσο, η τιμή του IV παραμένει σταθερή, τότε η περίπτωση δε διαφέρει από αυτή του στατικού κλειδιού.

Συμπεραίνουμε, λοιπόν, ότι το στατικό IV είναι επισφαλές, σε αντίθεση με τη χρήση διαφορετικού IV για *κάθε* πλαίσιο. Ωστόσο, υπάρχει περιορισμένος αριθμός δυνατών IV και, ενώ είναι εφικτή η χρήση διαφορετικών τιμών του για την πλειοψηφία των πλαισίων, τελικά θα γίνει επαναχρησιμοποίησή τους. Κάτι τέτοιο δεν είναι αποδεκτό, εν τούτοις, είναι γεγονός στην περίπτωση του WEP.

Στο πλαίσιο αυτό, θα μπορούσε να υποθέσει κανείς ότι η καλύτερη προσέγγιση θα ήταν να υπολογίζεται μια τυχαία τιμή. Ωστόσο, πολύ γρήγορα θα εμφανίζονταν επαναλαμβανόμενες τιμές του IV. Τελικά, ο καλύτερος τρόπος παραγωγής του IV είναι με αύξηση της τιμής κατά 1 για κάθε πλαίσιο, που καθιστά το χρόνο μεταξύ επαναλαμβανόμενων τιμών το μεγαλύτερο δυνατό. Εν τούτοις, με μήκος 24 bit, η **σύγκρουση τιμών IV** εμφανίζεται εγγυημένα μετά τη μετάδοση 2^{24} πλαισίων. Το IEEE 802.11b μπορεί να στείλει 500 πλαίσια πλήρους μήκους το δευτερόλεπτο και, με το ρυθμό αυτό, ο χώρος των IV εξαντλείται σε περίπου επτά ώρες.

Στην πράξη, μια σύγκρουση θα συμβεί πολύ νωρίτερα γιατί υπάρχουν πολλές συσκευές που μεταδίδουν ταυτόχρονα, κάθε μία αυξάνοντας μια ξεχωριστή τιμή IV με χρήση του ίδιου κλειδιού.

Όπως έχει αναφερθεί, με γνωστή τη ροή κλειδιού που αντιστοιχεί σε δεδομένη τιμή του IV, είναι δυνατή η αποκρυπτογράφηση κάθε επόμενου πλαισίου που χρησιμοποιεί το ίδιο IV (και μυστικό κλειδί). Αυτό είναι γεγονός ανεξάρτητα από το μήκος του κλειδιού. Για την αποκρυπτογράφηση κάθε μηνύματος, είναι απαραίτητη η γνώση της ροής κλειδιού για κάθε δυνατή τιμή του IV, περίπου 17 εκατομμύρια στο σύνολό τους. Η αποθήκευση ροών κλειδιού μήκους 1500 byte για κάθε IV χρειάζεται αποθηκευτικό χώρο της τάξης των 23 Gbytes, μέγεθος εφικτό για τους περισσότερους οικιακούς υπολογιστές σήμερα. Δεδομένης μια τέτοιας βάσης δεδομένων, είναι δυνατή η αποκρυπτογράφηση κάθε μηνύματος, χωρίς να είναι γνωστό το μυστικό κλειδί. Ωστόσο, απαιτείται η εύρεση όλων των ροών κλειδιού που δεν είναι τόσο εύκολη υπόθεση.

Ας υποθέσουμε ότι έχουν καταγραφεί δύο μηνύματα με κοινό IV και μυστικό κλειδί. Είναι γνωστό ότι η ροή κλειδιού είναι ίδια και στις δύο περιπτώσεις, ωστόσο δεν είναι γνωστή η τιμή της. Έστω, P_1 και P_2 τα δύο μη κρυπτογραφημένα μηνύματα, C_1 και C_2 τα αντίστοιχα κρυπτογραφήματα τους, ενώ K η κοινή ροή κλειδιού, τότε θα ισχύει:

$$C_1 = P_1 \oplus K$$

και

$$C_2 = P_2 \oplus K$$

Με εφαρμογή λογικού XOR μεταξύ τους προκύπτει:

$$C_1 \oplus C_2 = (P_1 \oplus K) \oplus (P_2 \oplus K) = P_1 \oplus P_2 \oplus K \oplus K = P_1 \oplus P_2$$

Επομένως, ο εισβολέας έχει στη διάθεσή του ένα μήνυμα που παράγεται από το λογικό XOR των δύο μη κρυπτογραφημένων μηνυμάτων. Ωστόσο, κάποιες από τις τιμές των μη κρυπτογραφημένων κειμένων είναι γνωστές, όπως ορισμένα πεδία της επικεφαλίδας. Σε άλλα πεδία δεν είναι γνωστή η τιμή αλλά ο σκοπός τους. Για παράδειγμα, τα πεδία της διεύθυνσης IP έχουν περιορισμένο σύνολο τιμών στα περισσότερα δίκτυα. Το τμήμα σώματος του κειμένου αποτελεί συχνά πληροφορίες κωδικοποιημένες σε ASCII, δίνοντας πάλι κάποιες πιθανές τιμές.

Αν, μετά από μια χρονική περίοδο, συλλεχθούν αρκετά δείγματα από επαναλαμβανόμενα IV, τότε μπορεί κανείς να μαντέψει σημαντικά τμήματα της ροής κλειδιού και, ως εκ τούτου, να αποκρυπτογραφήσει περισσότερες πληροφορίες. Άλλωστε, πρέπει να τονιστεί ξανά ότι αν προσδιοριστεί η ροή κλειδιού για κάποιο IV, είναι δυνατή η αποκρυπτογράφηση κάθε πλαισίου που ακολουθεί και χρησιμοποιεί το ίδιο IV, καθώς επίσης, και η παραγωγή πλαστών πλαισίων με αυτό το IV· όλα αυτά χωρίς να είναι γνωστό το κλειδί.

Εν τούτοις, η αδυναμία αυτή δε θεωρείται σημαντική απειλή, όσον αφορά την καθημερινή χρήση. Εξάλλου, θα χρειαζόταν πολλή προσπάθεια για την αποκρυπτογράφηση μεγάλου αριθμού πλαισίων και η ευφυία που απαιτείται για τον προσδιορισμό του μη κρυπτογραφημένου κειμένου καθιστά δύσκολη την ανάπτυξη αυτοματοποιημένου εργαλείου για το σκοπό αυτό.

Αδύναμα κλειδιά RC4

Το θεμελιώδες τμήμα του αλγορίθμου RC4 δεν είναι η κρυπτογράφηση αλλά η παραγωγή ψευδοτυχαίων αριθμών. Μόλις ετοιμαστεί η ακολουθία ψευδοτυχαίων byte, μπορεί να χρησιμοποιηθεί για την κρυπτογράφηση των δεδομένων με τη βοήθεια της πράξης XOR. Όπως έχει αναφερθεί, η χρήση του XOR αποτελεί αδυναμία αν δεν εφαρμοστεί σωστά, ωστόσο, ιδιαίτερο ενδιαφέρον παρουσιάζει η ψευδοτυχαία ακολουθία ή ροή κλειδιού.

Ο RC4 γεμίζει ένα πίνακα μεγέθους 256 byte με τιμές από 0 έως 255. Για την ακρίβεια, κάθε τιμή του εύρους αυτού, εμφανίζεται μια μόνο φορά μέσα στον πίνακα. Ωστόσο, η σειρά με την οποία εμφανίζονται οι αριθμοί είναι «τυχαία». Αυτό είναι γνωστό ως μετάθεση των τιμών. Επιπρόσθετα, οι τιμές αναδιατάσσονται συνεχώς με την παραγωγή κάθε ψευδοτυχαίου byte, ώστε να προκύπτει διαφορετική μετάθεση του πίνακα κάθε φορά.

Κάθε ψευδοτυχαίο byte παράγεται επιλέγοντας μια τιμή από τη μετάθεση, βάσει δύο δεικτών, του i και του j, οι οποίοι επίσης αλλάζουν κάθε φορά. Υπάρχουν πολλές μεταθέσεις 255 τιμών που μπορούν να παραχθούν. Μάλιστα, σε συνδυασμό με τους δύο δείκτες, προκύπτουν $512 * 256!$ πιθανές μεταθέσεις.

Αυτή η ιδιότητα του RC4 τον καθιστά πολύ ισχυρό, παρά την απλή υλοποίηση του αλγορίθμου. Είναι εξαιρετικά δύσκολο να γίνει διάκριση μεταξύ μιας ψευδοτυχαίας ακολουθίας RC4 και μιας πραγματικά τυχαίας ακολουθίας, διαδικασία που απαιτεί δείγμα μεγέθους 1 Gbyte.

Εν τούτοις, ο RC4 παρουσιάζει μια αδυναμία, που φαίνεται καλύτερα αν εξεταστεί η λειτουργία του: αρχικά δημιουργεί ένα πίνακα με όλες τις τιμές 0-255. Ο πίνακας είναι γνωστός ως κουτί-S. Στη συνέχεια δημιουργεί ένα δεύτερο πίνακα μεγέθους 256 byte με το κλειδί να επαναλαμβάνεται συνεχώς μέχρι να γεμίσει ο πίνακας. Ακολούθως, αναδιατάσσει το κουτί-S βάσει των τιμών στον πίνακα του κλειδιού. Αυτή είναι η φάση αρχικοποίησης. Το πρώτο ψευδοτυχαίο byte παράγεται αναδιατάσσοντας το κουτί-S ξανά και επιλέγοντας ένα byte.

Το πρόβλημα είναι ότι δεν υπάρχουν πολλές αναδιατάξεις μεταξύ της αρχικής μορφής του πίνακα κλειδιού και του πρώτου ψευδοτυχαίου byte. Αποδεικνύεται τελικά, ότι για ορισμένες τιμές κλειδιών, τα οποία λέγονται *αδύναμα κλειδιά*, ένας δυσανάλογος αριθμός bit στα πρώτα λίγα byte της ροής κλειδιού μπορεί να προσδιοριστεί από λίγα bit στο ίδιο το κλειδί [4].

Ιδανικά, αλλάζοντας ένα οποιοδήποτε bit στο κλειδί, τότε η ροή κλειδιού που προκύπτει θα έπρεπε να είναι τελείως διαφορετική. Κάθε bit θα έπρεπε να έχει 50% πιθανότητα να είναι

διαφορετικό σε σχέση με την προηγούμενη ροή κλειδιού. Ωστόσο, αυτό δεν ισχύει στον RC4. Κάποια bit του κλειδιού έχουν μεγαλύτερη επίδραση απά τα υπόλοιπα, ενώ κάποια άλλα δεν έχουν καμία επίδραση στα πρώτα λίγα byte της ροής κλειδιού. Προκύπτουν δύο προβλήματα από το γεγονός αυτό. Κατ΄ αρχήν, αν μειωθεί ο αριθμός των αποτελεσματικών bit, τότε είναι ευκολότερο να γίνει επίθεση στα κλειδιά. Επιπλέον, είναι συνήθως πιο εύκολο να γίνει υπόθεση για τα πρώτα λίγα byte του μη κρυπτογραφημένου κειμένου. Για παράδειγμα, στο WEP, είναι συνήθως η επικεφαλίδα LLC που αρχίζει με την ίδια δεκαεξαδική τιμή "AA". Αν είναι γνωστό το μη κρυπτογραφημένο κείμενο, τότε μπορεί να εξαχθεί η ροή κλειδιού και να αρχίσει επίθεση στο κλειδί.

Υπάρχει μια πολύ απλή προσέγγιση για την αντιμετώπιση αυτής της αδυναμίας: να αγνοηθούν τα πρώτα λίγα byte της ροής κλειδιού. Με άλλα λόγια, να χρησιμοποιείται η έξοδος του RC4, αφού εκτελεστεί πρώτα για λίγο ο αλγόριθμος. Έχει γίνει η πρόταση να αγνοούνται τα πρώτα 256 byte της ροής κλειδιού, αλλά φυσικά το WEP δεν το εφαρμόζει αυτό, καθώς μια τέτοια αλλαγή θα καθιστούσε τα παλιότερα συστήματα μη διαλειτουργικά.

Το παραπάνω πρόβλημα μπορεί να μη φαίνεται τόσο σοβαρό, καθώς θα μπορούσε κανείς να προσδιορίσει τα αδύναμα κλειδιά και να αποφεύγει τη χρήση τους. Κάτι τέτοιο όμως δεν είναι δυνατό, εξαιτίας του IV που προστίθεται στο μυστικό κλειδί. Δεδομένου ότι το IV μεταβάλλεται συνεχώς, αργά ή γρήγορα, θα παραχθεί κάποιο αδύναμο κλειδί.

Απ΄ ευθείας επιθέσεις κλειδιού

Η προσθήκη μιας γνωστής τιμής του IV στο μυστικό κλειδί θεωρείται επισφαλής, καθώς επιτρέπει σε ένα εισβολέα να περιμένει για ένα ενδεχομένως αδύναμο κλειδί και να επιτεθεί σε αυτό απ ευθείας [4]. Υπάρχουν δύο περιπτώσεις: στη μία το IV ακολουθεί το μυστικό κλειδί, ενώ στην άλλη προηγείται αυτού. Η δεύτερη περίπτωση είναι η πιο ευάλωτη και αφορά το WEP.

Η όλη ιδέα βασίζεται στην εκμετάλλευση των πρώτων byte του αδύναμου κλειδιού. Αρχικά, ας υποτεθεί ότι είναι γνωστά τα πρώτα byte του μη κρυπτογραφημένου κειμένου, όπως και ισχύει στην περίπτωση του IEEE 802.11, καθώς είναι συνήθως μια επικεφαλίδα IEEE 802.1LLC SNAP. Με συνεχή παρακολούθηση της μετάδοσης, αναζητείται ένα αδύναμο κλειδί, το οποίο έχει παραχθεί από το IV. Υπάρχει συσχετισμός μεταξύ των byte του μη κρυπτογραφημένου κειμένου, του κρυπτογραφημένου και του μυστικού κλειδιού. Είναι περιορισμένος ο αριθμός των πιθανών τιμών των πρώτων byte του μυστικού κλειδιού που αντιστοιχούν σε δεδομένο κείμενο και το κρυπτογράφημά αυτού. Έχοντας καταγράψει περίπου 60 τέτοια μηνύματα, ο εισβολέας μπορεί να μαντέψει το πρώτο byte του κλειδιού με σχετική βεβαιότητα.

Η μέθοδος αυτή, μπορεί να χρησιμοποιηθεί έτσι ώστε να γίνεται επίθεση διαδοχικά σε κάθε byte του μυστικού κλειδιού και τελικά να αποκαλύπτεται ολόκληρο το κλειδί. Ας σημειωθεί ότι η αύξηση του μήκους του κλειδιού από τα 40 στα 104 bit, σημαίνει μόνο ότι απαιτείται 2,5 φορές περισσότερος χρόνος να εξαχθεί το κλειδί, με άλλα λόγια, *ο χρόνος αυξάνεται γραμμικά και όχι εκθετικά.*

Όλες οι προηγούμενες αδυναμίες του WEP, που περιγράφηκαν στην ενότητα, φαίνονται ασήμαντες συγκρινόμενες με αυτή την επίθεση, καθώς απότελεί μια μέθοδο εξαγωγής των κλειδιών σε χρόνο γραμμικό. Και επειδή χρησιμοποιεί μηχανική προσέγγιση, είναι εύκολο να αναπτυχθούν αυτόματα εργαλεία που να εφαρμόζουν την επίθεση. Υπάρχει ήδη πληθώρα τέτοιων εργαλείων, τα οποία διατίθενται ελεύθερα στο διαδίκτυο.

Κεφάλαιο 4

Πρότυπο ασφαλείας IEEE 802.11i

Το IEEE 802.11i αποτελεί μια προσθήκη στο πρότυπο των Ασυρμάτων Τοπικών Δικτύων, η οποία σχετίζεται με τις νέες προδιαγραφές ασφαλείας. Στα πλαίσια αυτά, ορίζει ένα νέο τύπο δικτύου, το οποίο λέγεται **Δίκτυο Εύρωστης Ασφαλείας** (Robust Security Network – RSN) ή WPA2. Σε ορισμένες πτυχές του είναι όμοιο με τα συμβατικά δίκτυα ή το WEP. Ωστόσο, προκειμένου να συνδεθεί στο RSN, μια συσκευή απαιτείται να διαθέτει ένα αριθμό νέων δυνατοτήτων. Στο *πραγματικό* RSN, το σημείο πρόσβασης επιτρέπει μόνο σε κινητές συσκευές προδιαγραφών RSN να συνδεθούν και θέτει αυστηρούς κανόνες ασφαλείας κατά τη διαδικασία. Ωστόσο, επειδή η σχετική αναβάθμιση απαιτεί την πάροδο ορισμένου χρόνου, στον οποίο πρέπει να εξασφαλιστεί η διαλειτουργικότητα του συμβατικού εξοπλισμού, το IEEE 802.11i ορίζει το **Δίκτυο Μεταβατικής Ασφαλείας** (Transitional Security Network – TSN) ή WPA, στο οποίο συστήματα προδιαγραφών RSN, αλλά και WEP, μπορούν να λειτουργούν παράλληλα.

Δυστυχώς, οι περισσότερες από τις υπαρχουσες κάρτες ασύρματης δικτύωσης δεν είναι δυνατό να αναβαθμιστούν στο RSN, επειδή οι κρυπτογραφικές λειτουργίες που απαιτούνται δεν υποστηρίζονται από το υλισμικό και δε μπορούν να καλυφθούν από αναβάθμιση του λογισμικού. Ως εκ τούτου, απαιτείται η πάροδος ορισμένου χρόνου προκειμένου θα υπάρξουν πλήρως λειτουργικά δίκτυα RSN. Αντίθετα, τα δίκτυα TSN μπορούν να υλοποιηθούν άμεσα.

Το κεφάλαιο αυτό, πραγματεύεται τις δύο παραπάνω προσεγγίσεις του προτύπου IEEE 802.11i, ενώ στο τέλος γίνεται σύγκριση των τριών σχημάτων ασφαλείας, TSN, RSN και WEP.

4.1 Ασύρματη Προστατευμένη Πρόσβαση

Το πρότυπο IEEE 802.11i ανέπτυξε την Ασύρματη Προστατευμένη Πρόσβαση (Wi-Fi Protected Access – WPA) προκειμένου να καλυφθεί η ανάγκη για περισσότερη ασφάλεια από αυτή που παρέχει το WEP, χρησιμοποιώντας τις δυνατότητες του υπάρχοντος εξοπλισμού ασύρματης δικτύωσης. Για το σκοπό αυτό, ορίστηκε το **Πρωτόκολλο Χρονικής Ακεραιότητας Κλειδιού** (Temporal Key Integrity Protocol – TKIP), το οποίο παρουσιάζεται στην ενότητα 4.1.1. Το TKIP μπορεί να χρησιμοποιηθεί απλά με αναβάθμιση του λογισμικού των παλιότερων προϊόντων ασύρματης δικτύωσης, ενή η χρήση του στο RSN είναι προαιρετική.

Το πρότυπο WPA αποτελεί ουσιαστικά ένα υποσύνολο των προδιαγραφών ασφαλείας του RSN και υλοποιεί το TSN δίκτυο.

4.1.1 Πρωτόκολλο Χρονικής Ακεραιότητας Κλειδιού

Οι αδυναμίες του WEP, που παρουσιάστηκαν λεπτομερώς στην ενότητα 3.4, μπορούν να συνοψιστούν στον πίνακα 4.1.

Το Πρωτόκολλο Χρονικής Ακεραιότητας Κλειδιού (TKIP) εισάγει μια σειρά μέτρων που αντιμετωπίζουν κάθε ένα από τα ελαττώματα αυτά. Παρόλο που δεν είναι δυνατό να γίνουν μεγάλες αλλαγές, όπως για παράδειγμα να τροποποιηθεί ο τρόπος υλοποίησης του RC4 σε υλισμικό, εν τούτοις, προστίθενται μια σειρά από διορθωτικά εργαλεία γύρω από το υπάρχον υλισμικό. Οι αλλαγές που εφαρμόζονται στο WEP για την υλοποίηση του TKIP παρατίθενται στον πίνακα 4.2, όπου οι αριθμοί σε παρένθεση υποδεικνύουν τις αδυναμίες του πίνακα 4.1 που ανιμετωπίζει η κάθε αλλαγή.

1	Η τιμή του IV είναι πολύ μικρή και δεν αποτρέπεται η επαναχρησιμοποίησή της.
2	Ο τρόπος παραγωγής των κλειδιών από το IV καθιστά το WEP ευάλωτο σε επιθέσεις αδύναμων κλειδιών.
3	Δεν υπάρχει αποτελεσματικός τρόπος εντοπισμού της τροποποίησης των μηνυμάτων (ακεραιότητα μηνυμάτων).
4	Το WEP χρησιμοποιεί το κύριο κλειδί και δεν προβλέπει την ανανέωση των κλειδιών.
5	Δεν παρέχεται προστασία από την αναπαραγωγή των μηνυμάτων.

Πίνακας 4.1: Οι Αδυναμίες του WEP

Σκοπός	**Αλλαγή**	**Αδυναμίες που αντιμετωπίζονται**
Ακεραιότητα μηνυμάτων	Προσθήκη πρωτοκόλλου ακεραιότητας μηνυμάτων που να αποτρέπει την τροποποίηση και να μπορεί να υλοποιηθεί με λογισμικό σε μικροεπεξεργαστή μικρής ισχύος.	(3)
Επιλογή και χρήση IV	Αλλαγή των κανόνων επιλογής των τιμών IV και επαναχρησιμοποίηση του IV ως μετρητή αναπαραγωγής.	(1) (3)
Κλειδί ανά Πακέτο	Αλλαγή του κλειδιού κρυπτογράφησης για κάθε πλαίσιο.	(1) (2) (4)
Μέγεθος IV	Αύξηση του μεγέθους του IV ώστε να αποτρέπεται η επαναχρησιμοποίησή του.	(1) (4)
Διαχείριση Κλειδιών	Προσθήκη μηχανισμού διάθεσης και αλλαγής των κλειδιών που εκπέμπονται.	(4)

Πίνακας 4.2: Αλλαγές από το WEP στο TKIP

Ακεραιότητα μηνυμάτων

Όπως έχει αναφερθεί, η ακεραιότητα μηνυμάτων αποτελεί σημαντική παράμετρος στο θέμα της ασφάλειας. Το WEP διαθέτει το ICV για τον εντοπισμό της τροποποίησης μηνυμάτων, το οποίο, όμως, δεν είναι αποτελεσματικό. Αν και δεν αποτελεί μέρος της ασφάλειας του TKIP, ωστόσο η τιμή του εξακολουθεί να υπολογίζεται.

Μια απλή μέθοδος εντοπισμού τροποποιήσεων, είναι ο συνδυασμός όλων των byte ενός μηνύματος για την παραγωγή μιας τιμής ελέγχου και η αποστολή αυτής μαζί με το μήνυμα. Η λαμβάνουσα πλευρά μπορεί να κάνει τον αντίστοιχο υπολογισμό και να συγκρίνει το αποτέλεσμα, το οποίο θα διαφέρει εφόσον αλλάξει κάποιο bit.

Αυτή η απλή προσέγγιση δεν είναι αποτελεσματική, καθώς ένας κακόβουλος χρήστης είναι σε θέση να υπολογίσει ξανά την τιμή ελέγχου ώστε να συμφωνεί με τις τροποποιήσεις του στο μήνυμα. Ωστόσο, η βασική ιδέα είναι ίδια: Συνδυασμός όλων των byte του μηνύματος για την παραγωγή μιας τιμής ελέγχου που καλείται Κώδικας Ελέγχου Ακεραιότητας (Message Integrity Code – MIC) και αποστολής της μαζί με το μήνυμα. Εν τούτοις, στην περίπτωση του TKIP, ο MIC υπολογίζεται χρησιμοποιώντας μια μη αντιστρέψιμη επεξεργασία σε συνδυασμό με ένα μυστικό κλειδί. Ως εκ τούτου, ο επίδοξος εισβολέας δεν είναι σε θέση να υπολογίζει εκ νέου την τιμή του MIC, εφόσον δε γνωρίζει το μυστικό κλειδί. Μόνο ο παραλήπτης μπορεί να υπολογίσει και να ελέγξει την τιμή.

Υπάρχουν πολλές ασφαλείς μέθοδοι για την παραγωγή του MIC, οι οποίες όμως, απαιτούν είτε την εισαγωγή νέων κρυπτογραφικών αλγορίθμων ή ταχείς υπολογισμούς πολλαπλασιασμού. Ωστόσο, οι μικροεπεξεργαστές στις περισσότερες υπάρχουσες κάρτες ασύρματης δικτύωσης δε διαθέτουν μεγάλη ισχύ. Και ενώ μια προσέγγιση θα ήταν η μεταφορά του υπολογιστικού φόρτου στο λογισμικό του οδηγού, που αντέχουν οι επεξεργαστές των σύγχρονων προσωπικών υπολογιστών, ωστόσο η λύση αυτή δε μπορεί να εφαρμοστεί στα σημεία πρόσβασης, τα οποία, ως επί το πλείστον, δε διαθέτουν την απαιτούμενη επεξεργαστική ισχύ.

Συνεπώς, απαιτείται μια μέθοδος ασφαλής όσο οι ήδη γνωστές προσεγγίσεις, που να μην απαιτεί όμως, ούτε πολλαπλασιασμούς, ούτε νέους κρυπτογραφικούς αλγορίθμους. Μια καλή λύση συμβιβασμού δόθηκε από τον κρυπτογράφο Niels Ferguson με μια μέθοδο που ονόμασε Μιχάλη (Michael). Ο Μιχάλης είναι μια μέθοδος υπολογισμού του MIC που δε χρησιμοποιεί πολλαπλασιασμούς, παρά μόνο πράξεις ολίσθησης και πρόσθεσης. Ο Μιχάλης μπορεί να υλοποιηθεί από τα σύγχρονα σημεία πρόσβασης χωρίς να καταναλώνει ολόκληρη την υπολογιστική τους ισχύ. Ωστόσο, το κόστος της απλότητας είναι ότι ο Μιχάλης είναι ευάλωτος σε επιθέσεις ωμής δύναμης (brute force), κατά τις οποίες ο επίδοξος εισβολέας είναι σε θέση να κάνει πολλές αλλεπάλληλες επιθέσεις με ταχύ ρυθμό. Ο Μιχάλης αντιμετωπίζει την αδυναμία αυτή με την εισαγωγή της ιδέας των **αντιμέτρων** (countermeasures).

Η φιλοσοφία των αντιμέτρων είναι πολύ απλή: ανάπτυξη μιας αξιόπιστης μεθόδου εντοπισμού επιθέσεων και λήψης των κατάλληλων μέτρων. Το απλούστερο των αντιμέτρων είναι το κλείσιμο ολόκληρου του δικτύου όταν ανιχνευθεί μια επίθεση, ώστε ο επίδοξος εισβολέας να μην είναι σε θέση να κάνει επιπλέον απόπειρες.

Ο Μιχάλης επιτρέπει τον υπολογισμό της τιμής του MIC η οποία προστίθεται στο μήνυμα πριν την κρυπτογράφηση και ελέγχεται από τον παραλήπτη μετά την αποκρυπτογράφηση. Η τιμή αυτή, προσφέρει την ακεραιότητα μηνυμάτων που δεν παρέχει το WEP.

Ο Μιχάλης εφαρμόζεται στις MSDU καί όχι σε κάθε MPDU. Αυτό προσφέρει δύο πλεονεκτήματα. Κατ΄ αρχήν, όσον αφορά την πλευρά της ασύρματης συσκευής, επιτρέπει την υλοποίηση του υπολογισμού στον οδηγό της συσκευής που εκτελείται στον υπολογιστή πριν την προώθηση της MSDU στην κάρτα ασύρματης δικτύωσης. Επιπλέον, περιορίζει το επιπρόσθετο κόστος, καθώς δεν απαιτείται η προσθήκη της τιμής του MIC σε κάθε θραύσμα (MPDU) του μηνύματος. Αντίθετα, η *κρυπτογράφηση* TKIP λαμβάνει χώρα στο επίπεδο της

MPDU.

Ο Μιχάλης χρειάζεται το δικό του μυστικό κλειδί, το οποίο πρέπει να είναι διαφορετικό από το μυστικό κλειδί που χρησιμοποιείται στην κρυπτογράφηση. Η εξαγωγή τέτοιων κλειδιών επιτυγχάνεται εύκολα παράγωντας χρονικά κλειδιά από το κύριο κλειδί.

Επιλογή και χρησιμοποίηση IV

Στην ενότητα 3.4, παρουσιάσαμε τις αδυναμίες στον τρόπο χρήσης του IV από το WEP, οι οποίες συνοπτικά είναι οι εξής:

- Το IV είναι πολύ μικρό με αποτέλεσμα οι τιμές του να επαναχρησιμοποιούνται συχνά σε ένα δίκτυο με πολλή κίνηση.
- Το IV δεν είναι ειδικό για κάθε σταθμό και επομένως το ίδιο IV μπορεί να χρησιμοποιηθεί με το ίδιο μυστικό κλειδί από πολλαπλές ασύρματες συσκευές.
- Ο τρόπος που το IV εισάγεται πριν από το κλειδί καθιστά το σύστημα ευάλωτο σε επιθέσεις αδύναμων κλειδιών (επιθέσεις FMS).

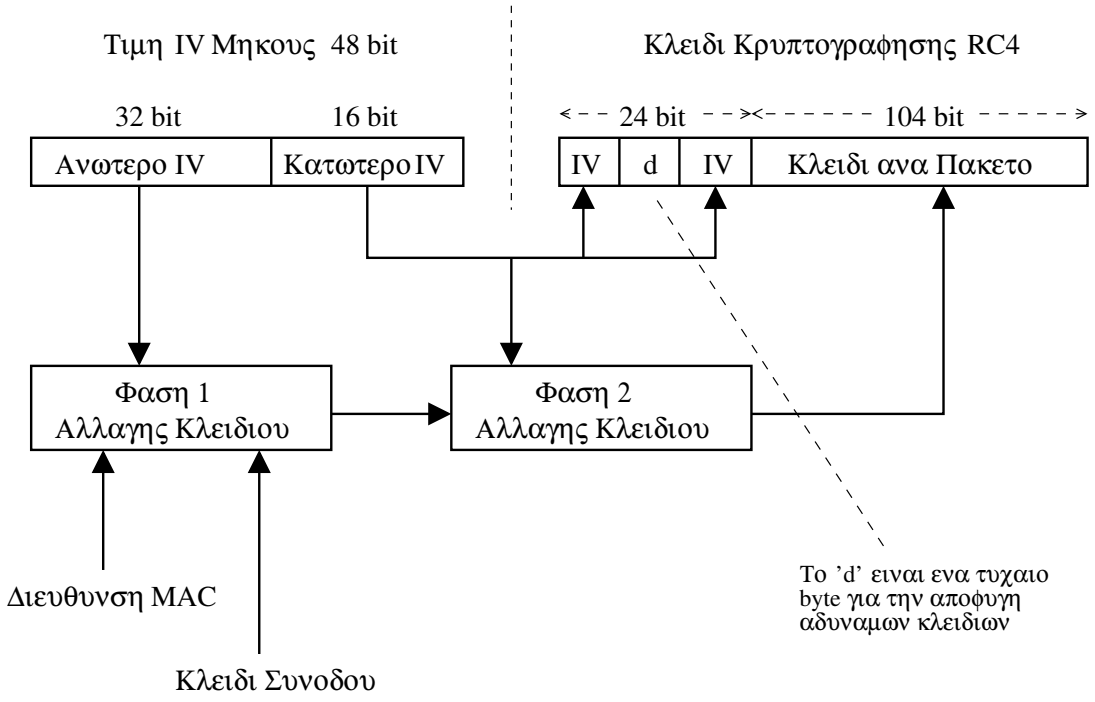
Στο πρότυπο του WEP, δεν υπάρχει η απαίτηση να αποφεύγεται η επαναχρησιμοποίηση του IV, σε αντίθεση με το TKIP. Η αύξηση του IV μπορεί να καθυστερεί τη σύγκρουση τιμών του IV, αυτή, ωστόσο, εμφανίζεται μετά από 16 εκατομμύρια πλαίσια. Ως εκ τούτου, το TKIP εισάγει νέους κανόνες όσον αφορά τη χρήση του IV. Ουσιαστικά, υπάρχουν τρεις διαφορές σε σύγκριση με το WEP:

1. Το μέγεθος του IV αυξάνεται από τα 24 στα 48 bit. Για την ακρίβεια προστίθενται 32 bit ακόμη, σχηματίζοντας ένα IV μήκους 56 bit. Ωστόσο, πρακτικά χρησιμοποιούνται μόνο τα 48 bit, καθώς ένα byte αξιοποιείται για την αποφυγή αδύναμων κλειδιών. Η αύξηση αυτή, εξαλείφει αποτελεσματικά τη σύγκρουση τιμών του IV, εν τούτοις, εξακολουθεί να υπάρχει η ανάγκη αποφυγής της χρήσης του ίδιου IV σε συνδυασμό με το ίδιο κλειδί από δύο διαφορετικές συσκευές.

   Από την άλλη πλευρά, η αλλαγή αυτή εισάγει ορισμένα προβλήματα όσον αφορά την υλοποίηση. Όπως είναι γνωστό, το IV στο WEP προστίθεται μπροστά από το μυστικό κλειδί προκειμένου να σχηματιστεί το κλειδί κρυπτογράφησης του αλγορίθμου RC4. Ως εκ τούτου, με το συνδυασμό του IV των 24 bit και ενός μυστικού κλειδιού των 40 bit, παράγεται ένα κλειδί RC4 μήκους 64 bit. Το υλισμικό του παλιότερου εξοπλισμού υποθέτει αυτή τη δομή κλειδιού και δε μπορεί να αναβαθμιστεί ξαφνικά ώστε να υποστηρίζει το νέο κλειδί των 88 bit για τις ανάγκες του TKIP. Στο πλαίσιο αυτό, ακολουθείται η εξής προσέγγιση: αντί να σχηματιστεί ένα νέο κλειδί RC4 από την ένωση του μυστικού κλειδιού και του IV, το IV χωρίζεται σε δύο τμήματα. Τα πρώτα 16 bit του νέου IV επεκτείνονται κατάλληλα στα 24 ώστε να αποφεύγονται γνωστά αδύναμα κλειδιά. Αυτή η τιμή μήκους 24 bit, χρησιμοποιείται όπως στα συστήματα WEP. Ωστόσο, αντί να ενωθεί με το μυστικό κλειδί, ένα νέο *ανάμεικτο κλειδί* (mixed key) παράγεται από το συνδυασμό του μυστικού κλειδιού με τα 32 bit που απομένουν στο IV. Ο τρόπος με τον οποίο το IV μεγάλου μήκους ενσωματώνεται στο κλειδί, ονομάζεται αλλαγή κλειδιού ανά πακέτο, και παρουσιάζεται στο Σχ. 4.1. Τέλος, πρέπει να σημειωθεί ότι η παραπάνω προσέγγιση επιτυγχάνει δύο στόχους:

   - Η τιμή του κλειδιού που χρησιμοποιείται στην κρυπτογράφηση RC4 είναι διαφορετική για κάθε τιμή του IV.

- Η δομή του κλειδιού του RC4 αποτελείται από το «παλιό» IV μήκους 24 bit και ένα πεδίο μυστικού κλειδιού των 104 bit.



Σχήμα 4.1: Δημιουργία του Κλειδιού Κρυπτογράφησης RC4 στο TKIP

2. Το IV αποκτάει ένα δευτερεύοντα ρόλο ως μετρητής ακολουθίας για την προστασία από επιθέσεις αναπαραγωγής. Αυτού του είδους η προστασία δεν παρέχεται από το WEP, όπου ένας επίδοξος εισβολέας είναι σε θέση να καταγράψει ένα έγκυρο πακέτο και να το αναπαράγει αργότερα. Σε μια τέτοια επίθεση, ο κακόβουλος χρήστης δεν επιχειρεί να αποκρυπτογραφήσει το μήνυμα, ωστόσο, προσπαθεί να υποθέσει το ρόλο αυτού. Για παράδειγμα, καταγράφοντας τα μηνύματα κατά τη διάρκεια της διαγραφής ενός αρχείου, είναι θεωρητικά δυνατό, αναπαράγωντας τα, να διαγραφεί ένα αρχείο με το ίδιο όνομα, χωρίς να παραβιαστεί καν η κρυπτογράφηση. Η προστασία από την αναπαραγωγή έχει ακριβώς ως στόχο την αποφυγή χρήσης παλιών μηνυμάτων με τον τρόπο αυτό. Το TKIP παρέχει ένα σχετικό μηχανισμό που ονομάζεται μετρητής ακολουθίας TKIP (TKIP Sequence Counter – TSC).

   Στην πραγματικότητα, ο TSC και το IV είναι το ίδιο. Η τιμή αρχίζει πάντα από το μηδέν και αυξάνεται κατά ένα για κάθε πακέτο που στέλνεται. Επειδή, είναι εγγυημένο ότι δεν πρόκειται να επαναληφθεί η τιμή του IV για ένα δοσμένο κλειδί, η αναπαραγωγή μπορεί να αποφευχθεί αγνοώντας οποιαδήποτε μηνύματα παρουσιάζουν τιμή του TSC που έχει ήδη ληφθεί. Οι κανόνες αυτοί εξασφαλίζουν ότι δεν είναι δυνατή μια επίθεση που θα στηρίζεται στην αναπαραγωγή παλιότερων καταγεγραμμένων μηνυμάτων.

   Ο απλούστερος τρόπος αποφυγής επιθέσεων αναπαραγωγής είναι η απόρριψη ληφθέντων μηνυμάτων στα οποία ο TSC δεν έχει αυξηθεί κατά 1 σε σχέση με το τελευταίο μήνυμα. Εν τούτοις, υπάρχουν αρκετοί πρακτικοί λόγοι που δεν επιτρέπουν αυτήν την προσέγγιση. Κατ΄ αρχήν, είναι δυνατό να χαθούν κάποια πλαίσια κατά τη μετάδοση λόγω παρεμβολών και θορύβου. Εξαιτίας ενός ενδεχόμενου χαμένου πλαισίου, όλα τα

πλαίσια που θα ακολουθούσαν θα απορρίπτονταν λανθασμένα επειδή ο TSC δε θα είχε αυξηθεί κατά 1.

Ως εκ τούτου, πρέπει να υιοθετηθεί μια προσέγγιση που να λαμβάνει υπόψη της τις επαναμεταδόσεις. Σύμφωνα με το πρότυπο, πρέπει να επιβεβαιώνεται η λήψη των πλαισίων με σύντομα μηνύματα ACK. Αν δε ληφθεί επιβεβαίωση, το μήνυμα πρέπει να επαναμεταδοθεί θέτοντας ένα bit που να υποδεικνύει ότι πρόκειται για αντίγραφο. Όντας μήνυμα επαναμετάδοσης, πρέπει να έχει την ίδια τιμή TSC με το αρχικό. Στην πράξη, η προσέγγιη αυτή είναι αποτελεσματική επειδή η λαμβάνουσα πλευρά χρειάζεται ένα μόνο έγκυρο αντίγραφο του μηνύματος και δεν υπάρχει πρόβλημα να απορρίπτονται τυχόν αντίγραφα κατά τον έλεγχο του TSC από τον παραλήπτη. Το ενδεχόμενο επαναμετάδοσης υποδηλώνει ότι ίδιες τιμές του TSC δεν πρέπει να εκλαμβάνονται απαραίτητα ως απόπειρα επίθεσης.

Ανακύπτει ένα ακόμη δυσκολότερο πρόβλημα εξαιτίας μιας νέας έννοιας γνωστής ως έκρηξη–ack (burst–ack). Σύμφωνα με το αρχικό πρότυπο IEEE 802.11, κάθε πλαίσιο δεδομένων που στέλνεται, πρέπει να επιβεβαιωθεί ξεχωριστά. Ενώ, η απαίτηση αυτή φαίνεται αποτελεσματική, δεν είναι, ωστόσο, επαρκής, καθώς ο αποστολέας πρέπει να σταματάει και να περιμένει μήνυμα ACK προτού συνεχίσει. Η έννοια της έκρηξης – ack είναι να στέλνονται διαδοχικά έως και 16 πλαίσια και στη συνέχεια να επιτρέπεται στον παραλήπτη να επιβεβαιώσει και τα 16 με ένα μήνυμα. Αν κάποια από τα μηνύματα δε ληφθούν επιτυχώς, ο παραλήπτης είναι σε θέση να υποδείξει ποια χρειάζονται επαναμετάδοση. Η έκρηξη–ack δεν είναι ακόμη μέρος του πρότυπου, εν τούτοις είναι πολύ πιθανό να συμπεριληφθει στο μέλλον.

- `ACCEPT`: Ο TSC είναι ο μεγαλύτερος που έχει εμφανιστεί έως τώρα.
- `REJECT`: Ο TSC είναι μικρότερος του μέγιστου -16.
- `WINDOW`: Ο TSC είναι μικρότερος του μέγιστου, αλλά μεγαλύτερος από το κατώτερο όριο (μέγιστο -16).

3. Το IV παράγεται κατάλληλα ώστε να αποφεύγονται ορισμένα αδύναμα κλειδιά, τα οποία καθιστούσαν το WEP ευάλωτο στην επίθεση FMS, που αποτελεί και τη μεγαλύτερη απειλή του προτύπου. Αυτή επιτρέπει την εξαγωγή του μυστικού κλειδιού παρακολουθώντας την κίνηση της δικτυακής κίνησης της ασύρματης ζεύξης με τη βοήθεια αυτοματοποιημένων εργαλείων, όπως είδαμε στην ενότητα 3.4.

   Ο Ron Rivest, σχεδιαστής του RC4, πρότεινε τη μη χρησιμοποίηση των πρώτων 256 byte που παράγονται από τον αλγόριθμο, προκειμένου να αντιμετωπιστεί αυτή η αδυναμία. Δεδομένου ότι το υλισμικό του υπάρχοντος εξοπλισμού ασύρματης δικτύωσης δεν υποστηρίζει την προτεινόμενη λύση, το TKIP θέτει τους εξής στόχους:

   - Προσπάθεια αποφυγής αδύναμων κλειδιών.
   - Προσπάθεια επιπρόσθετης απόκρυψης του μυστικού κλειδιού.

   Η επίθεση FMS βασίζεται στη δυνατότητα συλλογής πολλαπλών δειγμάτων πλαισίων που περιέχουν αδύναμα κλειδιά. Απαιτούνται μόλις 60 πλαίσια για να εξαχθούν τα πρώτα bit του ζητούμενου, ενώ η πλήρης αποκωδικοποίηση του κλειδιού μπορεί να γίνει μετά από λίγα εκατομμύρια πακέτα. Η προσέγγιση που υιοθετήθηκε από το TKIP είναι η αλλαγή του μυστικού κλειδιού για κάθε πακέτο. Με τον τρόπο αυτό, ο επίδοξος

εισβολέας δεν είναι σε θέση να συγκεντρώσει αρκετά δείγματα για να επιτεθεί σε κάποιο δοσμένο κλειδί.

Ένας επιπρόσθετος μηχανισμός άμυνας από την επίθεση FMS είναι η αποφυγή χρήσης αδύναμων κλειδιών. Το πρόβλημα είναι ότι κανείς δε γνωρίζει με ακρίβεια όλα τα αδύναμα κλειδιά. Ωστόσο, οι κρυπτογράφοι έχουν προσδιορίσει έναν τύπο κλειδιού που είναι αδύναμος. Αποδεικνύεται ότι θέτωντας κατάλληλα δύο bit του IV κατά τη φάση ανάμειξης κλειδιού, αποφεύγεται μια γνωστή κατηγορία αδύναμων κλειδιών.

Ορισμένοι κατασκευαστές έχουν τροποποιήσει την υλοποίηση του WEP ώστε να αποφεύγονται τιμές του IV που παράγουν αδύναμα κλειδιά. Εν τούτοις, προκύπτει ένα άλλο πρόβλημα με την προσέγγιση αυτή. Ως γνωστό, δεν υπάρχει επαρκής αριθμός τιμών του IV όταν αυτό έχει μήκος 24 bit. Μειώνοντας λοιπόν ακόμη περισσότερο το σύνολο τιμών του IV, περιορίζουμε το ένα πρόβλημα αλλά επιδεινώνουμε ένα άλλο. Στο TKIP δεν υπάρχει αυτός ο κίνδυνος, καθώς το μήκος του IV έχει διπλασιαστεί.

Η ενότητα αυτή, εστιάστηκε στις αλλαγές στον τρόπο χρήσης του IV από το TKIP. Συνοψίζοντας, υπάρχουν τρεις σημαντικές τροποποιήσεις: το μήκος αυξάνεται στα 48 bit, το IV χρησιμοποιείται ως μετρητής ακολουθίας (ο TSC), και το IV συνδυάζεται με το μυστικό κλειδί με περισσότερο πολύπλοκο τρόπο σε σχέση με το WEP. Η τελευταία αλλαγή επιτυγχάνει δύο στόχους: επιτρέπει την ενσωμάτωση του IV μήκους 48 bit που να υποστηρίζεται στις παρούσες υλοποιήσεις υλισμικού και επιπλέον αποτρέπει τη χρήση μιας γνωστής κατηγορίας αδύναμων κλειδιών. Οι τροποποιήσεις, όσον αφορά το IV, παρέχουν πολύ σημαντική επιπρόσθετη ασφάλεια σε σύγκριση με το WEP.

