**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ - ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ**

ΤΜΗΜΑ ΜΗΧ. Η/Υ, ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ & ΔΙΚΤΥΩΝ (ΤΜΗΥΤΔ)

# ΑΝΑΠΤΥΞΗ ΚΑΙ ΣΧΕΔΙΑΣΗ ΑΛΓΟΡΙΘΜΩΝ ΚΡΥΠΤΟΓΡΑΦΗΣΗΣ ΣΕ ΑΣΥΡΜΑΤΑ ΔΙΚΤΥΑ ΑΙΣΘΗΤΗΡΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Του

**ΝΟΥΤΣΗ Α. ΓΕΩΡΓΙΟΥ**

Εκπονήθηκε υπό την επίβλεψη Των καθηγητών

**Δρ. Παναγιώτη Κίκιρα**
**Δρ. Γεώργιο Σταμούλη**

ΒΟΛΟΣ, ΜΑΙΟΣ 2008

Λευκή σελίδα

## 1. ΕΙΣΑΓΩΓΗ



**ΜΕΡΟΣ Ι – ΘΕΩΡΗΤΙΚΑ ΘΕΜΑΤΑ**

## 2. ΑΣΦΑΛΕΙΑ ΣΤΑ ΑΣΥΡΜΑΤΑ ΔΙΚΤΥΑ ΑΙΣΘΗΤΗΡΩΝ



## 3. TINY ENCRYPTION ALGORITHM

**ΜΕΡΟΣ ΙΙ – ΕΦΑΡΜΟΓΗ**

**4. ΥΛΟΠΟΙΗΣΗ**



**5. ΕΚΤΙΜΗΣΗ ΑΠΟΔΟΣΗΣ ΠΡΩΤΟΚΟΛΛΟΥ**



**6. ΕΠΙΛΟΓΟΣ**



**ΠΑΡΑΡΤΗΜΑ**

Λευκή σελίδα

# ΕΥΧΑΡΙΣΤΙΕΣ

Για την εκπόνηση αυτής της εργασίας θα ήθελα να ευχαριστήσω θερμά τους επιβλέποντες καθηγητές κ. Παναγιώτη Κίκιρα και κ. Γεώργιο Σταμούλη για τη βοήθεια, τις συμβουλές και την υπομονή που επέδειξαν σε όλο αυτό το διάστημα. Η συμβολή τους στην εκπόνηση και τελική μορφή της παρούσας μεταπτυχιακής εργασίας ήταν καθοριστική κάνοντας την όλη διαδικασία πολύ πιο ενδιαφέρουσα.

Τέλος, επειδή με την εργασία αυτή, ολοκληρώνονται και οι σπουδές μου ως φοιτητή θα ήθελα να ευχαριστήσω την οικογένειά μου για την συμπαράσταση και την βοήθεια της.

Γεώργιος Α. Νούτσης

# 1. ΕΙΣΑΓΩΓΗ

## 1.1 ΑΝΤΙΚΕΙΜΕΝΟ ΤΗΣ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Σκοπός της παρούσας εργασίας είναι η ανάλυση της βιβλιογραφίας που καλύπτει τα θέματα της ασφάλειας των δικτύων αισθητήρων. Παράλληλα με τα θεωρητικά θέματα η εργασία πραγματεύεται την υλοποίηση του πρωτοκόλλου κρυπτογράφησης Tiny Encryption Algorithm (TEA) για ασύρματα δίκτυα αισθητήρων, με χρήση του εργαλείου TinyOS, σε κώδικα NesC. Για την υλοποίηση του πρωτοκόλλου δημιουργήθηκε δίκτυο αποτελούμενο από κόμβους οι οποίοι κρυπτογραφούν τα δεδομένα τους με τον αλγόριθμο κρυπτογράφησης Tiny Encryption και έναν σταθμό βάσης, ο οποίος αποκρυπτογραφεί τα δεδομένα που λαμβάνει από τους κόμβους του δικτύου.

## 1.2 ΔΙΑΡΘΡΩΣΗ ΤΗΣ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Η εργασία, χωρίζεται σε δύο ενότητες. Στην πρώτη, καλύπτεται μέρος της θεωρίας για την ασφάλεια στα ασύρματα δίκτυα αισθητήρων και για το πρωτόκολλο που υλοποιήθηκε, ενώ στη δεύτερη καταγράφονται λεπτομέρειες για την εφαρμογή και τον τρόπο λειτουργίας της.