Λεπτομέρειες υλοποίησης του TKIP

Στην ενότητα αυτή, περιγράφεται λεπτομερέστερα ο τρόπος υλοποίησης του αλγορίθμου TKIP. Αρχικά, υποθέτουμε ότι τα κύρια κλειδιά (master keys) έχουν διανεμηθεί, ενώ τα αντίστοιχα κλειδιά συνόδου (session keys) έχουν παραχθεί και στις δύο πλευρές της επικοινωνιακής ζεύξης. Τα κύρια κλειδιά μπορεί ενδεχομένως να έχουν αποκτηθεί χρησιμοποιώντας κάποια από τις μεθόδους επαλήθευσης ταυτότητας των ανωτέρων στρωμάτων που βασίζεται στο EAP, ή εναλλακτικά να αποτελούν προμεριζόμενα (preshared) κλειδιά. Η τελευταία περίπτωση είναι ανάλογη με την προσέγγιση του WEP, όπου τα κλειδιά προ-εγκαθίστανται στις διάφορες συσκευές. Προφανώς, κάτι τέτοιο μπορεί να βρει εφαρμογή μόνο σε δίκτυα περιορισμένων διαστάσεων ή κατά τη λειτουργία τύπου ad-hoc.

Στα πλαίσια του TKIP παράγονται τρεις τύποι κλειδιών:

1. Κλειδί για την προστασία της ανταλλαγής μηνυμάτων `EAPOL-Key`.
2. Κλειδί-Ζεύγος (pairwise) για την προστασία των ίδιων των μηνυμάτων με χρήση TKIP.
3. Ομαδικό κλειδί για την προστασία εκπομπών (broadcasts) που χρησιμοποιούν TKIP.

Από τα δεδομένα του κλειδιού-ζεύγους παράγονται τα χρονικά κλειδιά:

- Κλειδί Χρονικής Κρυπτογράφησης (128 bit): Αυτό χρησιμοποιείται ως είσοδος στο στάδιο αλλαγής κλειδιών πριν την κρυπτογράφηση RC4.
- Κλειδί Χρονικού Επαληθευτή Ταυτότητας TX MIC: Αυτό χρησιμοποιείται σε συνδυασμό με τη μέθοδο επαλήθευσης ταυτότητας Μιχάλης για την παραγωγή του MIC στα πλαίσια που μεταδίδονται από τον επαληθευτή ταυτότητας (σημείο πρόσβασης σε ένα δίκτυο υποδομής).

- Κλειδί Χρονικού Επαληθευτή Ταυτότητας RX MIC: Αυτό χρησιμοποιείται σε συνδυασμό με τη μέθοδο Μιχάλης για την παραγωγή του MIC στα πλαίσια που μεταδίδονται από την οντότητα του supplicant (συνήθως αυτή είναι η κινητή συσκευή).

Όσον αφορά τα ομαδικά κλειδιά, μόνο οι δύο πρώτοι τύποι χρειάζεται να παραχθούν, καθώς οι εκπομπές (broadcasts) στέλνονται αποκλειστικά από τον επαληθευτή ταυτότητας και όχι από την οντότητα του supplicant.

Ο στόχος του TKIP είναι, όπως έχει αναφερθεί, η παροχή μηχανισμών ασφαλείας, αφενός για την εξασφάλιση της ακεραιότητας των λαμβανομένων δεδομένων, αφετέρου για την προστασία των δεδομένων που αποστέλλονται. Στα πλαίσια αυτά, το πρωτόκολλο υλοποιεί τα ακόλουθά:

- Παραγωγή και έλεγχος IV
- Παραγωγή και έλεγχος MIC
- Κρυπτογράφηση και αποκρυπτογράφηση

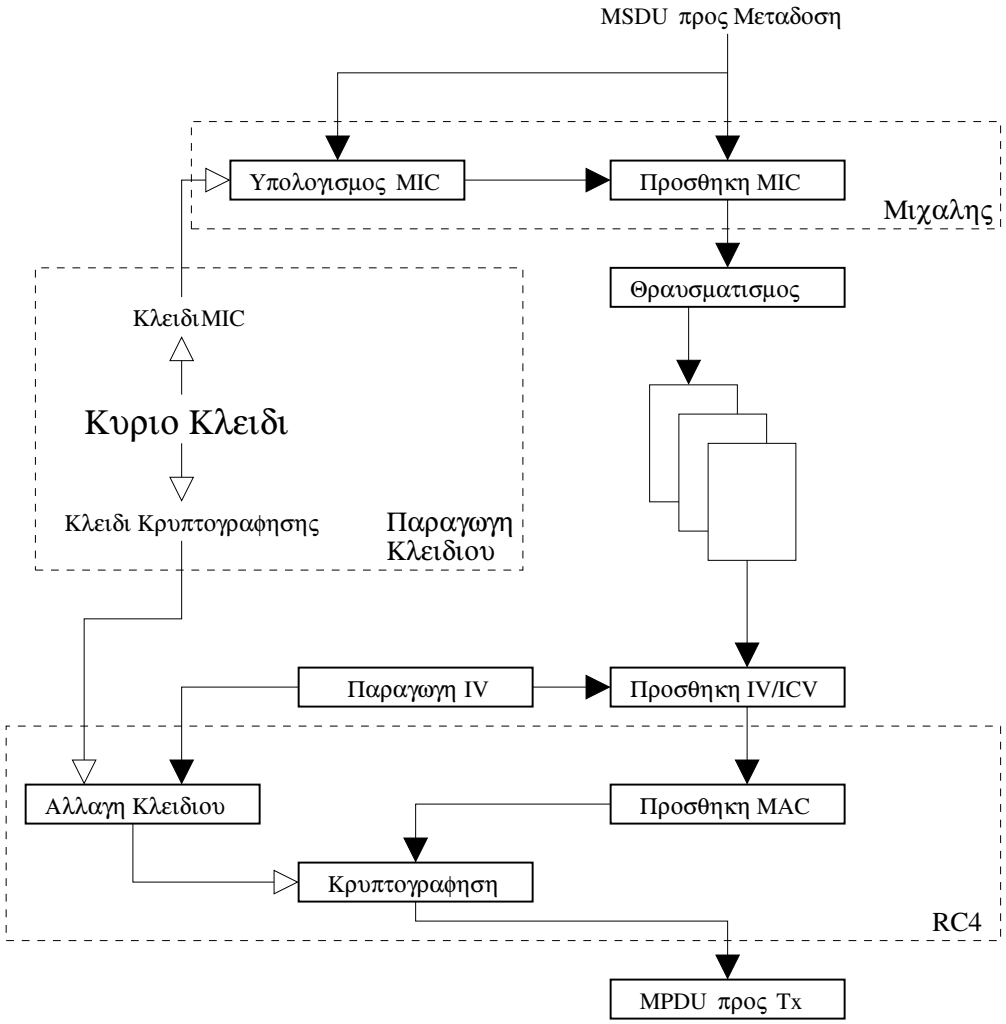
Η λειτουργία του TKIP κατά τη μετάδοση δεδομένων, φαίνεται στο Σχ. 4.2. Οι τέσσερις διεργασίες που χρησιμοποιεί το πρωτόκολλο είναι οι ακόλουθες:

1. Μιχάλης
2. Παραγωγή κλειδιών
3. IV/TSC
4. RC4

Ας σημειωθεί ότι η τιμή ελέγχου ακεραιότητας υπολογίζεται βάσει της MSDU και προστίθεται σε αυτή, πριν το θραυσματισμό. Ως αποτέλεσμα, τα byte της τιμής ελέγχου είναι παρόντα μόνο στην τελευταία MPDU και περιέχονται στα κρυπτογραφημένα δεδομένα. Η αρχική τιμή ελέγχου (του WEP), το ICV, εξακολουθεί να υπολογίζεται και να προστίθεται σε κάθε MPDU, παρόλο που δεν αποτελεί μέρος του ελέγχου ακεραιότητας πακέτων του TKIP.

Καθώς το MIC υπολογίζεται στο επίπεδο της MSDU, δεν είναι δυνατό να συμπεριληφθεί η τιμή του IV στον υπολογισμό του MIC για δύο λόγους. Κατ΄ αρχήν, επειδή η MSDU μπορεί να είναι θραυσματισμένη, ενδέχεται να χρησιμοποιούνται πολλαπλές τιμές του IV για την αποστολή των θραυσμάτων της MSDU. Επιπλεόν, δεν επιτρέπεται η επιλογή της τιμής του IV, παρά μόνο μετά την αφαίρεση του θραύσματος από τις ουρές μετάδοσης. Στο μέλλον, προκειμένου να υποστηριχθούν πολυμεσικές εφαρμογές, το πρότυπο IEEE 802.11e μπορεί να διαθέτει μέχρι και οκτώ ουρές προτεραιότητας για τα εξερχόμενα πλαίσια και η σειρά με την οποία τα θραύσματα επιλέγονται για μετάδοση εξαρτάται από πολλούς παράγοντες που καθορίζονται από περιορισμούς πραγματικού χρόνου και προτεραιοτήτες. Ως εκ τούτου, οι MSDU υψηλότερης προτεραιότητας ενδέχεται να προηγηθούν παλαιότερων MSDU ή ακόμη και να σταλούν μεταξύ θραυσμάτων των τελευταίων. Το TKIP διαθέται ένα μόνο μετρητή IV ανά ζεύξη -όχι ανά ουρά- και επομένως η ανάθεση της τιμής του IV πρέπει να περιμένει μέχρι την τελευταία στιγμή, δηλαδή πριν την επιλογή ενός θραύσματος για μετάδοση. Συνεπώς, η τιμή δε μπορεί να είναι γνωστή κατά τον υπολογισμό του MIC.

Ο υπολογισμός του MIC στο επίπεδο της MSDU, σε συνυδασμό με την έλλειψη προστασίας του IV, επιτρέπει σε ένα επίδοξο εισβολέα να «μπλοκάρει» ένα σταθμό αναπαράγοντας προηγούμενα πλαίσια με νέα τιμή του IV. Το πρόβλημα ανακύπτει καθώς το IV διπλασιάζεται,

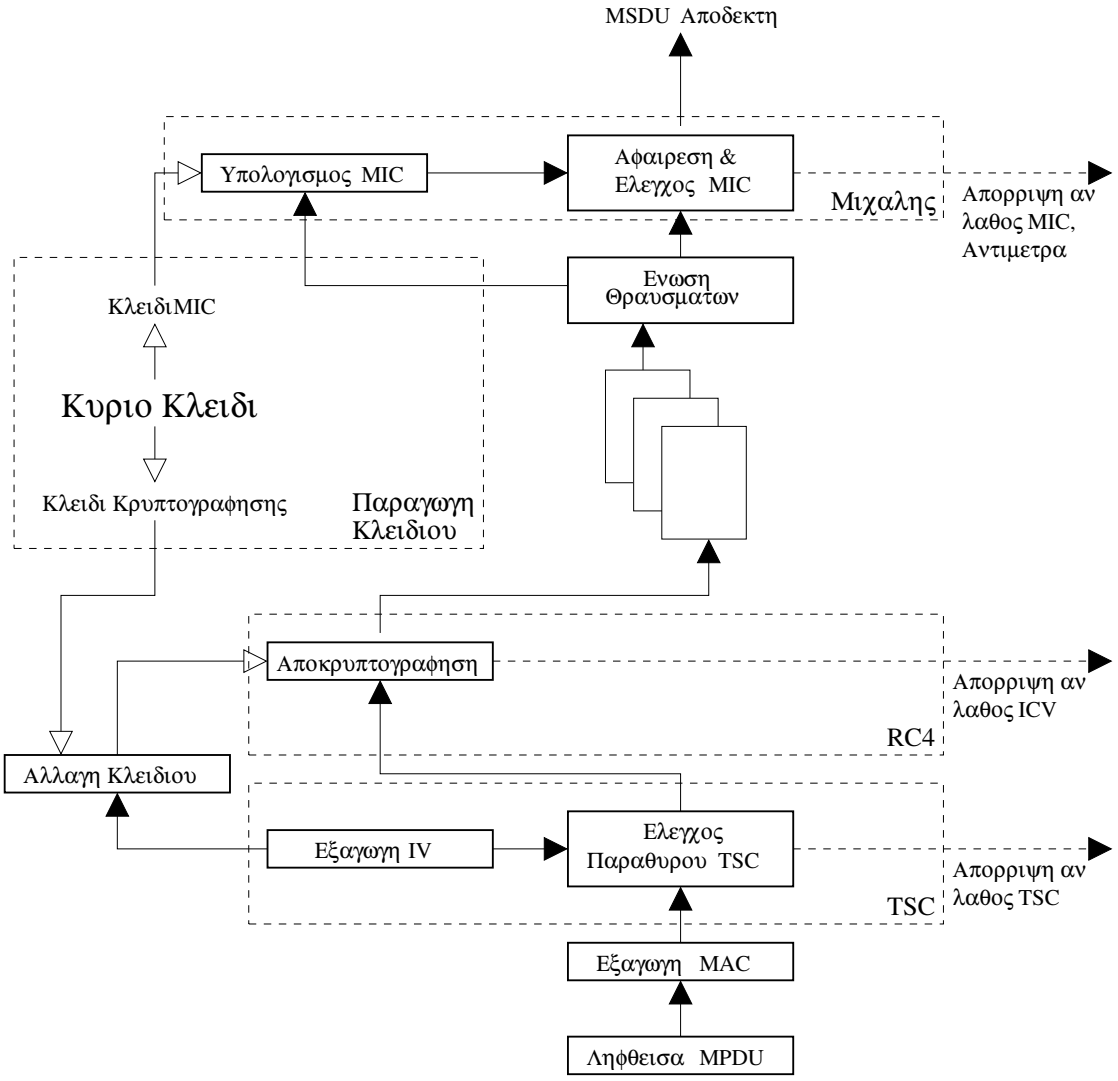


Σχήμα 4.2: Λειτουργία του TKIP κατά τη Μετάδοση

όπως ο μετρητής ακολουθίας TSC, προκειμένου να αποφευχθούν οι επιθέσεις αναπραγωγής. Προφανώς, θα αποτύχει η αποκρυπτογράφηση τέτοιων ψευδεπίγραφων πλαισίων, τα οποία και θα απορριφθούν. Δεν αποτελούν απειλή ως προς την ακεραιότητα του πρωτοκόλλου, ωστόσο καθιστούν τα έγκυρά πλαίσια που ακολουθούν να φαίνονται σαν επίθεση αναπαραγωγής. Όταν ένα έγκυρο πλαίσιο καταφθάνει, ενδέχεται να απορριφθεί επειδή η τιμή του TSC έχει εξαντληθεί από τον επιτιθέμενο σταθμό. Συνεπώς, ανήκει στην κατηγορία των επιθέσεων άρνησης - υπηρεσιών (denial-of-service). Στον κόσμο των ασύρματων επικοινωνιών υπάρχουν πολλοί απλοί τρόποι που επιτυγχάνουν ακριβώς αυτό και δεν είναι δυνατό να αντιμετωπισθούν αποτελώντας μόνιμα μια πιθανή απειλή.

Υποτίθεται ότι το τμήμα Κρυπτογράφησης, όπως παρουσιάζεται στο Σχ. 4.2, υλοποιεί τον ίδιο αλγόριθμο κρυπτογράφησης RC4 που χρησιμοποιείται και στο WEP. Οι περισσότεροι κατασκευαστές έχουν υλοποιήσει το τμήμα αυτό με τέτοιο τρόπο ώστε να μην είναι δυνατή η τροποποίησή του μέσω αναβαθμίσεων firmware. Ο υπάρχων εξοπλισμός WEP συχνά περιλαμβάνει μηχανισμό υλισμικού (hardware) για την αρχικοποίηση του κουτιού-S του RC4. Η αδυναμία αλλαγής της συγκεκριμένης μονάδας, αποτέλεσε και το μεγαλύτερο πρόβλημα κατά το σχεδιασμό του TKIP.

Η αντίστοιχη λειτουργία του πρωτοκόλλου κατά τη λήψη, παρουσιάζεται στο Σχ. 4.3.



Σχήμα 4.3: Λειτουργία του TKIP κατά τη Λήψη

Η διαδικασία λήψης δεν είναι η ακριβώς αντίστροφη αυτής της μετάδοσης. Κατ΄ αρχήν, η αποκρυπτογράφηση δεν είναι η πρώτη λειτουργία. Αντίθετα, ο TSC (που προκύπτει από το IV) ελέγχεται για την προστασία από αναπαραγωγές. Ας σημειωθεί ότι η τιμή του ICV ελέγχεται και χρησιμοποιείται για την απόρριψη του πακέτου. Δεν πρόκειται αυστηρά για έναν έλεγχο ακεραιότητας , ωστόσο αποτελεί μια γρήγορη ένδειξη για την επιτυχία ή μη της αποκρυπτογράφησης: Η αποκρυπτογράφηση ενός πακέτου με λανθασμένο κλειδί ή με χρήση μη έγκυρων τιμών του IV παράγει πάντα λανθασμένη τιμή του ICV.

Το MIC ελέγχεται μετά τη λήψη όλων των θραυσμάτων και τη σύνθεσή τους στην MSDU. Ας σημειωθεί ότι αν το MIC αποτύχει, δε θα απορριφθεί μόνο η MSDU, αλλά, επιπλέον, ενδέχεται να ενεργοποιηθούν αντιμέτρα. Αν και θεωρητικά δυνατό, είναι εξαιρετικά απίθανο να υπάρξουν λάθη κάτα τη μετάδοση, τέτοια που να επιτρέψουν σε ένα πλαίσιο να περάσει τον έλεγχο CRC και στη συνέχεια να αποκρυπτογραφηθεί για να παράγει ένα αποδεκτό ICV. Σε περίπτωση αποτυχίας του MIC , είναι σίγουρο ότι έχει προηγηθεί σκόπιμη τροποποίηση και όχι τυχαία σφάλματα μετάδοσης ή παρεμβολές.

4.2 Δίκτυο Εύρωστης Ασφαλείας

Στην ενότητα αυτή παρουσιάζουμε το Πρότυπο Προηγμένης Κρυπτογράφησης (Advanced Encryption Standard – AES) και το Πρωτόκολλο Counter-Mode/CBC-MAC (CCMP), που αποτελούν κύρια συστατικά του Δικτύου Εύρωστης Ασφαλείας (RSN), στα πλαίσια του προτύπου IEEE 802.11i. Θα αναδειχθούν οι λόγοι που επιλέχτηκε το AES και τα χαρακτηριστικά του ως αλγορίθμου κρυπτογράφησης. Τα συστήματα ασφαλείας κάνουν χρήση του AES σε συνδυασμό με διάφορους τρόπους λειτουργίας (operating modes), οι σημαντικότεροι των οποίων, περιγράφονται παρακάτω. Στην ενότητα 4.2.2, παρουσιάζουμε το CCMP και πιο συγκεκριμένα τη λειτουργικότητά του στο πρωτόκολλο IEEE 802.11.

4.2.1 Πρότυπο Προηγμένης Κρυπτογράφησης

Το AES είναι ένας κρυπτογραφικός μηχανισμός, ο οποίος, με τη βοήθεια μαθηματικών και λογικών πράξεων, συνδυάζει ένα κλειδί και ένα τμήμα δεδομένων (μη κρυπτογραφημένων) μήκους 128 bit, προκειμένου να παράγει ένα κρυπτογραφημένο τμήμα. Πρακτικά, είναι αδύνατο να πραγματοποιηθεί ο ίδιος μετασχηματισμός, εφόσον δεν είναι γνωστό το κλειδί. Το AES είναι αντιστρέψιμος αλγόριθμος, με την έννοια ότι είναι δυνατή η μετατροπή των κρυπτογραφημένων δεδομένων στα αρχικά με αποκρυπτογράφηση, ιδιότητα που είναι χρήσημη αλλά δε συναντάται στο σύνολο των πρωτοκόλλων ασφαλείας. Τα κρυπτογραφημένα και μη τμήματα δεδομένων είναι του ίδιου ακριβώς μεγέθους. Το AES πραγματοποιεί την κρυπτογράφηση με ιδιαίτερα αποδοτικό και ασφαλή τρόπο και θεωρείται απίθανο να ανακαλυφθεί κάποια θεμελιώδης αδυναμία του στο άμεσο μέλλον.

Το AES βασίζεται στον αλγόριθμο Rijndael [5], [6], που σχεδίασαν οι Joan Daeman και Vincent Rijmen. Ο αλγόριθμος αυτός επιτρέπει συγκεκριμένα μεγέθη τμημάτων και κλειδιών: 128, 192 ή 256 bit για το καθένα. Εν τούτοις, το IEEE 802.11i έχει υιοθετήσει το μήκος των 128 bit, τόσο για το τμήμα όσο και το κλειδί, απλοποιώντας με τον τρόπο αυτό την υλοποίηση.

Επειδή τα δεδομένα στα Ασύρματα Τοπικά Δίκτυα μεταδίδονται σε πλαίσια μεταβλητού μήκους -συνήθως από 512 έως 12000 bit ανά πλαίσιο-, προκειμένου να χρησιμοποιηθεί ένα αλγόριθμος κρυπτογράφησης τμημάτων σταθερού μεγέθους, όπως το AES, χρειάζεται ένας τρόπος μετατροπής ενός μηνύματος τυχαίου μήκους σε μια ακολουθία τμημάτων σταθερού μήκους πριν την κρυπτογράφηση. Προφανώς, η μέθοδος αυτή πρέπει να επιτρέπει την επανασύνθεση της ακολουθίας των τμημάτων στο αρχικό μήνυμα κατά την αποκρυπτογράφηση. Η μέθοδος μετασχηματισμού των μηνυμάτων σε τμήματα και αντίστροφα, ονομάζεται τρόπος λειτουργίας του κρυπτογραφημένου τμήματος.

Υπάρχει πλήθος διαφορετικών τρόπων λειτουργίας που υποστηρίζει το AES, ενώ το CCMP χρησιμοποιεί το CCM, που βασίζεται στον Τρόπο Μετρητή (Counter-Mode). Πριν προχωρήσουμε στην παρουσίαση των τρόπων λειτουργίας, θα σχολιάσουμε την αυθεντικότητα των μηνυμάτων στα πλαίσια του AES. Το τελευταίο παρέχει μια μέθοδο κρυπτογράφησης των δεδομένων ώστε να προστατεύονται από επίδοξους εισβολείς. Ωστόσο, είναι εξίσου σημαντικό για τον παραλήπτη να εξασφαλίζεται η αυθεντικότητα του μηνύματος, ότι, δηλαδή, δεν έχει τροποποιηθεί. Αυτό επιτυγχάνεται με την προσθήκη του κώδικα αυθεντικότητας μηνύματος (Message Authenticity Code – MAC). Για περισσότερη αποτελεσματικότητα, το MAC υπολογίζεται με τη βοήθεια του AES. Έτσι, γίνεται αντιληπτό ότι ο τρόπος λειτουργίας πρέπει να παρέχει κατάλληλους μηχανισμούς, τόσο για κρυπτογράφηση όσο και για αυθεντικότητα.

Ακολουθεί περιγραφή μερικών από τους σημαντικότερους τρόπους λειτουργίας, αρχίζοντας από το Βιβλίο Ηλεκτρονικού Κώδικα (Electronic Code Book – ECB) που είναι και ο

απλούστερος.

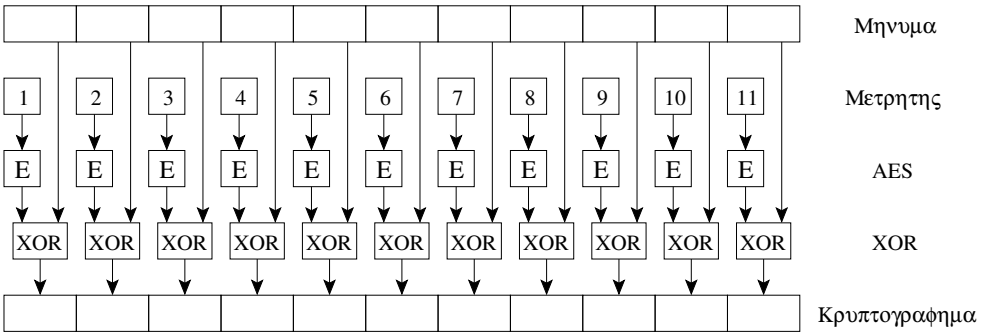
Βιβλίο Ηλεκτρονικού Κώδικα

Στα πλαίσια του τρόπου λειτουργίας του AES κατά το Βιβλίο Ηλεκτρονικού Κώδικα (Electronic Code Book – ECB), λαμβάνονται τμήματα από τα δεδομένα εισόδου και κρυπτογραφούνται ένα καθε φορά ακολουθιακά, χρησιμοποιώντας το ίδιο κλειδί, μέχρι να εξαντληθεί το σύνολο των τμημάτων [2], [7].

Η προσέγγιση αυτή είναι απλή, ωστόσο παρουσιάζει ορισμένα προβλήματα. Κατ΄ αρχήν, υπάρχει περίπτωση το μήνυμα να μην είναι ακριβές πολλαπλάσιο του μεγέθους τμήματος, επομένως χρειάζεται να συμπληρωθεί κατάλληλα το τελευταίο τμήμα και να καταγραφεί το πραγματικό του μήκος, ώστε να είναι διαθέσιμο στη φάση της αποκρυπτογράφησης. Επιπρόσθετα, ανακύπτει και ένα πρόβλημα ασφαλείας: αν δύο ή περισσότερα τμήματα περιέχουν τα ίδια δεδομένα, τότε θα παράγουν πανομοιότυπα κρυπτογραφημένα δεδομένα. Εξαιτίας της αδυναμίας αυτής το ECB δε χρησιμοποιείται συνήθως στα πλαίσια του AES.

Τρόπος Μετρητή

Ο Τρόπος Μετρητή είναι πιο περίπλοκος από το ECB και παρουσιάζει σημαντικές διαφορές. Δε χρησιμοποιεί απ΄ ευθείας το AES για την κρυπτογράφηση των δεδομένων. Αντίθετα, κρυπτογραφεί μια τυχαία τιμή που ονομάζεται μετρητής και στη συνέχεια συνδυάζει το αποτέλεσμα και τα δεδομένα εισόδου με τη λογική πράξη XOR, ώστε να παράγει το τελικό κρυπτογράφημα. Γενικά, ο μετρητής αυξάνεται κατά ένα για κάθε διαδοχικό τμήμα.



Σχήμα 4.4: Παράδειγμα AES με Τρόπο Μετρητή

Η διαδικασία αυτή φαίνεται στο Σχ. 4.4, όπου ο μετρητής ξεκινά από την τιμή 1 και αυξάνεται μέχρι το 11. Στην πράξη όμως, ο μετρητής αρχίζει από τυχαία τιμή και αυξάνεται με τυχαία βήματα που ακολουθούν κάποιο σχήμα. Προφανώς, είναι σημαντικό η πλευρά του παραλήπτη, που θα πραγματοποιήσει την αποκρυπτογράφηση, να γνωρίζει την αρχική τιμή και τους κανόνες που ακολουθούν οι διαδοχικές αυξήσεις της. Η διαφορετική τιμή του μετρητή για κάθε τμήμα εξαλείφει το πρόβλημα με τα επαναλμβανόμενα τμήματα που περιγράψαμε στο ECB.

Ο Τρόπος Μετρητή παρουσιάζει ορισμένες ενδιαφέρουσες ιδιότητες. Η αποκρυπτογράφηση είναι η ίδια ακριβώς διαδικασία με την κρυπτογράφηση, καθώς η εφαρμογή του λογικού XOR δύο φορές έχει ως αποτέλεσμα τα αρχικά δεδομένα. Αυτό σημαίνει ότι απαιτείται μόνο η υλοποίηση του μηχανισμού της κρυπτογράφησης. Μια επιπλέον σημαντική ιδιότητα είναι ότι η κρυπτογράφηση μπορεί να γίνει παράλληλα. Εφόσον όλες οι τιμές του μετρητή είναι γνωστές από την αρχή, μια διάταξη από μηχανισμούς AES είναι σε θέση να κρυπτογραφήσει

ένα μήνυμα με μία μόνο παράλληλη επεξεργασία. Αυτό δεν ισχύει για τους περισσότερους από τους άλλους τρόπους λειτουργίας του AES. Μια τελευταία χρήσιμη ιδιότητα είναι ότι δεν είναι απαραίτητο το αρχικό μήνυμα να είναι ακριβές πολλαπλάσιο του μεγέθους τμήματος. Απλά, το τελευταίο μη πλήρες τμήμα συνδυάζεται με XOR με τον αριθμό των απαιτούμενων bit από την κρυπτογραφημένη τιμή του μετρητή. Έτσι, προκύπτει κρυπτογράφημα μήκους ίσου με αυτό του αρχικού μηνύματος.

Ο Τρόπος Μετρητή χρησιμοποιείται για περισσότερο από δύο δεκαετίες και θεωρείται ασφαλής από την κρυπτογραφική κοινότητα. Ωστόσο, στη βασική του μορφή δεν παρέχει μηχανισμούς αυθεντικότητας μηνυμάτων, παρά μόνο κρυπτογράφησης. Ως αποτέλεσμα, στα πλαίσια του RSN, απαιτείται η προσθήκη επιπρόσθετων δυνατοτήτων.

Τρόπος Μετρητή/Αλύσωση Τμημάτων Κρυπτογραφημάτων-MAC: CCM

Το CCM αναπτύχθηκε αποκλειστικά για το RSN, από τρεις κρυπτογράφους που συμμετείχαν στο σχεδιασμό του IEEE 802.11i: Doug Whiting, Russ Housley και Niels Ferguson. Συνδυάζει τον Τρόπο Μετρητή με μια μέθοδο επαλήθευσης ταυτότητας μηνυμάτων που ονομάζεται Αλύσωση Τμημάτων Κρυπτογραφημάτων (Cipher Block Chaining – CBC). Το CBC χρησιμοποιείται για την παραγωγή του κώδικα ακεραιότητας μηνυμάτων MAC, εξού και η ονομασία CBC-MAC [8].

Η προσέγγιση που χρησιμοποιεί το CBC είναι απλή και έχει ως εξής:

1. Λήψη του πρώτου τμήματος του μηνύματος και κρυπτογράφηση αυτού με τη βοήθεια του AES.

2. Εφαρμογή XOR στο αποτέλεσμα σε συνδυασμό με το δεύτερο τμήμα και κρυπτογράφηση του νέου αποτελέσματος.

3. Εφαρμογή XOR στο αποτέλεσμα σε συνδυασμό με το επόμενο τμήμα και κρυπτογράφηση του νέου αποτελέσματος κ.ο.κ.

Το αποτέλεσμα που προκύπτει από την παραπάνω διαδικασία είναι ένα μόνο τμήμα, μήκους 128 bit, που συνδυάζει όλα τα δεδομένα του μηνύματος. Στην περίπτωση που μεταβαλόταν η τιμή ενός ή περισσότερων bit του μηνύματος, το αποτέλεσμα θα ήταν τελείως διαφορετικό. Το CBC-MAC είναι απλό, ωστόσο δεν είναι δυνατό να χρησιμοποιήσει παράλληλη επεξεργασία· οι λειτουργίες κρυπτογράφησης πρέπει να γίνονται ακολουθιακά. Επιπρόσθετα, ας σημειωθεί ότι το CBC-MAC μπορεί να εφαρμοστεί αποκλειστικά σε μηνύματα μεγέθους πολλαπλάσιου του μήκους τμήματος. Το CCMP προσφέρει λύση στο σχετικό πρόβλημα (περιγράφεται στην ενότητα 4.2.2), εν τούτοις, ορισμένοι κρυπτογράφοι έχουν ενστάσεις για την προσέγγιση που υιοθετείται.

Το CCM συνδυάζει δύο γνωστές μεθόδους: τον Τρόπο Μετρητή και το CBC-MAC. Προσθέτει, επιπλέον, ορισμένα χαρακτηριστικά που είναι χρήσιμα για συγκεκριμένες εφαρμογές, όπως είναι το RSN. Αυτά είναι:

- Ορισμός μιας τυχαίας τιμής προκειμένου να διαχωρίζονται κρυπτογραφικά τα διαδοχικά μηνύματα.

- Συνδυασμός της κρυπτογράφησης με την ακεραιότητα μηνυμάτων με χρήση ενός κοινού κλειδιού.

- Επέκταση της επαλήθευσης ταυτότητας ώστε να καλύπτεται η περίπτωση δεδομένων που δεν πρόκειται να κρυπτογραφηθούν.

Αξίζει να αναλυθεί το τελευταίο χαρακτηριστικό, καθώς είναι ιδιαίτερα σημαντικό για το RSN. Στην πλειοψηφεία των υπάρχουσων μεθόδων που πραγματοποιούν κρυπτογράφηση σε συνδυασμό με πιστοποίηση αυθεντικότητας, γίνεται η υπόθεση ότι θα κρυπτογραφηθεί ολόκληρο το μήνυμα. Ωστόσο, στα πλαίσια του IEEE 802.11i, μόνο ένα μέρος του μηνύματος απαιτείται να κρυπτογραφηθεί. Το τμήμα επικεφαλίδας του πλαισίου IEEE 802.11 περιέχει τις διευθύνσεις MAC, καθώς επίσης, και άλλες πληροφορίες σχετικές με τη λειτουργία του Ασύρματου Τοπικού Δικτύου. Τα πεδία αυτά πρέπει να μεταδίδονται σε μη κρυπτογραφημένη μορφή, προκειμένου να είναι εφικτή η επικοινωνία με άλλες ασύρματες συσκευές. Ως εκ τούτου, μόνο το τμήμα δεδομένων του πλαισίου κρυπτογραφείται. Ωστόσο, παρόλο που η επικεφαλίδα δεν κρυπτογραφείται, είναι αναγκαίο να εξασφαλίζεται στον παραλήπτη ότι αυτή δεν έχει τροποποιηθεί κατά τη μετάδοση. Για το σκοπό αυτό, το CCM επιτρέπει την κρυπτογράφηση ενός μόνο τμήματος του μηνύματος, του οποίου η αυθεντικότητα πιστοποιείται από το CBC-MAC.

Γενικά, η χρήση του ίδιου κλειδιού για δύο ξεχωριστές κρυπτογραφικές λειτουργίες δε συνίσταται. Στην προκειμένη περίπτωση, το ίδιο κλειδί χρησιμοποιείται τόσο για την κρυπτογράφηση, όσο και την πιστοποίηση αυθεντικότητας. Ωστόσο, παρόλο που το κλειδί είναι το ίδιο, χρησιμοποείται κάθε φορά σε συνδυασμό με κάποιο διάνυσμα αρχικοποίησης (Initialization Vector – IV). Η κατασκευή του IV είναι διαφορετική για τη διαδικασία του Τρόπου Μετρητή και του CBC-MAC, παράγοντας, τελικά, δύο διαφορετικά κλειδιά. Η αποτελεσματικότητα του διαχωρισμού αυτού έχει αποδειχθεί [9].

4.2.2 CCMP

Στην ενότητα αυτή, περιγράφεται ο τρόπος με τον οποίο τα πλαίσια του IEEE 802.11 κρυπτογραφούνται με τη βοήθεια του CCMP. Το πρώτο που πρέπει να τονιστεί είναι ότι το CCMP πραγματοποιεί την κρυπτογράφηση στο επίπεδο της MPDU. Στο Σχ. 4.5, φαίνεται η ροή των δεδομένων από το επίπεδο MSDU στο MPDU, και τελικά στη ραδιοζεύξη.

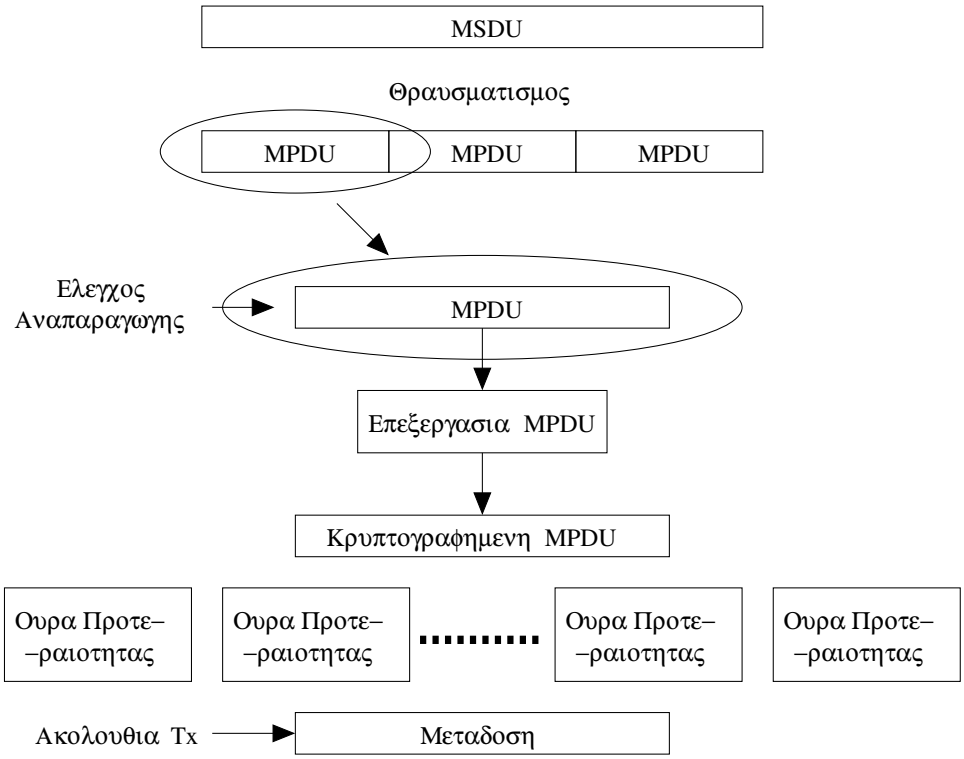
Τα δεδομένα καταφθάνουν ως μία MSDU και ενδέχεται να διασπαστούν σε θραύσματα. Κάθε θραύσμα μετασχήματίζεται σε MPDU, στην οποία ανατίθεται η κατάλληλη επικεφαλίδα 802.11 που περιέχει τις διευθύνσεις αποστολέα και παραλήπτη, καθώς και άλλες πληροφορίες. Στο σημείο αυτό, κάθε MPDU υφίσταται επεξεργασία στα πλαίσια του αλγορίθμου του CCMP, προκειμένου για την παραγωγή μιας νέας κρυπτογραφημένης MPDU. Μόνο το τμήμα δεδομένων κρυπτογραφείται, όχι η επικεφαλίδα. Εν τούτοις, το CCMP δεν περιορίζεται στην κρυπτογράφηση μόνο τμημάτων της MPDU. Επιπρόσθετα, προσθέτει νέα πεδία, με αποτέλεσμα η κρυπτογραφημένη MPDU που προκύπτει να είναι 16 byte μεγαλύτερη από την αρχική.

Τα βήματα που ακολουθεί το CCMP κατά την κρυπτογράφηση, φαίνονται στο Σχ. 4.6 και έχουν ως εξής:

1. Αρχικά η MPDU είναι μη κρυπτογραφημένη, συμπληρωμένη με την επικεφαλίδα MAC του IEEE 802.11. Η επικεφαλίδα αυτή περιλαμβάνει τη διεύθυνση αποστολέα και παραλήπτη, ενώ οι τιμές ορισμένων πεδίων δεν είναι ακόμη γνωστές και τίθενται προσωρινά στο 0.

2. Η επικεφαλίδα MAC αποσυνδέεται από την MPDU. Στο στάδιο αυτό, παράγεται η επικεφαλίδα CCMP, μήκους 8 byte, ώστε να συμπεριληφθεί αργότερα στην MPDU.

3. Η τιμή του MIC υπολογίζεται προκειμένου να προστατευτεί η επικεφαλίδα CCMP, τα δεδομένα και τμήματα της επικεφαλίδας IEEE 802.11. Το MIC προστίθεται στο πεδίο των δεδομένων.



Σχήμα 4.5: Λειτουργία του CCMP κατά τη Μετάδοση

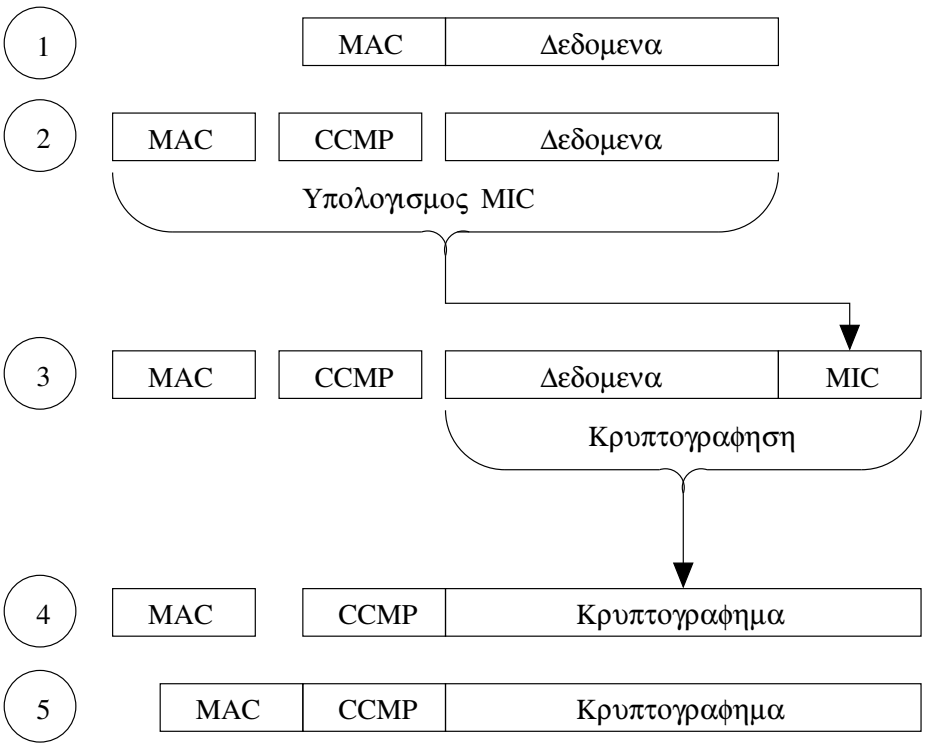
4. Τα δεδομένα σε συνδυασμό με την τιμή του MIC κρυπτογραφούνται. Μετά την κρυπτογράφηση, η επικεφαλίδα CCMP τοποθετείται μπροστά από το πεδίο δεδομένων.

5. Τελικά, η επικεφαλίδα MAC αποκαθίσταται μπροστά από τη νέα MPDU, η οποία μπορεί, πλέον, να τοποθετηθεί στην ουρά για μετάδοση. Η λογική της μετάδοσης είναι ανεξάρτητη από την επικεφαλίδα CCMP. Από το σημείο αυτό έως και τη μετάδοση, μόνο η επικεφαλίδα MAC θα ενημερωθεί.

Οι κρυπτογραφημένες MPDU τοποθετούνται σε μια ουρά πριν τη μετάδοση. Ενδέχεται να υπάρχουν και περισσότερες από μία ουρές, που να λειτουργούν βάσει κάποιας πολιτικής προτεραιοτήτων. Αυτό επιτρέπει μελλοντική επέκταση, τέτοια, ώστε να υποστηρίζονται διαφορετικές κλάσεις κίνησης, στα πλαίσια του IEEE 802.11e. Ακριβώς πριν από τη μετάδοση, κάποια από τα πεδία της επικεφαλίδας IEEE 802.11 ενημερώνονται κατάλληλα προκειμένου να συμμορφωθούν στους κανόνες μετάδοσης. Τα πεδία που υπόκεινται σε τέτοιες αλλαγές, εξαιρούνται κατά τον υπολογισμό της τιμής του MIC.

4.3 Σύγκριση TSN, RSN και WEP

Τα TSN και RSN διαθέτουν μια κοινή αρχιτεκτονική και προσέγγιση. Το TSN παρουσιάζει ένα υποσύνολο των δυνατοτήτων που εστιάζονται σε ένα μοναδικό τρόπο υλοποίησης του δικτύου, ενώ το RSN επιτρέπει περισσότερη ευελιξία στην υλοποίηση. Το RSN υποστηρίζει, εκτός του TKIP, και τον αλγόριθμο κρυπτογράφησης AES (Advanced Encryption Standard), σε αντίθεση με το TSN που εστιάζεται στο TKIP. Καθώς το WEP χρησιμοποιείται στα περισσότερα σύγχρονα εταιρικά δίκτυα, η πιο φυσική προσέγγιση είναι η υλοποίηση του



Σχήμα 4.6: Επεξεργασία MPDU στο CCMP

TSN σε πρώτη φάση και η σταδιακή μετάβαση στο RSN με την αναβάθμιση (αντικατάσταση) των υπάρχοντων συστημάτων ασύρματης δικτύωσης. Η τάση αυτή θα οδηγήσει σε δίκτυα βασισμένα πλήρως στο πρότυπο ασφαλείας IEEE 802.11i.