Ειδικότερα, στο πρώτο μέρος, γίνεται μια περιγραφή των ιδιοτήτων των δικτύων αισθητήρων που επηρεάζουν την ασφάλεια. Επίσης αναλύονται οι αρχές ασφάλειας για τα ασύρματα δίκτυα αισθητήρων και περιγράφονται διεξοδικά τα είδη των επιθέσεων και τα σημαντικότερα πρωτόκολλα ασφάλειας των δικτύων αυτού του τύπου (**Κεφάλαιο 2**).

Στο **κεφάλαιο 3** περιγράφεται ο αλγόριθμος κρυπτογράφησης Tiny Encryption Algorithm που υλοποιείται στο δεύτερο μέρος. Παρουσιάζονται τα βασικότερα σημεία του αλγορίθμου και παραθέτονται οι ρουτίνες κρυπτογράφησης και αποκρυπτογράφησης.

Το δεύτερο μέρος της εργασίας αφορά αποκλειστικά την υλοποίηση της εφαρμογής. Ειδικότερα, στο **Κεφάλαιο 4**, γίνεται μια σύντομη περιγραφή των εργαλείων που χρησιμοποιήθηκαν για τον προγραμματισμό των αισθητήρων και τα βασικά χαρακτηριστικά της πλατφόρμας Mica2 για την οποία έγινε η προσομοίωση. Επίσης, παρουσιάζονται τα βασικότερα κομμάτια της εφαρμογής.

Στο **κεφάλαιο 5** γίνεται η σύγκριση της απόδοσης τόσο σε ταχύτητα αλλά και σε κατανάλωση αποθηκευτικού χώρου κάποιων αλγορίθμων κρυπτογράφησης με τον αλγόριθμο TEA. Τέλος, στο **κεφάλαιο 6** εξάγονται κάποια βασικά συμπεράσματα όσον αφορά τον αλγόριθμο και θέτουμε το πλαίσιο της μελλοντικής έρευνας.

# ΜΕΡΟΣ Ι - ΘΕΩΡΗΤΙΚΑ ΘΕΜΑΤΑ

# 2. ΑΣΦΑΛΕΙΑ ΣΤΑ ΑΣΥΡΜΑΤΑ ΔΙΚΤΥΑ ΑΙΣΘΗΤΗΡΩΝ

## 2.1 ΕΙΣΑΓΩΓΗ – ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ

Τα δίκτυα αισθητήρων σχετίζονται με ένα ετερογενές σύστημα που συνδυάζει μικροσκοπικούς αισθητήρες και επεξεργαστές για τον υπολογισμό δεδομένων. Αυτά τα δίκτυα αποτελούνται από εκατοντάδες ή και χιλιάδες, χαμηλής ισχύος και χαμηλού κόστους ασύρματους κόμβους που αναπτύσσονται μαζικά για τον έλεγχο και τον επιρροή του περιβάλλοντος.

Όσο τα ασύρματα δίκτυα αισθητήρων συνεχίζουν να αυξάνονται, οι αποτελεσματικοί μηχανισμοί ασφάλειας γίνονται αναγκαίοι. Επειδή στα δίκτυα αισθητήρων μπορεί να διακινούνται ευαίσθητα δεδομένα ή/και να λειτουργούν σε εχθρικά και αφύλακτα περιβάλλοντα, είναι επιτακτικό τα θέματα ασφάλειας πρέπει να εξεταστούν από το αρχικό στάδιο σχεδίασης του δικτύου. Εντούτοις, λόγω της φύσης των ασύρματων κόμβων και των περιορισμών σε πόρους και υπολογιστική ισχύ, η ασφάλεια στα δίκτυα αισθητήρων θέτει διαφορετικές προκλήσεις σε σχέση με την ασφάλεια δικτύων και υπολογιστικών συστημάτων που είναι μια καλά εδραιωμένη επιστημονική περιοχή με πρωτόκολλα και πρότυπα τα οποία τυγχάνουν ευρείας αναγνώρισης.