Τα TSN και RSN μοιράζονται την ίδια αρχιτεκτονική ασφαλείας, υπό την οποία λειτουργούν τα πρωτόκολλα ασφαλείας TKIP και AES, αντίστοιχα. Η αρχιτεκτονική αυτή, περιλαμβάνει διαδικασίες, όπως η επαλήθευση ταυτότητας ανωτέρου επιπέδου, η διάθεση μυστικών κλειδιών και η ανανέωση κλειδιών. Η αρχιτεκτονική του RSN είναι αρκετά διαφορετική από εκείνη του WEP και περισσότερο πολύπλοκη. Ωστόσο, παρέχει μια λύση που είναι τόσο ασφαλής, όσο και κλιμακοθετήσιμη (scalable) για χρήση σε δίκτυα μεγάλων διαστάσεων. Άλλωστε, ένα από τα μεγάλα προβλήματα του WEP ήταν η δυσκολία διαχείρισης όσον αφορά τη διάθεση των κλειδιών για μεγάλο αριθμό χρηστών. Το πρόβλημα αυτό έχει αντιμετωπιστεί τόσο από το TSN, όσο και από το RSN.

Κεφάλαιο 5

Έλεγχος πρόσβασης

Το κεφάλαιο αυτό, παρουσιάζει τους διάφορους μηχανισμούς ελέγχου πρόσβασης που προτείνονται στα πλαίσια του προτύπου ασφαλείας 802.11i. Πιο συγκεκριμένα, περιγράφονται τα πρωτόκολλα IEEE 802.1X, Πρωτόκολλο Επεκτάσιμης Επαλήθευσης Ταυτότητας και Remote Access Dial-In User Service, ενώ παρουσιάζεται η αλληλεπίδρασή τους για την παροχή ελέγχου πρόσβασης στα Ασύρματα Τοπικά Δίκτυα.

5.1 Πρωτόκολλο IEEE 802.1X

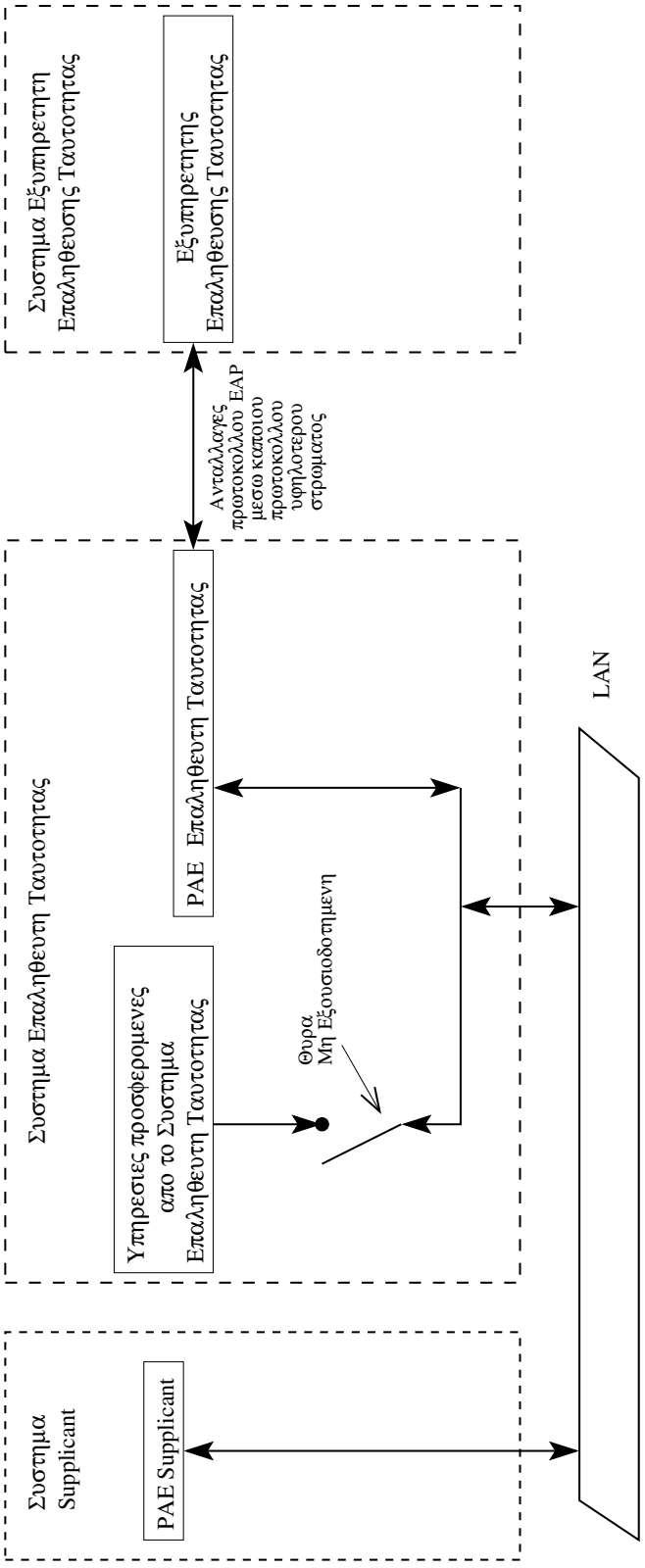
Το πρωτόκολλο IEEE 802.1X παρέχει μηχανισμούς ελέγχου πρόσβασης στο σημείο εισόδου ενός χρήστη στο δίκτυο. Ενέχει τρεις οντότητες:

- *Supplicant*, επιθυμεί να αποκτήσει πρόσβαση στο δίκτυο
- *Επαληθευτής ταυτότητας (Authenticator)*, ελέγχει την πρόσβαση
- *Εξυπηρετητής επαλήθευσης ταυτότητας (Authentication server)*, παίρνει αποφάσεις σχετικές με την επαλήθευση ταυτότητας

Το σημείο στο οποίο ένας χρήστης συνδέεται στο δίκτυο ονομάζεται *θύρα*. Ένα δίκτυο μπορεί να έχει πολλές θύρες· για παράδειγμα στην περίπτωση ενός τοπικού δικτύου Ethernet μεταγωγής, κάθε επαφή εισόδου του μεταγωγέα αποτελεί μια θύρα. Υπάρχει σχέση ένα - προς - ένα μεταξύ ενός supplicant και μιας θύρας, και σε κάθε θύρα αντιστοιχεί ένας επαληθευτής ταυτότητας για να ελέγχει την κατάστασή της. Υπάρχει σχέση πολλοί - προς - ένα μεταξύ των θυρών και του εξυπηρετητή ταυτότητας. Με άλλα λόγια, ένας εξυπηρετητής ταυτότητας είναι συνήθως υπεύθυνος για πολλές θύρες, καθεμία από τις οποίες έχει το δικό της επαληθευτή ταυτότητας.

Πρέπει να σημειωθεί επιπλέον ότι η κατάσταση κάθε θύρας είναι ανεξάρτητη από τις άλλες θύρες. Το διάγραμμα στο Σχ. 5.1 έχει εξαχθεί από τις προδιαγραφές του IEEE 802.1X και απεικονίζει τις σχέσεις μεταξύ των οντοτήτων του πρωτοκόλλου.

Στο σχήμα αυτό, παρουσιάζονται οι τρεις βασικές οντότητες: σύστημα supplicant, σύστημα επαληθευτή ταυτότητας και σύστημα εξυπηρετητή επαλήθευσης ταυτότητας. Ας σημειωθεί ότι ο διακόπτης παρέχει πρόσβαση στις «Υπηρεσίες προσφερόμενες από το Σύστημα Επαληθευτή Ταυτότητας», που συνήθως έχει την έννοια της σύνδεσης στο δίκτυο· ωστόσο θα μπορούσε να είναι και κάποια άλλη υπηρεσία. Τα αρχικά PAE σημαίνουν Θύρα Πρόσβασης Οντότητας (Port Access Entity), που είναι το πλήρες όνομα για μια θύρα.



Σχήμα 5.1: Το πρότυπο IEEE 802.1X

Το Σχ. 5.1, περιέχει αναφορά για κάποιο πρωτόκολλο υψηλότερου πρωτοκόλλου μεταξύ του επαληθευτή ταυτότητας και του εξυπηρετητή επαλήθευσης ταυτότητας. Πρόκειται για το πρωτόκολλο EAP. Μηνύματα EAP ανταλλάσσονται μεταξύ του supplicant και του επαληθευτή ταυτότητας. Ο επαληθευτής ταυτότητας μπορεί επίσης να τα προωθήσει στον εξυπηρετητή επαλήθευσης ταυτότητας ως μέρος της διαδικασίας επαλήθευσης ταυτότητας. Αν ο επαληθευτής ταυτότητας βρίσκεται σε απομακρυσμένη τοποθεσία, αυτά τα μηνύματα πρέπει να σταλούν δικτυακά με τη βοήθεια κάποιου πρωτοκόλλου υψηλότερου στρώματος. Στην περίπτωση αυτή χρησιμοποιείται το RADIUS (βλ. ενότητα 5.3).

Μια από τις διαφορές μεταξύ των δικτύων dial-in και του IEEE 802.1X είναι ότι στο δεύτερο δε χρειάζεται το Point-to-Point Protocol (PPP), καθώς τα τοπικά δίκτυα IEEE 802 έχουν σχεδιαστεί για τη μεταφορά πακέτων δεδομένων. Ωστόσο, εξακολουθεί να υπάρχει ανάγκη κάποιου πρωτοκόλλου έτσι ώστε ο παραλήπτης να είναι σε θέση να αναγνωρίσει την πληροφορία. Αυτό το πρωτόκολλο είναι το EAPOL (EAP over LAN), το οποίο περιγράφεται στην ενότητα 5.2.2. Περιλαμβάνει αρκετούς τύπους μηνυμάτων και εκτός από αυτά που σχετίζονται με τη μεταφορά των μηνυμάτων EAP, υπάρχουν άλλα που αφορούν τον επαληθευτή ταυτότητας. Επιπλέον, ορίζεται μήνυμα κατάλληλο για τη μεταφορά κλειδιών κρυπτογράφησης.

Η αρχική ιδέα πίσω από το IEEE 802.1X δεν αφορούσε τις ασύρματες επικοινωνίες αλλά την προστασία των θυρών σε ένα μεταγωγέα τοπικού δικτύου. Ο σκοπός ήταν να μην επιτρέπεται σε οποιονδήποτε να αποκτάει πρόσβαση στο δίκτυο απλά συνδέοντας ένα καλώδιο Ethernet αλλά, αντί αυτού, να απαιτείται η πιστοποίηση της ταυτότητας του χρήστη. Στη συνέχεια, αναγνωρίστηκε η ανάγκη επέκτασης της αρχής αυτής από τις ενσύρματες θύρες στις ασύρματες συνδέσεις, προσέγγιση που υιοθετήθηκε στα πλαίσια των TSN/RSN.

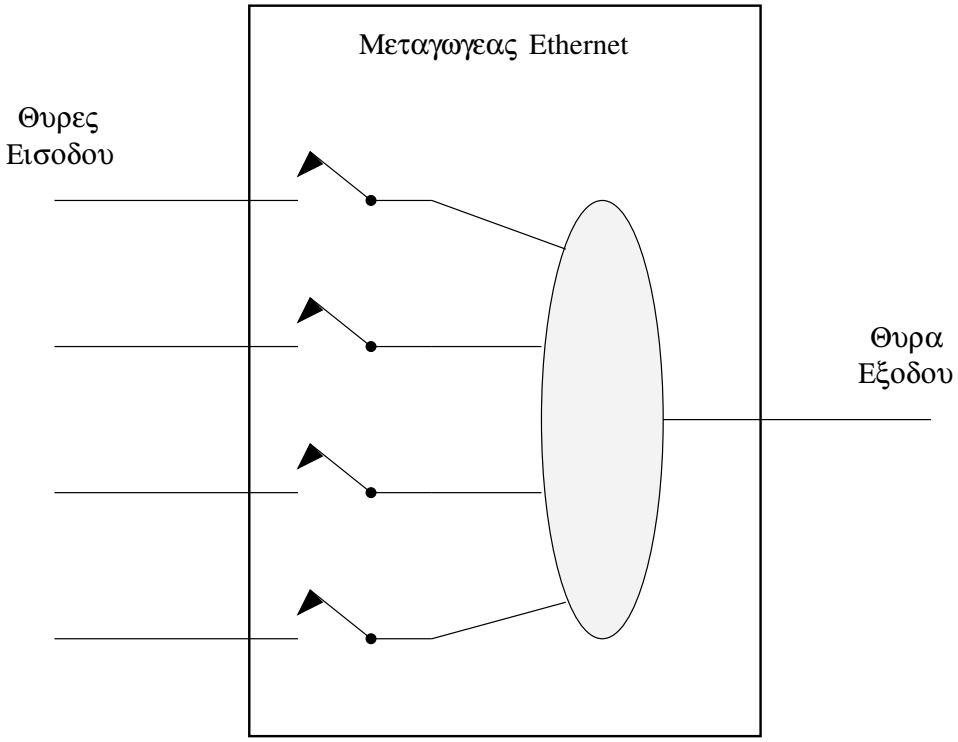
Η βασικός στόχος της ασφάλειας σε επίπεδο θύρας είναι η προστασία των δικτυακών συνδέσεων σε μη ασφαλείς περιοχές, όπως μπορεί να είναι για παράδειγμα μια αίθουσα υποδοχής ή ένας συνεδριακός χώρος. Στην περίπτωση ενός ενσύρματου δικτύου οι συνδέσεις αυτές είναι περιορισμένες, αντίθετα σε ασύρματα περιβάλλοντα κάθε σύνδεση είναι επισφαλής, καθώς η ίδια η φύση των ασύρματων ζεύξεων καθιστά *κάθε* σύνδεση προσβάσιμη από οπουδήποτε.

5.1.1 Ενσύρματο περιβάλλον μεταγωγής

Στην ενότητα αυτή, παρουσιάζεται η λειτουργία του πρωτοκόλλου IEEE 802.1X στην περίπτωση ενός απλού ενσύρματου τοπικού δικτύου μεταγωγής. Ο έλεγχος πρόσβασης στο περιβάλλον αυτό έχει την έννοια ενός διακόπτη σε κάθε θύρα, του οποίου η φυσική θέση είναι η ανοικτή, δηλαδή χωρίς σύνδεση. Ο διακόπτης κλείνει μόνο όταν εξουσιοδοτηθεί ο supplicant. Όπως φαίνεται στο Σχ. 5.2, όλες οι θύρες είναι αρχικά αποσυνδεδεμένες. Η πρόσβαση στο δίκτυο δεν παρέχεται αυτόματα με τη σύνδεση του καλωδίου συσκευής. Είναι σημαντικό να τονιστεί ότι η οντότητα του διακόπτη είναι λογική και όχι φυσική, δηλαδή υλοποιείται με λογισμικό. Όταν ο διακόπτης είναι ανοικτός, τα πακέτα δεδομένων δεν προωθούνται από και προς τη θύρα. Αντίθετα όταν είναι κλειστός στέλνονται κανονικά. Ωστόσο, η θύρα Ethernet παραμένει πάντοτε ηλεκτρικά ενεργή.

Στο μεταγωγέα του Σχ. 5.2 χρειάζεται ένας μηχανισμός που να επιτρέπει σε συσκευές που επιθυμούν να συνδεθούν, να ζητούν σχετική άδεια. Αυτή ακριβώς την ανάγκη καλύπτει το πρωτόκολλο IEEE 802.1X, το οποίο παρέχει έναν τρόπο επικοινωνίας μιας αιτούσας συσκευής με τον επαληθευτή ταυτότητας, ακόμη κι όταν ο διακόπτης είναι ανοικτός.

Σύμφωνα με την ορολογία του IEEE 802.1X, ο *επαληθευτής ταυτότητας* έχει τον έλεγχο της *κατάστασης θύρας* (είτε ο διακόπτης είναι ανοικτός ή κλειστός), όπως φαίνεται στο Σχ. 5.3. Σε αυτό παρατηρούμε ότι έχει επιτραπεί η σύνδεση της συσκευής στη θύρα 0 με το



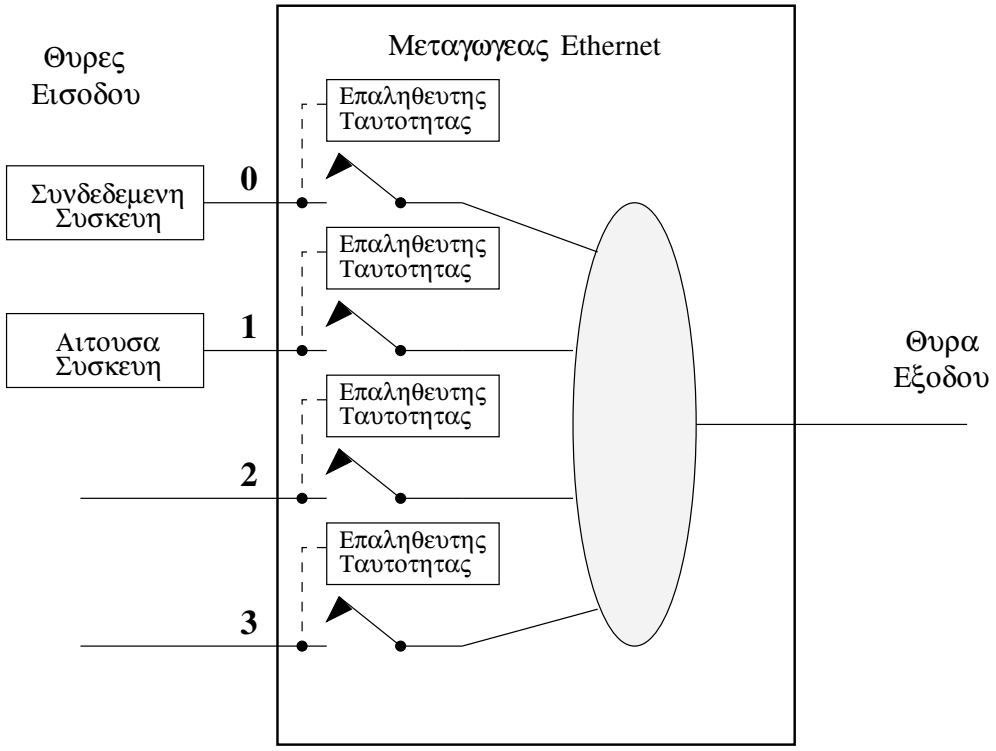
Σχήμα 5.2: Αρχική Κατάσταση Τοπικού Δικτύου Μεταγωγής IEEE 802.1X

δίκτυο, ενώ είναι σε εξέλιξη η αίτηση για πρόσβαση μια άλλης στη θύρα 1. Η δεύτερη συσκευή επικοινωνεί με τον επαληθευτή ταυτότητας, ωστόσο δεν της έχει δοθεί ακόμη άδεια για σύνδεση. Το πρωτόκολλο που χρησιμοποιείται για την επικοινωνία μεταξύ του supplicant και του επαληθευτή ταυτότητας βασίζεται στο EAP.
Στο Σχ. 5.3, φαίνεται ότι οι επαληθευτές ταυτότητας παίρνουν τις αποφάσεις επίτρεψης πρόσβασης. Στην πραγματικότητα, οι αποφάσεις αυτές λαμβάνονται βάσει μιας ελεγχόμενης βάσης δεδομένων με στοιχεία επαλήθευσης ταυτότητας. Για το σκοπό αυτό, ο επαληθευτής ταυτότητας πρέπει να επικοινωνεί με έναν εξυπηρετητή επαλήθευσης ταυτότητας κάθε φορά που δέχεται αίτηση από κάποιο supplicant για σύνδεση στο δίκτυο.

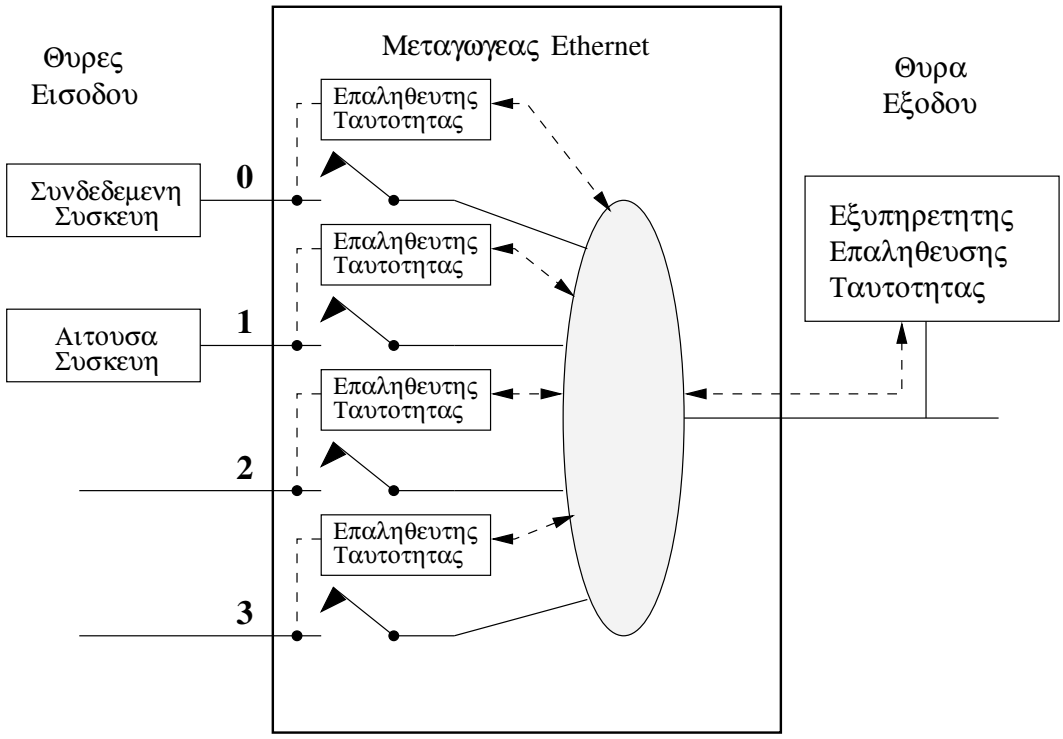
Σε ένα σύστημα μικρών διαστάσεων ο εξυπηρετητής επαλήθευσης ταυτότητας θα μπορούσε να βρίσκεται μέσα στον ίδιο το μεταγωγέα ως μία λίστα με χρήστες που έχουν δικαίωμα πρόσβασης. Η προσέγγιση αυτή δεν είναι ιδιαίτερα πρακτική, καθώς απαιτείται η ρύθμιση κάθε μεταγωγέα ξεχωριστά. Ως εκ τούτου, ο εξυπηρετητής επαλήθευσης ταυτότητας βρίσκεται συνήθως σε καποιο κεντρικό σημείο του δικτύου, όπως φαίνεται στο Σχ. 5.4.
Και οι τέσσερις επαληθευτές ταυτότητας του μεταγωγέα του Σχ. 5.4, επικοινωνούν με τον εξυπηρετητή επαλήθευσης ταυτότητας. Στην πράξη, αυτό ισχύει για όλους τους μεταγωγείς του δικτύου, ώστε ο εξυπηρετητής επαλήθευσης ταυτότητας να είναι σε θέση να λαμβάνει αποφάσεις για πλήθος θυρών.

5.1.2 Ασύρματα Τοπικά Δίκτυα

Στην ενότητα αυτή, περιγράφεται η λειτουργία του πρωτοκόλλου IEEE 802.1X στα πλαίσια των τοπικών ασυρμάτων δικτύων. Δεδομένου ότι στην περίπτωση των ενσυρμάτων τοπικών δικτύων το πρωτόκολλο ελέγχει μεμονωμένες θύρες, στα WLAN θα πρέπει να θεωρήσουμε



Σχήμα 5.3: Λειτουργία του Επαληθευτή Ταυτότητας IEEE 802.1X

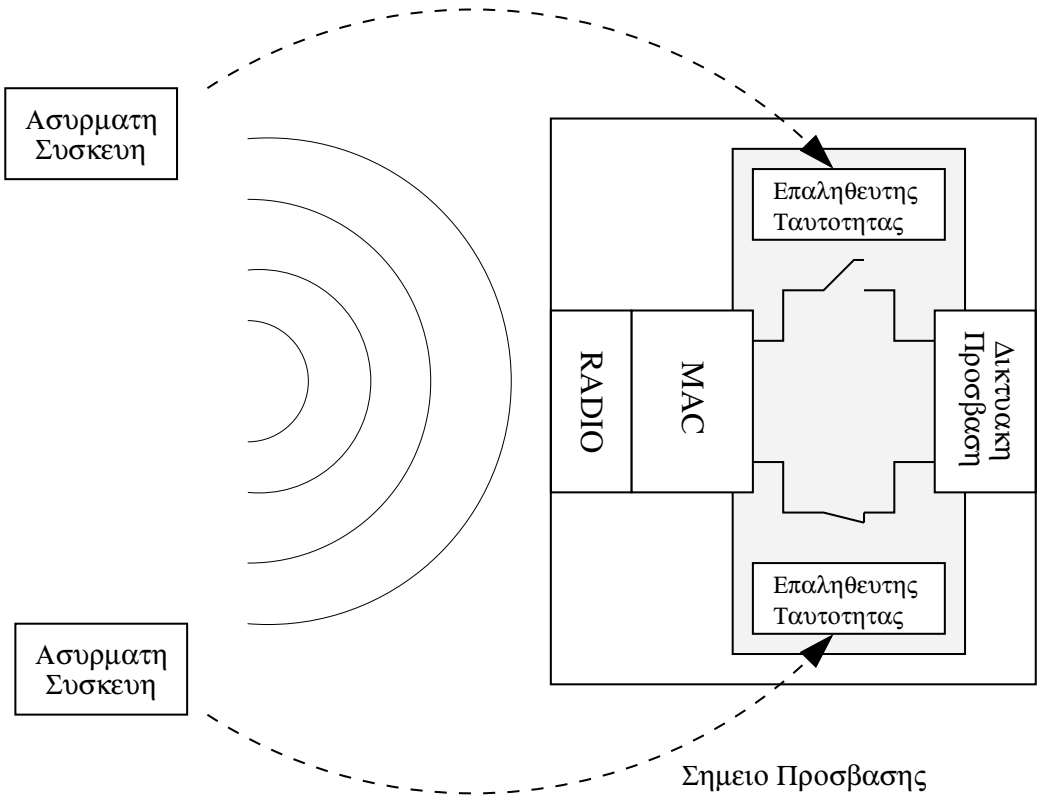


Σχήμα 5.4: Λειτουργία του Εξυπηρετητή Επαλήθευσης Ταυτότητας IEEE 802.1X

κάθε ασύρματη σύνδεση μεταξύ μιας κινητής συσκευής και του σημείου πρόσβασης ως μία ανεξάρτητη σύνδεση. Έτσι, αντικαθιστούμε τις φυσικές συνδέσεις του ενσυρμάτου δικτύου μεταγωγής με λογικές συνδέσεις ασύρματης επικοινωνίας.

Στα πλαίσια του IEEE 802.1X, κάθε κινητή συσκευή αποτελεί μία οντότητα supplicant που επιθυμεί τις υπηρεσίες του σημείου πρόσβασης, δηλαδή, τη σύνδεσή της στο ενσύρματο δίκτυο. Για το σκοπό αυτό, το σημείο πρόσβασης δημιουργεί για κάθε supplicant μία λογική θύρα με τον αντίστοιχο επαληθευτή ταυτότητας. Ο τελευταίος είναι υπεύθυνος για τον έλεγχο πρόσβασης του κινητού supplicant που του έχει ανατεθεί. Εκτός της λογικής θύρας και του επαληθευτή ταυτότητας, υπάρχει επιπρόσθετα και ένας λογικός διακόπτης ελέγχου σε ανοικτή αρχική κατάσταση.

Κάθε νέα ασύρματη συσκευή που παίζει το ρόλο του supplicant επιθυμώντας πρόσβαση, οφείλει να στείλει κατάλληλα μηνύματα στον επαληθευτή ταυτότητας, ο οποίος ελέγχει τη σύνδεση της συσκευής αυτής στο σημείο πρόσβασης. Η όλη διαδικασία υλοποιείται με λογισμικό, δηλαδή, δεν υπάρχει φυσική οντότητα επαληθευτή ταυτότητας ούτε διακόπτης. Επομένως, ο αριθμός των οντοτήτων του IEEE 802.1X σε λειτουργία είναι ίδιος με τον αριθμό των συνδεδεμένων κινητών συσκευών (βλ. Σχ. 5.5).



Σχήμα 5.5: Λογικές θύρες IEEE 802.1X σε Σημείο Πρόσβασης

Ο εξηπηρετητής επαλήθευσης ταυτότητας μπορεί να είναι ενσωματωμένος στο σημείο πρόσβασης, δηλαδή μια απλή διεργασία αυτού· για παράδειγμα μια λίστα με ονόματα και μυστικούς κωδικούς χρηστών. Στην περίπτωση αυτή δεν απαιτείται η χρήση RADIUS, καθώς ο επαληθευτής και ο εξυπηρετητής επαλήθευσης ταυτότητας βρίσκονται από κοινού στο σημείο πρόσβασης και επομένως δε χρειάζεται να επικοινωνούν μέσω δικτύου. Ωστόσο, αυτό σημαίνει ταυτόχρονα ότι ο αριθμός των μεθόδων επαλήθευσης ταυτότητας περιορίζεται σε αυτόν που υποστηρίζει ο κατασκευαστής του σημείου πρόσβασης.

Στην περίπτωση των θυρών των τοπικών δικτύων Ethernet ή στην πρόσβαση dial-up, όπου υπάρχει φυσική σύνδεση, ο ρόλος του IEEE 802.1X είναι απλός: ο supplicant ζητάει πρόσβαση και ο επαληθευτής ταυτότητας την παραχωρεί, αφού απευθυνθεί στον εξυπηρετητή επαλήθευσης ταυτότητας. Έτσι, είναι δύσκολο για έναν επίδοξο εισβολέα να πάρει υπό τον έλεγχο του μια σύνδεση από τη στιγμή που αυτή έχει εξουσιοδοτηθεί σε ένα έγκυρο χρήστη. Είναι φανερό ότι το ίδιο δεν ισχύει στα WLAN. Ελλείψει μηχανισμού προστασίας, θα ήταν ιδιαίτερα εύκολο για καοιον εισβολέα να περιμένει να παραχωρηθεί πρόσβαση σε ένα έγκυρο χρήστη και στη συνέχεια να χρησιμοποιήσει εκείνος τη σύνδεση με την ταυτότητα του χρήστη. Ως εκ τούτου, στα ασύρματα δίκτυα, ο μηχανισμός της επαλήθευσης ταυτότητας οφείλει να αποτρέπει τέτοιες κακόβουλες ενέργειες. Αυτο επιτυγχάνεται ενσωματώνοντας την ακεραιότητα των μηνυμάτων επαλήθευσης ταυτότητας στο διαδικασία εξουσιοδότησης. Πρέπει να εξασφαλίζεται ότι, τόσο το σημείο πρόσβασης όσο και η κινητή συσκευή, έχουν τα μυστικά κλειδιά τους εγκατεστημένα ελέγχοντας την αυθεντικότητα των μηνυμάτων και επιπλέον ότι έχουν ενεργοποιήσει την κρυπτογράφηση πριν παραχωρηθεί η πρόσβαση στο δίκτυο.

5.2 Πρωτόκολλο Επεκτάσιμης Επαλήθευσης Ταυτότητας

Στην ενότητα αυτή, παρουσιάζεται το Πρωτόκολλο Επεκτάσιμης Επαλήθευσης Ταυτότητας (Extensible Authentication Protocol – EAP). Το EAP περιλαμβάνει ένα σύνολο μηνυμάτων που χρησιμοποιείται κατά την έναρξη και το κλείσιμο των διαπραγματεύσεων που πραγματοποιούνται από όλες τις μεθόδους επαλήθευσης ταυτότητας των ανωτέρων στρωμάτων. Επιπλέον, το EAP επιτρέπει σε δύο πλευρές να ανταλλάξουν τις πληροφορίες που αφορούν τη συγκεκριμένη μέθοδο επαλήθευσης ταυτότητας που επιθυμούν να εφαρμόσουν. Το περιεχόμενο των μεθόδων αυτών δεν ορίζεται στο EAP. Ακριβώς αυτή η δυνατότητα του EAP να διεκπεραιώνει μέρος της επικοινωνίας με προτυποποιημένο τρόπο και το υπόλοιπο με ειδικό για κάθε μέθοδο τρόπο, αποτελεί το κλειδί της επεκτασιμότητας του πρωτοκόλλου. Αναφερόμαστε σε αυτά τα ειδικά μηνύματα ως *ενδιάμεσα*, επειδή παρουσιάζονται μετά την έναρξη και πριν τον τερματισμό.

Μεγάλος αριθμός αυτών των ενδιάμεσων μηνυμάτων μπορούν να ανταλλαχθούν μέχρι να ολοκληρωθεί η επαλήθευση ταυτότητας. Ο λόγος για τον οποίο το EAP είναι επεκτάσιμο είναι ότι οι λεπτομέρειες αυτών των ειδικών μηνυμάτων ορίζονται στα αντίστοιχα κείμενα Request For Comment – RFC της IETF. Για παράδειγμα, υπάρχει ειδικό RFC σχετικό με τη χρήση Ασφάλειας Επιπέδου Μεταφοράς πάνω από το EAP (EAP-TLS) και άλλο για το Σηραγγώδες TLS (EAP-TTLS). Το γεγονός αυτό επιτρέπει και την ανάπτυξη νέων μεθόδων οι οποίες μπορούν να υλοποιηθούν στα υπάρχοντα συστήματα.

Το EAP περιγράφεται στο κείμενο [10] της IETF, στο οποίο ορίζονται τέσσερις τύποι μηνυμάτων που μπορούν να σταλούν:

- **Request**: Χρησιμοποιείται για την αποστολή μηνυμάτων από τον επαληθευτή ταυτότητας στην οντότητα του supplicant
- **Response**: Χρησιμοποιείται για την αποστολή μηνυμάτων από την οντότητα του supplicant στον επαληθευτή ταυτότητας
- **Success**: Στέλνεται από τον επαληθευτή ταυτότητας ως ένδειξη παροχής πρόσβασης
- **Failure**: Στέλνεται από τον επαληθευτή ταυτότητας ως ένδειξη άρνησης πρόσβασης

Ας σημειωθεί ότι τα μηνύματα αυτά ορίζονται σε σχέση με τον επαληθευτή ταυτότητας. Ωστόσο, στο σενάριο του IEEE 802.1X, ο επαληθευτής ταυτότητας προωθεί τα μηνύματα στον εξυπηρετητή επαλήθευσης ταυτότητας, που συνήθως χρησιμοποιεί το RADIUS. Στην περίπτωση αυτή, ο εξυπηρετητής επαλήθευσης ταυτότητας είναι αυτός που παράγει μηνύματα request, success και failure, ενώ ο επαληθευτής ταυτότητας απλά τα αναμεταδίδει στην οντότητα του supplicant.

Τα μηνύματα request και response υποδιαιρούνται επιπλεόν με βάση το πεδίο `Τύπος` του EAP. Το πεδίο αυτό υποδεικνύει το είδος της πληροφορίας που μεταφέρεται στο μήνυμα EAP. Οι πρώτοι έξι τύποι ορίζονται στο πρότυπο, ενώ όλοι οι υπόλοιποι έχουν κρατηθεί για τις μεθόδους επαλήθευσης ταυτότητας. Ο πιο σημαντικός από τους βασικούς τύπους είναι ο `Identity` (ταυτότητα) με τιμή 1. Συνήθως, αυτός χρησιμοποιείται στη φάση έναρξης του EAP: το μήνυμα `EAP-Request/Identity` στέλνεται από τον επαληθευτή ταυτότητας σε ένα νέο supplicant. Ο τελευταίος απαντά με το μήνυμα `EAP-Response/Identity`, το οποίο περιέχει το όνομα χρήστη ή κάποιο άλλο αναγνωριστικό κατάλληλο για τον εξυπηρετητή επαλήθευσης ταυτότητας.

Οι τιμές του πεδίου `Τύπος` που είναι υψηλότερες του 6 δεν ορίζονται στο [10], αλλά εκχωρούνται από τον IANA και είναι μοναδικές για κάθε μέθοδο επαλήθευσης ταυτότητας. Ωστόσο, η χρήση του συγκεκριμένου πεδίου δεν έχει πάντα την ίδια έννοια. Γενικά, υποδεικνύει τη μέθοδο επαλήθευσης ταυτότητας. Για παράδειγμα ένα μήνυμα με τιμή του πεδίου `Τύπος` ίση με 2 ονομάζεται `Notification` (ανακοίνωση) και χρησιμοποιείται προκειμένου να εμφανιστεί κάποιο μήνυμα κειμένου στο χρήστη. Ένα μήνυμα με τιμή 3 στο εν λόγω πεδίο, ονομάζεται `NAK` και χρησιμοποιείται όταν γίνεται μια άιτηση για μέθοδο επαλήθευσης ταυτότητας που δεν υποστηρίζεται.

Στα πλαίσια του IEEE 802.1X μια αίτηση τύπου `Identity` αποτελεί συνήθως το πρώτο μήνυμα που στέλνεται και στο οποίο ο supplicant απαντά με πληροφορίες σχετικές με την ταυτότητά του. Μια πολύ απλοποιημένη διαδικασία επαλήθευσης ταυτότητας θα μπορούσε να έχει ως εξής:

1. `EAP-Identity request` (από τον επαληθευτή ταυτότητας)
2. `EAP-Identity response` (από την οντότητα του supplicant)
3. `EAP-Success` (από τον επαληθευτή ταυτότητας)

Ουσιαστικά στην απλή αυτή περίπτωση, η συσκευή δεν έχει επαληθεύσει την ταυτότητά της, καθώς ο επαληθευτής την αποδέχεται 'τυφλά'. Από την άλλη πλευρά, η απόδειξη αυθεντικότητας θα μπορούσε να παρέχεται με κάποιο άλλο μηχανισμό. Για παράδειγμα, η ταυτότητα ίσως να παράγεται από μια έξυπνη κάρτα (smart card) που μεταβάλλεται κάθε δευτερόλεπτο, όντας συγχρονισμένη με τον εξυπηρετητή επαλήθευσης ταυτότητας. Η μέθοδος αυτή αναφέρεται συχνά ως *κωδικός εισόδου μιας χρήσης* (one-time password). Η κενή επαλήθευση αυτού του τύπου, μπορεί να βρει εφαρμογή σε απλά ασύρματα τοπικά δίκτυα, τα οποία διαθέτουν ήδη εγκατεστημένα μυστικά κλειδιά (προμεριζόμενα κλειδιά), και στηρίζονται στην κρυπτογράφηση για την επίτευξη της ασφάλειας.

Καθώς η ανταλλαγή `EAP-Identity` μπορεί να θεωρηθεί από μόνη της ως μια ολοκληρωμένη μέθοδος επαλήθευσης ταυτότητας, όταν χρησιμοποιείται και μια άλλη μέθοδος, όπως για παράδειγμα η ασφάλεια επιπέδου μεταφοράς, τότε πρακτικά εκτελούνται δύο μέθοδοι στη σειρά. Αυτή η έννοια της **σειριακής επαλήθευσης ταυτότητας** έχει υιοθετηθεί στο πρότυπο EAP και επιτρέπει την εφαρμογή οποιουδήποτε αριθμού μεθόδων πριν το τελικό μήνυμα `EAP-Success` ή `EAP-Failure`. Η δυνατότητα αυτή, επιτρέπει στον πελάτη να επαληθευτεί από το δίκτυο προτού αποκαλύψει την ταυτότητα του.

5.2.1 Μορφή μηνυμάτων EAP

Όλα τα μηνύματα EAP έχουν την ίδια βασική μορφή, όπως φαίνεται στο Σχ. 5.6. Το πεδίο Κωδικός υποδεικνύει τον τύπο του μηνύματος:

- `Request (01)`
- `Response (02)`
- `Success (03)`
- `Failure (04)`

Κωδικος	Αναγνωριστικο	Μηκος	Δεδομενα

Σχήμα 5.6: Γενική Μορφή Μηνύματος EAP

Το πεδίο Αναγνωριστικό παίρνει τιμές εύρους 0-255 και το πρότυπο IEEE 802.1X ορίζει ότι πρέπει να αυξάνεται για κάθε μήνυμα που στέλνεται. Όταν στέλνεται μια απόκριση, το αναγνωριστικό της τίθεται ίσό με της τιμή της αντίστοιχης αίτησης. Αυτό βοηθά στην αντιστοίχιση των αποκρίσεων με τις αιτήσεις. Το πεδίο Τύπος είναι το συνολικό μέγεθος σε byte του μηνύματος EAP. Τέλος, το πεδίο Δεδομένα περιέχει τα πραγματικά δεδομένα αίτησης ή απόκρισης που στέλνονται.

Τα μηνύματα τύπου `Success` και `Failure` είναι σύντομα και δεν περιέχουν δεδομένα. Το ένα από αυτά χρησιμοποιείται κάθε φορά για να σηματοδοτήσει το αποτέλεσμα της διαδικασίας επαλήθευσης ταυτότητας. Επειδή είναι κοινά για όλες τις μεθόδους επαλήθευσης ταυτότητας, ενδιάμεσες συσκευές (όπως το σημείο πρόσβασης) είναι σε θέση να εντοπίζουν την ολοκλήρωση της διαδικασίας επαλήθευσης ταυτότητας, χωρίς να απαιτείται να γνωρίζουν τις σχετικές λεπτομέρειες. Το σημείο πρόσβασης πρέπει να περιμένει μήνυμα RADIUS τύπου `Access-Accept` πριν πάρει οποιαδήποτε απόφαση που αφορά δικαιώματα πρόσβασης.

Οι λεπτομέρειες της μεθόδου επαλήθευσης ταυτότητας μεταφέρονται στα μηνύματα `request` και `response`. Αυτά έχουν ένα επιπρόσθετο πεδίο το οποίο καλείται Τύπος. Η μορφή ενός μηνύματος `EAP-Request` ή `EAP-Response` φαίνεται στο Σχ. 5.7.

Κωδικος	Αναγνωριστικο	Μηκος	Τυπος	Δεδομενα Αιτησης / Αποκρισης

Σχήμα 5.7: Μορφή Μηνύματος `EAP-Request/Response`

Το πεδίο Τύπος αποτελεί το κλειδί της *επεκτασιμότητας* του EAP. Σε κάθε μέθοδος επαλήθευσης ταυτότητας έχει εκχωρηθεί μία μοναδική τιμή, ώστε το σύστημα να διαβάζει κατάλληλα τις σχετικές πληροφορίες που μεταφέρονται.

5.2.2 EAP πάνω από Τοπικό Δίκτυο

Το σχετικό με το EAP RFC δεν προσδιορίζει τον τρόπο μεταφοράς των μηνυμάτων του. Μάλιστα, το EAP δεν είναι πρωτόκολλο τοπικού δικτύου, γιατί σχεδιάστηκε αρχικά για την επαλήθευση ταυτότητας μέσω διαποδιαμορφωτών (modem) dial-up. Επομένως, προκειμένου να μεταφερθούν τα μηνύματα EAP, χρειάζεται ένα πρωτόκολλο δικτύου για την ενθυλάκωσή

τους. Το IEEE 802.1X ορίζει το **EAP πάνω από Τοπικό Δίκτυο** (EAP over LAN – EAPOL) για την αποστολή μηνυμάτων μεταξύ supplicant και επαληθευτή ταυτότητας.

Το IEEE 802.1X περιέχει τις προδιαγραφές του EAPOL και ορίζει μορφές πλαισίου κατάλληλες για Ethernet (IEEE 802.3) και Τοπικά Δίκτυα Δακτυλίου με Κουπόνι (Token Ring LANs), αλλά όχι και για το IEEE 802.11. Η μορφή ενός πλαισίου EAPOL για το Ethernet φαίνεται στο Σχ. 5.8. Αξίζει να σημειωθεί ότι:

- Το πεδίο ΄Εκδοση Πρωτοκόλλου έχει πάντοτε τιμή 1.
- Το πεδίο Τύπος Πακέτου δηλώνει τον τύπο του μηνύματος.
- Για κάποιους τύπους μηνυμάτων, δε χρειάζονται επιπλέον πληροφορίες και το Μήκος Σώματος Πακέτου τίθεται στο 0, ενώ το Σώμα Πακέτου παραλείπεται. Όταν, ωστόσο, υπάρχει Σώμα Πακέτου, όπως για παράδειγμα ένα μήνυμα EAP, τότε το μήκος του και τα δεδομένα του προστίθενται κατάλληλα.

Επικεφαλιδα Ethernet MAC	Εκδοση Πρωτοκολλου	Τυπος Πακετου	Μηκος Σωματος Πακετου	Σωμα Πακετου

Σχήμα 5.8: Μορφή Πλαισίου EAPOL

Δε μεταφέρουν EAP μηνύματα όλα τα πλαίσια EAPOL· κάποια χρησιμοποιούνται για διαχειριστικά καθήκοντα. Οι πέντε τύποι μηνυμάτων EAPOL είναι οι εξής:

- `EAPOL-Start`
- `EAPOL-Key`
- `EAPOL-Packet`
- `EAPOL-Logoff`
- `EAPOL-Encapsulated-ASF-Alert`

Στη συνέχεια παρουσιάζουμε τους τύπους αυτούς, εκτός από τον τελευταίο, ο οποίος δε χρησιμοποιείται στα TSN/RSN.