### 2.1.1 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΔΙΚΤΥΩΝ ΑΙΣΘΗΤΗΡΩΝ

Τα δίκτυα αισθητήρων έχουν συχνά ένα ή περισσότερα σημεία ελέγχου, τους γνωστούς σταθμούς βάσης. Ένας σταθμός βάσης είναι ουσιαστικά μια πύλη σε ένα άλλο δίκτυο, ένα ισχυρό κέντρο επεξεργασίας ή αποθήκευσης, ή ένα σημείο πρόσβασης για μια ανθρώπινη διεπαφή. Μπορούν να χρησιμοποιηθούν ως ένας σύνδεσμος για να διαδώσει τις πληροφορίες ελέγχου και στοιχεία του δικτύου σε έναν ή περισσότερους κόμβους.

Οι κόμβοι αισθητήρων καθιερώνουν ένα δάσος δρομολόγησης, με έναν σταθμό βάσης στη ρίζα κάθε δέντρου. Οι σταθμοί βάσεων είναι πολύ ισχυρότεροι από τους κόμβους αισθητήρων. Χαρακτηριστικά, οι σταθμοί βάσεων έχουν αρκετή ενεργειακή ισχύ για να ξεπεράσουν τη διάρκεια ζωής όλων των κόμβων αισθητήρων, ικανοποιητική μνήμη για να

υποθηκεύσουν τα κρυπτογραφικά κλειδιά, ισχυρότερους επεξεργαστές καθώς και μέσα για επικοινωνία με τα εξωτερικά δίκτυα.

Γενικά, οι κόμβοι αισθητήρων επικοινωνούν χρησιμοποιώντας RF, έτσι η ραδιοφωνική μετάδοση είναι η θεμελιώδης επικοινωνία. Τα βασικά πρωτόκολλα πρέπει να αξιοποιούνται έτσι ώστε από την μια μεριά να μην βλάπτεται η εμπιστευτικότητα και από την άλλη να ελαχιστοποιείται η ενεργειακή σπατάλη.

Στις εφαρμογές που αναπτύσσονται μέχρι τώρα, τα σχέδια επικοινωνίας μέσα στο δίκτυο εμπίπτουν στις ακόλουθες κατηγορίες:

- *Κόμβος με σταθμό βάσης (node to base station),* π.χ. ερεθίσματα αισθητήρων, συναγερμοί
- *Σταθμό βάσης με κόμβο (base station to node),* π.χ. ιδιαίτερες αιτήσεις, updates κλειδιών
- *Σταθμός βάσεων με όλους τους κόμβους (base station to all nodes),* π.χ. αναγνωριστικά σήματα δρομολόγησης, ερωτηματολόγια ή επανα-προγραμματισμός ολόκληρου του δικτύου. *Επικοινωνία μεταξύ μιας καθορισμένης ομάδας κόμβων (communication amongst a defined cluster of nodes)* (π.χ, ένας κόμβος και όλοι οι γείτονές του). Ο διαχωρισμός των κόμβων σε ομάδες μπορεί να μειώσει το συνολικό αριθμό μηνυμάτων που στέλνονται και έτσι να διασφαλίσει ενέργεια με τη χρησιμοποίηση ενδο-δικτυακών τεχνικών επεξεργασίας στοιχείων, όπως η συσσώρευση δεδομένων (data aggregation) και η *παθητική συμμετοχή* (passive participation). Ένα σημείο συνάθροισης μπορεί να συλλέξει δεδομένα από γειτονικούς κόμβους και να διαβιβάσει ένα ενιαίο μήνυμα που περιέχει μια καθολική τιμή. Σύμφωνα με την παθητική συμμετοχή, στην περίπτωση που ένας κόμβος γνωρίζει ότι ένας γειτονικό κόμβος προώθησε την ίδια με τη δική του τρέχουσα ανάγνωση μπορεί να επιλέξει να μην διαβιβάσει το ίδιο μήνυμα.

## 2.2 ΠΕΡΙΟΡΙΣΜΟΙ ΣΤΗΝ ΑΣΦΑΛΕΙΑ ΤΩΝ ΔΙΚΤΥΩΝ ΑΙΣΘΗΤΗΡΩΝ

Ένα ασύρματο δίκτυο αισθητήρων είναι ένα ειδικό δίκτυο που έχει πολλούς περιορισμούς σε σχέση με ένα παραδοσιακό δίκτυο υπολογιστών. Λόγω αυτών των περιορισμών είναι δύσκολο να υιοθετηθούν άμεσα οι υπάρχουσες προσεγγίσεις ασφάλειας σε τέτοιου είδους δίκτυα.. Επομένως, για να αναπτυχθούν χρήσιμοι και αποτελεσματικοί μηχανισμοί ασφάλειας είναι απαραίτητη η κατανόηση τους.

### 2.2.1 ΠΕΡΙΟΡΙΣΜΕΝΟΙ ΠΟΡΟΙ

Όλες οι προσεγγίσεις ασφάλειας απαιτούν κάποιους πόρους για την υλοποίηση των εφαρμογών όπως μνήμη, χώρο για τον κώδικα και ηλεκτρική ισχύ.

- ***Περιορισμένη μνήμη και χώρος αποθήκευσης***. Ο αισθητήρας είναι μια μικροσκοπική συσκευή με μόνο ένα μικρό μέγεθος διαθέσιμης μνήμης και αποθηκευτικού χώρου για τον κώδικα. Έτσι είναι αναγκαίο, για να αναπτυχθεί ένας αποτελεσματικός μηχανισμός ασφάλειας, να περιοριστεί το μέγεθος του κώδικα του αλγορίθμου ασφάλειας. Για παράδειγμα, ένας κοινός τύπος αισθητήρων (TelosB) έχει έναν δεκαεξάμπιτο, 8 MHz RISC KME με μόνο 10K RAM, μνήμη προγράμματος 48K, και 1024K μνήμη flash [1]. Με τέτοιου είδους περιορισμούς πρέπει και το λογισμικό του αισθητήρα πρέπει να είναι εξίσου μικρό. Το συνολικό τμήμα του κώδικα του TinyOS είναι περίπου 4K [2] και ο χρονοπρογραμματιστής καταλαμβάνει μόνο 178 ψηφιολέξεις. Επομένως, το συνολικό μέγεθος κώδικα για την εφαρμογή ασφάλειας πρέπει να είναι επίσης μικρό.
- ***Περιορισμένη Ενέργεια***. *Η ενέργεια είναι το μεγαλύτερο εμπόδιο για τον σχεδιασμό αποτελεσματικών εφαρμογών ασφάλειας στα δίκτυα αισθητήρων.* Υποθέτουμε ότι μόλις επεκταθούν οι κόμβοι σε ένα δίκτυο αισθητήρων, δεν μπορούν να αντικατασταθούν εύκολα (υψηλή λειτουργική δαπάνη) ή να επαναφορτιστούν (υψηλό κόστος). Επομένως, η ηλεκτρική ισχύς των

μπαταριών πρέπει να διατηρηθεί όσο το δυνατόν περισσότερο και σε υψηλά επίπεδα έτσι ώστε να επεκταθεί η διάρκεια ζωής τόσο μεμονωμένα των κόμβων όσο και ολόκληρου του δικτύου. Έτσι, κατά τον σχεδιασμό και προγραμματισμό ενός αλγορίθμου ασφάλειας πρέπει να ληφθεί σοβαρά υπόψη και το αντίτυπο σε σπατάλη ενέργειας.

Κατά προσθήκη ενός αλγορίθμου ασφάλειας σε έναν αισθητήρα ενδιαφερόμαστε για την επίδραση που θα έχει στη διάρκεια ζωής του (δηλ., η διάρκεια μπαταριών του). Οι επιπλέον απαιτήσεις σε ενέργεια από τους κόμβους αισθητήρων λόγω του αλγορίθμου ασφάλειας συσχετίζονται με την επεξεργασία που απαιτείται για την κρυπτογράφηση και αποκρυπτογράφηση, για την υπογραφή των δεδομένων – μηνυμάτων, την παραπάνω ενέργεια που απαιτείται για να διαβιβαστούν τα μεγαλύτερα μηνύματα (διανύσματα έναρξης που απαιτούνται για την κρυπτογράφηση/αποκρυπτογράφηση), και την ενέργεια που απαιτείται για την αποθήκευση των όποιων κρυπτογραφικών κλειδιών (δημόσιων ή ιδιωτικών).

### 2.2.2 ΑΝΑΞΙΟΠΙΣΤΗ ΕΠΙΚΟΙΝΩΝΙΑ

Η αναξιόπιστη επικοινωνία είναι μια ακόμα απειλή για τα δίκτυα αισθητήρων. Η ασφάλεια του δικτύου στηρίζεται σε ένα μεγάλο ποσοστό στο πρωτόκολλο το οποίο με την σειρά του εξαρτάται από την επικοινωνία.