`EAPOL-Start`

Όταν η οντότητα του supplicant συνδέεται για πρώτη φορά στο τοπικό δίκτυο, δε γνωρίζει τη διεύθυνση MAC του επαληθευτή ταυτότητας. Στην πραγματικότητα, δε γνωρίζει αν υπάρχει καν επαληθευτής ταυτότητας. Στο πλαίσιο αυτό, το IEEE 802.1X ορίζει το μήνυμα `EAPOL-Start`, η αποστολή του οποίου σε μια ειδική ομαδική διεύθυνση πολυεκπομπής (multicast) MAC μπορεί να αποκαλύψει την ύπαρξη ενός επαληθευτή ταυτότητας και παράλληλα να γνωστοποιήσει την παρουσία του supplicant. Η ειδική αυτή διεύθυνση, έχει κρατηθεί για τους επαληθευτές ταυτότητας IEEE 802.1X. Σε πολλές περιπτώσεις, ο επαληθευτής ταυτότητας θα έχει ήδη ειδοποιηθεί σχετικά με τη σύνδεση μιας νέας συσκευής μέσω κάποιου μηχανισμού του υλισμικού (hardware). Για παράδειγμα, ένας μεταγωγέας είναι σε θέση να γνωρίζει τη σύνδεση ενός καλωδίου προτού η συσκευή στείλει οποιαδήποτε δεδομένα. Στην περίπτωση αυτή, ο επαληθευτής ταυτότητας μπορεί να αντικαταστήσει το μήνυμα `EAPOL-Start` με το δικό του. Σε κάθε περίπτωση, ο επαληθευτής ταυτότητας στέλνει ένα μήνυμα `EAP-Request Identity` χρησιμοποιώντας το πλαίσιο `EAPOL-Packet`.

`EAPOL-Key`

Με τη βοήθεια αυτού του τύπου μηνύματος, ο επαληθευτής ταυτότητας στέλνει τα κλειδιά κρυπτογράφησης στην οντότητα του supplicant, αμέσως μολίς αποφασίσει ότι του επιτρέπει την πρόσβαση στο δίκτυο. Φυσικά, απαιτείται η κρυπτογράφηση των ίδιων των κλειδιών πριν την αποστολή τους, διαδικασία που δεν ορίζεται στις προδιαγραφές του IEEE 802.1X. Γενικά, οι μέθοδοι που σχετίζονται με το συνδυασμό της κρυπτογραφίας και της διαδικασίας επαλήθευσης ταυτότητας περιγράφονται στο πρότυπο IEEE 802.1AA, που αποτελεί την εξέλιξη του IEEE 802.1X.

`EAPOL-Packet`

Αυτό το πλαίσιο EAPOL χρησιμοποιείται για την αποστολή των πραγματικών μηνυμάτων EAP. Αποτελεί απλά το δοχείο μεταφοράς ενός μηνύματος EAP μέσα από το τοπικό δίκτυο, που αποτελεί και το σκοπό του EAPOL.

`EAPOL-Logoff`

Αυτός ο τύπος μηνύματος δηλώνει την πρόθεση του supplicant να αποσυνδεθεί από το δίκτυο.

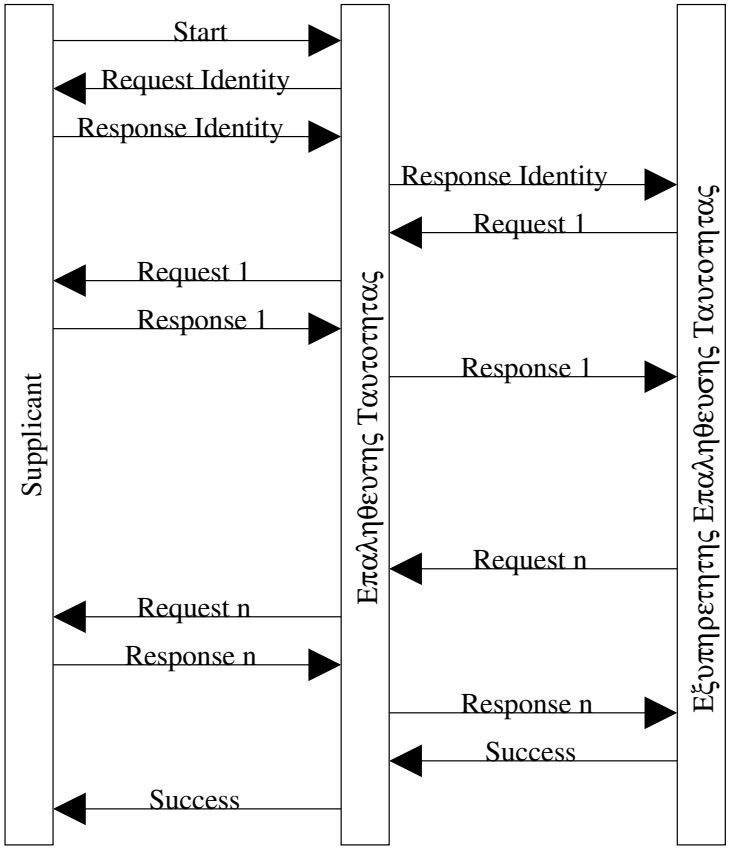
5.2.3 Επαλήθευση ταυτότητας

Στην ενότητα αυτή, παρουσιάζεται η διαδικασία επαλήθευσης ταυτότητας. Η ακολουθία των μηνυμάτων EAP που ανταλλάσσονται για το σκοπό αυτό, φαίνεται στο Σχ. 5.9.

Όταν μια οντότητα supplicant επιθυμεί να συνδεθεί, πρέπει αρχικά να γνωστοποιήσει τον επαληθευτή ταυτότητας. Ως επί το πλείστον, ο επαληθευτής ταυτότητας προειδοποιείται σχετικά κατά τη διαδικασία σύνδεσης, είτε με τη σύνδεση του καλωδίου, ή στην περίπτωση της ασύρματης επικοινωνίας, με το συσχετισμό στο σημείο πρόσβασης. Εναλλακτικά, μπορεί να χρησιμοποιηθεί το μήνυμα `EAPOL-Start`.

Ο επαληθευτής ταυτότητας αποκρίνεται αρχικά με μήνυμα `EAP-Request/Identity`. Το βήμα αυτό μπορεί να παραλειφθεί, στην περίπτωση που η ταυτότητα του supplicant είναι ήδη γνωστή στον επαληθευτή από κάποια άλλη μέθοδο. Ο supplicant πρέπει να απαντήσει με μήνυμα `EAP-Response/Identity`. Εδώ ανακύπτει θέμα ασφαλείας, καθώς η οντότητα του supplicant δε μπορεί να είναι σίγουρη για τον επαληθευτή ταυτότητας, ειδικά σε ένα ασύρματο δίκτυο. Άλλωστε θα μπορούσε κάποιος επίδοξος εισβολέας να έχει τοποθετήσει κάποιο μη εξουσιοδοτημένο σημείο πρόσβασης. Στο πλαίσιο αυτό, ο supplicant μπορεί να μη θέλει να αποκαλύψει την ταυτότητά του στην παρούσα φάση και να χρησιμοποιήσει κάποιο ψευδώνυμο εναλλακτικά. Υπάρχουν μέθοδοι που υποστηρίζουν τη χρήση ψευδωνύμων, ωστόσο υποθέτουμε ότι στο σενάριο του Σχ. 5.9, ο supplicant είναι διατεθιμένος να στείλει την πραγματική του ταυτότητα στον επαληθευτή.

Έως τώρα, όλα τα μηνύματα ανταλλάσσονται μεταξύ του supplicant και του επαληθευτή ταυτότητας, δηλαδή στην περίπτωση του IEEE 802.11, μεταξύ της κινητής συσκευής και του σημείου πρόσβασης. Είναι σημαντικό να μην εμπλακεί ο εξυπηρετητής επαλήθευσης ταυτότητας προτού εξασφαλιστεί ότι η οντότητα του supplicant υποστηρίζει το 802.1X, στέλνοντας το πρώτο μήνυμα `EAP-Request`. Έχοντας αποκτήσει την ταυτότητα του supplicant, ο επαληθευτής πρέπει να επικοινωνήσει με τον εξυπηρετητή επαλήθευσης ταυτότητας για να διαπιστώσει αν θα του επιτραπεί η πρόσβαση. Προκειμένου να μην απαιτείται ο επαληθευτής να υποστηρίζει όλες τις μεθόδους επαλήθευσης ταυτότητας, τα σχετικά μηνύματα EAP προωθούνται απ΄ ευθείας στον εξυπηρετητή επαλήθευσης ταυτότητας. Επομένως, στην πραγματικότητα, ο



Σχήμα 5.9: Ακολουθία Μηνυμάτων EAP

supplicant και ο εξυπηρετητής επικοινωνούν απ΄ ευθείας στη φάση αυτή. Κατά τη διάρκεια της διαδικασίας επαλήθευσης ταυτότητας, ο επαληθευτής εξετάζει κάθε μήνυμα που μεταφέρεται μεταξύ supplicant και εξυπηρετητή επαλήθευσης ταυτότητας, προκειμένου να εντοπίσει μηνύματα τύπου `EAP-Success` ή `EAP-Failure`. Πρέπει, λοιπόν, να περιμένει την απόφαση του εξυπηρετητή ταυτότητας σχετικά με το αν ο supplicant έχει γίνει αποδεκτός ή έχει απορριφθεί. Στην περίπτωση που χρησιμοποιείται το RADIUS, η αντίστοιχη ένδειξη παρέχεται με κατάλληλο μήνυμα RADIUS.

5.2.4 Ελαφρύ EAP

Το Ελαφρύ EAP (Lightweight EAP – LEAP), παρόλο που αποτελεί ένα ιδιοκτησιακό πρωτόκολλο που αναπτύχθηκε από τη Cisco, ωστόσο η ευρεία του χρήση οδήγησε και άλλους κατασκευαστές να το υποστηρίζουν στους εξυπηρετητές RADIUS.

Συνεπές με το μοντέλο IEEE 802.1X, το LEAP χωρίζει το σύστημα σε τρεις οντότητες: supplicant, επαληθευτής ταυτότητας και εξυπηρετητής επαλήθευσης ταυτότητας. Ο supplicant βρίσκεται στην κινητή συσκευή, ενώ ο επαληθευτής ταυτότητας στο σημείο πρόσβασης. Ο εξυπηρετητής επαλήθευσης ταυτότητας υλοποιείται από ένα εξυπηρετητή RADIUS. Για τη μεταφορά των κλειδιών χρησιμοποιούνται κάποια ιδιοκτησιακά γνωρίσματα του RADIUS.

Το LEAP είναι ένα πρωτόκολλο αμφίδρομης πρόκλησης – απόκρισης που βασίζεται σε ένα μεριζόμενο μυστικό κλειδί μεταξύ του εξυπηρετητή επαλήθευσης ταυτότητας και της κινητής

συσκευής· *όχι* του σημείου πρόσβασης. Στηρίζεται γενικά στο MS-CHAPv1, το οποίο χρησιμοποιείται συνήθως για απομακρυσμένη επαλήθευση ταυτότητας μέσω dial-up. Σε αντίθεση με το συμβατικό MS-CHAP, η επαλήθευση ταυτότητας είναι αμοιβαία, με ξεχωριστές προκλήσεις να εκδίδονται από τον εξυπηρετητή επαλήθευσης ταυτότητας και την κινητή συσκευή. Στο πλαίσιο αυτό, δεν εξασφαλίζεται η αυθεντικότητα του ίδιου του σημείου πρόσβασης. Αν ένα μη εξουσιοδοτημένο σημείο πρόσβασης μπορούσε με κάποιο τρόπο να αποκτήσει πρόσβαση στο ενσύρματο δίκτυο με σύνδεση στον εξυπηρετητή επαλήθευσης ταυτότητας, θα μπορούσε να δράσει ως 'ενδιάμεσος ' (man in the middle) στη διαδικασία επαλήθευσης ταυτότητας. Ωστόσο, το σημείο πρόσβασης πρέπει να διαθέτει ήδη εγκαταστημένη σχέση εμπιστοσύνης με τον εξυπηρετητή επαλήθευσης ταυτότητας προκειμένου να λάβει το κλειδί κρυπτογράφησης συνόδου, επομένως ένα παράνομο σημείο πρόσβασης δε θα ήταν σε θέση να στείλει ή να λάβει κρυπτογραφημένα δεδομένα από την κινητή συσκευή.

Μόλις ολοκληρωθεί η αμοιβαία επαλήθευση ταυτότητας, το κλειδί κρυπτογράφησης συνόδου στέλνεται στο σημείο πρόσβασης μέσα σε ένα γνώρισμα RADIUS. Αυτό το γνώρισμα κρυπτογραφείται χρησιμοποιώντας ένα μεριζόμενο μυστικό μεταξύ του σημείου πρόσβασης και του εξυπηρετητή. Ο πελάτης υπολογίζει επίσης ένα αντίγραφο του κλειδιού συνόδου. Το κλειδί δε μεταδίδεται μέσω της ασύρματης ζεύξης αλλά υπολογίζεται βάσει μιας τυχαίας τιμής. Το σημείο πρόσβασης σηματοδοτεί μια επιτυχημένη επαλήθευση ταυτότητας με μήνυμα `EAPOL-Success` προς την κινητή συσκευή. Στη συνέχεια ενεργοποιεί την κρυπτογράφηση στέλνοντας ένα μήνυμα `EAPOL-Key`. Ακολουθούν τα βήματα της όλης διαδικασίας, ενώ σχηματικά παρουσιάζεται στο Σχ. 5.10:

1. Ο εξυπηρετητής επαλήθευσης ταυτότητας στέλνει μια τυχαία ακολουθία χαρακτήρων στην κινητή συσκευή ως πρόκληση. Η κινητή συσκευή πρέπει να αποδείξει ότι γνωρίζει το κλειδί στέλνοντας μια ακολουθία χαρακτήρων που προκύπτει από την πρόκληση.

2. Η κινητή συσκευή στέλνει μια πρόκληση στον εξυπηρετητή επαλήθευσης ταυτότητας, ο οποίος πρέπει επίσης να αποκριθεί σωστά.

3. Ο εξυπηρετητής επαλήθευσης ταυτότητας παράγει και στέλνει ένα κλειδί συνόδου στο σημείο πρόσβασης με το μήνυμα `EAP-Success` ενθυλακωμένο σε RADIUS.

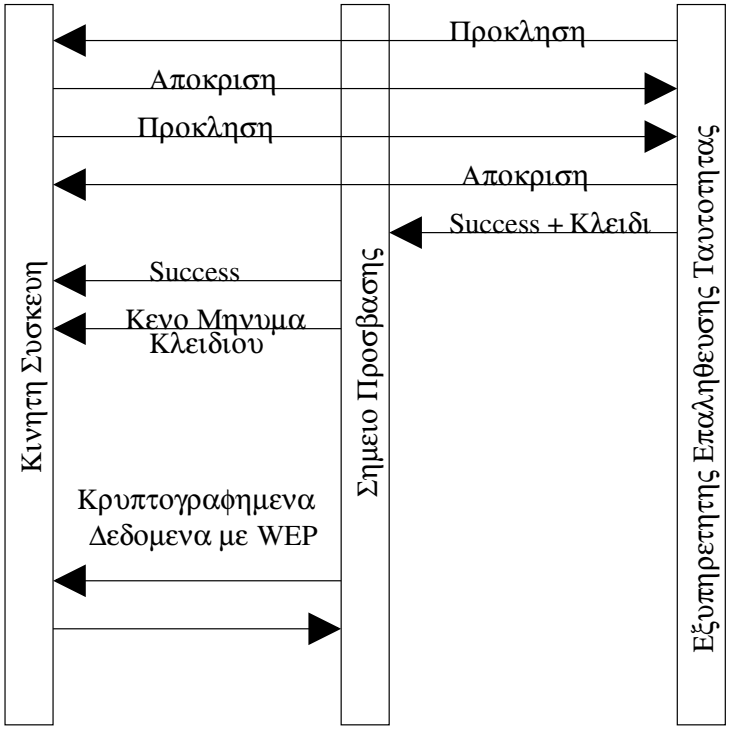
4. Το σημείο πρόσβασης ειδοποιεί την κινητή συσκευή σχετικά με την επαλήθευση ταυτότητας χρησιμοποιώντας το μήνυμα `EAPOL-Success`. Στο σημείο αυτό ο πελάτης υπολογίζει το κλειδί συνόδου που απαιτείται.

5. Το σημείο πρόσβασης στέλνει ένα μήνυμα `EAPOL-Key` για την ενεργοποίηση της κρυπτογράφησης. Ας σημειωθεί ότι δε στέλνεται το πραγματικό κλειδί· είναι απλά ένα μήνυμα γνωστοποίησης.

6. Η κινητή συσκευή και το σημείο πρόσβασης επικοινωνούν χρησιμοποιώντας κρυπτογράφηση WEP.

Στην ασύρματη πλευρά, το LEAP χρησιμοποιεί το IEEE 802.1X και το EAPOL, όπως περιγράφεται στις ενότητες 5.1 και 5.2.2, αντίστοιχα. Στην ενσύρματη πλευρά, το LEAP χρησιμοποιεί το EAP πάνω από RADIUS.

Το LEAP εμπεριέχει πολλές από τις βασικές έννοιες των TSN και RSN. Ωστόσο, τα τελευταία έχουν προσθέσει επιπλέον λεπτομέρειες που βελτιώνουν σημαντικά τη συνολική ασφάλεια. Το LEAP αρχικά χρησιμοποιούσε το WEP, το οποίο παρουσιάζει αρκετές αδυναμίες. Ωστόσο, η δυνατότητα του LEAP να παράγει προσωρινά κλειδία συνόδου μειώνει σε κάποιο βαθμό την αποτελεσματικότητα των επιθέσεων. Επιπλέον, το LEAP χρησιμοποιεί το



Σχήμα 5.10: Ακολουθία Μηνυμάτων LEAP

πρωτόκολλο MS-CHAPv1, το οποίο είναι ευάλωτο σε μερικές επιθέσεις με λεξικό. Συνολικά, όμως, το LEAP παρέχει σχετική ασφάλεια παρουσιάζοντας τα εξής πλεονεκτήματα:

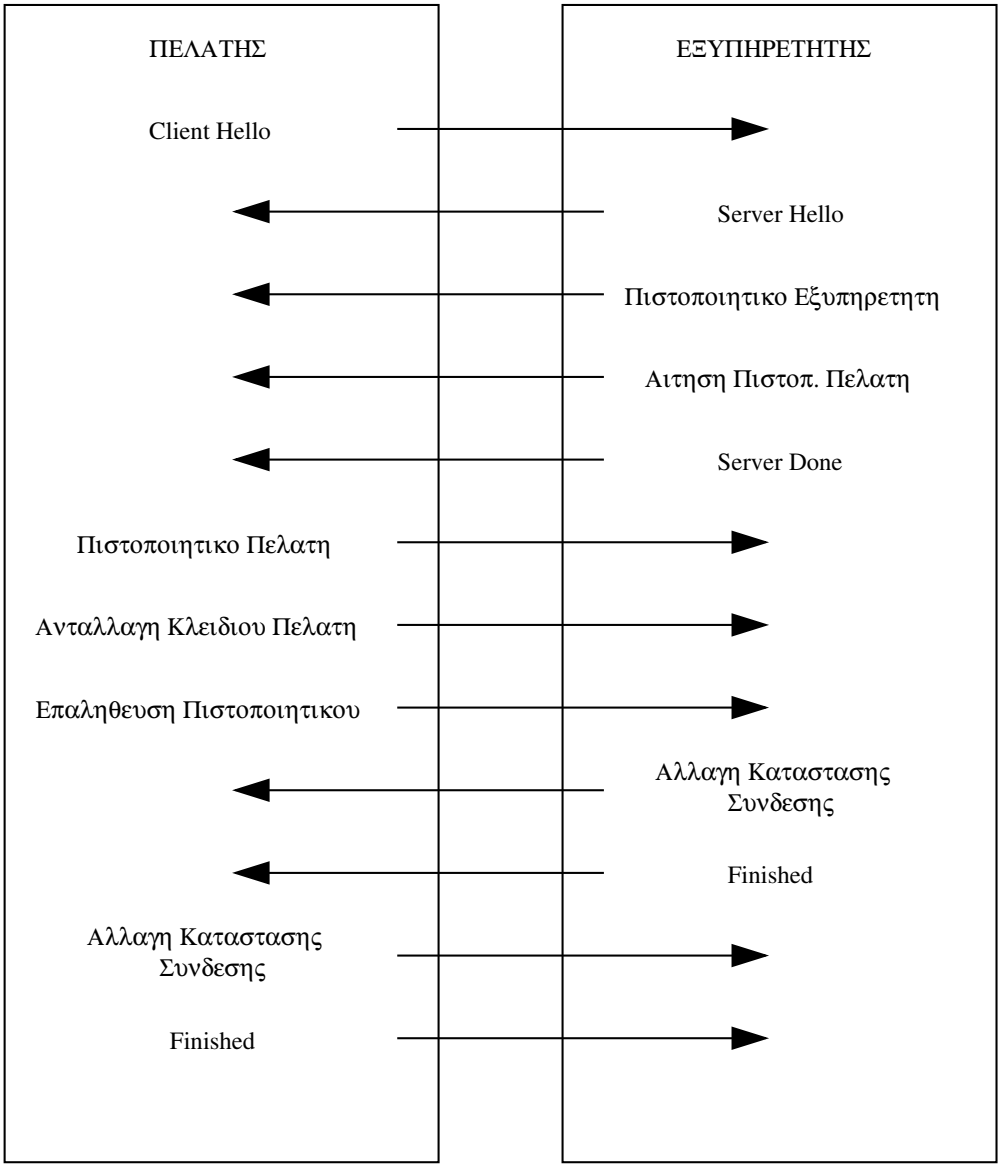
- Αμοιβαία επαλήθευση ταυτότητας
- Προσωρινά κλειδιά συνόδου
- Κεντρικοποιημένη διαχείριση κλειδιών

5.2.5 Ασφάλεια Επιπέδου Μεταφοράς και EAP

Το πρωτόκολλο Ασφάλειας Επιπέδου Μεταφοράς (Transport Layer Security – TLS) προσφέρει περισσότερες υπηρεσίες από αυτές που απαιτούνται στα πλαίσια του TSN/RSN. Πιο συγκεκριμένα παρέχει μηχανισμούς επαλήθευσης ταυτότητας, κρυπτογράφησης και συμπίεσης δεδομένων. Τόσο το TSN, όσο και το RSN, διαθέτουν δικές τους μεθόδους κρυπτογράφησης, όπως είναι το TKIP και το AES-CCMP αντίστοιχα, ενώ στις προδιαγραφές τους δεν ανήκει η χρήση συμπίεσης των δεδομένων. Αντίθετα, η μέθοδος επαλήθευσης ταυτότητας του TLS είναι ιδανική για το μοντέλο του EAP/IEEE 802.1X

Γενικές αρχές του TLS

Η σχέση μεταξή των δύο πλευρών που επικοινωνούν, εγκαθίσταται στο TLS με τη σύναψη μιας χειραψίας. Αυτή περιλαμβάνει μια σειρά μηνυμάτων που ανταλλάσσονται με καθορισμένη σειρά και φαίνεται στο Σχ. 5.11. Μπορούμε να παρατηρήσουμε ότι κατά την έναρξη της χειραψίας, οι δυο πλευρές στέλνουν μηνύματα χαιρετισμού (`Client Hello`/`Server Hello`), ενώ πριν το τέλος της, ελέγχεται η εγκυρότητα κάθε μηνύματος.



Σχήμα 5.11: Χειραψία TLS

Η διαδικασία της χειραψίας του TLS επιτυγχάνει τρεις στόχους:

1. Την επαλήθευση ταυτότητας του εξυπηρετητή (και προαιρετικά του πελάτη).
2. Την παραγωγή ενός μυστικού κυρίου κλειδιού (master key) για τη σύνοδο.
3. Την αρχικοποίηση και ενεργοποίηση κρυπτογραφικής λειτουργίας για την προστασία των επικοινωνιών.

Στα πλαίσια του TSN/RSN, οι μόνες λειτουγίες του TLS που απαιτούνται, είναι η επαλήθευση ταυτότητας και η παραγωγή του κυρίου κλειδιού, καθώς, όπως αναφέρθηκε, διαθέτουν δικές τους κρυπτογραφικές μεθόδους. Το TSN/RSN λαμβάνει το κύριο κλειδί που παράγεται από το TLS και από αυτό υπολογίζει ένα σύνολο κλειδιών που χρησιμοποιεί για την κρυπτογράφηση της ασύρματης ζεύξης. Με αυτόν τον τρόπο, το TLS ενσωματώνεται στο μοντέλο κατά IEEE 802.1X και λειτουργεί πάνω από το EAP, οπως θα δούμε αναλυτικά στη συνέχεια.

EAP-TLS

Το TLS σχεδιάστηκε για να λειτουργεί στο στρώμα πάνω από ένα αξιόπιστο πρωτόκολλο μεταφοράς γενικά, και όχι αποκλειστικά πάνω από το TCP/IP. Έτσι, στα πλαίσια των TSN/RSN, το TLS λειτουργεί πάνω από το EAP. Οι σχετικοί ορισμοί βρίσκονται στο κείμενο [11] της IETF, όπου περιγράφεται η χειραψία TLS για το EAP.

Το EAP αρχίζει και ολοκληρώνεται πάντα με την ίδια ακολουθία. Συνήθως, ανταλάσσεται ένα μήνυμα αίτησης/απόκρισης `EAP-Identity`. Στη συνέχεια στέλνεται μια σειρά αιτήσεων και αποκρίσεων μηνυμάτων EAP που σχετίζονται με τη συγκεκριμένη μέθοδο επαλήθευσης ταυτότητας και αναγνωρίζονται από το πεδίο `Τύπος` κάθε μηνύματος. Τελικά, ένα μήνυμα `EAP-Success/Fail` στέλνεται ανάλογα με την έκβαση. Η γενική μορφή ενός μηνύματος EAP απεικονίζεται στο Σχ. 5.12.

Κωδικος	Αναγνωριστικο	Μηκος	Τυπος	Δεδομενα Αιτησης / Αποκρισης

Σχήμα 5.12: Μορφή Μηνύματος EAP

Στην περίπτωση του TLS, το RFC ορίζει ότι το πεδίο `Τύπος` για τις αιτήσεις και αποκρίσεις του EAP παίρνει την τιμή 13. Μόνο οι πελάτες και οι εξυπηρετητές που υποστηρίζουν το EAP-TLS θα επιχειρήσουν να αποκωδικοποιήσουν αυτά τα μηνύματα. Επιπλέον, ορίζονται δύο νέα πεδία που ακολουθούν το `Τύπος`. Τα πεδία αυτά είναι τα `Σημαίες` και `Μήκος`, όπως φαίνεται στο Σχ. 5.13.

Κωδικος	Αναγνωριστικο	Μηκος	'13'	Σημαιες	Μηκος	Δεδομενα EAP–TLS

Σχήμα 5.13: Μορφή Μηνύματος EAP-TLS

Το πρώτο πεδίο `Μήκος` αναφέρεται στο μήκος του πλαισίου EAP, ενώ το δεύτερο στο μήκος του πακέτου EAP-TLS. Τα πακέτα αυτά μπορεί να είναι αρκετά μεγάλα σε μέγεθος, υπερβαίνοντας το μέγιστο μέγεθος ενός μηνύματος EAP. Σε μια τέτοια περίπτωση, το πακέτο EAP-TLS θραυσματίζεται και στέλνεται διαδοχικά. Η δεύτερη τιμή του πεδίου `Μήκος`, αναφέρεται συνολικά στο μήνυμα TLS και όχι στο τρέχον πλαίσιο. Μάλιστα, το δεύτερο πεδίο `Μήκος` είναι προαιρετικό και συνήθως παραλείπεται όταν τα δεδομένα του EAP-TLS χωρούν στο τρέχον πλαίσιο.

Το πεδίο `Σημαίες` περιλαμβάνει τρία bit:

- **Σημαία Μήκους:** Υποδεικνύει την παρουσία ή μη του πεδίου `Μήκος`
- **Σημαία Θραυσμάτων:** Ενεργοποιείται όταν ακολουθούν θραύσματα.
- **Σημαία Έναρξης:** Σηματοδοτεί την έναρξη της χειραψίας

Η ακολουθία των μηνυμάτων που ανταλάσσονται κατά τη χειραψία του EAP-TLS παρουσιάζονται στο Σχ. 5.14. Έχει υποτεθεί ότι ο εξυπηρετητής έχει αρχίσει την επικοινωνία του με τον πελάτη μέσω κάποιας μεθόδου, όπως για παράδειγμα με ένα μήνυμα `EAP-Start`. Τα βήματα έχουν ως εξής:

1. {request} Αυτή είναι η αρχή της συναλλαγής EAP. Ο εξυπηρετητής ζητά την ταυτότητα του πελάτη.

2. {response} Εδώ ο πελάτης στέλνει ένα μήνυμα με την ταυτότητα του. Για εταιρικό περιβάλλον, θα μπορούσε να προσδιοριστεί η ταυτότητα του ιδιοκτήτη του πιστοποιητικού του πελάτη που θα σταλεί. Αν ο πελάτης δεν προτίθεται να στείλει πιστοποιητικό, μπορεί να στείλει ένα ανώνυμο αναγνωριστικό, για παράδειγμα η ακολουθία `'anonymous'`.

3. {request} Ο εξυπηρετητής στέλνει μια κενή αίτηση EAP-TLS με τη σημαία έναρξης ενεργοποιημένη. Αυτή είναι και η μοναδική περίπτωση που η σημαία αυτή τίθεται.

4. {response} Ο πελάτης στέλνει μήνυμα `Client Hello` το οποίο περιλαμβάνει τις ίδιες πληροφορίες με το σύνηθες TLS.

5. {request} Ο εξυπηρετητής στέλνει δύο ή τρία μηνύματα TLS σε μία μόνο αίτηση: το `Server Hello`, προαιρετικά την αίτηση για πιστοποιητικό πελάτη, και το μήνυμα τερματισμού του εξυπηρετητή.

6. {response} Ο πελάτης απαντά τώρα με πολλά μηνύματα TLS σε μία μόνο απόκριση:

   - Πιστοποιητικό πελάτη (αν ζητηθεί)
   - Προ-κύριο μυστικό του μηνύματος ανταλλαγής κλειδιού
   - Πληροφορίες επαλήθευσης πιστοποιητικού πελάτη
   - Αλλαγή κρυπτογραφήματος
   - Τερματισμός

   Παρατηρούμε ότι ο πελάτης δημιουργεί το προ-κύριο μυστικό, υπολογίζει το κύριο μυστικό και ενεργοποιεί το κρυπτογράφημα στο ίδιο βήμα. Ωστόσο, πρέπει να σημειωθεί ότι ολόκληρο το μήνυμα EAP στέλνεται μέσα στην αρχική κρυπτογραφική ακολουθία, η οποία είναι συνήθως ανοικτή, δηλαδή χωρίς κρυπτογράφηση. Η ακολουθία αυτή δεν ενεργοποιείται προτού ολοκληρωθούν τα μηνύματα EAP.

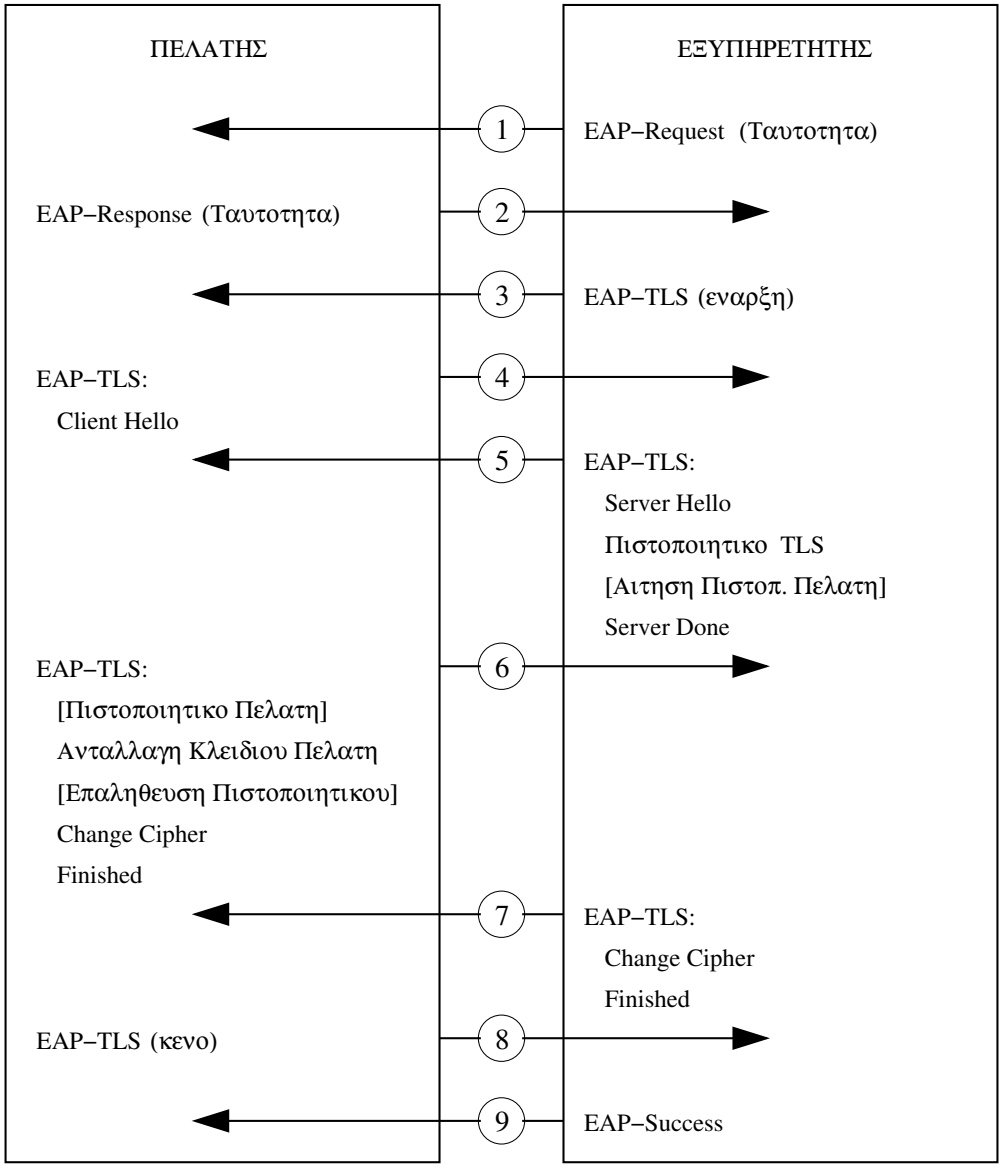
7. {request} Ο εξυπηρετητής στέλνει όλα τα μηνύματα που απομένουν σε μία μόνο αίτηση EAP.

8. {response} Ο πελάτης δεν έχει επιπλέον πληροφορίες να στείλει αλλά απαιτείται από το πρωτόκολλο να αποκριθεί και στο πλαίσιο αυτό απαντά με ένα κενό μήνυμα `EAP-Response`.

9. Τελικά για την ολοκλήρωση της χειραψίας EAP, ο εξυπηρετητής στέλνει ένα μήνυμα `EAP-Success`, υποθέτοντας ότι όλα έχουν πάει καλά. Αν οποιοδήποτε από τα βήματα είχαν αποτύχει, ο εξυπηρετητής θα είχε στείλει μήνυμα `EAP-Failure` στο σημείο που εντοπίστηκε το πρόβλημα.

Η χρήση του EAP αποτελεί κλειδί στην υλοποίηση του TLS στο TSN ή το RSN. Κατ' αρχήν, συνεπάγεται ότι δε χρειάζεται διεύθυνση IP, επομένως η ασύρματη συσκευή μπορεί να ανταλλάξει μηνύματα EAP με το σημείο πρόσβασης και να πραγματοποιήσει τη χειραψία προτού της δοθεί πρόσβαση στο ενσύρματο δίκτυο. Το σημείο πρόσβασης δεν απαιτείται να υποστηρίζει το πρωτόκολλο TLS για την ολοκλήρωση της συναλλαγής, εφ' όσον έχει στη διάθεσή του έναν εξυπηρετητή επαλήθευσης ταυτότητας για να του στείλει τα μηνύματα EAP. Το σημείο πρόσβασης μπορεί να περιμένει για μήνυμα `EAP-Success` το οποίο σηματοδοτεί την επίτρεψη πρόσβασης στο δίκτυο.

Ο τρόπος με το οποίο το σημείο πρόσβασης στέλνει τα μηνύματα EAP στον εξυπηρετητή επαλήθευσης ταυτότητας έγκειται στη χρήση του RADIUS. Πρόκειται για ένα πρωτόκολλο



Σχήμα 5.14: Χειραψία EAP-TLS

που επιτρέπει την επικοινωνία με τον εξυπηρετητή επαλήθευσης ταυτότητας. Έχει επεκταθεί σε μεγάλο βαθμό σε σχέση με τον αρχικό του σχεδιασμό, ωστόσο οι βασικές αρχές δεν έχουν μεταβληθεί. Μία από τις κύριες επεκτάσεις του σε σχέση με τα TSN/RSN, αποτελεί η υποστήριξη της προώθησης των αιτήσεων και αποκρίσεων EAP απ΄ ευθείας στον εξυπηρετητή. Το RADIUS και οι σχετικοί μηχανισμοί περιγράφονται αναλυτικά στην ενότητα 5.3.

5.2.6 Προστατευμένο EAP

Το Προστατευμένο EAP (Protected EAP – PEAP), όπως δηλώνει και το όνομά του, παρέχει έναν ασφαλή μηχανισμό για τις διαδικασίες του EAP. Το αρχικό κίνητρο ήταν να εξασφαλιστεί η ασφάλεια των κωδικών πρόσβασης των χρηστών προστατεύοντας τους από επιθέσεις με λεξικό. Για να επιτευχθεί αυτό, κάθε σύνοδος EAP είναι απόλυτα μυστική από τους επίδοξους εισβολείς.

Αρχικά πρέπει να εξεταστούν οι αδυναμίες σε επίπεδο ασφαλείας του EAP. Υπάρχει ο κεντρικός μηχανισμός επαλήθευσης ταυτότητας μεταξύ του πελάτη και του εξυπηρετητή. Ο μηχανισμός αυτός μπορεί να χρησιμοποιεί μέθοδο TLS και να θεωρείται ασφαλής, όπως έχουμε ήδη αναφέρει. Ωστόσο, αυτό που είναι κοινό σε όλες τις μεθόδους EAP είναι η φάση `EAP-Identity` και τα μηνύματα `EAP-Success` ή `EAP-Fail` στο τέλος. Σε αυτές τις φάσεις συναντώνται και οι αδυναμίες στην ασφάλεια:

- Επειδή το μήνυμα `EAP-Identity` δεν προστατεύεται, μπορεί να υποκλαπεί, αποκαλύπτοντας την ταυτότητα του χρήστη που επιχειρεί να συνδεθεί.
- Το μήνυμα `EAP-Success/Fail` δεν προστατεύεται και θα μπορούσε να υποκλαπεί.

Μία λύση και στα δύο αυτά προβλήματα θα ήταν να πραγματοποιούνται οι διαπραγματεύσεις του EAP μέσα σε μία απόρρητη κρυπτογραφημένη σήραγγα (tunnel). Αν υπάρχει ασφαλής σύνδεση μεταξύ του πελάτη και του εξυπηρετητή, τότε οι διαπραγματεύσεις του EAP μπορούν να λάβουν χώρα με αρκετή ασφάλεια και η ταυτότητα του πελάτη δε θα αποκαλυφθεί. Ταυτόχρονα η ευελιξία που προσφέρει το EAP δε χάνεται, καθώς όλες οι μέθοδοι επαλήθευσης ταυτότητας ανωτέρων στρωμάτων εξακολουθούν να μπορούν να χρησιμοποιηθούν. Αυτή είναι η βάσικη ιδέα του PEAP: όλες οι διαπραγματεύσεις του EAP προστατεύονται.

Το απόρρητο (privacy) και η αυθεντικότητα (authenticity) αποτελούν βασικές αρχές της ασφάλειας. Το απόρρητο έχει την έννοια ότι προστατεύεται η μυστικότητα της επικοινωνίας. Η αυθεντικότητα σημαίνει ότι δύο (ή περισσότερες) πλευρές μπορούν να αποδείξουν αμοιβαία την ταυτότητά τους. Στόχος του EAP αποτελεί η αυθεντικότητα: Πρωτόκολλο Επεκτάσιμης Επαλήθευσης Ταυτότητας. Στόχος του PEAP είναι να εξασφαλιστεί το απόρρητο κατά τη διαδικασία επαλήθευσης ταυτότητας. Για την επίτευξη και των δύο στόχων, αρχικά εξασφαλίζουμε το απόρρητο *χωρίς* αυθεντικότητα, και στη συνέχεια πραγματοποιείται η επαλήθευση ταυτότητας χρησιμοποιώντας την απόρρητη σύνδεση. Η προσέγγιση που περιγράφηκε αποτελείται από δύο φάσεις:

i Κατά την πρώτη φάση, το EAP χρησιμοποιείται συμβατικά για την εγκατάσταση ασφαλούς σύνδεσης με τη βοήθεια του TLS. Μόνο η ταυτότητα του εξυπηρετητή επαληθεύεται σε αυτή τη φάση.

ii Κατά τη δεύτερη φάση, η ασφαλής σύνδεση χρησιμοποιείται για την πραγματοποίηση των διαπραγματεύσεων του EAP, στα πλαίσια των οποίων, λαμβάνει χώρα πλήρης επαλήθευση ταυτοτήτων.

Το TLS είναι η μέθοδος που έχει επιλεγεί για την εξασφάλιση του απορρήτου κατά την πρώτη φάση. Ωστόσο, μόλις το ασφαλές κανάλι εγκατασταθεί, οποιαδήποτε μέθοδος που υποστηρίζεται από το EAP θα μπορούσε να χρησιμοποιηθεί για τις διαπραγματεύσεις· δεν πρέπει να είναι απαραίτητα το TLS.

Πρέπει να τονιστεί ότι, η πρώτη φάση του PEAP περιλαμβάνει μέρος των διαδικασιών που εξασφαλίζουν την αυθεντικότητα· ο εξυπηρετητής απαιτείται πάντα να αποδεικνύει την ταυτότητά του. Αυτό μπορεί να γίνει με τη χρήση κάποιου πιστοποιητικού, όπως περιγράφηκε στην ενότητα 5.2.5 για το TLS. Με τον τρόπο αυτό επιτρέπει στον πελάτη να είναι σίγουρος για τη νομιμότητα του εξυπηρετητή. Αυτό είναι ιδιαίτερα σημαντικό για τα ασύρματα τοπικά δίκτυα γιατί είναι σχετικά εύκολο να εγκατασταθούν σημεία πρόσβασης τα οποία να διαφημίζουν ψευδώς ότι ανήκουν σε κάποιο έγκυρο δίκτυο.

Στη συνέχεια περιγράφουμε αναλυτικά τις δύο φάσεις του PEAP.

Φάση 1

Εξωτερικά, η φάση 1, φαίνεται όπως οι συνήθεις διαπραγματεύσεις του EAP. Η διαφορά σε σχέση με την περίπτωση του EAP-TLS, είναι ότι στο τέλος της φάσης, αντί να σταλεί μήνυμα `EAP-Success`, οι διαπραγματεύσεις προχωρούν στη δεύτερη φάση, όπου αρχίζει μια εξ΄ ολοκλήρου νέα σύνοδος, κρυπτογραφημένη χρησιμοποιώντας τα κλειδιά που έχουν ήδη διαπραγματευτεί.

Στην αρχή, τόσο της φάσης 1 όσο και της 2, ο εξυπηρετητής στέλνει μήνυμα `EAP-Request/Identity`. Ο πελάτης οφείλει να απαντήσει με απόκριση ταυτότητας. Ωστόσο, επιτρέπεται στον πελάτη να στείλει μια ανώνυμη ταυτότητα στον πρώτο γύρο του EAP. Στο συμβατικό EAP, η ταυτότητα χρησιμοποιείται συχνά για τον προσδιορισμό της μεθόδου επαλήθευσης ταυτότητας ανωτέρων στρωμάτων που θα χρησιμοποιηθεί. Αυτό ισχύει και στο PEAP, όπου η ταυτότητα που στέλνεται στην πρώτη φάση προσδιορίζει στον εξυπηρετητή τη χρήση του PEAP. Μπορεί να είναι κάποιο τυχαίο αναγνωριστικό, για παράδειγμα `'peap@anonymous.com'`. Η πραγματική ταυτότητα του πελάτη στέλνεται κατά τη διάρκεια της φάσης 2. Μερικές φορές ο επαληθευτής ταυτότητας χρησιμοποιεί το αναγνωριστικό αυτό για να προσδιορίσει σε ποιον εξυπηρετητή θα απευθυνθεί για τις αποφάσεις επαλήθευσης ταυτότητας. Αυτό μπορεί να βρει εφαρμογή σε ένα σημείο πρόσβασης που εξυπηρετεί πολλά δίκτυα που δε διαχειρίζονται από την ίδια αρχή. Στην περίπτωση αυτή, μπορεί να σταλεί ένα αναγνωριστικό του τύπου `'anonymous@Network_A.com'`, όπου το τμήμα του αναγνωριστικού που αναφέρεται στο δίκτυο είναι πραγματικό, ενώ το όνομα του χρήστη δίνεται αργότερα, όταν εγκατασταθεί η ασφαλής σύνδεση με τον εξυπηρετητή του δικτύου αυτού.

Κατά τη διάρκεια των διαπραγματεύσεων του TLS, ο εξυπηρετητής μπορεί να ζητήσει πιστοποιητικό από τον πελάτη. Παρέχοντας ένα τέτοιο πιστοποιητικό εξασφαλίζεται η αυθεντικότητα του πελάτη. Ωστόσο, στο PEAP, ο χρήστης έχει το δικαίωμα να αρνηθεί να στείλει αυτό το πιστοποιητικό, οπότε ο εξυπηρετητής περνάει στη φάση 2. Σε αντίθετη περίπτωση δεν υπάρχει νόημα να προχωρήσει η διαδικασία στη δεύτερη φάση, καθώς η αμοιβαία επαλήθευση ταυτοτήτων θα έχει ολοκληρωθεί από τη φάση 1.

Φάση 2

Η δεύτερη φάση αποτελεί μια συμβατική διαδικασία EAP, που επιτρέπει να χρησιμοποιηθεί οποιαδήποτε μέθοδος επαλήθευσης ταυτότητας ανωτέρων στρωμάτων υποστηρίζει ο εξυπηρετητής. Η μοναδική διαφορά είναι ότι όλα τα μηνύματα EAP στέλνονται χρησιμοποιώντας την κρυπτογραφημένη σύνοδο που έχει δημιουργηθεί από τη φάση 1. Συνεπώς, είναι αρκετά ασφαλές να σταλεί η πραγματική ταυτότητα του πελάτη. Ο επαληθευτής ταυτότητας δε συγκρίνει την ταυτότητα αυτή με εκείνη που έχει λάβει από την πρώτη φάση, η οποία, όπως αναφέραμε παραπάνω, δεν έχει νόημα.

Το PEAP επιτρέπει σε έναν κακόβουλο χρήστη να περάσει τη φάση 1 χωρίς πρόκληση. Επειδή δεν υπάρχει επαλήθευση ταυτότητας, οποιοσδήποτε μπορεί να κάνει διαπραγματεύσεις TLS και να εγκαταστήσει ασφαλή σύνδεση με τον εξυπηρετητή επαλήθευσης ταυτότητας. Επομένως, στην αρχή της φάσης 2, ο πελάτης δε θεωρείται έμπιστος και είναι υποχρεωμένος να επαληθεύσει την ταυτότητά του, παρόλο που χρησιμοποιεί ασφαλή σύνοδο. Έτσι, σε περίπτωση αποτυχίας ο εξυπηρετητής τον αποσυνδέει.

5.3 Remote Access Dial-In User Service (RADIUS)

Στην ενότητα αυτή, παρουσιάζεται το πρωτόκολλο RADIUS, το οποίο, αν και δεν αποτελεί μέρος του προτύπου IEEE 802.11i, χρησιμοποιείται πολύ στην πράξη για την επικοινωνία μεταξύ του σημείου πρόσβασης και του εξυπηρετητή επαλήθευσης ταυτότητας. Συναντάται κυρίως σε εταιρικά και ευρείας κλίμακας δίκτυα και σπάνια σε οικιακές εγκαταστάσεις, καθώς, στις τελευταίες περιπτώσεις, ο εξυπηρετητής επαλήθευσης ταυτότητας είναι συνήθως ενσωματωμενος στο σημείο πρόσβασης.

Το RADIUS ορίζει, κατ΄ αρχήν, ένα σύνολο λειτουργιών που οφείλουν να είναι κοινές μεταξύ όλων των εξυπηρετητών επαλήθευσης ταυτότητας. Επιπλέον, ορίζει ένα πρωτόκολλο που επιτρέπει σε άλλες συσκευές να έχουν πρόσβαση στις παραπάνω λειτουργίες. Όταν αναφερόμαστε στον *εξυπηρετητή* RADIUS, εννοούμε το τμήμα του εξυπηρετητή επαλήθευσης ταυτότητας το οποίο υποστηρίζει λειτουργίες RADIUS· το RADIUS γενικά αναφέρεται στο πρωτόκολλο που χρησιμοποιείται για την επικοινωνία με τον εξυπηρετητή. Ο τελευταίος μπορεί να είναι **πλεονάζων**, δηλαδή να υπάρχουν εφεδρικές μονάδες έτοιμες να αναλάβουν σε περίπτωση αποτυχίας του κυρίου εξυπηρετητή, και **κατανεμημένος**, που σημαίνει ότι πολλοί εξυπηρετητές λειτουργούν σε διαφορετικές τοποθεσίες χρησιμοποιώντας μια κοινή βάση δεδομένων.

Το RADIUS, που ορίζεται από την IETF, έχει σχεδιαστεί για δίκτυα τύπου TCP/IP· υποθέτει ότι οι συσκευές χρησιμοποιούν ένα δίκτυο IP για να επικοινωνούν με τον εξυπηρετητή RADIUS. Το αρχικό κίνητρο πίσω από την ανάπτυξη του RADIUS ήταν η υποστήριξη δεξαμενών διαποδιαμορφωτών (modem pools) για υπηρεσίες dial-in. Ένας ISP που παρέχει πρόσβαση dial-up σε εθνικό επίπεδο οφείλει να έχει εγκαταστήσει από μια δεξαμενή διαποδιαμορφωτών σε κάθε τοπική τηλεφωνική περιοχή, ώστε οι πελάτες να μην πληρώνουν υπεραστικές κλήσεις. Σε κάθε τέτοια δεξαμενή, ο εξυπηρετητής dial-in απαντά στις κλήσεις, επαληθεύει την ταυτότητα του χρήστη για να πιστοποιήσει την εγκυρότητά του, και στη συνέχεια εκτελεί το πρωτόκολλο PPP για να επιτρέψει τη σύνδεση του πελάτη στο διαδίκτυο. Το πρόβλημα είναι ότι ο εξυπηρετητής σε κάθε δεξαμενή διαποδιαμορφωτών πρέπει να γνωριζει όλους τους έγκυρους χρήστες προκειμένου να πραγματοποιήσει τη διαδικασία επαλήθευσης ταυτότητας. Η λογική του RADIUS είναι ότι υπάρχει ένας κεντρικός εξυπηρετητής επαλήθευσης ταυτότητας, ο οποίος γνωρίζει όλους τους πελάτες και επιτρέπει στους εξυπηρετητές των δεξαμενών διαποδιαμορφωτών να προωθούν τις πληροφορίες σχετικές με την επαλήθευση ταυτότητας στα κεντρικά για έλεγχο. Στην ορολογία του RADIUS, ο εξυπηρετητής των δεξαμενών διαποδιαμορφωτών αποτελεί τον εξυπηρετητή πρόσβασης δικτύου (Network Access Server – NAS) και ο εξυπηρετητής επαλήθευσης ταυτότητας (Authentication Server – AS) είναι ο εξυπηρετητής RADIUS.

Η αναλογία στα ασύρματα τοπικά δίκτυα είναι φανερή· το σημείο πρόσβασης αποτελεί μια οντότητα NAS και, προφανώς, δεν είναι θεμιτό *κάθε* σημείο πρόσβασης να γνωρίζει τη βάση δεδομένων που περιέχει τις πληροφορίες επαλήθευσης ταυτότητας. Στο πλαίσο αυτό, χρησιμοποιείται ο εξυπηρετητής RADIUS ο οποίος παρέχει ένα κεντρικοποιημένο μηχανισμό επαλήθευσης ταυτότητας.

5.3.1 Οι μηχανισμοί του RADIUS

Στην ενότητα αυτή, παρουσιάζονται οι μηχανισμοί του πρωτοκόλλου. Το βασικό σύνολο των μηνυμάτων RADIUS είναι περιορισμένο, ωστόσο υπάρχει σχετική πολυπλοκότητα όσον αφορά τα διάφορα γνωρίσματα (attributes) που μεταφέρουν τα μηνύματα.

Βασικά Μηνύματα

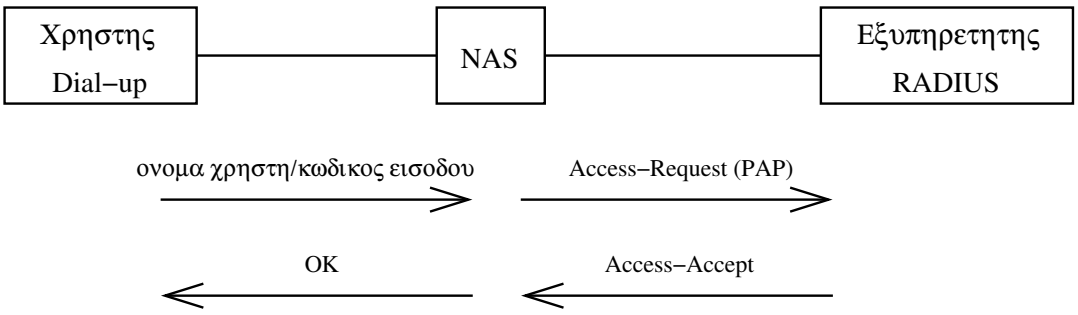
Το πρωτόκολλο RADIUS υποστηρίζει τους παρακάτω τέσσερις τύπους μηνυμάτων:

- `Access-Request` (NAS ⟶ AS)
- `Access-Challenge` (NAS ⟵ AS)
- `Access-Accept` (NAS ⟵ AS)
- `Access-Reject` (NAS ⟵ AS)

όπου στα πλαίσια των TSN/RSN, το σημείο πρόσβασης είναι το ισοδύναμο της οντότητας του NAS και AS είναι ο επαληθευτής ταυτότητας RADIUS.

Τα τέσσερα αυτά μηνύματα, σχεδιάστηκαν για τις ανάγκες του PPP, του πρωτοκόλλου για τις υπηρεσίες dial-in. Υποστηρίζονται δύο επιλογές για επαλήθευση ταυτότητας: το PAP και το CHAP. Το PAP αποτελεί μια απλή προσέγγιση χρήσης ονόματος χρήστη/κωδικού εισόδου. Το CHAP από την άλλη πλευρά, απαιτεί από τον εξυπηρετητή να στείλει τυχαία δεδομένα που αποτελούν την **πρόκληση**, και την οποία το σύστημα dial-in πρέπει να κρυπτογραφήσει και να επιστρέψει στη συνέχεια για έλεγχο.

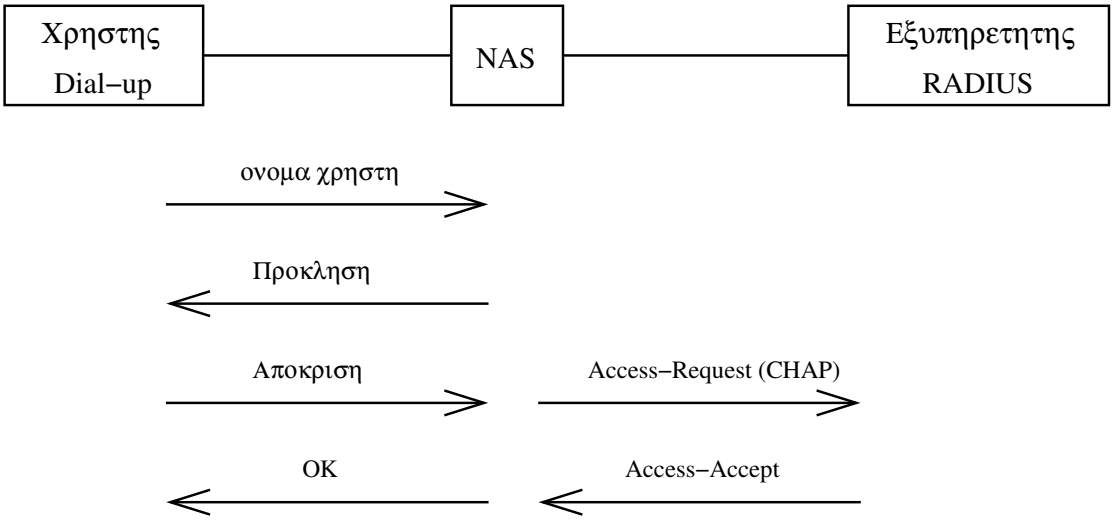
Αρχικά εξετάζουμε την περίπτωση του PAP, που φαίνεται στο Σχ. 5.15. Ο χρήστης συνδέεται με το διαποδιαμορφωτή και ο NAS δηλώνει ότι χρησιμοποιείται το PAP για επαλήθευση ταυτότητας. Το σύστημα του πελάτη αποκρίνεται στέλνοντας το όνομα χρήστη και τον κωδικό εισόδου του λογαριασμού του. Ο NAS στέλνει τώρα μήνυμα `Access-Request` στον εξυπηρετητή RADIUS, το οποίο περιέχει τις πληροφορίες του χρήστη κρυπτογραφημένες. Ο εξυπηρετητής RADIUS αποκρίνεται είτε με `Access-Accept` ή με `Access-Reject` και ο NAS ενεργεί ανάλογα. Αυτή είναι μια πολύ απλή προσέγγιση και είναι ευάλωτη σε πολλών ειδών κακόβουλες επιθέσεις. Η μεγαλύτερη αδυναμία του PAP είναι ότι ο κωδικός εισόδου στέλνεται χωρίς κρυπτογράφηση από την τηλεφωνική γραμμή δίνοντας την ευκαιρία, σε οποιονδήποτε παρακολουθεί τη γραμμή, να το υποκλέψει. Το πρωτόκολλο αυτό, λοιπόν, υποθέτει ότι οι τηλεπικοινωνιακές γραμμές είναι ασφαλείς.



Σχήμα 5.15: Ο Μηχανισμός του PAP

Το CHAP είναι λίγο πιο βελτιωμένο κάνοντας μια απόπειρα για πιο ασφαλή επαλήθευση ταυτότητας, όπως φαίνεται στο Σχ. 5.16. Αντί να στέλνει τον κωδικό εισόδου χωρίς κρυπτογράφηση από την τηλεφωνική γραμμή, στέλνει μόνο το αναγνωριστικό του χρήστη στο NAS. Ο τελευταίος πρέπει τώρα να αποκριθεί με πρόκληση. Στο πλαίσιο αυτό, ο NAS στέλνει το αναγνωριστικό χρήστη στον εξυπηρετητή με μήνυμα `Access-Challenge`, αντί για `Access-Request`. Ωστόσο, στις περισσότερες υλοποιήσεις, ο NAS δεν απευθύνεται στον εξυπηρετητή και παράγει την πρόκληση μόνος του, σύμφωνα και με το Σχ. 5.16. Η πρόκληση

στέλνεται στο σύστημα του χρήστη το οποίο την κρυπτογραφεί με βάση τον κωδικό εισόδου και τη στέλνει με τη σειρά του πίσω. Τελικά, ο NAS είναι σε θέση να στείλει την πρόκληση, απάντηση και ταυτότητα στον AS, δηλώνοντας ότι χρησιμοποιεί το CHAP.



Σχήμα 5.16: Ο Μηχανισμός του CHAP

Σύμφωνα με την προσέγγιση αυτή, ο κωδικός εισόδου στέλνεται κρυπτογραφημένος· επιπλέον παρουσιάζει το πλεονέκτημα ότι η πρόκληση αλλάζει σε κάθε απόπειρα πρόσβασης. Ωστόσο, ο όλος μηχανισμός εξακολουθεί να είναι ευάλωτος σε επίθεση με λεξικό καθώς, τόσο οι κρυπτογραφημένες όσο και μη κρυπτογραφημένες εκδόσεις της πρόκλησης είναι προσβάσιμες για ένα ενδεχόμενο εισβολέα.

Αυτή η αδυναμία της επίθεσης με λεξικό, οδήγησε τη Microsoft στην ανάπτυξη μιας τροποποιημένης έκδοσης του CHAP, το MS-CHAP που χρησιμοποιείται ευρέως σήμερα. Τα γνωρίσματα RADIUS που σχετίζονται με το πρωτόκολλο αυτό, ορίζονται στο [12].

Το RADIUS σχεδιάστηκε, αρχικά, αποκλειστικά για το PPP και προέβλεπε δύο σενάρια επαλήθευσης ταυτότητας: την απλή αίτηση με κωδικό εισόδου του PAP και την πρόκληση-απόκριση του CHAP. Στα πλαίσια των TSN/RSN, απαιτείται η χρήση του RADIUS με ένα πιο προηγμένο πρωτόκολλο ασφαλείας σε σχέση με τα PAP και CHAP. Για το σκοπό αυτό, πρέπει να αλλάξει η λειτουργικότητα κάποιων από τα μηνύματα του RADIUS. Έτσι, για την υποστήριξη του EAP, χρησιμοποιείται η μέθοδος πρόσβασης – πρόκλησης, όχι απλά σαν πρόκληση, αλλά για την αποστολή των αιτήσεων και αποκρίσεων του EAP. Το RADIUS με την ευελιξία που του προσφέρει η χρήση των γνωρισμάτων έχει προσαρμοστεί στις ανάγκες αυτές.

Παρόλο που υπάρχουν ουσιαστικά μόνο τέσσερα μηνύματα RADIUS σχετικά με τη διαδικασία επαλήθευσης ταυτότητας, το νόημα αυτών μπορεί να αλλάξει ανάλογα με τη χρήση των αναγνωριστικών. Στα Σχ. 5.15 και 5.16, φαίνεται ότι το μήνυμα `Access-Request` μπορεί να έχει τρεις διαφορετικές λειτουργίες, καθώς τα γνωρίσματα που μεταφέρει κάθε φορά είναι διαφορετικά. Κάθε μήνυμα RADIUS έχει τη βασική μορφή που απεικονίζεται στο Σχ. 5.17.

Κωδικος	Αναγνωριστικο	Μηκος	Επαληθευτης Ταυτοτητας	Γνωρισματα...

Σχήμα 5.17: Βασική μορφή μηνυμάτων RADIUS

Το πεδίο Κωδικός υποδεικνύει τον τύπο του μηνύματος:

- Access-Request: 1
- Access-Challenge: 2
- Access-Accept: 3
- Access-Reject: 11

Το Αναγνωριστικό είναι ένας τυχαίος αριθμός που αντιστοιχεί τις αιτήσεις με τις απαντήσεις, ενώ το Μήκος υποδεικνύει το συνολικό μέγεθος του μηνύματος σε byte. Το πεδίο Επαληθευτής Ταυτότητας είναι σημαντικό, καθώς σχετίζεται με τους μηχανισμούς ασφαλείας. Ανάλογα με τον τύπο του μηνύματος, μπορεί να έχει τις ακόλουθες χρήσεις:

Στην περίπτωση του μηνύματος Access-Request, το Επαληθευτής Ταυτότητας παίρνει μια τιμή που είναι πάντα διαφορετική για κάθε αίτηση. Αυτό γίνεται για δύο λόγους. Κατ' αρχήν, αν το Access-Request μεταφέρει έναν κωδικό εισόδου σε κάποιο του γνώρισμα, τότε αυτό κρυπτογραφείται με βάση το συνδυασμό ενός μυστικού κλειδιού και του εν λόγω αριθμού. Επιπλέον, τα μηνύματα-απαντήσεις χρησιμοποιούν τον αριθμό του πεδίου Επαληθευτής Ταυτότητας για να ελέγχουν την ακεραιότητα των μηνυμάτων, διαδικασία που περιγράφεται στη συνέχεια.

Ένα εκ των μηνυμάτων, Access-Accept, Access-Reject και Access-Challenge, στέλνεται ως απόκριση στο Access-Request. Είναι σημαντικό να γίνει κατάλληλος έλεγχος ώστε να εξασφαλιστεί ότι η απάντηση προέρχεται από το νόμιμο εξυπηρετητή RADIUS και ότι δεν έχει τροποποιηθεί στη διαδρομή. Για το σκοπό αυτό, υπολογίζεται κατάλληλα η τιμή ελέγχου ακεραιότητας και τοποθετείται στο πεδίο Επαληθευτής Ταυτότητας του μηνύματος της απάντησης.

Ο NAS και ο εξυπηρετητής RADIUS μοιράζονται ένα μυστικό κλειδί. Για τη δημιουργία της τιμής ελέγχου, ο εξυπηρετητής RADIUS συνδυάζει ολόκληρο το μήνυμα της απάντησης με το μυστικό κλειδί. Πριν, τον υπολογισμό αυτό, εισάγει την τυχαία τιμή από το μήνυμα της αίτησης στο πεδίο Επαληθευτής Ταυτότητας του μηνύματος της απάντησης, και μόλις ολοκληρωθεί ο υπολογισμός της τιμής ελέγχου ακεραιότητας, δημιουργεί τη νέα τιμή του Επαληθευτής Ταυτότητας και αντικαθιστά την παλιά. Στα πλαίσια αυτά, είναι δύσκολη η παραγωγή μια ψευδούς απάντησης αν δεν είναι γνωστό το μυστικό κλειδί, ενώ η χρήση του τυχαίου αριθμού δεν επιτρέπει την αναπαραγωγή παλιών μηνυμάτων.

Γνωρίσματα

Η χρήσιμη πληροφορία που μεταφέρεται στα μηνύματα RADIUS περιέχεται στα γνωρίσματα. Κάθε μήνυμα μπορεί να μεταφέρει ένα ή περισσότερα γνωρίσματα και καθένα από αυτά αποτελεί ένα αυτόνομο πακέτο πληροφορίας. Επέκταση του πρωτοκόλλου RADIUS σημαίνει ουσιαστικά ορισμός και υποστήριξη νέων γνωρισμάτων. Το πρότυπο του WPA βασίζεται σε ήδη ορισμένα γνωρίσματα του πρωτοκόλλου και συνεπώς η υποστήριξη RADIUS στα ασύρματα τοπικά δίκτυα δεν παρουσιάζει προβλήματα.

Υπάρχει μεγάλο πλήθος γνωρισμάτων RADIUS. Στον Πίνακα 5.1, παρουσιάζονται κάποια από τα πιο κοινά γνωρίσματα.

Όνομα	Περιγραφή
User-Name	Το αναγνωριστικό όνομα του χρήστη.
User-Password	Περιέχει τον κωδικό πρόσβασης. Τα δεδομένα του κωδικού πρόσβασης κρυπτογραφούνται χρησιμοποιώντας ένα μεριζόμενο μυστικό κλειδί και την τιμή του πεδίου Authenticator του Access-Request.
CHAP-Password	Κατά τη διάρκεια της διαδικασίας CHAP, ο χρήστης κρυπτογραφεί την πρόκληση και επιστρέφει την τιμή. Αυτή προωθείται από το NAS στον εξυπηρετητή RADIUS μέσα σε αυτό το γνώρισμα.
NAS-IP-Address	Η διεύθυνση IP του NAS στον οποίο πρέπει να αποκρίνεται ο εξυπηρετητής RADIUS.
ReplyMessage	Περιέχει κείμενο το οποίο μπορεί να εμφανισθεί στο χρήστη για να δηλώσει κάποιο γεγονός ή ενέργεια που πρέπει να ληφθεί.

Πίνακας 5.1: Παραδείγματα γνωρισμάτων RADIUS

5.3.2 EAP και RADIUS

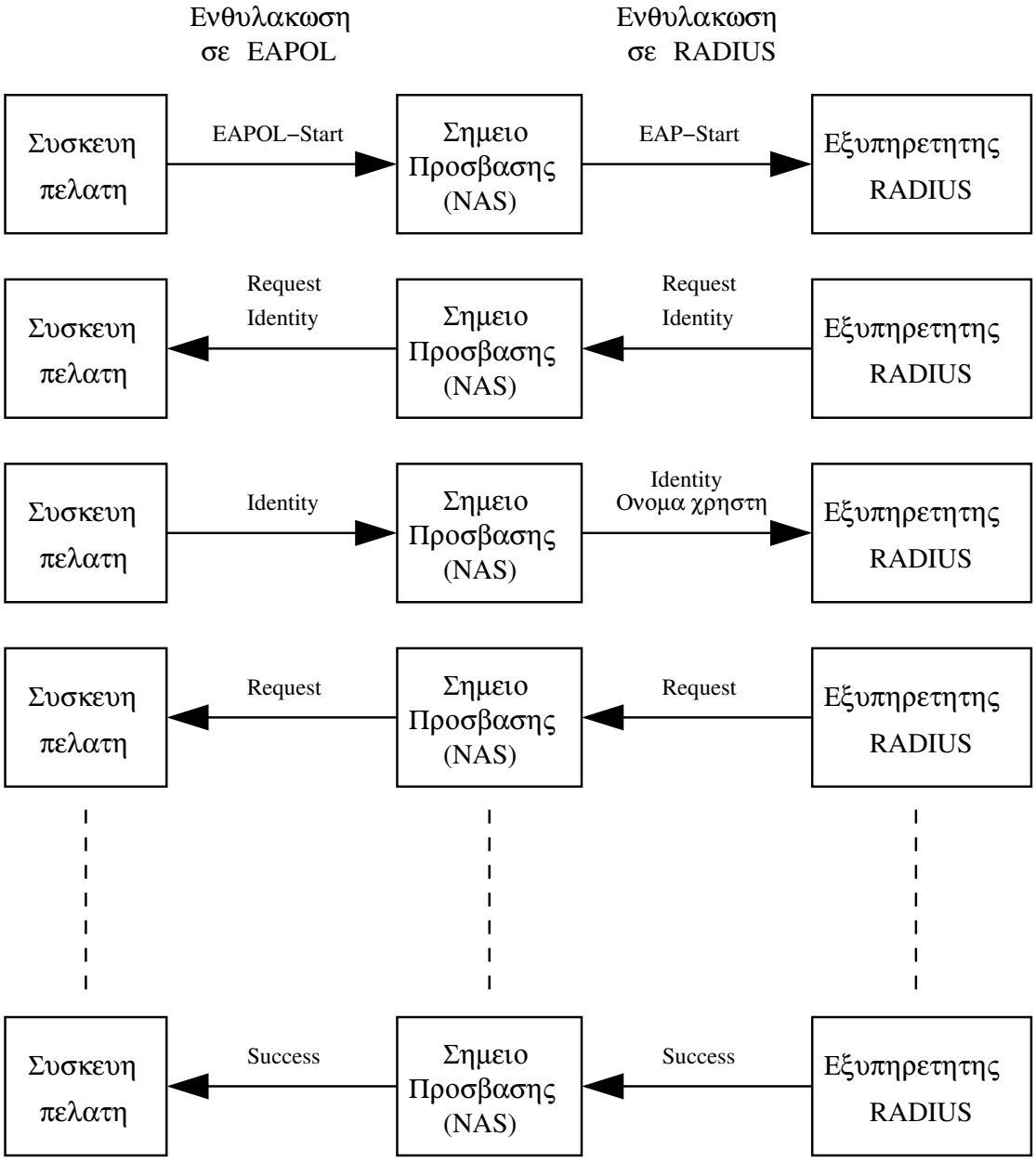
Επειδή το EAP σχεδιάστηκε για να επεκτείνει την επαλήθευση ταυτότητας μέσω των διαποδιαμορφωτών dial-in, και δεδομένου ότι μεγάλο πλήθος δεξαμενών διαποδιαμορφωτών χρησιμοποιούν το RADIUS, δημιουργήθηκε η ανάγκη ανάπτυξης μια μεθόδου για τη μεταφορά του EAP πάνω από το RADIUS. Οι επεκτάσεις του RADIUS που υλοποιούν αυτή τη μέθοδο ορίζονται στα κείμενα [13] και [14] της IETF. Οι επεκτάσεις αυτές σχετίζονται και με τα ασύρματα τοπικά δίκτυα, γιατί τα TSN και RSN χρησιμοποιούν επίσης το EAP. Το ίδιο RFC ορίζει ακόμη διαδικασίες για τη μεταφορά πληροφοριών accounting.

Στο αρχικό πρότυπο του RADIUS, μόνο δύο μηνύματα ήταν διαθέσιμα για την αποστολή πληροφοριών επαλήθευσης ταυτότητας: το Access-Request για την αποστολή δεδομένων από τον NAS στον εξυπηρετητή RADIUS και το Access-Challenge για τη μεταφορά από τον εξυπηρετητή RADIUS στον NAS. Όπως υποδηλώνει και το ίδιο το όνομα, το Access-Challenge σχετίζεται άμεσα με την πρόκληση ποθ χρησιμοποιείται στο πρωτόκολλο CHAP. Ωστόσο, το [13] χρησιμοποιεί αυτόν τον τύπο μηνύματος με περισσότερο γενικό τρόπο για τη λήψη δεδομένων από τον εξυπηρετητή RADIUS. Ως εκ τούτου, τα μηνύματα EAP στέλνονται στα μηνύματα επαλήθευσης ταυτότητας μέσα σε ένα μήνυμα Access-Request και οι απαντήσεις επιστρέφονται μέσα σε ένα Access-Challenge.

Το ίδιο το μήνυμα EAP στέλνεται μέσα σε ένα ή περισσότερα ειδικά γνωρίσματα των οποίων ο τύπος έχει την τιμή 79. Υποστηρίζεται η αποστολή όλων των συνήθων μηνυμάτων EAP. Υπάρχουν ορισμένοι κανόνες που επιτρέπουν στις υπάρχουσες υλοποιήσεις RADIUS να αντιστοιχούν τις αιτήσεις αυτές στις υπαρχουσες συμβάσεις. Για παράδειγμα, η ταυτότητα του χρήστη dial-in στέλνεται συνήθως μέσα σε ένα μήνυμα EAP-Response/Identity. Το μήνυμα αυτό προωθείται από τον εξυπηρετητή RADIUS μέσα σε γνώρισμα EAP, αλλά οι πληροφορίες οι σχετικές με την ταυτότητα πρέπει επίσης να αντιγραφούν στο γνώρισμα User-Name, ώστε οι εξυπηρετητές RADIUS, συμπεριλαμβανομένων των παλιότερων εκδόσεων, να είναι σε θέση να καταλάβουν και να προωθήσουν κατάλληλα το μήνυμα.

Όπως αναφέρθηκε, το πρωτόκολλο EAPOL συμπεριλαμβάνει ένα μήνυμα που λέγεται EAPOL-Start και σηματοδοτεί στον επαληθευτή ταυτότητας την άφιξη μιας νέας συσκευής που επιθυμεί να συνδεθεί. Το [13] ορίζει ένα ανάλογο μήνυμα που λέγεται EAP-Start, το οποίο αποτελεί ένα γνώρισμα EAP χωρίς δεδομένα. Αυτό χρησιμοποιείται από τον NAS για

να σημάνει την έναρξη του εξυπηρετητή RADIUS, όπως φαίνεται στο Σχ. 5.18.



Σχήμα 5.18: Διαδικασία Επαλήθευσης Ταυτότητας Χρησιμοποιώντας EAP πάνω από RADIUS

Στο Σχ. 5.18, έχει σχεδιαστεί το σημείο πρόσβασης στη θέση του NAS υπηρεσιών dial-up, ωστόσο ισχύουν οι ίδιες αρχές. Το σημείο πρόσβασης περιλαμβάνει επίσης ένα επαληθευτή ταυτότητας IEEE 802.1X, ο οποίος επικοινωνεί με το νέο πελάτη (supplicant) χρησιμοποιώντας το EAP. Τα μηνύματα EAP τα οποία θέλει να περάσει ο επαληθευτής ταυτότητας IEEE 802.1X στον εξυπηρετητή επαλήθευσης ταυτότητας ενθυλακώνονται σε RADIUS και στέλνονται στον εξυπηρετητή RADIUS.

Η διαδικασία έχει ως εξής: Αρχικά η νέα συσκευή στέλνει ένα μήνυμα `EAPOL-Start` στον επαληθευτή ταυτότητας του σημείου πρόσβασης. Αν το σημείο πρόσβασης γνωρίζει

ότι ο εξυπηρετητής RADIUS υποστηρίζει το EAP, τότε μπορεί να συνεχίσει και να εκδόσει μήνυμα `EAP-Request/Identity` στη συσκευή του πελάτη και να στείλει την απάντηση απ΄ ευθείας στον εξυπηρετητή. Αν, ωστόσο, δεν είναι σίγουρος για τον εξυπηρετητή, τότε μπορεί να ζητήσει από τον εξυπηρετητή RADIUS να εκκινήσει τη διαδικασία EAP στέλνοντας του μήνυμα `EAP-Start` μέσα σε `Access-Request`. Αν ο εξυπηρετητής RADIUS δεν υποστηρίζει EAP, τότε απαντά με μήνυμα απόρριψης. Σε αντίθετη περίπτωση, στέλνει μήνυμα `EAP-request/Identity` μέσα σε `Access-Challenge` μήνυμα RADIUS. Το Σχ. 5.18, απεικονίζει ένα σενάριο στο οποίο η μέθοδος επαλήθευσης ταυτότητας είναι το TLS. Στο τέλος της διαδικασίας, ένα μήνυμα `EAP-Success` ή `EAP-Fail` γνωστοποιεί το αποτέλεσμα.

5.3.3 Χρήση του RADIUS στο TSN και RSN

Όπως είδαμε στην ενότητα 5.3.2, η λειτουργία του EAP πάνω από το RADIUS ταιριάζει στην αρχιτεκτονικη των TSN/RSN. Ωστόσο, υπάρχει μια σημαντική διαφορά μεταξύ των υπηρεσιών dial-up και Wi-Fi: στην πρώτη περίπτωση, μας ενδιαφέρει μόνο η αρχική επαλήθευση ταυτότητας, ενώ στα TSN/RSN σημασία έχει η δημιουργία ενός γενικότερου ασφαλούς πλαισίου. Για την υπηρεσία dial-up αρκεί να καθοριστεί αν θα επιτραπεί ή όχι η πρόσβαση του χρήστη στο σύστημα. Εξαιτίας της φύσης των τηλεφωνικών γραμμών είναι απίθανο για κάποιον εισβολέα να υποκλέψει ένα διαποδιαμορφωτή dial-in που μόλις έχει συνδεθεί, αν και θεωρητικά η προσέγγιση αυτή είναι εφικτή. Ως εκ τούτου, μόλις η επαλήθευση ταυτότητας έχει ολοκληρωθεί, ο NAS μπορεί να υποθέσει ότι ένας χρήστης εμπιστοσύνης έχει συνδεθεί χωρίς περαιτέρω ανησυχίες. Ωστόσο, στο WLAN είναι πολύ εύκολο να υποκλαπεί μια εγκατεστημένη σύνδεση απλά χρησιμοποιώντας ψευδώς μια νόμιμη διεύθυνση MAC.

Η προστασία από υποκλοπές συνόδου παρέχεται με επαλήθευση ταυτότητας ανά πακέτο και προστασία ακεραιότητας. Για το σκοπό αυτό, ο εξυπηρετητής ταυτότητας πρέπει να περάσει ένα κύριο μυστικό κλειδί στο σημείο πρόσβασης.

Συνεπώς, όσον αφορά τη χρήση του RADIUS στα TSN και RSN, απαιτείται η υποστήριξη του RADIUS και των επεκτάσεων EAP από το σημείο πρόσβασης. Επιπλέον, ο εξυπηρετητής RADIUS δεν πρέπει να είναι σε θέση να καταλαβαίνει μόνο αυτά τα πρωτόκολλα αλλά και να μπορεί να στέλνει το κατάλληλο κλειδί στο σημείο πρόσβασης. Δεν είναι υποχρεωτική η χρήση του RADIUS στην περίπτωση του RSN, ωστόσο στο TSN είναι.

Κεφάλαιο 6

Υλοποίηση Δικτύου Μεταβατικής Ασφάλειας

Στο κεφάλαιο αυτό, παρουσιάζεται η σχεδίαση και η υλοποίηση ενός Ασύρματου Τοπικού Δικτύου, βάσει του προτύπου IEEE 802.11i και πιο συγκεκριμένα της προσέγγισης του Δικτύου Μεταβατικής Ασφάλειας. Η εφαρμογή αυτού του δικτύου TSN, έγινε για το Ασύρματο Τοπικό Δίκτυο του Εργαστηρίου Δικτύων Υπολογιστών του Ε.Μ.Π. Στις επόμενες ενότητες, αναλύεται η αρχιτεκτονική του, οι λειτουργίες του, καθώς και, οι μηχανισμοί πρόσβασης και διαχείρισής του. Τέλος, ελέγχεται η σωστή λειτουργία του, όσον αφορά το επίπεδο ασφάλειας που επιτυγχάνει, και γίνεται σύγκριση με το αρχικό πρότυπο ασφαλείας WEP.

6.1 Ανάλυση και σχεδίαση

Στην ενότητα αυτή, παρουσιάζονται οι σχεδιαστικές επιλογές που έγιναν για την υλοποίηση του Ασύρματου Τοπικού Δικτύου TSN του Εργαστηρίου.

6.1.1 Επίπεδα ασφάλειας

Στα πλαίσια σχεδίασης των μηχανισμών ασφαλείας ενός Ασύρματου Τοπικού Δικτύου, μπορούμε να διακρίνουμε τρία επίπεδα:

- Επίπεδο Ασύρματου Τοπικού Δικτύου (WLAN)
- Επίπεδο Ελέγχου Πρόσβασης
- Επίπεδο Επαλήθευσης Ταυτότητας

Μάλιστα, η οργάνωση στα επίπεδα αυτά, ισχύει και σε οποιαδήποτε σύστημα ασφαλείας το οποίο αφορά Τοπικά Δίκτυα γενικότερα.

Το *Επίπεδο WLAN* σχετίζεται με την ίδια την επικοινωνία των κόμβων, στα πλαίσια της οποίας, διαφημίζονται οι διάφορες δυνατότητες του δικτύου και απαντώνται οι αιτήσεις για σύνδεση σε αυτό. Επιπρόσθετα, το επίπεδο αυτό είναι υπεύθυνο για την κρυπτογράφηση και αποκρυπτογράφηση των δεδομένων από τη στιγμή που εγκαθίσταται κάποιο σχήμα ασφαλείας.

Το *Επίπεδο Ελέγχου Πρόσβασης* αναλαμβάνει τη διαχείριση του σχήματος ασφαλείας. Πρέπει να εμποδίζει τη μετάδοση δεδομένων από/προς οποιονδήποτε δεν έχει εγκαταστημένο το παρόν σχήμα ασφαλείας. Το επίπεδο αυτό είναι ευμετάβλητο, με την έννοια ότι ανά πάσα

στιγμή μπορεί να αλλάξει η αντιμετώπιση ενός κινητού τερματικού από φιλική σε εχθρική, ανάλογα με το αποτέλεσμα των διαδικασιών επαλήθευσης ταυτότητας και εγκατάστασης του σχήματος ασφαλείας. Το Επίπεδο Ελέγχου Πρόσβασης επικοινωνεί με εκείνο της Επαλήθευσης Ταυτότητας, προκειμένου να είναι σε θέση να γνωρίζει την ισχύουσα κατάσταση των κινητών τερματικών ως προς το σχήμα ασφαλείας και, επιπλέον, συμμετέχει στην παραγωγή των συσχετισμένων χρονικών κλειδιών.

Το *Επίπεδο Επαλήθευσης Ταυτότητας* βρίσκεται υψηλότερα στην ιεραρχία. Στο επίπεδο αυτό, λαμβάνονται οι αποφάσεις που σχετίζονται με τις ισχύουσες πολιτικές, ενώ τα διάφορα πιστοποιητικά ταυτότητας γίνονται αποδεκτά ή απορρίπτονται. Τελικά, το επίπεδο αυτό είναι σε θέση είτε να αποκλείσει μια κινητή συσκευή ή να παραδώσει τον έλεγχο στο Επίπεδο Ελέγχου Πρόσβασης αν αποδεχτεί την αίτησή της να συνδεθεί στο δίκτυο. Προφανώς, το Επίπεδο WLAN εδρεύει στην ασύρματη συσκευή που περιλαμβάνει το σημείο πρόσβασης. Συνήθως, το Επίπεδο Ελέγχου Πρόσβασης βρίσκεται επίσης στο σημείο πρόσβασης. Παρόλο που σε συστήματα μικρής κλίμακας το Επίπεδο Επαλήθευσης Ταυτότητας ενδέχεται να είναι και αυτό στο σημείο πρόσβασης, στα μεγαλύτερα συστήματα, υλοποιείται από ένα εξυπηρετητή επαλήθευσης ταυτότητας και διαχωρίζεται από τα σημεία πρόσβασης. Η ύπαρξη ενός κεντρικού εξυπηρετητή επαλήθευσης ταυτότητας, διευκολύνει τη διαχείριση της βάσης δεδομένων των χρηστών. Με άλλα λόγια, η προσέγγιση αυτή επιλύει το πρόβλημα της διαχείρισης κλειδιών του WEP και επιτρέπει την ενσωμάτωση των Ασύρματων Τοπικών Δικτύων στο συνολικό σύστημα διαχείρισης ασφαλείας.

Όσον αφορά την κινητή συσκευή, τα επίπεδα είναι όμοια. Συνήθως, το Επίπεδο WLAN υλοποιείται από την ασύρματη κάρτα δικτύωσης και το σχετικό λογισμικό (οδηγοί - drivers). Ο Έλεγχος Πρόσβασης και οι υπηρεσίες Επαλήθευσης Ταυτότητας μπορούν να υλοποιηθούν από το λειτουργικό σύστημα. Στο Σχ. 6.1, φαίνεται η σχέση των επιπέδων ασφαλείας και ο τρόπος που αλληλεπιδρούν μεταξύ τους. Ας σημειωθεί ότι η οντότητα supplicant του σχήματος, αναφέρεται στο τμήμα του λειτουργικού συστήματος της κινητής συσκευής που κάνει την αίτηση προκειμένου να συνδεθεί στο δίκτυο.

Το Ασύρματο Τοπικό Δίκτυο TSN του Εργαστηρίου, σχεδιάστηκε ώστε να υλοποιεί τα τρία αυτά επίπεδα ασφαλείας.

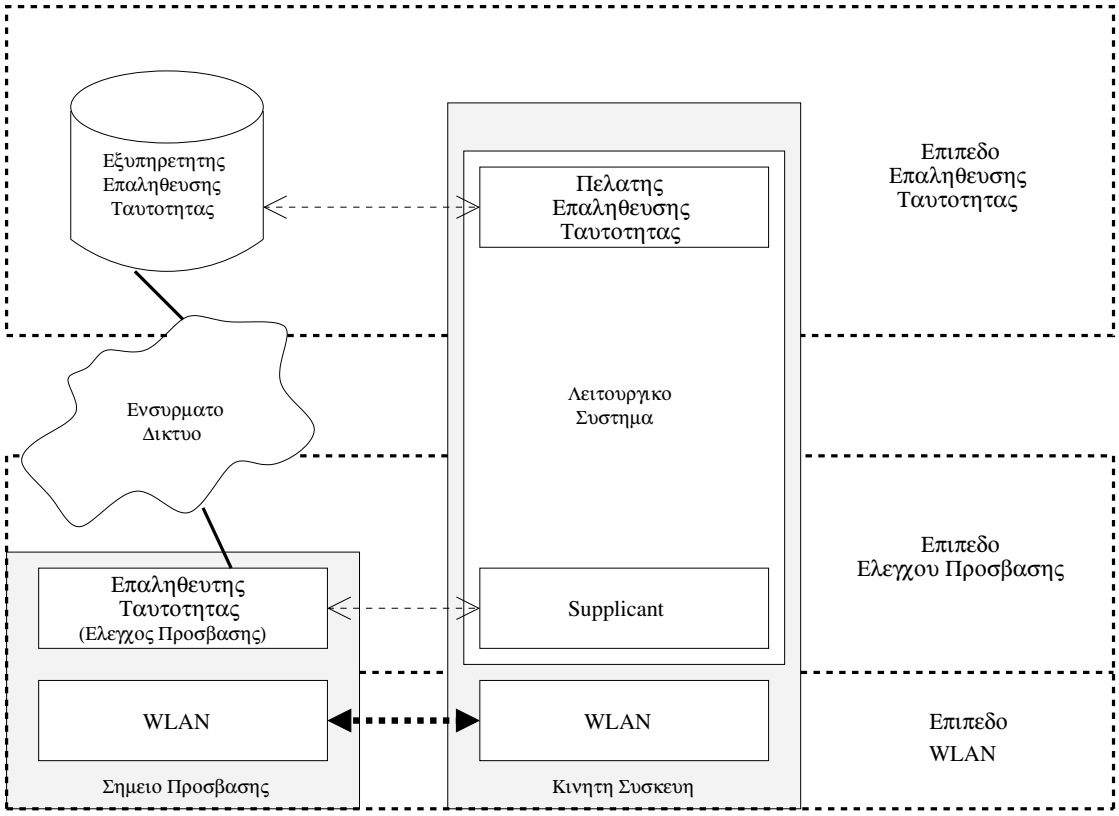
6.1.2 Περιγραφή αρχιτεκτονικής

Ακολουθεί περιγραφή της αρχιτεκτονικής του δικτύου TSN του Εργαστηρίου Υπολογιστών που υλοποιήθηκε και αναλύονται οι λόγοι που χρησιμοποιήθηκαν οι συγκεκριμένες τεχνολογίες.

Πρότυπο IEEE 802.11i

Δεδομένου των αδυναμιών του WEP, που παρουσιάστηκαν λεπτομερώς στην ενότητα 3.4, η υλοποίηση ενός ασφαλούς Ασύρματου Τοπικού Δικτύου απαιτεί τη μετάβαση σε ένα από τα δύο σχήματα ασφαλείας TSN και RSN, τα οποία προτείνονται στα πλαίσια του πρωτοκόλλου IEEE 802.11i. Τόσο το TSN, όσο και το RSN, προβλέπουν μηχανισμούς ελέγχου πρόσβασης και διαχείρισης κλειδιών που δεν υποστηρίζονται στα πλαίσια του WEP. Οι λόγοι που υιοθετήθηκε, τελικά, το TSN, αντί του RSN, είναι οι εξής:

- Το πρωτόκολλο TKIP, που χρησιμοποιείται στο TSN, εισάγει μια σειρά μέτρων που αντιμετωπίζει όλα τα γνωστά ελαττώματα του WEP. Έτσι, παρόλο που το CCMP/AES του RSN θεωρείται ως το *πλέον* ασφαλές πρωτόκολλο ασφαλείας στα πλαίσια των Ασύρματων Τοπικών Δικτύων, εν τούτοις, δεν έχουν αναφερθεί αδυναμίες του TKIP.