- ***Αναξιόπιστη αποστολή.*** Στις περισσότερες περιπτώσεις το πρωτόκολλο δρομολόγησης του δικτύου αισθητήρων δεν προκαθορίζει τις συνέσεις ανάμεσα στους κόμβους (connectionless) με αποτέλεσμα να είναι αναξιόπιστο. Τα πακέτα μπορεί να τροποποιηθούν λόγω των διάφορων λαθών του καναλιού ή και να ληφθούν από κορεσμένους κόμβους, Το αποτέλεσμα είναι χαμένα ή αγνοούντα πακέτα. Επιπλέον, το, εκ φύσεως, αναξιόπιστο ασύρματο κανάλι οδηγεί επίσης τέτοιου είδους φαινόμενα.. Το υψηλό ποσοστό λαθών και απωλειών πακέτων αναγκάζει τον υπεύθυνο για τον χειρισμό λαθών να

αφιερώσει πόρους στο χειρισμό λαθών. Είναι σημαντικό να δοθεί ιδιαίτερη προσοχή στο πεδίο του χειρισμού των λαθών γιατί είναι επικίνδυνο να χαθούν κρίσιμα για την ασφάλεια στοιχεία (π.χ κρυπτογραφικά κλειδιά).

- ***Συγκρούσεις.*** Ακόμα κι αν το κανάλι ήταν αξιόπιστο, η επικοινωνία μπορεί να συνεχίζει να είναι αναξιόπιστη. Αυτό οφείλεται στη φύση της broadcast μετάδοσης των ασύρματων δικτύων αισθητήρων. Εάν τα πακέτα συναντηθούν κατά την διάρκεια της μεταφοράς θα δημιουργηθούν συγκρούσεις και η μεταφορά θα αποτύχει. Σε έναν αισθητήρα με μεγάλη κίνηση πακέτων (μπορεί να είναι κόμβος με μεγάλη σημασία στο δέντρο δρομολόγησης ή/και ρίζα δέντρου) αυτό μπορεί να έχει σαν αποτέλεσμα πολλά προβλήματα στην λειτουργία του δικτύου.
- ***Καθυστέρηση (latency).*** Η multi-hop δρομολόγηση, η συμφόρηση του δικτύου και η επεξεργασία δεδομένων στους κόμβους μπορεί να οδηγήσει το δίκτυο σε μεγάλα επίπεδα καθυστέρησης με αποτέλεσμα ο συγχρονισμός ανάμεσα στους κόμβους να είναι δύσκολο έως αδύνατον να επιτευχθεί. Τα ζητήματα συγχρονισμού του δικτύου μπορεί να είναι κρίσιμα για την ασφάλεια των αισθητήρων, αφού οι μηχανισμοί της ασφάλειας στηρίζονται σε κρίσιμες αναφορές γεγονότων (report events) και στην κατανεμημένη διανομή κλειδιών κρυπτογράφησης.

### 2.2.3 ΛΕΙΤΟΥΡΓΙΑ ΧΩΡΙΣ ΠΑΡΑΚΟΛΟΥΘΗΣΗ

Ανάλογα με τον σκοπό και το λόγω δημιουργίας του εκάστοτε δικτύου, οι κόμβοι αισθητήρων μπορεί να λειτουργούν χωρίς κάποιον διαχειριστή του δικτύου να παρακολουθεί για μεγάλα χρονικά διαστήματα.

- ***Έκθεση σε επιθέσεις στο φυσικό επίπεδο.*** Οι αισθητήρες μπορούν να τοποθετηθούν σε περιβάλλοντα προσβάσιμα στους υποκλοπείς, με άσχημα καιρικά φαινόμενα κλπ. Η πιθανότητα για επιθέσεις στο φυσικό επίπεδο είναι

πολύ μεγαλύτερη στα δίκτυα αισθητήρων σε σχέση με τα τυπικά δίκτυα υπολογιστών που βρίσκονται σε ελεγχόμενους χώρους,