Σχήμα 6.1: Σχέση Επιπέδων Ασφαλείας

Μάλιστα, η κρυπτογραφική κοινότητα δεν προβλέπει να γίνει κάτι τέτοιο στο άμεσο μέλλον, γεγονός που καθιστά την επιλογή του TSN ως μια ασφαλή λύση.

- Η προσέγγιση του TSN δεν απαιτεί τροποποιήσεις, όσον αφορά τον τρόπο υλοποίησης του αλγορίθμου κρυπτογράφησης RC4 που επίσης χρησιμοποιεί το WEP. Ως εκ τούτου, τα διορθωτικά μέτρα του TKIP μπορούν να εφαρμοστούν στο υπάρχον υλισμικό απλά με κατάλληλη αναβάθμιση του firmware των σημείων πρόσβασης και των οδηγών (drivers) των ασύρματων δικτυακών διεπαφών. Σε συνδυασμό με το γεγονός ότι οι συσκευές Ασύρματων Τοπικών Δικτύων που είναι συμβατές με το RSN, υποστηρίζουν στο σύνολό τους και το TSN, εξασφαλίζεται η διαλειτουργικότητα (interoperability) μεταξύ παλιότερου και νεότερου εξοπλισμού ασύρματης δικτύωσης.

Πρωτόκολλο Επεκτάσιμης Επαλήθευσης Ταυτότητας

Η τελική επιλογή της μεθόδου EAP που χρησιμοποιήθηκε στην υλοποίηση μας, στηρίζεται στους παρακάτω λόγους:

- Το LEAP είναι ιδιοκτησιακό (proprietary) πρωτόκολλο που σχεδιάστηκε από τη Cisco και έχει γνωστές αδυναμίες που το καθιστούν αδύναμο.

- Η αποστολή των πιστοποιητικών ταυτότητας των χρηστών, που λαμβάνει χώρα κατά την επαλήθευση της ταυτότητας τους, πρέπει να γίνεται σε ασφαλές περιβάλλον, προκειμένου να προστατεύονται ευαίσθητα δεδομένα, όπως είναι οι μυστικοί κωδικοί πρόσβασης. Ως εκ τούτου, προκύπτει η απαίτηση για κρυπτογράφηση της συνόδου της

διαδικασίας της επαλήθευσης ταυτότητας. Η απαίτηση αυτή, ικανοποιείται τόσο από το EAP-TLS, όσο και από το PEAP.

- Η αυθεντικότητα του εξυπηρετητή επαλήθευσης ταυτότητας, μπορεί να εξασφαλιστεί με τη βοήθεια ψηφιακών πιστοποιητικών. Τη δυνατότητα αυτή, προσφέρουν τόσο το EAP-TLS, όσο και από το PEAP.

- Η κύρια διαφορά μεταξύ του EAP-TLS και του PEAP είναι ότι στο πρώτο απαιτείται η *αμοιβαία* αυθεντικότητα εξυπηρετητή και πελάτη, ενώ στο δεύτερο η αυθεντικότητα των πελατών είναι προαιρετική. Παρόλο που το χαρακτηριστικό αυτό, ενισχύει την ασφάλεια του πλαισίου, εν τούτοις, προϋποθέτει ότι κάθε πελάτης έχει προμηθευτεί κατάλληλο ψηφιακό πιστοποιητικό από κάποια αρχή πιστοποίησης (Certification Authority – CA). Προφανώς, το διαχειριστικό κόστος για την έκδοση πιστοποιητικών για το σύνολο των χρηστών ενός δικτύου μεσαίας κλίμακας και άνω, είναι ιδιαίτερα σημαντικό. Επομένως, η λειτουργία του PEAP με πιστοποίηση της αυθεντικότητας του εξυπηρετητή μόνο και όχι των πελατών, αποτελεί την πιο αποδοτική λύση με αποτέλεσμα να προτιμάται η χρήση του έναντι του EAP-TLS.

- Στο παρόν κείμενο έχει τονιστεί η σπουδαιότητα της ασφάλειας στα Ασύρματα Τοπικά Δίκτυα, ωστόσο, ιδιαίτερη σημασία έχει επίσης ο βαθμός ευκολίας της διαδικασίας σύνδεσης των χρηστών σε αυτά. Εξάλλου, δεν πρέπει να παραγνωρίζεται η φιλικότητα που οφείλει να διακρίνει οποιοδήποτε περιβάλλον παροχής υπηρεσιών, που στην προκειμένη περίπτωση αφορά την πρόσβαση σε ένα ασφαλές Ασύρματο Τοπικό Δίκτυο. Από τη μια πλευρά καθιστά τις προσφερόμενες υπηρεσίες περισσότερο ελκυστικές στους χρήστες, ενώ από την άλλη μειώνεται το διαχειριστικό κόστος. Στα πλαίσια αυτά, είναι σημαντικό πλεονέκτημα της μεθόδου PEAP, το γεγονός ότι υποστηρίζεται από το λειτουργικό σύστημα Microsft Windows έκδοσης XP[1], χωρίς εγκατάσταση επιπρόσθετου λογισμικού. Αυτό, αντίθετα, δεν ισχύει στην περίπτωση του EAP-TLS. Ταυτόχρονα, το PEAP υποστηρίζεται και από λειτουργικά συστήματα ανοιχτού κώδικα, όπως είναι το GNU/Linux και η οικογένεια *BSD.

Με βάση τα παραπάνω, η μέθοδος EAP που χρησιμοποιήθηκε στην παρούσα υλοποίηση, είναι το PEAP.

Εξυπηρετητής Επαλήθευσης Ταυτότητας

Παρόλο που δεν αποτελεί μέρος του προτύπου ασφαλείας IEEE 802.11i των Ασύρματων Τοπικών Δικτύων, το πρωτόκολλο RADIUS είναι κατάλληλο για την επικοινωνία μεταξύ του σημείου πρόσβασης και του εξυπηρετητή επαλήθευσης ταυτότητας. Όπως αναφέρθηκε στην ενότητα 5.3.2, οι επεκτάσεις του πρωτοκόλλου επιτρέπουν τη μεταφορά των μηνυμάτων EAP που απαιτούνται για το σκοπό αυτό. Εξάλλου, η χρήση του RADIUS παρουσιάζει τα παρακάτω πλεονεκτήματα:

- Διαχείριση των πιστοποιητικών ταυτότητας των χρηστών σε κάποιον *κεντρικό* εξυπηρετητή επαλήθευσης ταυτότητας και όχι σε κάθε σημείο πρόσβασης. Κατά συνέπεια, το διαχειριστικό κόστος μειώνεται σημαντικά για ένα δίκτυο μεσαίας και άνω κλίμακας.

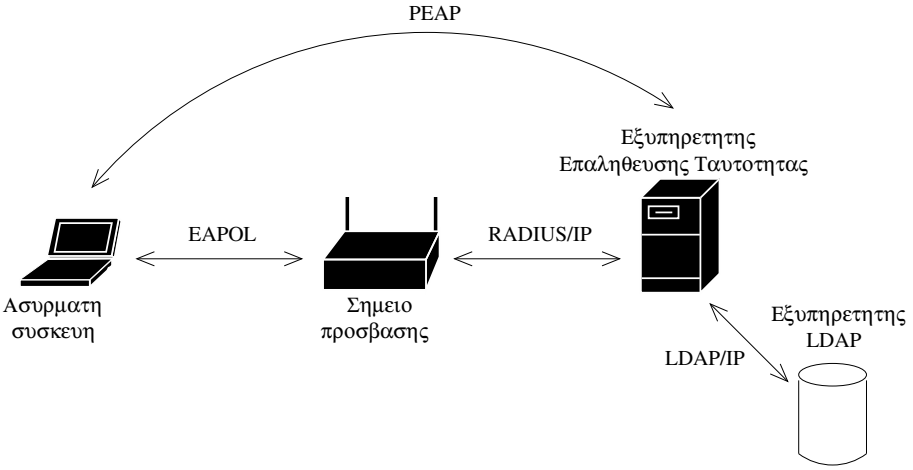
- Διατήρηση πληροφοριών *accounting* σχετικών με τον επιτυχή ή μη συσχετισμό των ασύρματων συσκευών με τα σημεία πρόσβασης, αλλά και καταγραφή των διευθύνσεων IP – MAC των χρηστών, του χρόνου σύνδεσης τους κ.ά.

[1] Προϋποθέτει την εγκατάσταση του Service Pack 2 – SP2.

Τα πιστοποιητικά ταυτότητας των χρηστών διαχειρίζεται μια βάση δεδομένων **Lightweight Directory Access Protocol – LDAP**. Το πρωτόκολλο LDAP αποτελεί το πλέον σύγχρονο πρότυπο, όσον αφορά τις υπηρεσίες καταλόγου, παρέχοντας γρήγορη προσπέλαση στις πληροφορίες που διαχειρίζεται και δυνατότητα κατανεμημένης οργάνωσης τους, ώστε να εξασφαλίζεται λειτουργία με πλεονασμό (redundance).

Αλληλεπίδραση πρωτοκόλλων των τριών οντοτήτων

Στο Σχ. 6.2, φαίνεται η αρχιτεκτονική του δικτύου TSN που υλοποιήθηκε και η αλληλεπίδραση των πρωτοκόλλων των τριών οντοτήτων: ασύρματη συσκευή (supplicant), σημείο πρόσβασης (επαληθευτής ταυτότητας) και εξυπηρετητής επαλήθευσης ταυτότητας (RADIUS).



Σχήμα 6.2: Αρχιτεκτονική πρωτοκόλλων του Ασύρματου Τοπικού Δικτύου TSN

6.1.3 Περιγραφή λειτουργιών

Το Ασύρματο Τοπικό Δίκτυο του Εργαστηρίου Δικτύων Υπολογιστών, παρέχει πρόσβαση σε δύο κατηγορίες χρηστών: *εξουσιοδοτημένα μέλη* και *επισκέπτες*.

Εξουσιοδοτημένα μέλη

Πρόκειται για χρήστες που διαθέτουν λογαριασμό για πρόσβαση στις υπόλοιπες υπηρεσίες του δικτύου δεδομένων του Εργαστηρίου (ηλεκτρονική αλληλογραφία, προσωπική ιστοσελίδα, εξυπηρετητής αρχείων). Κατά τη διαδικασία επαλήθευσης της ταυτότητας τους προκειμένου να συνδεθούν στο Ασύρματο Τοπικό Δίκτυο, χρησιμοποιούνται τα προσωπικά τους αναγνωριστικά (ψευδώνυμο χρήστη και μυστικός κωδικός πρόσβασης) που ισχύουν για το σύνολο των υπηρεσιών. Ουσιαστικά, η ασύρματη πρόσβαση αποτελεί επέκταση των υπηρεσιών δικτύου των εγγεγραμμένων μελών του Εργαστηρίου.

Επισκέπτες

Η κατηγορία των επισκεπτών αφορά χρήστες οι οποίοι δεν είναι εγγεγραμμένα μέλη του Εργαστηρίου και επομένως δε διαθέτουν ειδικό λογαριασμό. Από το σύνολο των υπηρεσιών δικτύου, τους παρέχεται μόνο ασύρματη πρόσβαση στο Διαδίκτυο. Για την επαλήθευση ταυτότητας που λαμβάνει χώρα, αρχικά, κατά τη σύνδεσή τους στο Ασύρματο Τοπικό Δίκτυο,

χρησιμοποιείται *κοινό* αναγνωριστικό που θεωρείται γνωστό στα πλαίσια της υπηρεσίας παροχής ασύρματης πρόσβασης σε επισκέπτες. Επειδή με τον τρόπο αυτό, οποιοσδήποτε μπορεί να συνδεθεί ως επισκέπτης στο Ασύρματο Τοπικό Δίκτυο, η χρήση του δικτύου δεδομένων από τη συγκεκριμένη κατηγορία υπόκειται σε περιορισμούς για λόγους ασφαλείας.

6.2 Υλοποίηση

Στην ενότητα αυτή, παρουσιάζονται οι λεπτομέρειες υλοποίησης του δικτύου TSN του Εργαστηρίου Δικτύων Υπολογιστών που σχεδιάσαμε.

6.2.1 Εικονικά Τοπικά Δίκτυα

Για την υποστήριξη διαφορετικών κατηγοριών χρηστών από το Ασύρματο Τοπικό Δίκτυο του Εργαστηρίου, εφαρμόστηκε η τεχνολογία των Εικονικών Τοπικών Δικτύων (Virtual Local Area Networks – VLAN) και πιο συγκεκριμένα η ενθυλάκωση των πλαισίων σύμφωνα με το πρότυπο IEEE 802.1Q [15]. Η τεχνολογία αυτή, επιτρέπει την υποδιαίρεση ενός φυσικού (υπο)δικτύου σε διαφορετικές περιοχές εκπομπής (broadcast domains), καθεμιά από τις οποίες, αποτελεί ένα VLAN. Σκοπός μας ήταν ο περιορισμός της δικτυακής κίνησης που παράγεται από κάθε κατηγορία χρηστών σε διαφορετικό VLAN, χρησιμοποιώντας όμως τον ίδιο φυσικό δίαυλο, τόσο όσον αφορά την ασύρματη επικοινωνία IEEE 802.11, όσο και την ενσύρματη. Στην πρώτη περίπτωση, ο διαχωρισμός αυτός επιτυγχάνεται με χρήση διαφορετικού ESSID, αλλά του ίδιου καναλιού της συχνότητας των 2,4 GHz (πρόκειται για δίκτυο IEEE 802.11b/g). Όσον αφορά το ενσύρματο δίκτυο, η ταμπέλα (tag) του πλαισίου ενθυλάκωσης IEEE 802.1Q καθορίζει το VLAN που αυτό ανήκει, καθιστώντας δυνατή την ύπαρξη πολλαπλών περιοχών εκπομπής, και άρα (υπο)δικτύων, στον ίδιο μεταγωγέα (switch) Ethernet. Οι τεχνικές λεπτομέρειες των VLAN είναι εκτός του αντικειμένου της παρούσας διπλωματικής εργασίας και δεν αναλύονται περαιτέρω.

Στα πλαίσια του δικτύου TSN του Εργαστηρίου, υλοποιήθηκαν τα εξής VLAN:

- `VLAN 40` (`cn`): Περιλαμβάνει τη δικτυακή κίνηση που παράγεται από τα εξουσιοδοτημένα μέλη. Το ESSID που έχει αποδοθεί στο VLAN αυτό, είναι `cn` και αποτελεί ασύρματη επέκταση του δημόσιου υποδικτύου IP του Εργαστηρίου. Η εκπομπή του ESSID είναι απενεργοποιημένη για λόγους ασφαλείας.

- `VLAN 101` (`cn-guest`): Περιλαμβάνει τη δικτυακή κίνηση που παράγεται από τους επισκέπτες. Το ESSID που έχει αποδοθεί στο VLAN αυτό, είναι `cn-guest` και χρησιμοποιεί ιδιωτικό (private) χώρο διευθύνσεων IP. Με τον τρόπο αυτό, οι επισκέπτες δεν είναι σε θέση να εγκαταστήσουν οποιοδήποτε λογισμικό εξυπηρετητή, καθώς δεν είναι προσπελάσιμοι από το Διαδίκτυο. Άλλος περιορισμός που ισχύει, επίσης, για λόγους ασφαλείας στο δίκτυο αυτό, είναι ότι απορρίπτεται η κίνηση με προορισμό τη θύρα 25, που αντιστοιχεί στο πρωτόκολλο αποστολής ηλεκτρονικού ταχυδρομείου SMTP, προκειμένου να περιορίζονται φαινόμενα αλληλογραφίας spam. Η εκπομπή του ESSID είναι απενεργοποιημένη.

- `VLAN 100` (`cn-management`): Περιλαμβάνει τη δικτυακή κίνηση των *διαχειριστών*. Πρόκειται για ένα ιδιωτικό υποδίκτυο IP, στο οποίο ανήκουν οι διαχειριστικές διεπαφές των διαφόρων εξυπηρετητών, των ευφυών μεταγωγέων/πλημνών (hubs) Ethernet, καθώς επίσης, και των σημείων πρόσβασης. Είναι προφανές, ότι, για λόγους ασφαλείας, αυτό το VLAN δεν έχει αντίστοιχο ασύρματο δίκτυο.

6.2.2 Λογισμικό

Ένας σημαντικός στόχος της υλοποίησής μας, ήταν η χρήση Ελεύθερου Λογισμικού / Λογισμικού Ανοιχτού Κώδικα – ΕΛ/ΛΑΚ (Open Source Software – OSS). Στο πλαίσιο αυτό, επιλέχθηκαν ελεύθερες υλοποιήσεις, τόσο του εξυπηρετητή RADIUS, όσο και του LDAP. Πιο συγκεκριμένα, εγκαταστήσαμε το FreeRADIUS [16] και το OpenLDAP [17], που αποτελούν ιδιαίτερα διαδεδομένες και δοκιμασμένες λύσεις. Προκειμένου να υπάρχει μια κοινή βάση δεδομένων των χρηστών για όλες τις υπηρεσίες του Εργαστηρίου χρησιμοποιήθηκε εξυπηρετητής Samba [18] σε συνδυασμό με το OpenLDAP. Τα αντίστοιχα αρχεία ρυθμίσεων παρατίθενται στο Παράρτημα Β΄.

Όπως αναφέρθηκε, στα πλαίσια του PEAP, απαιτείται η πιστοποίηση της αυθεντικότητας του εξυπηρετητή επαλήθευσης ταυτότητας RADIUS. Για το σκοπό αυτό, χρειάζεται κατάλληλο πιστοποιητικό (certificate), το οποίο πρέπει να έχει υπογραφεί από κάποια αρχή (Signing Authority – SA) που εμπιστεύεται ο πελάτης. Στην υλοποίησή μας, δε χρησιμοποιήθηκε κάποια επίσημη αρχή πιστοποίησης και δημιουργήσαμε μόνοι μας το πιστοποιητικό (self-signed certificate) [19]. Η διαδικασία εγκατάστασης του πιστοποιητικού από τον πελάτη περιγράφεται στο Παράρτημα Γ΄.

6.2.3 Δικτυακή τοπολογία

Στο Σχ. 6.3, φαίνεται η τοπολογία του Ασύρματου Τοπικού Δικτύου του Εργαστηρίου Δικτύων Υπολογιστών που υλοποιήσαμε, με βάση τις αρχές του Δικτύου Μεταβατικής Ασφάλειας.

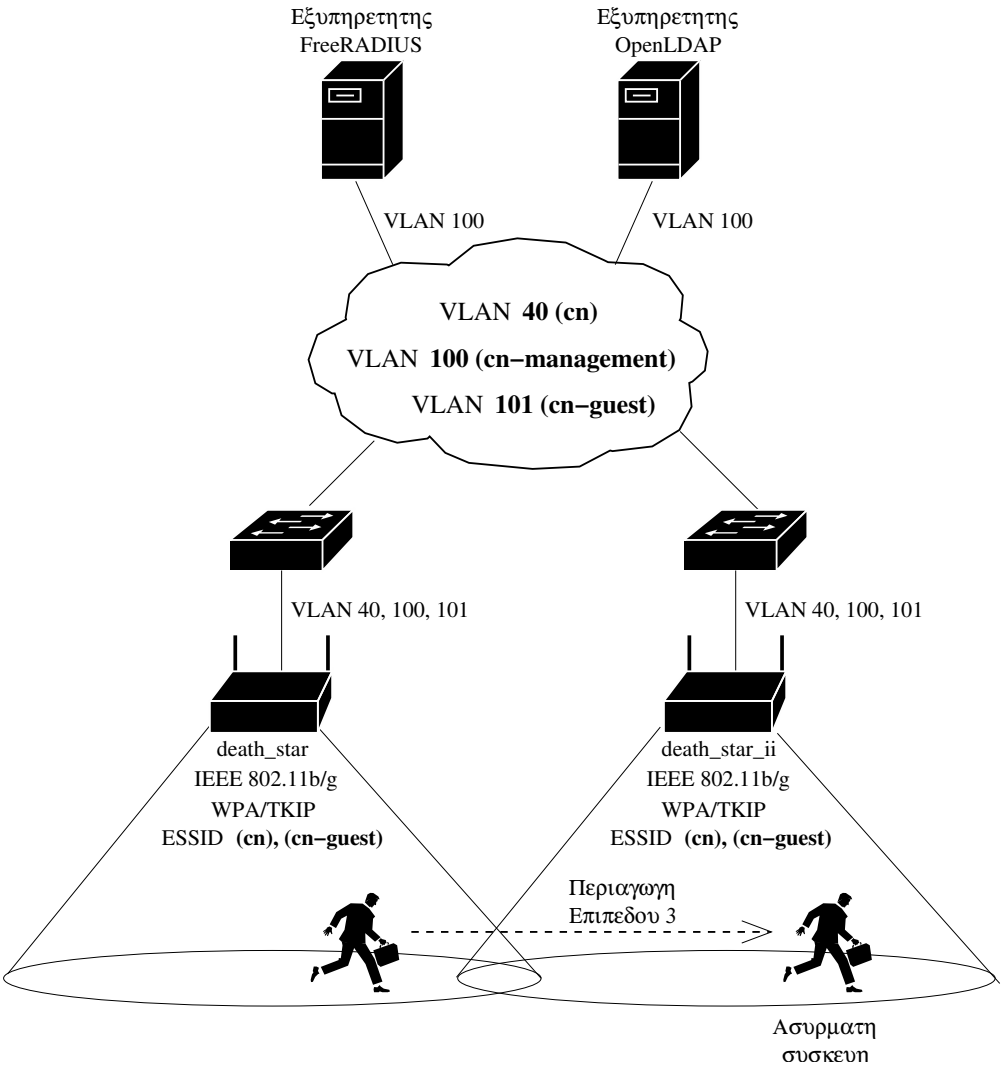
Περιλαμβάνει δύο σημεία πρόσβασης, `death_star` και `death_star_ii`, που υποστηρίζουν τα πρωτόκολλα IEEE 802.11b/g και ανήκουν στη σειρά 1200 της Cisco. Στην τοπολογία, φαίνονται ακόμη δύο μεταγωγείς Ethernet της σειράς 2900 της Cisco, που υποστηρίζουν την ενθυλάκωση πλαισίων IEEE 802.1Q για τη λειτουργία των VLAN. Όλες οι απαραίτητες ρυθμίσεις των δικτυακών συσκευών παρατίθενται στο Παράρτημα Α΄.

Οι διαχειριστικές δικτυακές διεπαφές των παραπάνω συσκευών ανήκουν, όπως αναφέρθηκε, στο `VLAN 100` του ενσύρματου δικτύου, όπου είναι, επίσης, συνδεδεμένος ο εξυπηρετητής RADIUS που επικοινωνεί με τη βάση δεδομένων LDAP των χρηστών για την επαλήθευση της ταυτότητάς τους.

Όταν ένας ασύρματος χρήστης μετακινείται από τη ζώνη κάλυψης του ενός σημείου πρόσβασης στην άλλη, τότε λαμβάνει χώρα περιαγωγή (roaming) επιπέδου 3 (IP) της συσκευής του με τρόπο διαφανή που ο ίδιος δεν αντιλαμβάνεται.

6.3 Έλεγχος

Στη συνέχεια θα επαληθευτεί η ασφαλής λειτουργία του TSN δικτύου που υλοποιήθηκε και θα γίνουν μετρήσεις για την εύρεση του επιπέδου ασφαλείας που αυτό επιτυγχάνει, ώστε να ελέγξουμε την ορθότητα της επιλογής μας και να αποφανθούμε αν ανταποκρίνεται στις προσδοκίες μας. Στα πλαίσια αυτά, θα συγκρίνουμε τα αποτελέσματα από την επίθεση κλειδιού [4], αρχικά, σε ένα Ασύρματο Τοπικό Δίκτυο συμβατικής ασφάλειας WEP και στη συνέχεια στο TSN του Εργαστηρίου Υπολογιστών.



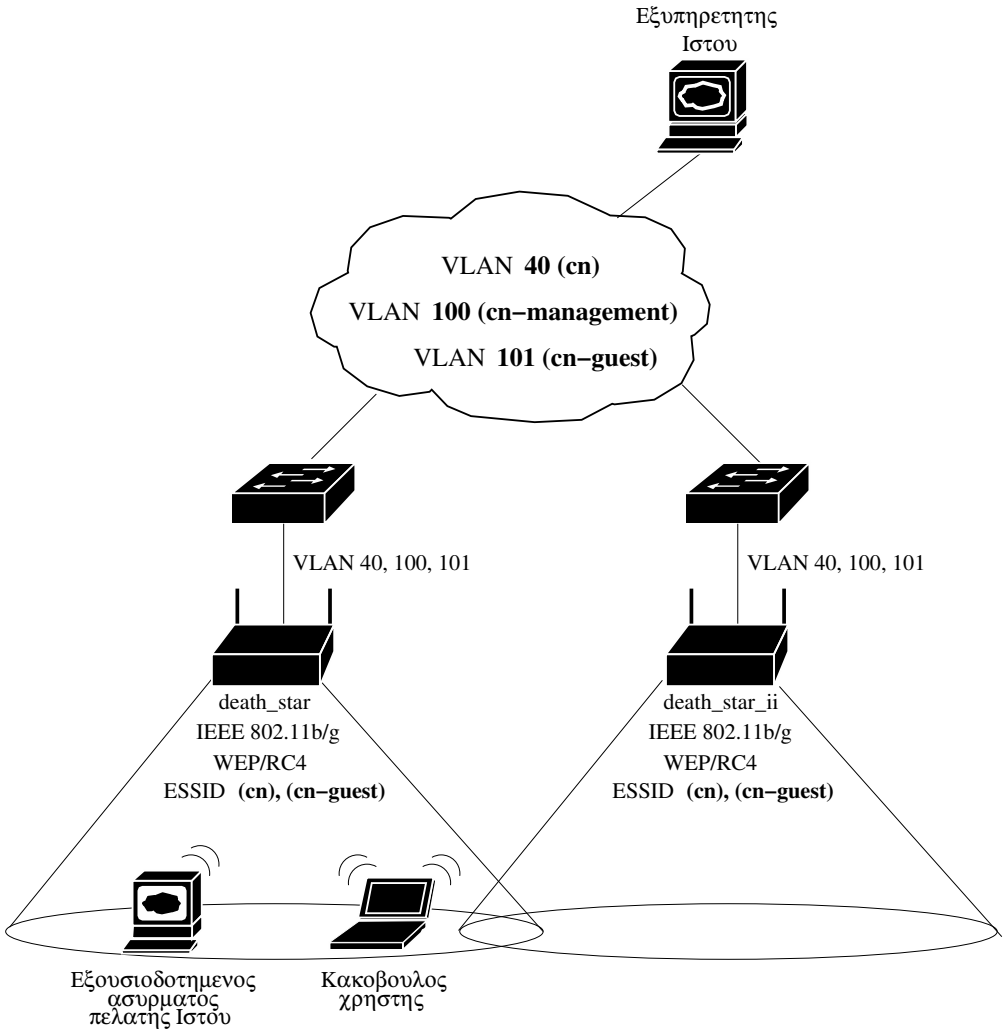
Σχήμα 6.3: Τοπολογία TSN Εργαστηρίου Δικτύων Υπολογιστών

6.3.1 Πειραματική πλατφόρμα

Στο Σχ. 6.4, φαίνεται το Ασύρματο Τοπικό Δίκτυο που υλοποιήθηκε προκειμένου να ληφθούν μετρήσεις στην περίπτωση του WEP. Στο σενάριο αυτό, δεν υπάρχει ο εξυπηρετητής επαλήθευσης ταυτότητας RADIUS, καθώς, όπως αναφέρθηκε, το WEP δεν υποστηρίζει το συγκεκριμένο μηχανισμό επαλήθευσης ταυτότητας. Ας σημειωθεί ότι στα πειράματα που πραγματοποιήσαμε, χρησιμοποιήθηκε ο ανοιχτός τρόπος επαλήθευσης ταυτότητας του WEP.

Υπάρχει πληθώρα εργαλείων που πραγματοποιούν επιθέσεις και «σπάνε» (crack) το κλειδί κρυπτογράφησης του WEP. Κάποια από αυτά αποτελούν εμπορικά προϊόντα, ωστόσο, υπάρχουν πολλά τα οποία είναι ανοιχτού κώδικα και διατίθενται ελεύθερα στο διαδίκτυο. Το πρώτο εργαλείο που δημοσιεύθηκε και εκμεταλλεύεται με επιτυχία τις αδυναμίες του WEP, είναι το WEPcrack. Ωστόσο, στα πειράματά μας, χρησιμοποιήσαμε το **Airsnort** [20], που θεωρείται από τα πλέον γρήγορα και του οποίου η χρήση είναι ιδιαίτερα διαδεδομένη μεταξύ των διαχειριστών δικτύων και των επίδοξων εισβολέων.

Το Airsnort θέτει την κάρτα ασύρματης δικτύωσης σε τρόπο λειτουργίας παρακολούθησης (monitor mode), ώστε να καταγράφει όλη την ασύρματη κίνηση που βρίσκεται στην

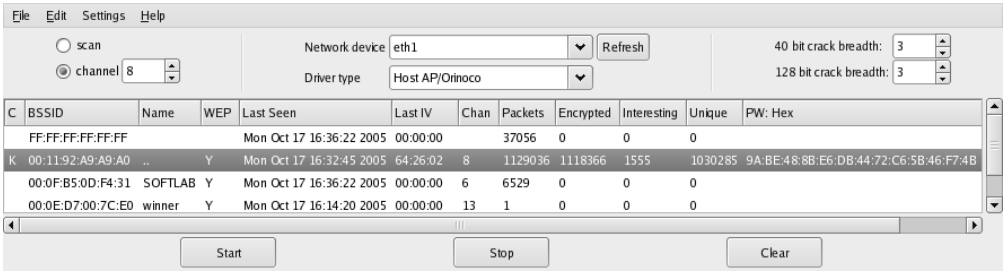


Σχήμα 6.4: « Σπάζοντας » το WEP

εμβέλεια της. Από το σύνολο των κρυπτογραφημένων πακέτων (**encrypted packets**) ενός Ασύρματου Τοπικού Δικτύου WEP, επιλέγει εκείνα για τα οποία χρησιμοποιήθηκαν *αδύναμα κλειδιά* (**interesting packets**). Συνδυάζοντας τα πακέτα αυτά και εκμεταλλευόμενο τη συγκεκριμένη αδυναμία του WEP, το Airsnort αποκαλύπτει στο χρήστη του το μυστικό κλειδί κρυπτογράφησης.

Στο Σχ. 6.5, φαίνεται ένα παράδειγμα εκτέλεσης του εργαλείου αυτού, για ένα κλειδί μήκους 104 bit. Μπορούμε να παρατηρήσουμε ότι το μυστικό κλειδί αποκαλύφθηκε μετά τη σύλληψη 1.129.036 κρυπτογραφημένων πακέτων, εκ των οποίων τα 1.555 είχαν κρυπτογραφηθεί με αδύναμο IV. Το Airsnort ολοκλήρωσε τη λειτουργία σε 30 λεπτά, κατά τη διάρκεια των οποίων διακινήθηκαν περίπου 3 GB δεδομένων.

Στα πειράματα που πραγματοποιήσαμε, υπήρχαν δύο οντότητες: από τη μια πλευρά ο *εξουσιοδοτημένος* χρήστης ενός επιτραπέζιου σταθμού εργασίας συνδεδεμένου με το Ασύρματο Τοπικό Δίκτυο σε IEEE 802.11g mode, ο οποίος μετέφερε αρχεία πάνω από το πρωτόκολλο HTTP και από την άλλη, ο *κακόβουλος* χρήστης ενός φορητού υπολογιστή εξοπλισμένου με μια ασύρματη κάρτας δικτύου, που με τη βοήθεια του Airsnort, επιχειρούσε επιθέσεις κλειδιού. Για την περίπτωση του WEP χρησιμοποιήθηκε η τοπολογία του Σχ. 6.4 με κλειδιά

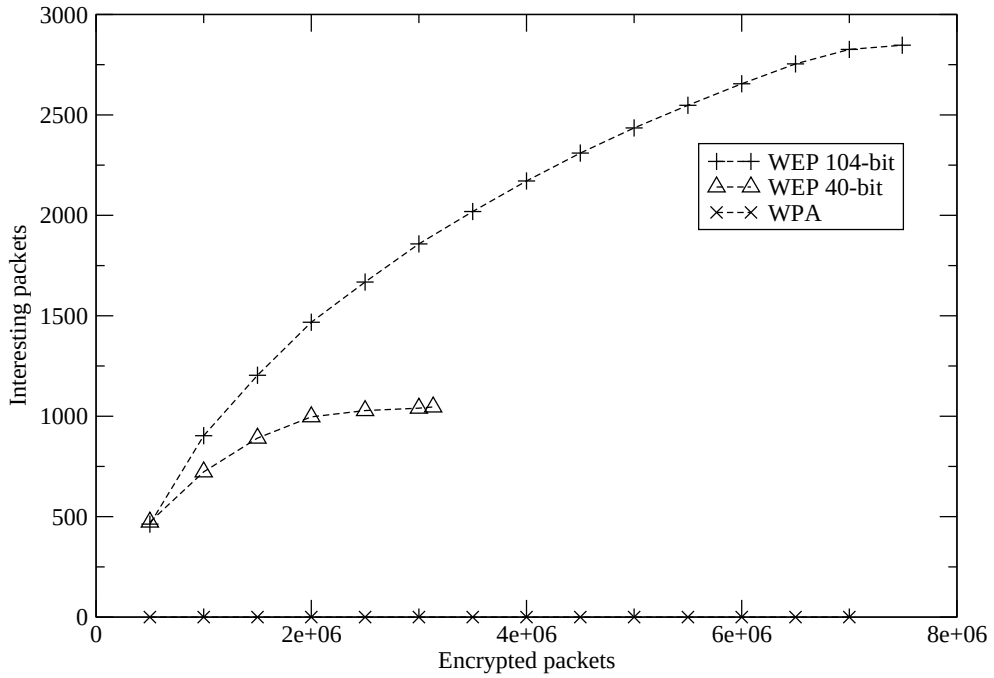


Σχήμα 6.5: Παράδειγμα επίθεσης στο WEP

κρυπτογράφησης μήκους 40 και 104 bit και 10 διαφορετικές τιμές τους, ανά περίπτωση. Αντίστοιχα, για τον έλεγχο της ασφάλειας του TSN, εκτελέστηκε το Airsnort στο δίκτυο που υλοποιήσαμε (Σχ. 6.3).

6.3.2 Αποτελέσματα

Στο Σχ. 6.6, φαίνονται τα αποτελέσματα των μετρήσεων που λάβαμε με τη βοήθεια του Airsnort. Στο γράφημα αυτό, έχουμε σημειώσει τον αριθμό των πακέτων με αδύναμο κλειδί, ως συνάρτηση του συνολικού αριθμού κρυπτογραφημένων πακέτων.



Σχήμα 6.6: Αποτελέσματα επιθέσεων στο WEP

Από το γράφημα μπορούμε να παρατηρήσουμε ότι:

1. Στην περίπτωση του κλειδιού WEP μήκους 40 bit, το μυστικό κλειδί αποκαλύπτεται κατά μέσο όρο μετά τη σύλληψη $3,5$ εκατομμυρίων κρυπτογραφημένων πακέτων, όταν δηλαδή έχουν καταγραφεί περίπου 1000 αδύναμα κλειδιά.

2. Στην περίπτωση του κλειδιού WEP μήκους 104 bit, το μυστικό κλειδί αποκαλύπτεται κατά μέσο όρο μετά τη σύλληψη 8 εκατομμυρίων κρυπτογραφημένων πακέτων, όταν δηλαδή έχουν καταγραφεί περίπου 3000 αδύναμα κλειδιά.

3. Στην περίπτωση του WPA, λόγω της χρησιμοποίησης του πρωτοκόλλου TKIP βάσει του οποίου το κλειδί κρυπτογράφησης αλλάζει με μεγάλη συχνότητα, το Airsnort δεν ήταν σε θέση να καταγράψει *κανένα* πακέτο κρυπτογραφημένο με αδύναμο κλειδί.

Με βάση τα παραπάνω αποτελέσματα, συμπεραίνουμε ότι το WEP είναι ιδιαίτερα ευάλωτο σε επιθέσεις κλειδιού, καθώς σε κάθε περίπτωση, το μυστικό κλειδί κρυπτογράφησης αποκαλύπτεται στον επίδοξο εισβολέα. Επιπλέον, επιβεβαιώνεται ότι η αύξηση του μήκους κλειδιού προκαλεί απλά *γραμμική* αύξηση του χρόνου εύρεσης του και όχι εκθετική. Τέλος, αποδεικνύεται ότι το WPA, παρόλο που δε θεωρείται τόσο ισχυρό όσο το RSN, εν τούτοις, επιτυγχάνει ένα επαρκές επίπεδο ασφαλείας για τα σημερινά δεδομένα.

Κεφάλαιο 7

Συμπεράσματα

Στο κεφάλαιο αυτό, περιγράφεται εν συντομία η πορεία της παρούσας διπλωματικής εργασίας, συνοψίζονται τα αποτελέσματα της και αναφέρονται τα συμπεράσματα που εξήχθησαν. Τέλος, γίνεται αναφορά σε επεκτάσεις που θα είχαν ερευνητικό και πρακτικό ενδιαφέρον.

7.1 Σύνοψη και συμπεράσματα

Στη διπλωματική αυτή εργασία, υλοποιήθηκε ένα Ασύρματο Τοπικό Δίκτυο στηριζόμενο στις αρχές του προτύπου ασφαλείας IEEE 802.11i και συγκεκριμένα στις προδιαγραφές του Δικτύου Μεταβατικής Ασφάλειας. Επίσης, δοκιμάστηκε το επίπεδο ασφαλείας που αυτό επιτυγχάνει, με σκοπό να αποδειχθεί η υπεροχή του έναντι του υπάρχοντος σχήματος ασφαλείας WEP.

Πιο συγκεκριμένα, αρχικά, αναφέρθηκαν οι αρχές λειτουργίας του πρωτοκόλλου IEEE 802.11, που προδιαγράφει την τεχνολογία των Ασυρμάτων Τοπικών Δικτύων, και παρουσιάστηκαν οι επιθέσεις στις οποίες αυτό είναι ευάλωτο. Επιπρόσθετα, αναλύθηκαν οι μηχανισμοί του αρχικού προτύπου ασφαλείας WEP και δόθηκε ιδιαίτερη έμφαση στις επιδόσεις του όσον αφορά την ασφάλεια. Στο πλαίσιο αυτό, αναφέρθηκαν λεπτομερώς οι αδυναμίες του WEP και γνωστές επιθέσεις που μπορούν να εφαρμοστούν για την παραβίασή του. Στη συνέχεια, παρουσιάστηκαν οι δύο προσεγγίσεις που προτείνει το νέο πρότυπο ασφαλείας για τα Ασύρματα Τοπικά Δίκτυα, IEEE 802.11i, δηλαδή, το Δίκτυο Μεταβατικής Ασφαλείας και το Δίκτυο Εύρωστης Ασφαλείας, και υποστηρίχθηκε το πρώτο λόγω του επαρκούς επιπέδου ασφαλείας που επιτυγχάνει σε συνδυασμό με τη διαλειτουργικότητα που εξασφαλίζει μεταξύ του συμβατικού και του νεότερου εξοπλισμού ασύρματης δικτύωσης. Επιπλέον, παρουσιάστηκαν οι διάφοροι μηχανισμοί ελέγχου πρόσβασης που προτείνονται στα πλαίσια του προτύπου ασφαλείας IEEE 802.11i. Πιο συγκεκριμένα, αναλύθηκαν τα πρωτόκολλα IEEE 802.1X, Πρωτόκολλο Επεκτάσιμης Επαλήθευσης Ταυτότητας και Remote Access Dial-In User Service, ενώ περιγράφηκε η αλληλεπίδρασή τους για την παροχή ελέγχου πρόσβασης στα Ασύρματα Τοπικά Δίκτυα.

Ακολούθησε η σχεδίαση και η υλοποίηση του Ασύρματου Τοπικού Δικτύου του Εργαστηρίου Δικτύων Υπολογιστών του Ε.Μ.Π. με εφαρμογή των αρχών του Δικτύου Μεταβατικής Ασφαλείας και εξηγήθηκαν οι μηχανισμοί πρόσβασης και διαχείρισής του. Επιπρόσθετα, πραγματοποιήθηκαν γνωστές επιθέσεις, τόσο σε Ασύρματα Τοπικά Δίκτυα συμβατικής ασφάλειας WEP, όσο και στην υλοποίησή μας. Τα αποτελέσματα της σύγκρισης και τα συμπεράσματα για την απόδοση, όσον αφορά την ασφάλεια, συνοψίζονται στα παρακάτω:

- Το WEP δεν παρέχει κανένα μηχανισμό επαλήθευσης ταυτότητας των χρηστών, προκειμένου να συνδεθούν στο Ασύρματο Τοπικό Δίκτυο, με εξαίρεση το ίδιο το κλειδί

κρυπτογράφησής του. Αντίθετα, το σχήμα IEEE 802.1X/PEAP, σε συνδυασμό με το RADIUS, αποτελεί ένα ολοκληρωμένο σύστημα κεντρικής πιστοποίησης της αυθεντικότητας των πελατών και διαχείρισης κλειδιών.

- Το WEP δεν είναι σε θέση να αντιμετωπίσει τις επιθέσεις που, νομοτελειακά, θα αποκαλύψουν το κλειδί κρυπτογράφησης σε ένα επίδοξο εισβολέα. Εξάλλου, η αύξηση του μήκους κλειδιού κρυπτογράφησης από τα 40 στα 104 bit αυξάνει *γραμμικά*, και όχι εκθετικά, το χρόνο εύρεσης του.

- Αποδεικνύεται και στην πράξη, ότι το TSN καλύπτει τις αδυναμίες του WEP, αντιμετωπίζοντας με επιτυχία τις επιθέσεις κλειδιού εναντίον του.

- Το RSN χρησιμοποιεί το πρωτόκολλο CCMP και τον αλγόριθμο κρυπτογράφησης AES που είναι ιδιαίτερα ισχυρός. Ως εκ τούτου, είναι το πιο ασφαλές σχήμα ασφαλείας για τα Ασύρματα Τοπικά Δίκτυα, εν τούτοις, *δεν* είναι συμβατό με τον υπάρχοντα εξοπλισμό ασύρματης δικτύωσης.

Από τα παραπάνω, γίνεται φανερή η ανάγκη για αντικατάσταση του σχήματος ασφαλείας WEP στα Ασύρματα Τοπικά Δίκτυα, ενώ συμπεραίνουμε ότι για το σκοπό αυτό, η μόνη εφικτή και ασφαλής λύση σήμερα, πριν την πλήρη μετάβαση στο RSN, είναι το TSN.

7.2 Μελλοντικές επεκτάσεις

Στην παρούσα διπλωματική εργασία, έχουμε υλοποιήσει ένα Ασύρματο Τοπικό Δίκτυο στηριζόμενο στις αρχές του Δικτύου Μεταβατικής Ασφάλειας, εν τούτοις, το Δίκτυο Εύρωστης Ασφάλειας εξετάζεται μόνο θεωρητικά. Θα παρουσίαζε ιδιαίτερο ενδιαφέρον, η υλοποίηση ενός δικτύου RSN και η σύγκριση της επίδοσής του, όσον αφορά το επίπεδο της ασφάλειας, με το TSN. Προφανώς, προϋπόθεση για την υλοποίηση αυτή, θα αποτελούσε η ύπαρξη του κατάλληλου εξοπλισμού ασύρματης δικτύωσης, καθώς επίσης, και ειδικού λογισμικού που να υλοποιεί επιθέσεις κλειδιού κατάλληλες για το πρωτόκολλο κρυπτογράφησης CCMP/AES του RSN.