- ***Απομακρυσμένη διαχείριση.*** Η απομακρυσμένη διαχείριση των δικτύων αισθητήρων καθιστά αδύνατη την ανίχνευση οποιονδήποτε παραβιάσεων σε φυσικό επίπεδο καθώς και των ζητημάτων συντήρησης (π.χ αλλαγή μπαταριών). Ίσως το πιο ακραίο παράδειγμα είναι ένας κόμβος να χρησιμοποιηθεί σε μακρινές αποστολές αναγνώρισης πίσω από εχθρικές γραμμές. Σε μια τέτοια περίπτωση ο κόμβος μπορεί να μην έχει οποιαδήποτε επαφή στο φυσικό επίπεδο από την στιγμή της τοποθέτησής του.
- ***Μη κεντρικοποιημένη διαχείριση.*** Ένα δίκτυο αισθητήρων πρέπει να είναι ένα κατανεμημένο δίκτυο χωρίς κεντρικό κόμβο διαχείρισης. Αυτό θα έχει σαν αποτέλεσμα την αύξηση της διάρκειας ζωής του δικτύου. Εντούτοις, αν ο σχεδιασμός του δικτύου είναι ανακριβής θα έχει σαν αποτέλεσμα η οργάνωση του να είναι δύσκολη, ανεπαρκής και εύθραυστη.

## 2.3 ΠΡΟΫΠΟΘΕΣΕΙΣ ΑΣΦΑΛΕΙΑΣ

Τα δίκτυα αισθητήρων είναι μια ειδική περίπτωση δικτύων. Έχει ορισμένες ομοιότητες με ένα τυπικό δίκτυο υπολογιστών, αλλά επιπροσθέτως έχουν κάποια μοναδικά χαρακτηριστικά όπως είδαμε στην υποενότητα 2.2. Επομένως μπορούμε να αναλογιστούμε τις απαιτήσεις ασφάλειας για αυτά τα δίκτυα σαν συνάθροιση των απαιτήσεων των παραδοσιακών δικτύων και των μοναδικών χαρακτηριστικών των δικτύων αισθητήρων.

### 2.3.1 ΕΜΠΙΣΤΕΥΤΙΚΟΤΗΤΑ ΔΕΔΟΜΕΝΩΝ

Η εμπιστευτικότητα των δεδομένων είναι το σημαντικότερο ζήτημα στην ασφάλεια των δικτύων. Κάθε δίκτυο με οποιαδήποτε επίπεδο ασφάλειας θα πρέπει να έχει εξετάσει το συγκεκριμένο θέμα πρωταρχικά. Στα δίκτυα αισθητήρων, η εμπιστευτικότητα έχει να κάνει με τα παρακάτω [3,4]:

- Ένας αισθητήρας δεν θα πρέπει να γνωστοποιεί ανεπίσημα δεδομένα στους γειτονικούς κόμβους του. Ειδικά σε μια στρατιωτική εφαρμογή, η αποθηκευμένη πληροφορία στον κόμβο αισθητήρα μπορεί να είναι ιδιαίτερα ευαίσθητη.
- Σε πολλές εφαρμογές οι αισθητήρες ανταλλάσουν ιδιαίτερα κρίσιμα δεδομένα (π.χ κλειδιά κρυπτογράφησης). Έτσι, είναι πολύ σημαντικό να αναπτυχθεί ένα ασφαλές κανάλι σε ένα ασύρματο δίκτυο αισθητήρων.
- Δημόσιες πληροφορίες των αισθητήρων, όπως η ταυτότητα του αισθητήρα και τα δημόσια κλειδιά, θα πρέπει να κρυπτογραφούνται έως ένα βαθμό έτσι ώστε να θωρακιστούν σε επιθέσεις ανάλυσης κίνησης (traffic analysis attacks).

Η τυπική προσέγγιση για την διατήρηση ευαίσθητων πληροφοριών μυστικών είναι να κρυπτογραφηθούν τα δεδομένα με ένα μυστικό κλειδί που μόνο οι συμβαλλόμενοι κόμβοι γνωρίζουν. Αυτό έχει σαν συνέπεια την εμπιστευτικότητα των δεδομένων.