Παράρτημα Α΄

Α΄.1 Ρυθμίσεις IOS σημείου πρόσβασης

```
version 12.2
no service pad
service timestamps debug datetime localtime show-timezone
service timestamps log datetime localtime show-timezone
service password-encryption
!
hostname death-star
!
logging queue-limit 100
no logging console
enable secret 5 $1$ulN5$eVvWcnYWexXE1MQx2s5iu/
!
clock timezone EEET 2
clock summer-time EEST recurring
ip subnet-zero
!
aaa new-model
!
!
aaa group server radius freeradius_peap
 server 172.24.100.9 auth-port 1812 acct-port 1813
!
aaa authentication login peap group freeradius_peap
aaa authentication login no_radius line
aaa session-id common
!
bridge irb
!
interface Dot11Radio0
 no ip address
 no ip route-cache
 !
 encryption mode ciphers tkip
 !
 encryption vlan 100 mode ciphers tkip
 !
```

```
 encryption vlan 101 mode ciphers tkip
 !
 encryption vlan 40 mode ciphers tkip
 !
 ssid cn-management
    vlan 100
    authentication open eap peap
    authentication key-management wpa
 !
 ssid cn-guest
    vlan 101
    authentication open eap peap
    authentication key-management wpa
 !
 ssid cn
    vlan 40
    authentication open eap peap
    authentication key-management wpa
 !
 speed basic-1.0 2.0 5.5 6.0 9.0 11.0 12.0 18.0 24.0 36.0 48.0 54.0
 rts threshold 2312
 station-role root
 no dot11 extension aironet
!
interface Dot11Radio0.101
 encapsulation dot1Q 101
 no ip route-cache
 bridge-group 101
 bridge-group 101 subscriber-loop-control
 bridge-group 101 block-unknown-source
 no bridge-group 101 source-learning
 no bridge-group 101 unicast-flooding
 bridge-group 101 spanning-disabled
!
interface Dot11Radio0.40
 encapsulation dot1Q 40
 no ip route-cache
 bridge-group 40
 bridge-group 40 subscriber-loop-control
 bridge-group 40 block-unknown-source
 no bridge-group 40 source-learning
 no bridge-group 40 unicast-flooding
 bridge-group 40 spanning-disabled
!
interface Dot11Radio0.100
 encapsulation dot1Q 100 native
 no ip route-cache
 bridge-group 1
```

```
 bridge-group 1 subscriber-loop-control
 bridge-group 1 block-unknown-source
 no bridge-group 1 source-learning
 no bridge-group 1 unicast-flooding
 bridge-group 1 spanning-disabled
!
interface FastEthernet0
 no ip address
 no ip route-cache
 duplex auto
 speed auto
!
interface FastEthernet0.101
 encapsulation dot1Q 101
 no ip route-cache
 bridge-group 101
 no bridge-group 101 source-learning
 bridge-group 101 spanning-disabled
!
interface FastEthernet0.40
 encapsulation dot1Q 40
 no ip route-cache
 bridge-group 40
 no bridge-group 40 source-learning
 bridge-group 40 spanning-disabled
!
interface FastEthernet0.100
 encapsulation dot1Q 100 native
 no ip route-cache
 bridge-group 1
 no bridge-group 1 source-learning
 bridge-group 1 spanning-disabled
!
interface BVI1
 ip address 172.24.100.241 255.255.255.0
 no ip route-cache
!
no ip http server
ip http help-path http://www.cisco.com/warp/public/779/smbiz/prodconfig/help/eag/122-
ip radius source-interface BVI1
radius-server attribute 32 include-in-access-req format %h
radius-server host 172.24.100.9 auth-port 1812 acct-port 1813 key 7 045F58551F1E5F1D0
radius-server authorization permit missing Service-Type
bridge 1 route ip
!
!
!
line con 0
```

```
line vty 0 4
 exec-timeout 180 0
 password 7 0226134F0D572F60
 login authentication no_radius
line vty 5 15
 exec-timeout 180 0
 password 7 0226134F0D572F60
 login authentication no_radius
!
end
```

Α΄.2 Ρυθμίσεις IOS μεταγωγέα Ethernet

```
version 12.0
service nagle
no service pad
service timestamps debug datetime localtime show-timezone
service timestamps log datetime localtime show-timezone
service password-encryption
!
hostname kamino
!
logging buffered 16384 debugging
no logging console
enable secret 5 $1$PcMY$fiwSeXlcCCQrbGEPGpegd.
!
clock timezone EET 2
clock summer-time EEST recurring
!
no spanning-tree vlan 1
ip subnet-zero
no ip finger
!
interface FastEthernet0/1
 switchport trunk encapsulation dot1q
 switchport trunk native vlan 100
 switchport trunk allowed vlan 1,40,100,101,1002-1005
 switchport mode trunk
 spanning-tree portfast
!
interface FastEthernet0/2
 switchport access vlan 40
 spanning-tree portfast
!
! The rest of the interfaces' configurations are omitted
!
interface FastEthernet0/8
```

```
 description Uplink to 15b-00-sc-b1 (NOC)
 switchport access vlan 40
!
interface VLAN1
 no ip directed-broadcast
 no ip route-cache
 shutdown
!
interface VLAN100
 ip address 172.24.100.223 255.255.255.0
 no ip directed-broadcast
 no ip route-cache
!
no ip http server

!
line con 0
 exec-timeout 0 0
 transport input none
 stopbits 1
line vty 0 4
 exec-timeout 30 0
 password 7 052B111B271D6E48
 login
line vty 5 15
 exec-timeout 30 0
 password 7 052B111B271D6E48
 login
!
end
```

Παράρτημα Β΄

Β΄.1 Ρυθμίσεις εξυπηρετητή FreeRADIUS

Β΄.1.1 Αρχείο radiusd.conf

```
##
## radiusd.conf -- FreeRADIUS server configuration file.
##
## http://www.freeradius.org/
## $Id: radiusd.conf.in,v 1.188.2.3 2005/02/07 19:52:05 aland Exp $
##

#   The location of other config files and
#   logfiles are declared in this file
#
#   Also general configuration for modules can be done
#   in this file, it is exported through the API to
#   modules that ask for it.
#
# The configuration variables defined here are of the form ${foo}
# They are local to this file, and do not change from request to
# request.
#
# The per-request variables are of the form %{Attribute-Name}, and
# are taken from the values of the attribute in the incoming
# request.  See 'doc/variables.txt' for more information.

prefix = /usr/local
exec_prefix = ${prefix}
sysconfdir = ${prefix}/etc
localstatedir = /var
sbindir = ${exec_prefix}/sbin
logdir = /var/log
raddbdir = ${sysconfdir}/raddb
radacctdir = ${logdir}/radacct

#  Location of config and logfiles.
confdir = ${raddbdir}
run_dir = ${localstatedir}/run/radiusd
```

```
#
#  The logging messages for the server are appended to the
#  tail of this file.
#
log_file = ${logdir}/radius.log

#
# libdir: Where to find the rlm_* modules.
#
#   This should be automatically set at configuration time.
#
#   If the server builds and installs, but fails at execution time
#   with an 'undefined symbol' error, then you can use the libdir
#   directive to work around the problem.
#
#   The cause is usually that a library has been installed on your
#   system in a place where the dynamic linker CANNOT find it.  When
#   executing as root (or another user), your personal environment MAY
#   be set up to allow the dynamic linker to find the library.  When
#   executing as a daemon, FreeRADIUS MAY NOT have the same
#   personalized configuration.
#
#   To work around the problem, find out which library contains that symbol,
#   and add the directory containing that library to the end of 'libdir',
#   with a colon separating the directory names.  NO spaces are allowed.
#
#   e.g. libdir = /usr/local/lib:/opt/package/lib
#
#   You can also try setting the LD_LIBRARY_PATH environment variable
#   in a script which starts the server.
#
#   If that does not work, then you can re-configure and re-build the
#   server to NOT use shared libraries, via:
#
# ./configure --disable-shared
# make
# make install
#
libdir = ${exec_prefix}/lib

#  pidfile: Where to place the PID of the RADIUS server.
#
#  The server may be signalled while it's running by using this
#  file.
#
#  This file is written when ONLY running in daemon mode.
#
```

```
#  e.g.:  kill -HUP `cat /var/run/radiusd/radiusd.pid`
#
pidfile = ${run_dir}/radiusd.pid


# user/group: The name (or #number) of the user/group to run radiusd as.
#
#   If these are commented out, the server will run as the user/group
#   that started it.  In order to change to a different user/group, you
#   MUST be root ( or have root privleges ) to start the server.
#
#   We STRONGLY recommend that you run the server with as few permissions
#   as possible.  That is, if you're not using shadow passwords, the
#   user and group items below should be set to 'nobody'.
#
#    On SCO (ODT 3) use "user = nouser" and "group = nogroup".
#
#  NOTE that some kernels refuse to setgid(group) when the value of
#  (unsigned)group is above 60000; don't use group nobody on these systems!
#
#  On systems with shadow passwords, you might have to set 'group = shadow'
#  for the server to be able to read the shadow password file.  If you can
#  authenticate users while in debug mode, but not in daemon mode, it may be
#  that the debugging mode server is running as a user that can read the
#  shadow info, and the user listed below can not.
#
#user = nobody
#group = nobody

#  max_request_time: The maximum time (in seconds) to handle a request.
#
#  Requests which take more time than this to process may be killed, and
#  a REJECT message is returned.
#
#  WARNING: If you notice that requests take a long time to be handled,
#  then this MAY INDICATE a bug in the server, in one of the modules
#  used to handle a request, OR in your local configuration.
#
#  This problem is most often seen when using an SQL database.  If it takes
#  more than a second or two to receive an answer from the SQL database,
#  then it probably means that you haven't indexed the database.  See your
#  SQL server documentation for more information.
#
#  Useful range of values: 5 to 120
#
max_request_time = 30

#  delete_blocked_requests: If the request takes MORE THAN 'max_request_time'
```

```
#  to be handled, then maybe the server should delete it.
#
#  If you're running in threaded, or thread pool mode, this setting
#  should probably be 'no'.  Setting it to 'yes' when using a threaded
#  server MAY cause the server to crash!
#
delete_blocked_requests = no

#  cleanup_delay: The time to wait (in seconds) before cleaning up
#  a reply which was sent to the NAS.
#
#  The RADIUS request is normally cached internally for a short period
#  of time, after the reply is sent to the NAS.  The reply packet may be
#  lost in the network, and the NAS will not see it.  The NAS will then
#  re-send the request, and the server will respond quickly with the
#  cached reply.
#
#  If this value is set too low, then duplicate requests from the NAS
#  MAY NOT be detected, and will instead be handled as seperate requests.
#
#  If this value is set too high, then the server will cache too many
#  requests, and some new requests may get blocked.  (See 'max_requests'.)
#
#  Useful range of values: 2 to 10
#
cleanup_delay = 5

#  max_requests: The maximum number of requests which the server keeps
#  track of.  This should be 256 multiplied by the number of clients.
#  e.g. With 4 clients, this number should be 1024.
#
#  If this number is too low, then when the server becomes busy,
#  it will not respond to any new requests, until the 'cleanup_delay'
#  time has passed, and it has removed the old requests.
#
#  If this number is set too high, then the server will use a bit more
#  memory for no real benefit.
#
#  If you aren't sure what it should be set to, it's better to set it
#  too high than too low.  Setting it to 1000 per client is probably
#  the highest it should be.
#
#  Useful range of values: 256 to infinity
#
max_requests = 1024

#  bind_address:  Make the server listen on a particular IP address, and
#  send replies out from that address.  This directive is most useful
```

```
#  for machines with multiple IP addresses on one interface.
#
#  It can either contain "*", or an IP address, or a fully qualified
#  Internet domain name.  The default is "*"
#
#  As of 1.0, you can also use the "listen" directive.  See below for
#  more information.
#
bind_address = *

#  port: Allows you to bind FreeRADIUS to a specific port.
#
#  The default port that most NAS boxes use is 1645, which is historical.
#  RFC 2138 defines 1812 to be the new port.  Many new servers and
#  NAS boxes use 1812, which can create interoperability problems.
#
#  The port is defined here to be 0 so that the server will pick up
#  the machine's local configuration for the radius port, as defined
#  in /etc/services.
#
#  If you want to use the default RADIUS port as defined on your server,
#  (usually through 'grep radius /etc/services') set this to 0 (zero).
#
#  A port given on the command-line via '-p' over-rides this one.
#
#  As of 1.0, you can also use the "listen" directive.  See below for
#  more information.
#
port = 0

#
#  By default, the server uses "bind_address" to listen to all IP's
#  on a machine, or just one IP.  The "port" configuration is used
#  to select the authentication port used when listening on those
#  addresses.
#
#  If you want the server to listen on additional addresses, you can
#  use the "listen" section.  A sample section (commented out) is included
#  below.  This "listen" section duplicates the functionality of the
#  "bind_address" and "port" configuration entries, but it only listens
#  for authentication packets.
#
#  If you comment out the "bind_address" and "port" configuration entries,
#  then it becomes possible to make the server accept only accounting,
#  or authentication packets.  Previously, it always listened for both
#  types of packets, and it was impossible to make it listen for only
#  one type of packet.
#
```

```
#listen {
#  IP address on which to listen.
#  Allowed values are:
# dotted quad (1.2.3.4)
#       hostname    (radius.example.com)
#       wildcard    (*)
# ipaddr = *

#  Port on which to listen.
#  Allowed values are:
# integer port number (1812)
# 0 means "use /etc/services for the proper port"
# port = 0

#  Type of packets to listen for.
#  Allowed values are:
# auth listen for authentication packets
# acct listen for accounting packets
#
# type = auth
#}


#  hostname_lookups: Log the names of clients or just their IP addresses
#  e.g., www.freeradius.org (on) or 206.47.27.232 (off).
#
#  The default is 'off' because it would be overall better for the net
#  if people had to knowingly turn this feature on, since enabling it
#  means that each client request will result in AT LEAST one lookup
#  request to the nameserver.   Enabling hostname_lookups will also
#  mean that your server may stop randomly for 30 seconds from time
#  to time, if the DNS requests take too long.
#
#  Turning hostname lookups off also means that the server won't block
#  for 30 seconds, if it sees an IP address which has no name associated
#  with it.
#
#  allowed values: {no, yes}
#
hostname_lookups = no

#  Core dumps are a bad thing.  This should only be set to 'yes'
#  if you're debugging a problem with the server.
#
#  allowed values: {no, yes}
#
allow_core_dumps = no
```

```
#  Regular expressions
#
#  These items are set at configure time.  If they're set to "yes",
#  then setting them to "no" turns off regular expression support.
#
#  If they're set to "no" at configure time, then setting them to "yes"
#  WILL NOT WORK.  It will give you an error.
#
regular_expressions = yes
extended_expressions = yes

#  Log the full User-Name attribute, as it was found in the request.
#
# allowed values: {no, yes}
#
log_stripped_names = no

#  Log authentication requests to the log file.
#
#  allowed values: {no, yes}
#
log_auth = no

#  Log passwords with the authentication requests.
#  log_auth_badpass  - logs password if it's rejected
#  log_auth_goodpass - logs password if it's correct
#
#  allowed values: {no, yes}
#
log_auth_badpass = no
log_auth_goodpass = no

# usercollide:  Turn "username collision" code on and off.  See the
# "doc/duplicate-users" file
#
#  WARNING
#  !!!!!!!  Setting this to "yes" may result in the server behaving
#  !!!!!!!  strangely.  The "username collision" code will ONLY work
#  !!!!!!!  with clear-text passwords.  Even then, it may not do what
#  !!!!!!!  you want, or what you expect.
#  !!!!!!!
#  !!!!!!!  We STRONGLY RECOMMEND that you do not use this feature,
#  !!!!!!!  and that you find another way of acheiving the same goal.
#  !!!!!!!
#  !!!!!!!  e,g. module fail-over.  See 'doc/configurable_failover'
#  WARNING
#
usercollide = no
```

```
# lower_user / lower_pass:
# Lower case the username/password "before" or "after"
# attempting to authenticate.
#
#  If "before", the server will first modify the request and then try
#  to auth the user.  If "after", the server will first auth using the
#  values provided by the user.  If that fails it will reprocess the
#  request after modifying it as you specify below.
#
#  This is as close as we can get to case insensitivity.  It is the
#  admin's job to ensure that the username on the auth db side is
#  *also* lowercase to make this work
#
# Default is 'no' (don't lowercase values)
# Valid values = "before" / "after" / "no"
#
lower_user = no
lower_pass = no

# nospace_user / nospace_pass:
#
#  Some users like to enter spaces in their username or password
#  incorrectly.  To save yourself the tech support call, you can
#  eliminate those spaces here:
#
# Default is 'no' (don't remove spaces)
# Valid values = "before" / "after" / "no" (explanation above)
#
nospace_user = no
nospace_pass = no

#  The program to execute to do concurrency checks.
checkrad = ${sbindir}/checkrad

# SECURITY CONFIGURATION
#
#  There may be multiple methods of attacking on the server.  This
#  section holds the configuration items which minimize the impact
#  of those attacks
#
security {
#
#  max_attributes: The maximum number of attributes
#  permitted in a RADIUS packet.  Packets which have MORE
#  than this number of attributes in them will be dropped.
#
#  If this number is set too low, then no RADIUS packets
```

```
#  will be accepted.
#
#  If this number is set too high, then an attacker may be
#  able to send a small number of packets which will cause
#  the server to use all available memory on the machine.
#
#  Setting this number to 0 means "allow any number of attributes"
max_attributes = 200

#
#  delayed_reject: When sending an Access-Reject, it can be
#  delayed for a few seconds.  This may help slow down a DoS
#  attack.  It also helps to slow down people trying to brute-force
#  crack a users password.
#
#  Setting this number to 0 means "send rejects immediately"
#
#  If this number is set higher than 'cleanup_delay', then the
#  rejects will be sent at 'cleanup_delay' time, when the request
#  is deleted from the internal cache of requests.
#
#  Useful ranges: 1 to 5
reject_delay = 1

#
#  status_server: Whether or not the server will respond
#  to Status-Server requests.
#
#  Normally this should be set to "no", because they're useless.
#  See: http://www.freeradius.org/rfc/rfc2865.html#Keep-Alives
#
#  However, certain NAS boxes may require them.
#
#  When sent a Status-Server message, the server responds with
#  an Access-Accept packet, containing a Reply-Message attribute,
#  which is a string describing how long the server has been
#  running.
#
status_server = no
}

# PROXY CONFIGURATION
#
#  proxy_requests: Turns proxying of RADIUS requests on or off.
#
#  The server has proxying turned on by default.  If your system is NOT
#  set up to proxy requests to another server, then you can turn proxying
#  off here.  This will save a small amount of resources on the server.
```

```
#
#  If you have proxying turned off, and your configuration files say
#  to proxy a request, then an error message will be logged.
#
#  To disable proxying, change the "yes" to "no", and comment the
#  $INCLUDE line.
#
#  allowed values: {no, yes}
#
proxy_requests  = yes
$INCLUDE  ${confdir}/proxy.conf


# CLIENTS CONFIGURATION
#
#  Client configuration is defined in "clients.conf".
#

#  The 'clients.conf' file contains all of the information from the old
#  'clients' and 'naslist' configuration files.  We recommend that you
#  do NOT use 'client's or 'naslist', although they are still
#  supported.
#
#  Anything listed in 'clients.conf' will take precedence over the
#  information from the old-style configuration files.
#
$INCLUDE  ${confdir}/clients.conf


# SNMP CONFIGURATION
#
#  Snmp configuration is only valid if SNMP support was enabled
#  at compile time.
#
#  To enable SNMP querying of the server, set the value of the
#  'snmp' attribute to 'yes'
#
snmp = no
$INCLUDE  ${confdir}/snmp.conf


# THREAD POOL CONFIGURATION
#
#  The thread pool is a long-lived group of threads which
#  take turns (round-robin) handling any incoming requests.
#
#  You probably want to have a few spare threads around,
#  so that high-load situations can be handled immediately.  If you
```

```
#  don't have any spare threads, then the request handling will
#  be delayed while a new thread is created, and added to the pool.
#
#  You probably don't want too many spare threads around,
#  otherwise they'll be sitting there taking up resources, and
#  not doing anything productive.
#
#  The numbers given below should be adequate for most situations.
#
thread pool {
#  Number of servers to start initially --- should be a reasonable
#  ballpark figure.
start_servers = 5

#  Limit on the total number of servers running.
#
#  If this limit is ever reached, clients will be LOCKED OUT, so it
#  should NOT BE SET TOO LOW.  It is intended mainly as a brake to
#  keep a runaway server from taking the system with it as it spirals
#  down...
#
#  You may find that the server is regularly reaching the
#  'max_servers' number of threads, and that increasing
#  'max_servers' doesn't seem to make much difference.
#
#  If this is the case, then the problem is MOST LIKELY that
#  your back-end databases are taking too long to respond, and
#  are preventing the server from responding in a timely manner.
#
#  The solution is NOT do keep increasing the 'max_servers'
#  value, but instead to fix the underlying cause of the
#  problem: slow database, or 'hostname_lookups=yes'.
#
#  For more information, see 'max_request_time', above.
#
max_servers = 32

#  Server-pool size regulation.  Rather than making you guess
#  how many servers you need, FreeRADIUS dynamically adapts to
#  the load it sees, that is, it tries to maintain enough
#  servers to handle the current load, plus a few spare
#  servers to handle transient load spikes.
#
#  It does this by periodically checking how many servers are
#  waiting for a request.  If there are fewer than
#  min_spare_servers, it creates a new spare.  If there are
#  more than max_spare_servers, some of the spares die off.
#  The default values are probably OK for most sites.
```

```
#
min_spare_servers = 3
max_spare_servers = 10

#  There may be memory leaks or resource allocation problems with
#  the server.  If so, set this value to 300 or so, so that the
#  resources will be cleaned up periodically.
#
#  This should only be necessary if there are serious bugs in the
#  server which have not yet been fixed.
#
#  '0' is a special value meaning 'infinity', or 'the servers never
#  exit'
max_requests_per_server = 0
}

# MODULE CONFIGURATION
#
#  The names and configuration of each module is located in this section.
#
#  After the modules are defined here, they may be referred to by name,
#  in other sections of this configuration file.
#
modules {
#
#  Each module has a configuration as follows:
#
# name [ instance ] {
# config_item = value
# ...
# }
#
#  The 'name' is used to load the 'rlm_name' library
#  which implements the functionality of the module.
#
#  The 'instance' is optional.  To have two different instances
#  of a module, it first must be referred to by 'name'.
#  The different copies of the module are then created by
#  inventing two 'instance' names, e.g. 'instance1' and 'instance2'
#
#  The instance names can then be used in later configuration
#  INSTEAD of the original 'name'.  See the 'radutmp' configuration
#  below for an example.
#

# PAP module to authenticate users based on their stored password
#
#  Supports multiple encryption schemes
```

```
#  clear: Clear text
#  crypt: Unix crypt
#    md5: MD5 ecnryption
#   sha1: SHA1 encryption.
#  DEFAULT: crypt
pap {
encryption_scheme = crypt
}

# CHAP module
#
#  To authenticate requests containing a CHAP-Password attribute.
#
chap {
authtype = CHAP
}

# Pluggable Authentication Modules
#
#  For Linux, see:
# http://www.kernel.org/pub/linux/libs/pam/index.html
#
#  WARNING: On many systems, the system PAM libraries have
#           memory leaks!  We STRONGLY SUGGEST that you do not
#     use PAM for authentication, due to those memory leaks.
#
pam {
#
#  The name to use for PAM authentication.
#  PAM looks in /etc/pam.d/${pam_auth_name}
#  for it's configuration.  See 'redhat/radiusd-pam'
#  for a sample PAM configuration file.
#
#  Note that any Pam-Auth attribute set in the 'authorize'
#  section will over-ride this one.
#
pam_auth = radiusd
}

# Unix /etc/passwd style authentication
#
unix {
#
#  Cache /etc/passwd, /etc/shadow, and /etc/group
#
#  The default is to NOT cache them.
#
#  For FreeBSD and NetBSD, you do NOT want to enable
```

```
#  the cache, as it's password lookups are done via a
#  database, so set this value to 'no'.
#
#  Some systems (e.g. RedHat Linux with pam_pwbd) can
#  take *seconds* to check a password, when th passwd
#  file containing 1000's of entries.  For those systems,
#  you should set the cache value to 'yes', and set
#  the locations of the 'passwd', 'shadow', and 'group'
#  files, below.
#
# allowed values: {no, yes}
cache = no

# Reload the cache every 600 seconds (10mins). 0 to disable.
cache_reload = 600

#
#  Define the locations of the normal passwd, shadow, and
#  group files.
#
#  'shadow' is commented out by default, because not all
#  systems have shadow passwords.
#
#  To force the module to use the system password functions,
#  instead of reading the files, leave the following entries
#  commented out.
#
#  This is required for some systems, like FreeBSD,
#  and Mac OSX.
#
# passwd = /etc/passwd
# shadow = /etc/shadow
# group = /etc/group

#
#  The location of the "wtmp" file.
#  This should be moved to it's own module soon.
#
#  The only use for 'radlast'.  If you don't use
#  'radlast', then you can comment out this item.
#
radwtmp = ${logdir}/radwtmp
}

#  Extensible Authentication Protocol
#
#  For all EAP related authentications.
#  Now in another file, because it is very large.
```

```
#
$INCLUDE ${confdir}/eap.conf

# Microsoft CHAP authentication
#
#  This module supports MS-CHAP and MS-CHAPv2 authentication.
#  It also enforces the SMB-Account-Ctrl attribute.
#
mschap {
#
#  As of 0.9, the mschap module does NOT support
#  reading from /etc/smbpasswd.
#
#  If you are using /etc/smbpasswd, see the 'passwd'
#  module for an example of how to use /etc/smbpasswd

# authtype value, if present, will be used
# to overwrite (or add) Auth-Type during
# authorization. Normally should be MS-CHAP
authtype = MS-CHAP

# if use_mppe is not set to no mschap will
# add MS-CHAP-MPPE-Keys for MS-CHAPv1 and
# MS-MPPE-Recv-Key/MS-MPPE-Send-Key for MS-CHAPv2
#
#use_mppe = no
use_mppe = yes

# if mppe is enabled require_encryption makes
# encryption moderate
#
#require_encryption = yes
require_encryption = yes

# require_strong always requires 128 bit key
# encryption
#
require_strong = yes

# Windows sends us a username in the form of
# DOMAIN\user, but sends the challenge response
# based on only the user portion.  This hack
# corrects for that incorrect behavior.
#
#with_ntdomain_hack = no

# The module can perform authentication itself, OR
# use a Windows Domain Controller.  This configuration
```

```
# directive tells the module to call the ntlm_auth
# program, which will do the authentication, and return
# the NT-Key.  Note that you MUST have "winbindd" and
# "nmbd" running on the local machine for ntlm_auth
# to work.  See the ntlm_auth program documentation
# for details.
#
# Be VERY careful when editing the following line!
#
#ntlm_auth = "/path/to/ntlm_auth --request-nt-key --username=%{Stripped-User-Name:-%{User-Name
}

# Lightweight Directory Access Protocol (LDAP)
#
#  This module definition allows you to use LDAP for
#  authorization and authentication (Auth-Type := LDAP)
#
#  See doc/rlm_ldap for description of configuration options
#  and sample authorize{} and authenticate{} blocks
ldap {
server = localhost
identity = "cn=toor,dc=cn,dc=ntua,dc=gr"
password = deepest_secret
basedn = "ou=people,dc=cn,dc=ntua,dc=gr"
filter = "(uid=%{Stripped-User-Name:-%{User-Name}})"
# base_filter = "(objectclass=radiusprofile)"

# set this to 'yes' to use TLS encrypted connections
# to the LDAP database by using the StartTLS extended
# operation.
# The StartTLS operation is supposed to be used with normal
# ldap connections instead of using ldaps (port 689) connections
start_tls = no

# tls_cacertfile = /path/to/cacert.pem
# tls_cacertdir = /path/to/ca/dir/
# tls_certfile = /path/to/radius.crt
# tls_keyfile = /path/to/radius.key
# tls_randfile = /path/to/rnd
# tls_require_cert = "demand"

# default_profile = "cn=radprofile,ou=dialup,o=My Org,c=UA"
# profile_attribute = "radiusProfileDn"
################access_attr = "dialupAccess"

# Mapping of RADIUS dictionary attributes to LDAP
# directory attributes.
dictionary_mapping = ${raddbdir}/ldap.attrmap
```

```
ldap_connections_number = 5

#
# NOTICE: The password_header directive is NOT case insensitive
#
# password_header = "{clear}"
#
# Set:
# password_attribute = nspmPassword
#
# to get the user's password from a Novell eDirectory
# backend. This will work *only if* freeRADIUS is
# configured to build with --with-edir option.
#
#
#  The server can usually figure this out on its own, and pull
#  the correct User-Password or NT-Password from the database.
#
#  Note that NT-Passwords MUST be stored as a 32-digit hex
#  string, and MUST start off with "0x", such as:
#
# 0x000102030405060708090a0b0c0d0e0f
#
# password_attribute = nspmPassword
#  Without the leading "0x", NT-Passwords will not work.
#  This goes for NT-Passwords stored in SQL, too.
#
# Un-comment the following to disable Novell eDirectory account
# policy check and intruder detection. This will work *only if*
# FreeRADIUS is configured to build with --with-edir option.
#
# edir_account_policy_check=no
#
# groupname_attribute = cn
# groupmembership_filter = "(|(&(objectClass=GroupOfNames)(member=%{Ldap-UserDn}))(&(
# groupmembership_attribute = radiusGroupName
timeout = 4
timelimit = 3
net_timeout = 1
# compare_check_items = yes
# do_xlat = yes
# access_attr_used_for_allow = yes
}

# Realm module, for proxying.
#
#  You can have multiple instances of the realm module to
```

```
#  support multiple realm syntaxs at the same time.  The
#  search order is defined by the order in the authorize and
#  preacct sections.
#
#  Four config options:
# format         -  must be 'prefix' or 'suffix'
# delimiter      -  must be a single character
# ignore_default -  set to 'yes' or 'no'
#       ignore_null    -  set to 'yes' or 'no'
#
#  ignore_default and ignore_null can be set to 'yes' to prevent
#  the module from matching against DEFAULT or NULL realms.  This
#  may be useful if you have have multiple instances of the
#  realm module.
#
#  They both default to 'no'.
#

#  'realm/username'
#
#  Using this entry, IPASS users have their realm set to "IPASS".
realm IPASS {
format = prefix
delimiter = "/"
ignore_default = no
ignore_null = no
}

#  'username@realm'
#
realm suffix {
format = suffix
delimiter = "@"
ignore_default = no
ignore_null = no
}

#  'username%realm'
#
realm realmpercent {
format = suffix
delimiter = "%"
ignore_default = no
ignore_null = no
}

#
#  'domain\user'
```

```
#
realm ntdomain {
format = prefix
delimiter = "\\"
ignore_default = no
ignore_null = no
}

#  A simple value checking module
#
#  It can be used to check if an attribute value in the request
#  matches a (possibly multi valued) attribute in the check
#  items This can be used for example for caller-id
#  authentication.  For the module to run, both the request
#  attribute and the check items attribute must exist
#
#  i.e.
#  A user has an ldap entry with 2 radiusCallingStationId
#  attributes with values "12345678" and "12345679".  If we
#  enable rlm_checkval, then any request which contains a
#  Calling-Station-Id with one of those two values will be
#  accepted.  Requests with other values for
#  Calling-Station-Id will be rejected.
#
#  Regular expressions in the check attribute value are allowed
#  as long as the operator is '=~'
#
checkval {
# The attribute to look for in the request
item-name = Calling-Station-Id

# The attribute to look for in check items. Can be multi valued
check-name = Calling-Station-Id

# The data type. Can be
# string,integer,ipaddr,date,abinary,octets
data-type = string

# If set to yes and we dont find the item-name attribute in the
# request then we send back a reject
# DEFAULT is no
#notfound-reject = no
}

# Preprocess the incoming RADIUS request, before handing it off
# to other modules.
#
#  This module processes the 'huntgroups' and 'hints' files.
```

```
#  In addition, it re-writes some weird attributes created
#  by some NASes, and converts the attributes into a form which
#  is a little more standard.
#
preprocess {
huntgroups = ${confdir}/huntgroups
hints = ${confdir}/hints

# This hack changes Ascend's wierd port numberings
# to standard 0-??? port numbers so that the "+" works
# for IP address assignments.
with_ascend_hack = no
ascend_channels_per_line = 23

# Windows NT machines often authenticate themselves as
# NT_DOMAIN\username
#
# If this is set to 'yes', then the NT_DOMAIN portion
# of the user-name is silently discarded.
#
# This configuration entry SHOULD NOT be used.
# See the "realms" module for a better way to handle
# NT domains.
with_ntdomain_hack = no

# Specialix Jetstream 8500 24 port access server.
#
# If the user name is 10 characters or longer, a "/"
# and the excess characters after the 10th are
# appended to the user name.
#
# If you're not running that NAS, you don't need
# this hack.
with_specialix_jetstream_hack = no

# Cisco (and Quintum in Cisco mode) sends it's VSA attributes
# with the attribute name *again* in the string, like:
#
#   H323-Attribute = "h323-attribute=value".
#
# If this configuration item is set to 'yes', then
# the redundant data in the the attribute text is stripped
# out.  The result is:
#
#  H323-Attribute = "value"
#
# If you're not running a Cisco or Quintum NAS, you don't
# need this hack.
```

```
with_cisco_vsa_hack = no
}

# Livingston-style 'users' file
#
files {
usersfile = ${confdir}/users
acctusersfile = ${confdir}/acct_users

#  If you want to use the old Cistron 'users' file
#  with FreeRADIUS, you should change the next line
#  to 'compat = cistron'.  You can the copy your 'users'
#  file from Cistron.
compat = no
}

# Write a detailed log of all accounting records received.
#
detail {
#  Note that we do NOT use NAS-IP-Address here, as
#  that attribute MAY BE from the originating NAS, and
#  NOT from the proxy which actually sent us the
#  request.  The Client-IP-Address attribute is ALWAYS
#  the address of the client which sent us the
#  request.
#
#  The following line creates a new detail file for
#  every radius client (by IP address or hostname).
#  In addition, a new detail file is created every
#  day, so that the detail file doesn't have to go
#  through a 'log rotation'
#
#  If your detail files are large, you may also want
#  to add a ':%H' (see doc/variables.txt) to the end
#  of it, to create a new detail file every hour, e.g.:
#
#   ..../detail-%Y%m%d:%H
#
#  This will create a new detail file for every hour.
#
detailfile = ${radacctdir}/%{Client-IP-Address}/detail-%Y%m%d

#
#  The Unix-style permissions on the 'detail' file.
#
#  The detail file often contains secret or private
#  information about users.  So by keeping the file
#  permissions restrictive, we can prevent unwanted
```

```
#  people from seeing that information.
detailperm = 0600
}

#
#  Many people want to log authentication requests.
#  Rather than modifying the server core to print out more
#  messages, we can use a different instance of the 'detail'
#  module, to log the authentication requests to a file.
#
#  You will also need to un-comment the 'auth_log' line
#  in the 'authorize' section, below.
#
# detail auth_log {
# detailfile = ${radacctdir}/%{Client-IP-Address}/auth-detail-%Y%m%d

#
#  This MUST be 0600, otherwise anyone can read
#  the users passwords!
# detailperm = 0600
# }

#
#  This module logs authentication reply packets sent
#  to a NAS.  Both Access-Accept and Access-Reject packets
#  are logged.
#
#  You will also need to un-comment the 'reply_log' line
#  in the 'post-auth' section, below.
#
# detail reply_log {
# detailfile = ${radacctdir}/%{Client-IP-Address}/reply-detail-%Y%m%d

#
#  This MUST be 0600, otherwise anyone can read
#  the users passwords!
# detailperm = 0600
# }

# Create a unique accounting session Id.  Many NASes re-use or
# repeat values for Acct-Session-Id, causing no end of
# confusion.
#
#  This module will add a (probably) unique session id
#  to an accounting packet based on the attributes listed
#  below found in the packet.  See doc/rlm_acct_unique for
#  more information.
#
```

```
acct_unique {
key = "User-Name, Acct-Session-Id, NAS-IP-Address, Client-IP-Address, NAS-Port"
}


#  Include another file that has the SQL-related configuration.
#  This is another file only because it tends to be big.
#
#  The following configuration file is for use with MySQL.
#
#  For Postgresql, use: ${confdir}/postgresql.conf
#  For MS-SQL, use:   ${confdir}/mssql.conf
#  For Oracle, use:   ${confdir}/oraclesql.conf
#
$INCLUDE  ${confdir}/sql.conf

#  Write a 'utmp' style file, of which users are currently
#  logged in, and where they've logged in from.
#
#  This file is used mainly for Simultaneous-Use checking,
#  and also 'radwho', to see who's currently logged in.
#
radutmp {
#  Where the file is stored.  It's not a log file,
#  so it doesn't need rotating.
#
filename = ${logdir}/radutmp

#  The field in the packet to key on for the
#  'user' name,  If you have other fields which you want
#  to use to key on to control Simultaneous-Use,
#  then you can use them here.
#
#  Note, however, that the size of the field in the
#  'utmp' data structure is small, around 32
#  characters, so that will limit the possible choices
#  of keys.
#
#  You may want instead: %{Stripped-User-Name:-%{User-Name}}
username = %{User-Name}


#  Whether or not we want to treat "user" the same
#  as "USER", or "User".  Some systems have problems
#  with case sensitivity, so this should be set to
#  'no' to enable the comparisons of the key attribute
#  to be case insensitive.
#
```

```
case_sensitive = yes

#  Accounting information may be lost, so the user MAY
#  have logged off of the NAS, but we haven't noticed.
#  If so, we can verify this information with the NAS,
#
#  If we want to believe the 'utmp' file, then this
#  configuration entry can be set to 'no'.
#
check_with_nas = yes

# Set the file permissions, as the contents of this file
# are usually private.
perm = 0600

callerid = "yes"
}

# "Safe" radutmp - does not contain caller ID, so it can be
# world-readable, and radwho can work for normal users, without
# exposing any information that isn't already exposed by who(1).
#
# This is another 'instance' of the radutmp module, but it is given
# then name "sradutmp" to identify it later in the "accounting"
# section.
radutmp sradutmp {
filename = ${logdir}/sradutmp
perm = 0644
callerid = "no"
}

# attr_filter - filters the attributes received in replies from
# proxied servers, to make sure we send back to our RADIUS client
# only allowed attributes.
attr_filter {
attrsfile = ${confdir}/attrs
}

#  counter module:
#  This module takes an attribute (count-attribute).
#  It also takes a key, and creates a counter for each unique
#  key.  The count is incremented when accounting packets are
#  received by the server.  The value of the increment depends
#  on the attribute type.
#  If the attribute is Acct-Session-Time or of an integer type we add the
#  value of the attribute. If it is anything else we increase the
#  counter by one.
#
```

```
#  The 'reset' parameter defines when the counters are all reset to
#  zero.  It can be hourly, daily, weekly, monthly or never.
#
#  hourly: Reset on 00:00 of every hour
#  daily: Reset on 00:00:00 every day
#  weekly: Reset on 00:00:00 on sunday
#  monthly: Reset on 00:00:00 of the first day of each month
#
#  It can also be user defined. It should be of the form:
#  num[hdwm] where:
#  h: hours, d: days, w: weeks, m: months
#  If the letter is ommited days will be assumed. In example:
#  reset = 10h (reset every 10 hours)
#  reset = 12  (reset every 12 days)
#
#
#  The check-name attribute defines an attribute which will be
#  registered by the counter module and can be used to set the
#  maximum allowed value for the counter after which the user
#  is rejected.
#  Something like:
#
#  DEFAULT Max-Daily-Session := 36000
#          Fall-Through = 1
#
#  You should add the counter module in the instantiate
#  section so that it registers check-name before the files
#  module reads the users file.
#
#  If check-name is set and the user is to be rejected then we
#  send back a Reply-Message and we log a Failure-Message in
#  the radius.log
#  If the count attribute is Acct-Session-Time then on each login
#  we send back the remaining online time as a Session-Timeout attribute
#
#  The counter-name can also be used instead of using the check-name
#  like below:
#
#  DEFAULT  Daily-Session-Time > 3600, Auth-Type = Reject
#      Reply-Message = "You've used up more than one hour today"
#
#  The allowed-servicetype attribute can be used to only take
#  into account specific sessions. For example if a user first
#  logs in through a login menu and then selects ppp there will
#  be two sessions. One for Login-User and one for Framed-User
#  service type. We only need to take into account the second one.
#
#  The module should be added in the instantiate, authorize and
```

```
#  accounting sections.  Make sure that in the authorize
#  section it comes after any module which sets the
#  'check-name' attribute.
#
counter daily {
filename = ${raddbdir}/db.daily
key = User-Name
count-attribute = Acct-Session-Time
reset = daily
counter-name = Daily-Session-Time
check-name = Max-Daily-Session
allowed-servicetype = Framed-User
cache-size = 5000
}

# The "always" module is here for debugging purposes. Each
# instance simply returns the same result, always, without
# doing anything.
always fail {
rcode = fail
}
always reject {
rcode = reject
}
always ok {
rcode = ok
simulcount = 0
mpp = no
}

#
#  The 'expression' module currently has no configuration.
#
#  This module is useful only for 'xlat'.  To use it,
#  put 'exec' into the 'instantiate' section.  You can then
#  do dynamic translation of attributes like:
#
#  Attribute-Name = `%{expr:2 + 3 + %{exec: uid -u}}`
#
#  The value of the attribute will be replaced with the output
#  of the program which is executed.  Due to RADIUS protocol
#  limitations, any output over 253 bytes will be ignored.
expr {
}

#
#  The 'digest' module currently has no configuration.
#
```

```
#  "Digest" authentication against a Cisco SIP server.
#  See 'doc/rfc/draft-sterman-aaa-sip-00.txt' for details
#  on performing digest authentication for Cisco SIP servers.
#
digest {
}

#
#  Execute external programs
#
#  This module is useful only for 'xlat'.  To use it,
#  put 'exec' into the 'instantiate' section.  You can then
#  do dynamic translation of attributes like:
#
#  Attribute-Name = `%{exec:/path/to/program args}`
#
#  The value of the attribute will be replaced with the output
#  of the program which is executed.  Due to RADIUS protocol
#  limitations, any output over 253 bytes will be ignored.
#
#  The RADIUS attributes from the user request will be placed
#  into environment variables of the executed program, as
#  described in 'doc/variables.txt'
#
exec {
wait = yes
input_pairs = request
}

#
#  This is a more general example of the execute module.
#
#  This one is called "echo".
#
#  Attribute-Name = `%{echo:/path/to/program args}`
#
#  If you wish to execute an external program in more than
#  one section (e.g. 'authorize', 'pre_proxy', etc), then it
#  is probably best to define a different instance of the
#  'exec' module for every section.
#
exec echo {
#
#  Wait for the program to finish.
#
#  If we do NOT wait, then the program is "fire and
#  forget", and any output attributes from it are ignored.
#
```

```
#  If we are looking for the program to output
#  attributes, and want to add those attributes to the
#  request, then we MUST wait for the program to
#  finish, and therefore set 'wait=yes'
#
# allowed values: {no, yes}
wait = yes

#
#  The name of the program to execute, and it's
#  arguments.  Dynamic translation is done on this
#  field, so things like the following example will
#  work.
#
program = "/bin/echo %{User-Name}"

#
#  The attributes which are placed into the
#  environment variables for the program.
#
#  Allowed values are:
#
# request attributes from the request
# config attributes from the configuration items list
# reply attributes from the reply
# proxy-request attributes from the proxy request
# proxy-reply attributes from the proxy reply
#
#  Note that some attributes may not exist at some
#  stages.  e.g. There may be no proxy-reply
#  attributes if this module is used in the
#  'authorize' section.
#
input_pairs = request

#
#  Where to place the output attributes (if any) from
#  the executed program.  The values allowed, and the
#  restrictions as to availability, are the same as
#  for the input_pairs.
#
output_pairs = reply

#
#  When to execute the program.  If the packet
#  type does NOT match what's listed here, then
#  the module does NOT execute the program.
#
```

```
#  For a list of allowed packet types, see
#  the 'dictionary' file, and look for VALUEs
#  of the Packet-Type attribute.
#
#  By default, the module executes on ANY packet.
#  Un-comment out the following line to tell the
#  module to execute only if an Access-Accept is
#  being sent to the NAS.
#
#packet_type = Access-Accept
}

#  Do server side ip pool management. Should be added in post-auth and
#  accounting sections.
#
#  The module also requires the existance of the Pool-Name
#  attribute. That way the administrator can add the Pool-Name
#  attribute in the user profiles and use different pools
#  for different users. The Pool-Name attribute is a *check* item not
#  a reply item.
#
# Example:
# radiusd.conf: ippool students { [...] }
# users file  : DEFAULT Group == students, Pool-Name := "students"
#
# ********* IF YOU CHANGE THE RANGE PARAMETERS YOU MUST *********
# ********* THEN ERASE THE DB FILES                     *********
#
ippool main_pool {

#  range-start,range-stop: The start and end ip
#  addresses for the ip pool
range-start = 192.168.1.1
range-stop = 192.168.3.254

#  netmask: The network mask used for the ip's
netmask = 255.255.255.0

#  cache-size: The gdbm cache size for the db
#  files. Should be equal to the number of ip's
#  available in the ip pool
cache-size = 800

# session-db: The main db file used to allocate ip's to clients
session-db = ${raddbdir}/db.ippool

# ip-index: Helper db index file used in multilink
ip-index = ${raddbdir}/db.ipindex
```

```
# override: Will this ippool override a Framed-IP-Address already set
override = no

# maximum-timeout: If not zero specifies the maximum time in seconds an
# entry may be active. Default: 0
maximum-timeout = 0
}

# ANSI X9.9 token support.  Not included by default.
# $INCLUDE  ${confdir}/x99.conf

}

# Instantiation
#
#  This section orders the loading of the modules.  Modules
#  listed here will get loaded BEFORE the later sections like
#  authorize, authenticate, etc. get examined.
#
#  This section is not strictly needed.  When a section like
#  authorize refers to a module, it's automatically loaded and
#  initialized.  However, some modules may not be listed in any
#  of the following sections, so they can be listed here.
#
#  Also, listing modules here ensures that you have control over
#  the order in which they are initalized.  If one module needs
#  something defined by another module, you can list them in order
#  here, and ensure that the configuration will be OK.
#
instantiate {
#
#  Allows the execution of external scripts.
#  The entire command line (and output) must fit into 253 bytes.
#
#  e.g. Framed-Pool = `%{exec:/bin/echo foo}`
exec

#
#  The expression module doesn't do authorization,
#  authentication, or accounting.  It only does dynamic
#  translation, of the form:
#
# Session-Timeout = `%{expr:2 + 3}`
#
#  So the module needs to be instantiated, but CANNOT be
#  listed in any other section.  See 'doc/rlm_expr' for
#  more information.
```

```
#
expr

#
# We add the counter module here so that it registers
# the check-name attribute before any module which sets
# it
# daily
}

#  Authorization. First preprocess (hints and huntgroups files),
#  then realms, and finally look in the "users" file.
#
#  The order of the realm modules will determine the order that
#  we try to find a matching realm.
#
#  Make *sure* that 'preprocess' comes before any realm if you
#  need to setup hints for the remote radius server
authorize {
#
#  The preprocess module takes care of sanitizing some bizarre
#  attributes in the request, and turning them into attributes
#  which are more standard.
#
#  It takes care of processing the 'raddb/hints' and the
#  'raddb/huntgroups' files.
#
#  It also adds the %{Client-IP-Address} attribute to the request.
preprocess

#
#  If you want to have a log of authentication requests,
#  un-comment the following line, and the 'detail auth_log'
#  section, above.
# auth_log

# attr_filter

#
#  If the users are logging in with an MS-CHAP-Challenge
#  attribute for authentication, the mschap module will find
#  the MS-CHAP-Challenge attribute, and add 'Auth-Type := MS-CHAP'
#  to the request, which will cause the server to then use
#  the mschap module for authentication.
mschap

#
#  If you are using multiple kinds of realms, you probably
```

```
#  want to set "ignore_null = yes" for all of them.
#  Otherwise, when the first style of realm doesn't match,
#  the other styles won't be checked.
#
suffix
# ntdomain

#
#  This module takes care of EAP-MD5, EAP-TLS, and EAP-LEAP
#  authentication.
#
#  It also sets the EAP-Type attribute in the request
#  attribute list to the EAP type from the packet.
eap

#
#  Read the 'users' file
files

#
#  The ldap module will set Auth-Type to LDAP if it has not
#  already been set
ldap

#
#  Enforce daily limits on time spent logged in.
# daily

#
# Use the checkval module
# checkval
}


#  Authentication.
#
#
#  This section lists which modules are available for authentication.
#  Note that it does NOT mean 'try each module in order'.  It means
#  that a module from the 'authorize' section adds a configuration
#  attribute 'Auth-Type := FOO'.  That authentication type is then
#  used to pick the apropriate module from the list below.
#

#  In general, you SHOULD NOT set the Auth-Type attribute.  The server
#  will figure it out on its own, and will do the right thing.  The
#  most common side effect of erroneously setting the Auth-Type
#  attribute is that one authentication method will work, but the
```

```
#  others will not.
#
#  The common reasons to set the Auth-Type attribute by hand
#  is to either forcibly reject the user, or forcibly accept him.
#
authenticate {
#
#  MSCHAP authentication.
Auth-Type MS-CHAP {
mschap
}

#
#  Allow EAP authentication.
eap
}


#
#  Pre-accounting.  Decide which accounting type to use.
#
preacct {
preprocess

#
#  Ensure that we have a semi-unique identifier for every
#  request, and many NAS boxes are broken.
acct_unique

#
#  Look for IPASS-style 'realm/', and if not found, look for
#  '@realm', and decide whether or not to proxy, based on
#  that.
#
#  Accounting requests are generally proxied to the same
#  home server as authentication requests.
# IPASS
suffix
# ntdomain

#
#  Read the 'acct_users' file
files
}

#
#  Accounting.  Log the accounting data.
#
```

```
accounting {
#
#  Create a 'detail'ed log of the packets.
#  Note that accounting requests which are proxied
#  are also logged in the detail file.
detail
# daily

#  Update the wtmp file
#
#  If you don't use "radlast", you can delete this line.
unix

#
#  For Simultaneous-Use tracking.
#
#  Due to packet losses in the network, the data here
#  may be incorrect.  There is little we can do about it.
radutmp
# sradutmp

#  Return an address to the IP Pool when we see a stop record.
# main_pool

}


#  Session database, used for checking Simultaneous-Use. Either the radutmp
#  or rlm_sql module can handle this.
#  The rlm_sql module is *much* faster
session {
radutmp

#
#  See "Simultaneous Use Checking Querie" in sql.conf
# sql
}


#  Post-Authentication
#  Once we KNOW that the user has been authenticated, there are
#  additional steps we can take.
post-auth {
#  Get an address from the IP Pool.
# main_pool

#
#  If you want to have a log of authentication replies,
```

```
#  un-comment the following line, and the 'detail reply_log'
#  section, above.
# reply_log

#
#  After authenticating the user, do another SQL qeury.
#
#  See "Authentication Logging Queries" in sql.conf
# sql

#
#  Un-comment the following if you have set
#  'edir_account_policy_check = yes' in the ldap module sub-section of
#  the 'modules' section.
#
# ldap
#
#  Access-Reject packets are sent through the REJECT sub-section of the
#  post-auth section.
#  Uncomment the following and set the module name to the ldap instance
#  name if you have set 'edir_account_policy_check = yes' in the ldap
#  module sub-section of the 'modules' section.
#
# Post-Auth-Type REJECT {
# insert-module-name-here
# }

}

#
#  When the server decides to proxy a request to a home server,
#  the proxied request is first passed through the pre-proxy
#  stage.  This stage can re-write the request, or decide to
#  cancel the proxy.
#
#  Only a few modules currently have this method.
#
pre-proxy {
# attr_rewrite

#  If you want to have a log of packets proxied to a home
#  server, un-comment the following line, and the
#  'detail pre_proxy_log' section, above.
# pre_proxy_log
}

#
#  When the server receives a reply to a request it proxied
```

```
#  to a home server, the request may be massaged here, in the
#  post-proxy stage.
#
post-proxy {
#

#  If you want to have a log of replies from a home server,
#  un-comment the following line, and the 'detail post_proxy_log'
#  section, above.
# post_proxy_log

# attr_rewrite

#  Uncomment the following line if you want to filter replies from
#  remote proxies based on the rules defined in the 'attrs' file.

# attr_filter

#
#  If you are proxying LEAP, you MUST configure the EAP
#  module, and you MUST list it here, in the post-proxy
#  stage.
#
#  You MUST also use the 'nostrip' option in the 'realm'
#  configuration.  Otherwise, the User-Name attribute
#  in the proxied request will not match the user name
#  hidden inside of the EAP packet, and the end server will
#  reject the EAP request.
#
eap
}
```

Β΄.1.2 Αρχείο eap.conf

```
#
#  Whatever you do, do NOT set 'Auth-Type := EAP'.  The server
#  is smart enough to figure this out on its own.  The most
#  common side effect of setting 'Auth-Type := EAP' is that the
#  users then cannot use ANY other authentication method.
#
# $Id: eap.conf,v 1.4 2004/04/15 18:34:41 aland Exp $
#
eap {
#  Invoke the default supported EAP type when
#  EAP-Identity response is received.
#
#  The incoming EAP messages DO NOT specify which EAP
#  type they will be using, so it MUST be set here.
```

```
#
#  For now, only one default EAP type may be used at a time.
#
#  If the EAP-Type attribute is set by another module,
#  then that EAP type takes precedence over the
#  default type configured here.
#
#default_eap_type = md5
default_eap_type = peap

#  A list is maintained to correlate EAP-Response
#  packets with EAP-Request packets.  After a
#  configurable length of time, entries in the list
#  expire, and are deleted.
#
timer_expire     = 60

#  There are many EAP types, but the server has support
#  for only a limited subset.  If the server receives
#  a request for an EAP type it does not support, then
#  it normally rejects the request.  By setting this
#  configuration to "yes", you can tell the server to
#  instead keep processing the request.  Another module
#  MUST then be configured to proxy the request to
#  another RADIUS server which supports that EAP type.
#
#  If another module is NOT configured to handle the
#  request, then the request will still end up being
#  rejected.
ignore_unknown_eap_types = no

# Cisco AP1230B firmware 12.2(13)JA1 has a bug.  When given
# a User-Name attribute in an Access-Accept, it copies one
# more byte than it should.
#
# We can work around it by configurably adding an extra
# zero byte.
cisco_accounting_username_bug = no

## EAP-TLS
#
#  To generate ctest certificates, run the script
#
# ../scripts/certs.sh
#
#  The documents on http://www.freeradius.org/doc
#  are old, but may be helpful.
#
```

```
#  See also:
#
#  http://www.dslreports.com/forum/remark,9286052~mode=flat
#
tls {
private_key_password = d33p_s3cr3t
private_key_file = ${raddbdir}/certs/cert-srv.pem

#  If Private key & Certificate are located in
#  the same file, then private_key_file &
#  certificate_file must contain the same file
#  name.
certificate_file = ${raddbdir}/certs/cert-srv.pem

#  Trusted Root CA list
CA_file = ${raddbdir}/certs/demoCA/cacert.pem

dh_file = ${raddbdir}/certs/dh
random_file = /dev/urandom

#
#  This can never exceed the size of a RADIUS
#  packet (4096 bytes), and is preferably half
#  that, to accomodate other attributes in
#  RADIUS packet.  On most APs the MAX packet
#  length is configured between 1500 - 1600
#  In these cases, fragment size should be
#  1024 or less.
#
# fragment_size = 1024

#  include_length is a flag which is
#  by default set to yes If set to
#  yes, Total Length of the message is
#  included in EVERY packet we send.
#  If set to no, Total Length of the
#  message is included ONLY in the
#  First packet of a fragment series.
#
# include_length = yes

#  Check the Certificate Revocation List
#
#  1) Copy CA certificates and CRLs to same directory.
#  2) Execute 'c_rehash <CA certs&CRLs Directory>'.
#    'c_rehash' is OpenSSL's command.
#  3) Add 'CA_path=<CA certs&CRLs directory>'
#      to radiusd.conf's tls section.
```

```
#  4) uncomment the line below.
#  5) Restart radiusd
# check_crl = yes

                        #
                        #  If check_cert_cn is set, the value will
                        #  be xlat'ed and checked against the CN
                        #  in the client certificate.  If the values
                        #  do not match, the certificate verification
                        #  will fail rejecting the user.
                        #
                #       check_cert_cn = %{User-Name}
}

#
#  The tunneled EAP session needs a default EAP type
#  which is separate from the one for the non-tunneled
#  EAP module.  Inside of the TLS/PEAP tunnel, we
#  recommend using EAP-MS-CHAPv2.
#
#  The PEAP module needs the TLS module to be installed
#  and configured, in order to use the TLS tunnel
#  inside of the EAP packet.  You will still need to
#  configure the TLS module, even if you do not want
#  to deploy EAP-TLS in your network.  Users will not
#  be able to request EAP-TLS, as it requires them to
#  have a client certificate.  EAP-PEAP does not
#  require a client certificate.
#
peap {
#  The tunneled EAP session needs a default
#  EAP type which is separate from the one for
#  the non-tunneled EAP module.  Inside of the
#  PEAP tunnel, we recommend using MS-CHAPv2,
#  as that is the default type supported by
#  Windows clients.
default_eap_type = mschapv2
}

#
#  This takes no configuration.
#
#  Note that it is the EAP MS-CHAPv2 sub-module, not
#  the main 'mschap' module.
#
#  Note also that in order for this sub-module to work,
#  the main 'mschap' module MUST ALSO be configured.
#
```

```
#  This module is the *Microsoft* implementation of MS-CHAPv2
#  in EAP.  There is another (incompatible) implementation
#  of MS-CHAPv2 in EAP by Cisco, which FreeRADIUS does not
#  currently support.
#
mschapv2 {
}
}
```

Β΄.1.3 Αρχείο ldap.attrmap

```
#
# Mapping of RADIUS dictionary attributes to LDAP directory attributes
# to be used by LDAP authentication and authorization module (rlm_ldap)
#
# Format:
#   ItemType RADIUS-Attribute-Name ldapAttributeName
#
# Where:
#   ItemType              = checkItem or replyItem
#   RADIUS-Attribute-Name = attribute name in RADIUS dictionary
#   ldapAttributeName     = attribute name in LDAP schema
#
# If $GENERIC$ is specified as RADIUS-Attribute-Name, the line specifies
# a LDAP attribute which can be used to store any RADIUS
# attribute/value-pair in LDAP directory.
#
# You should edit this file to suit it to your needs.
#

checkItem $GENERIC$ radiusCheckItem
replyItem $GENERIC$ radiusReplyItem

checkItem Auth-Type radiusAuthType
checkItem Simultaneous-Use radiusSimultaneousUse
checkItem Called-Station-Id radiusCalledStationId
checkItem Calling-Station-Id radiusCallingStationId
checkItem LM-Password sambaLMPassword
checkItem NT-Password sambaNTPassword
checkItem User-Password sambaNTPassword
checkItem SMB-Account-CTRL-TEXT acctFlags
checkItem Expiration radiusExpiration
checkItem Login-Time sambaLogonHours
replyItem Service-Type radiusServiceType
replyItem Framed-Protocol radiusFramedProtocol
replyItem Framed-IP-Address radiusFramedIPAddress
replyItem Framed-IP-Netmask radiusFramedIPNetmask
```

```
replyItem Framed-Route radiusFramedRoute
replyItem Framed-Routing radiusFramedRouting
replyItem Filter-Id radiusFilterId
replyItem Framed-MTU radiusFramedMTU
replyItem Framed-Compression radiusFramedCompression
replyItem Login-IP-Host radiusLoginIPHost
replyItem Login-Service radiusLoginService
replyItem Login-TCP-Port radiusLoginTCPPort
replyItem Callback-Number radiusCallbackNumber
replyItem Callback-Id radiusCallbackId
replyItem Framed-IPX-Network radiusFramedIPXNetwork
replyItem Class radiusClass
replyItem Session-Timeout radiusSessionTimeout
replyItem Idle-Timeout radiusIdleTimeout
replyItem Termination-Action radiusTerminationAction
replyItem Login-LAT-Service radiusLoginLATService
replyItem Login-LAT-Node radiusLoginLATNode
replyItem Login-LAT-Group radiusLoginLATGroup
replyItem Framed-AppleTalk-Link radiusFramedAppleTalkLink
replyItem Framed-AppleTalk-Network radiusFramedAppleTalkNetwork
replyItem Framed-AppleTalk-Zone radiusFramedAppleTalkZone
replyItem Port-Limit radiusPortLimit
replyItem Login-LAT-Port radiusLoginLATPort
```

Β′.1.4 Αρχείο clients.conf

```
#######################################################################
#
#  Definition of a RADIUS client (usually a NAS).
#
#  The information given here over rides anything given in the
#  'clients' file, or in the 'naslist' file.  The configuration here
#  contains all of the information from those two files, and allows
#  for more configuration items.
#
#  The "shortname" is be used for logging.  The "nastype", "login" and
#  "password" fields are mainly used for checkrad and are optional.
#

#
#  Defines a RADIUS client.  The format is 'client [hostname|ip-address]'
#

client 172.24.100.241 {
nastype     = cisco
shortname   = death-star
secret      = DEEP_SECRET
}
```

```
client 147.102.40.242 {
nastype     = cisco
shortname   = death-star-ii
secret      = DEEP_SECRET
}
```

Β΄.2 Ρυθμίσεις εξυπηρετητή SAMBA

Β΄.2.1 Αρχείο smb.conf

```
#======================= Global Settings =====================================
[global]

   workgroup = CN
   netbios name = PSYCHE
   server string = PDC
   enable privileges = yes
#   username map = /etc/samba/smbusers
   security = user
   encrypt passwords = yes
   log file = /var/log/samba/log.%m
   max log size = 1000
   time server = yes
   socket options = TCP_NODELAY
#   logon script = logon.bat
#   logon drive = H:
#   logon home =
#   logon path =
   domain logons = yes
   os level = 65
   preferred master = yes
   domain master = yes
   wins support = yes

   ldap passwd sync = yes
   passdb backend = ldapsam:ldap://127.0.0.1/
#   ldap filter = (&(objectclass=sambaSamAccount)(uid=%u))
   ldap admin dn = cn=toor,dc=cn,dc=ntua,dc=gr
   ldap suffix = dc=cn,dc=ntua,dc=gr
   ldap group suffix = ou=groups
   ldap user suffix = ou=people
   ldap machine suffix = ou=computers
#   ldap ssl = start_tls

   add machine script = /usr/local/sbin/smbldap-useradd -w "%u"
   add user script = /usr/local/sbin/smbldap-useradd -m "%u"
```

```
   ldap delete dn = Yes
#   delete user script = /usr/local/sbin/smbldap-userdel "%u"
   add group script = /usr/local/sbin/smbldap-groupadd -p "%g"
#   delete group script = /usr/local/sbin/smbldap-groupdel "%g"
   add user to group script = /usr/local/sbin/smbldap-groupmod -m "%u" "%g"
   delete user from group script = /usr/local/sbin/smbldap-groupmod -x "%u" "%g"
   set primary group script = /usr/local/sbin/smbldap-usermod -g "%g" "%u"


# charset settings
;   display charset = ASCII
;   unix charset = ASCII
;   dos charset = ASCII


#============================ Share Definitions ==============================
[homes]
   comment = Home Directories
   browseable = no
   writable = yes

# Un-comment the following and create the netlogon directory for Domain Logons
; [netlogon]
;   comment = Network Logon Service
;   path = /usr/local/samba/lib/netlogon
;   guest ok = yes
;   writable = no
;   share modes = no


# Un-comment the following to provide a specific roving profile share
# the default is to use the user's home directory
;[Profiles]
;    path = /usr/local/samba/profiles
;    browseable = no
;    guest ok = yes


# NOTE: If you have a BSD-style print system there is no need to
# specifically define each individual printer
[printers]
   comment = All Printers
   path = /var/spool/samba
   browseable = no
# Set public = yes to allow user 'guest account' to print
   guest ok = no
   writable = no
   printable = yes
```

```
# This one is useful for people to share files
;[tmp]
;   comment = Temporary file space
;   path = /tmp
;   read only = no
;   public = yes

# A publicly accessible directory, but read only, except for people in
# the "staff" group
;[public]
;   comment = Public Stuff
;   path = /home/samba
;   public = yes
;   writable = yes
;   printable = no
;   write list = @staff
```

Β΄.3 Ρυθμίσεις εξυπηρετητή OpenLDAP

Β΄.3.1 Αρχείο slapd.conf

```
include /usr/local/etc/openldap/schema/core.schema
include     /usr/local/etc/openldap/schema/cosine.schema
include /usr/local/etc/openldap/schema/nis.schema
include /usr/local/etc/openldap/schema/inetorgperson.schema
include /usr/local/etc/openldap/schema/qmail.schema
include /usr/local/etc/openldap/schema/samba.schema

pidfile /var/run/openldap/slapd.pid
argsfile /var/run/openldap/slapd.args

# rootdn can always write!

access to dn.one="ou=people,dc=cn,dc=ntua,dc=gr" attrs=userPassword,sambaNTPassword,sambaLMPas
  by self write
  by anonymous auth
  by * none

access to dn.one="ou=people,dc=cn,dc=ntua,dc=gr" attrs=mailForwardingAddress,deliveryMode,tele
  by self write
  by * read

access to *
  by * read

loglevel 256
```

```
allow bind_v2

########################################################################
# ldbm database definitions
########################################################################

database bdb
suffix "dc=cn,dc=ntua,dc=gr"
rootdn "cn=toor,dc=cn,dc=ntua,dc=gr"
# Cleartext passwords, especially for the rootdn, should
# be avoid.  See slappasswd(8) and slapd.conf(5) for details.
# Use of strong authentication encouraged.
rootpw {SSHA}8qfgA9g3SHQsCLHZeCAKLQXpNdLCtFNj
# The database directory MUST exist prior to running slapd AND
# should only be accessible by the slapd and slap tools.
# Mode 700 recommended.
directory /var/db/openldap-data


# Indices to maintain
index   objectClass,uidNumber,gidNumber                 eq
index   cn,sn,uid,displayName                           pres,sub,eq
index   memberUid,mail,givenname                        eq,subinitial
index   sambaSID,sambaPrimaryGroupSID,sambaDomainName   eq

index   mailAlternateAddress                            eq
index   accountStatus                                   eq
index   deliveryMode                                    eq
```

Β′.4 Ρυθμίσεις εργαλείων smbldap

Β′.4.1 Αρχείο smbldap.conf

```
# $Source: /opt/cvs/samba/smbldap-tools/smbldap.conf,v $
# $Id: smbldap.conf,v 1.17 2005/01/29 15:00:54 jtournier Exp $
#
# smbldap-tools.conf : Q & D configuration file for smbldap-tools

#  This code was developped by IDEALX (http://IDEALX.org/) and
#  contributors (their names can be found in the CONTRIBUTORS file).
#
#                 Copyright (C) 2001-2002 IDEALX
#
#  This program is free software; you can redistribute it and/or
#  modify it under the terms of the GNU General Public License
#  as published by the Free Software Foundation; either version 2
```

```
#  of the License, or (at your option) any later version.
#
#  This program is distributed in the hope that it will be useful,
#  but WITHOUT ANY WARRANTY; without even the implied warranty of
#  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
#  GNU General Public License for more details.
#
#  You should have received a copy of the GNU General Public License
#  along with this program; if not, write to the Free Software
#  Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307,
#  USA.

#  Purpose :
#       . be the configuration file for all smbldap-tools scripts

##############################################################################
#
# General Configuration
#
##############################################################################

# Put your own SID
# to obtain this number do: net getlocalsid
SID="S-1-5-21-1580129704-204451263-851912754"

##############################################################################
#
# LDAP Configuration
#
##############################################################################

# Notes: to use to dual ldap servers backend for Samba, you must patch
# Samba with the dual-head patch from IDEALX. If not using this patch
# just use the same server for slaveLDAP and masterLDAP.
# Those two servers declarations can also be used when you have
# . one master LDAP server where all writing operations must be done
# . one slave LDAP server where all reading operations must be done
#   (typically a replication directory)

# Ex: slaveLDAP=127.0.0.1
slaveLDAP="127.0.0.1"
slavePort="389"

# Master LDAP : needed for write operations
# Ex: masterLDAP=127.0.0.1
masterLDAP="127.0.0.1"
masterPort="389"
```

```
# Use TLS for LDAP
# If set to 1, this option will use start_tls for connection
# (you should also used the port 389)
ldapTLS="0"

# How to verify the server's certificate (none, optional or require)
# see "man Net::LDAP" in start_tls section for more details
verify="require"

# CA certificate
# see "man Net::LDAP" in start_tls section for more details
cafile="/usr/local/etc/smbldap-tools/ca.pem"

# certificate to use to connect to the ldap server
# see "man Net::LDAP" in start_tls section for more details
clientcert="/usr/local/etc/smbldap-tools/smbldap-tools.pem"

# key certificate to use to connect to the ldap server
# see "man Net::LDAP" in start_tls section for more details
clientkey="/usr/local/etc/smbldap-tools/smbldap-tools.key"

# LDAP Suffix
# Ex: suffix=dc=IDEALX,dc=ORG
suffix="dc=cn,dc=ntua,dc=gr"

# Where are stored Users
# Ex: usersdn="ou=people,dc=IDEALX,dc=ORG"
usersdn="ou=people,${suffix}"

# Where are stored Computers
# Ex: computersdn="ou=Computers,dc=IDEALX,dc=ORG"
computersdn="ou=computers,${suffix}"

# Where are stored Groups
# Ex groupsdn="ou=Groups,dc=IDEALX,dc=ORG"
groupsdn="ou=groups,${suffix}"

# Where are stored Idmap entries (used if samba is a domain member server)
# Ex groupsdn="ou=Idmap,dc=IDEALX,dc=ORG"
idmapdn="ou=Idmap,${suffix}"

# Where to store next uidNumber and gidNumber available
sambaUnixIdPooldn="sambaDomainName=CN,${suffix}"

# Default scope Used
scope="sub"

# Unix password encryption (CRYPT, MD5, SMD5, SSHA, SHA, CLEARTEXT)
```

```
hash_encrypt="SSHA"

# if hash_encrypt is set to CRYPT, you may set a salt format.
# default is "%s", but many systems will generate MD5 hashed
# passwords if you use "$1$%.8s". This parameter is optional!
crypt_salt_format="%s"

##############################################################################
#
# Unix Accounts Configuration
#
##############################################################################

# Login defs
# Default Login Shell
# Ex: userLoginShell="/bin/bash"
userLoginShell="/bin/bash"

# Home directory
# Ex: userHome="/home/%U"
userHome="/home/%U"

# Gecos
userGecos="System User"

# Default User (POSIX and Samba) GID
defaultUserGid="513"

# Default Computer (Samba) GID
defaultComputerGid="515"

# Skel dir
skeletonDir="/etc/skel"

# Default password validation time (time in days) Comment the next line if
# you don't want password to be enable for defaultMaxPasswordAge days (be
# careful to the sambaPwdMustChange attribute's value)
#defaultMaxPasswordAge="99"

##############################################################################
#
# SAMBA Configuration
#
##############################################################################

# The UNC path to home drives location (%U username substitution)
# Ex: \\My-PDC-netbios-name\homes\%U
# Just set it to a null string if you want to use the smb.conf 'logon home'
```

```
# directive and/or disable roaming profiles
userSmbHome="\\PSYCHE\homes\%U"

# The UNC path to profiles locations (%U username substitution)
# Ex: \\My-PDC-netbios-name\profiles\%U
# Just set it to a null string if you want to use the smb.conf 'logon path'
# directive and/or disable roaming profiles
userProfile="\\PCYCHE\profiles\%U"

# The default Home Drive Letter mapping
# (will be automatically mapped at logon time if home directory exist)
# Ex: H: for H:
userHomeDrive="H:"

# The default user netlogon script name (%U username substitution)
# if not used, will be automatically username.cmd
# make sure script file is edited under dos
# Ex: %U.cmd
# userScript="startup.cmd" # make sure script file is edited under dos
userScript="%U.cmd"

# Domain appended to the users "mail"-attribute
# when smbldap-useradd -M is used
mailDomain="cn.ntua.gr"

##############################################################################
#
# SMBLDAP-TOOLS Configuration (default are ok for a RedHat)
#
##############################################################################

# Allows not to use smbpasswd (if with_smbpasswd == 0 in smbldap_conf.pm) but
# prefer Crypt::SmbHash library
with_smbpasswd="0"
smbpasswd="/usr/bin/smbpasswd"

# Allows not to use slappasswd (if with_slappasswd == 0 in smbldap_conf.pm)
# but prefer Crypt:: libraries
with_slappasswd="0"
slappasswd="/usr/sbin/slappasswd"
```

Β′.4.2 Αρχείο smbldap_bind.conf

```
############################
# Credential Configuration #
############################
# Notes: you can specify two differents configuration if you use a
```

```
# master ldap for writing access and a slave ldap server for reading access
# By default, we will use the same DN (so it will work for standard Samba
# release)
slaveDN="cn=toor,dc=cn,dc=ntua,dc=gr"
slavePw="deepest_secret"
masterDN="cn=toor,dc=cn,dc=ntua,dc=gr"
masterPw="deepest_secret"
```

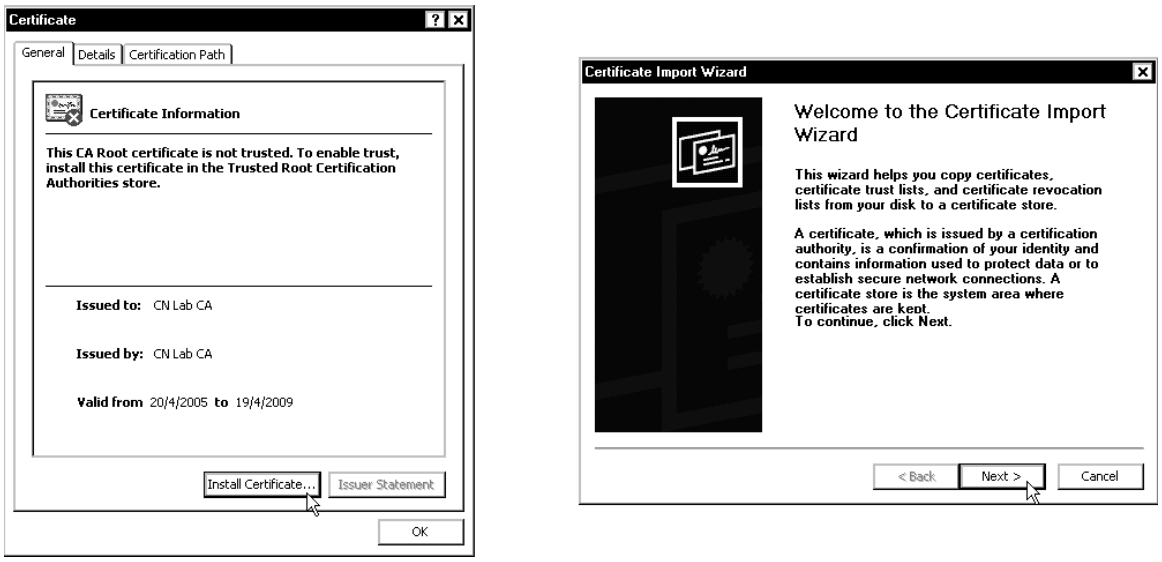
Παράρτημα Γ΄

Στο παράρτημα αυτό, παρουσιάζεται η διαδικασία εγκατάστασης του πιστοποιητικού του εξυπηρετητή RADIUS από ένα ασύρματο πελάτη, καθώς επίσης, και οι κατάλληλες ρυθμίσεις για τη σύνδεση στο Ασύρματο Τοπικό Δίκτυο TSN του Εργαστηρίου Δικτύων Υπολογιστών.

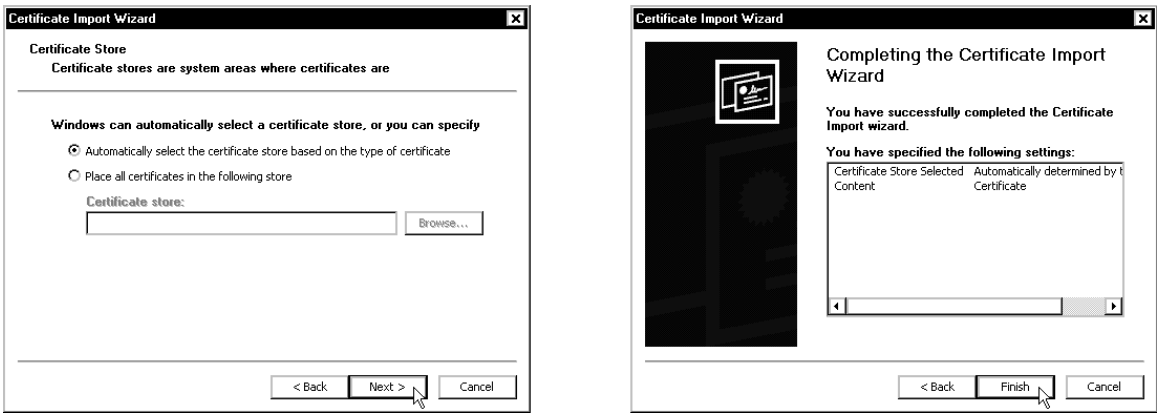
Γ΄.1 Εγκατάσταση πιστοποιητικού εξυπηρετητή RADIUS

Η ενότητα αυτή, παρουσιάζει τη διαδικασία εγκατάστασης του πιστοποιητικού του εξυπηρετητή RADIUS του Ασύρματου Τοπικού Δικτύου TSN του Εργαστηρίου Δικτύων Υπολογιστών από ένα χρήστη λειτουργικού συστήματος Microsoft Windows έκδοσης XP.

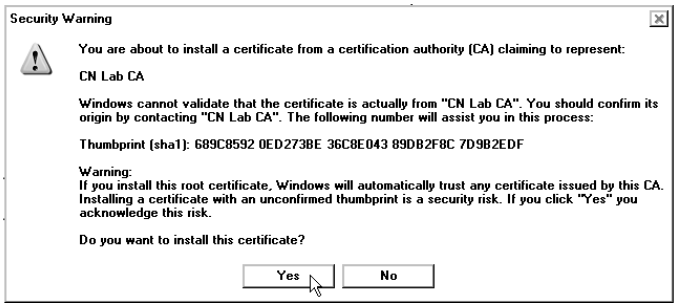
Ο ασύρματος πελάτης, αφού προμηθευτεί το πιστοποιητικό (αρχείο `cn-cert.der`), ακολουθεί τα παρακάτω βήματα:



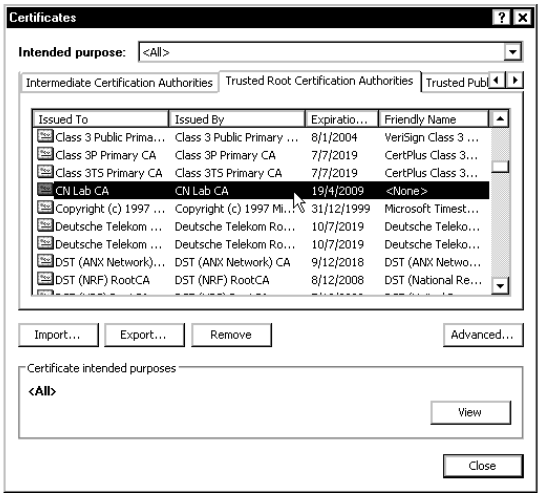
Σχήμα Γ΄.1: Εγκατάσταση πιστοποιητικού εξυπηρετητή RADIUS (βήματα 1-2)



Σχήμα Γ΄.2: Εγκατάσταση πιστοποιητικού εξυπηρετητή RADIUS (βήματα 3-4)



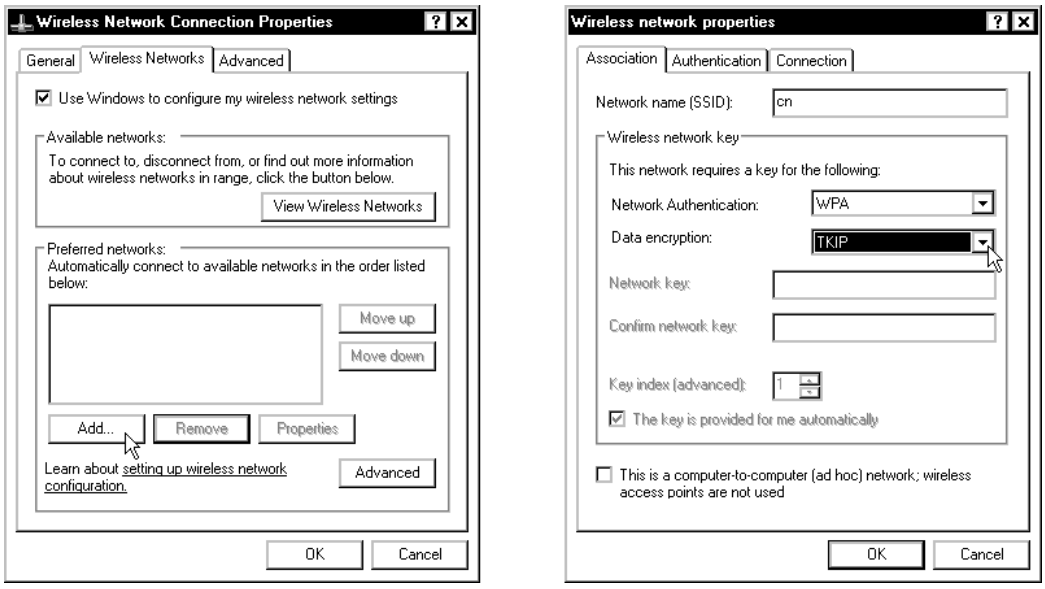
Σχήμα Γ΄.3: Εγκατάσταση πιστοποιητικού εξυπηρετητή RADIUS (βήμα 5^o)



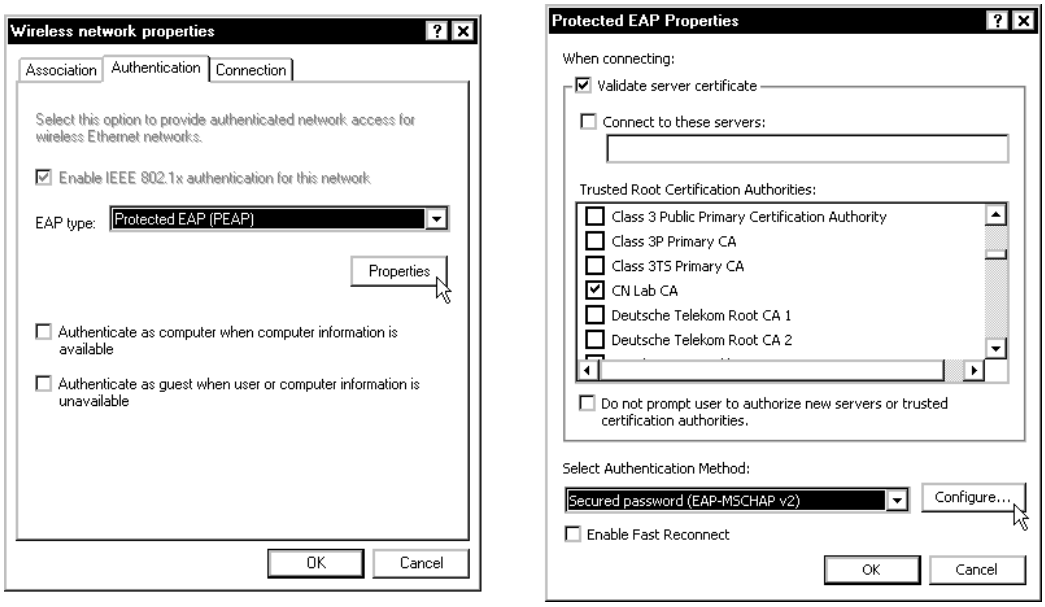
Σχήμα Γ΄.4: Επιβεβαίωση εγκατάστασης πιστοποιητικού εξυπηρετητή RADIUS

Γ′.2 Ρυθμίσεις σύνδεσης ασύρματου πελάτη στο TSN

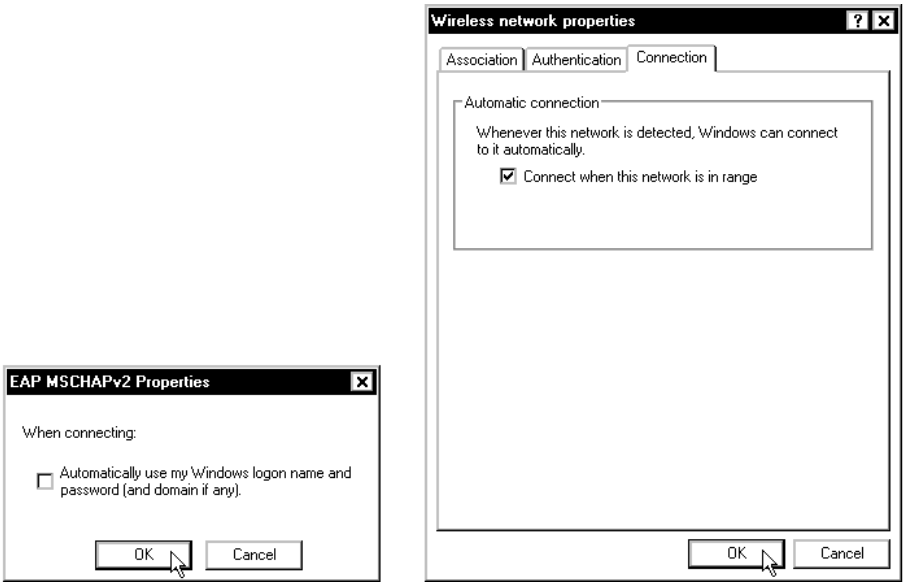
Η ενότητα αυτή, παρουσιάζει τις απαραίτητες ρυθμίσεις που πρέπει να κάνει ένας χρήστης, προκειμένου να συνδεθεί στο Ασύρματο Τοπικό Δίκτυο TSN του Εργαστηρίου Δικτύων Υπολογιστών. Η διαδικασία που ακολουθεί, αφορά *αποκλειστικά* χρήστες λειτουργικού συστήματος Microsoft Windows έκδοσης XP, με εγκατεστημένο το Service Pack 2 – SP2.



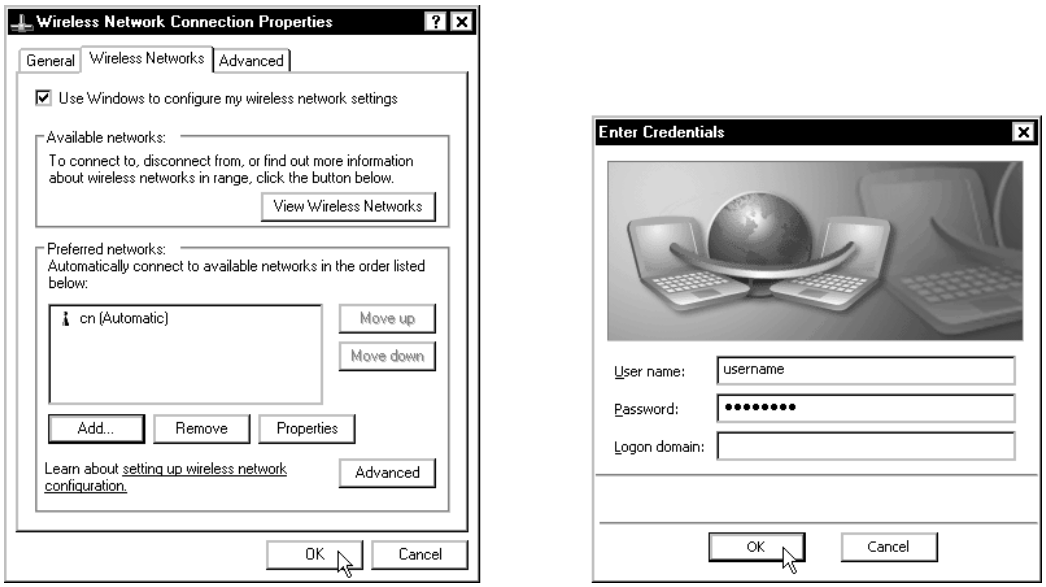
Σχήμα Γ′.5: Ρυθμίσεις σύνδεσης ασύρματου πελάτη στο TSN (βήματα 1-2)



Σχήμα Γ′.6: Ρυθμίσεις σύνδεσης ασύρματου πελάτη στο TSN (βήματα 3-4)



Σχήμα Γ΄.7: Ρυθμίσεις σύνδεσης ασύρματου πελάτη στο TSN (βήματα 5-6)



Σχήμα Γ΄.8: Ρυθμίσεις σύνδεσης ασύρματου πελάτη στο TSN (βήματα 7-8)

Όσον αφορά τα λειτουργικά συστήματα Unix, υπάρχει η ανοιχτή υλοποίηση Open1x, που επιτρέπει τη σύνδεση σε ένα Ασύρματο Τοπικό Δίκτυο IEEE 802.1X/PEAP [21].

Βιβλιογραφία

[1] A. S. Tanenbaum, *Computer Networks, 4th Edition*, Prentice Hall, 2002.

[2] B. Schneier, *Applied Cryptography, 2nd Edition*, 1996.

[3] N. Borisov, I. Goldberg and D. Wagner, "Intercepting Mobile Communications: The Insecurity of 802.11", *7th Annual International Conference on Mobile Computing and Networking*, pp. 180-188, 2001.

[4] S. Fluhrer, I. Mantin and A. Shamir, "Weaknesses in the Key Scheduling Algorithm of RC4", *8th Annual Workshop on Selected Areas in Cryptography*, 2001.

[5] J. Daemen and V. Rijmen, "The Block Cipher Rijdael", *International Conference on Smart Card Research and Applications*, pp. 288-296, 2000.

[6] J. Daemen and V. Rijmen, "Rijdael, the Advanced Encryption Standard", *Dr. Dobb's Journal*, vol. 26, no. 3, pp. 137-139, March 2001.

[7] A. J. Menezes, P. C. Van Oorschot and S. A. Vanstone, *Handbook of Applied Cryptography*, 1996.

[8] M. Bellare, J. Kilian and P. Rogaway, "The Security of the Cipher Block Chaining Message Authentication Code", *Journal of Computer and System Sciences*, vol. 61, no. 3, pp. 362-399, 2000.

[9] J. Jonsson, "On the Security of CTR + CBC-MAC", *9th Annual Workshop on Selected Areas of Cryptography*, 2002.

[10] L. Blunk and J. Vollbrecht, "PPP Extensible Authentication Protocol (EAP)", *RFC2284*, IETF, March 1998.

[11] B. Aboba and D. Simon, "PPP EAP TLS Authentication Protocol", *RFC2716*, IETF, October 1999.

[12] G. Zorn, "Microsoft Vendor-specific RADIUS Attributes", *RFC2548*, IETF, March 1999.

[13] C. Rigney, W. Willats and P. Calhoun, "RADIUS Extensions", *RFC2869*, IETF, June 2000.

[14] B. Aboba and P. Calhoun, "RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP)", *RFC3579*, IETF, September 2003.

[15] IEEE Std, "Virtual Bridged Local Area networks", *IEEE Std 802.1Q*, 1998.

[16] FreeRADIUS Server Project, "FreeRADIUS – building the perfect RADIUS server", June 2005, http://www.freeradius.org.

[17] OpenLDAP Project, "OpenLDAP", June 2005, http://www.openldap.org.

[18] The Samba Team, "Samba – Opening Windows to a Wider World", June 2005, http://www.samba.org.

[19] OpenSSL Project, "HOWTO certificates", May 2005, http://www.openssl.org/docs/HOWTO/certificates.txt.

[20] The Shmoo Group, "AirSnort Homepage", October 2005, http://airsnort.shmoo.com.

[21] The Open1x Team, "Open1x – Open Source Implementation of IEEE 802.1x", October 2005, http://www.open1x.org.