### 2.3.2 ΑΚΕΡΑΙΟΤΗΤΑ ΔΕΔΟΜΕΝΩΝ

Με την υλοποίηση της εμπιστευτικότητας, ένας επιτιθέμενος είναι πιθανόν να είναι αδύνατον να υποκλέψει πληροφορίες. Παρόλα αυτά αυτό δεν σημαίνει ταυτόχρονα ότι τα δεδομένα είναι ασφαλή. Ο κακόβουλος χρήστης μπορεί να τροποποιήσει τα δεδομένα έτσι ώστε να αποσταλούν στο δίκτυο μεταλλαγμένα με όποιες συνέπειες μπορεί αυτό να προκαλέσει. Για παράδειγμα, ένας κόμβος που ελέγχεται από ένα επιτιθέμενο μπορεί να προσθέσει κάποια κομμάτια να τροποποιήσει μέρη του μηνύματος. Αυτό το μεταλλαγμένο πακέτο μπορεί να αποσταλεί στον δέκτη που προοριζόταν. Απώλειες μηνυμάτων ή και μετατροπές μπορούν να συμβούν και χωρίς την παρουσία του επιτιθεμένου λόγω του αντίξοου περιβάλλοντος επικοινωνίας. Έτσι, η ακεραιότητα των δεδομένων εξασφαλίζει ότι οποιαδήποτε πληροφορία ληφθεί από έναν κόμβο δεν έχει τροποποιηθεί κατά την διαδικασία αποστολής.

### 2.3.3 ΦΡΕΣΚΑΔΑ ΔΕΔΟΜΕΝΩΝ

Ακόμα και αν η εμπιστευτικότητα και η ακεραιότητα των δεδομένων είναι δεδομένα συστατικά του δικτύου, πρέπει να εξασφαλιστεί και η φρεσκάδα κάθε μηνύματος. Η φρεσκάδα των δεδομένων (data freshness) εξασφαλίζει ότι τα δεδομένα είναι πρόσφατα και ότι δεν έχουν επαναληφθεί παλιά μηνύματα (replay attacks). Αυτό η προϋπόθεση είναι ιδιαίτερη σημαντική όταν έχουν υλοποιηθεί στρατηγικές κοινού κλειδιού. Τα κοινά κλειδιά πρέπει να αλλάζουν με την πάροδο του χρόνου, Εντούτοις, η διάδοση των κλειδιών σε ολόκληρο το δίκτυο είναι μι ιδιαίτερα χρονοβόρα διαδικασία. Σε αυτή την περίπτωση, είναι εύκολο ένας επιτιθέμενος να χρησιμοποιήσει μια επίθεση επανάληψης και να διακόψει την κανονική λειτουργία ενός κόμβου που δεν γνωρίζει το νέο κλειδί. Για να επιλυθεί αυτό το πρόβλημα μπορεί να εισαχθεί στο πακέτο ένας μοναδικός μετρητής χρόνου (nonce) έτσι ώστε να διασφαλιστεί η φρεσκάδα των μηνυμάτων.

### 2.3.4 ΔΙΑΘΕΣΙΜΟΤΗΤΑ

Η τροποποίηση και η χρήση παραδοσιακών αλγορίθμων κρυπτογράφησης, έτσι ώστε να ταιριάζουν στα δίκτυα αισθητήρων, θα εισάγει πρόσθετες δαπάνες, Μερικές προσεγγίσεις προσπαθούν να τροποποιήσουν τον κώδικα για να τον επαναχρησιμοποιούν όσο το δυνατόν περισσότερο. Άλλες προσπαθούν να χρησιμοποιήσουν πρόσθετη επικοινωνία για να επιτύχουν το παραπάνω και άλλες προωθούν περιορισμούς για την πρόσβαση των δεδομένων ή προτείνουν ένα ακατάλληλο σχέδιο (όπως ένα κεντρικοποιημένο δίκτυο) έτσι ώστε να απλοποιηθεί ο αλγόριθμος. Παρόλα αυτά όλες οι παραπάνω προσεγγίσεις μικραίνουν τον χρόνο που ο ένας κόμβος μπορεί να είναι διαθέσιμος για τους παρακάτω λόγους:

- Επιπρόσθετοι υπολογισμοί έχουν σαν αποτέλεσμα την κατανάλωση επιπλέον ενέργειας. Αν εξαντληθούν οι μπαταρίες τα δεδομένα δεν θα μπορούν να προσπελαστούν.
- Επιπλέον επικοινωνία έχουν επίσης σαν αποτέλεσμα την κατανάλωση επιπλέον ενέργειας. Επίσης όσο αυξάνονται τα μηνύματα προς αποστολή τόσο αυξάνεται και η πιθανότητα για συγκρούσεις.
- Με την χρησιμοποίηση του κεντρικοποιημένου σχήματος εισάγεται στο δίκτυο ένα μοναδικό σημείο βλάβης (single point of failure) που αν καταρρεύσει σταματά η λειτουργία του συνολικού δικτύου.

Η προϋπόθεση για ασφάλεια δεν έχει να κάνει άμεσα μόνο με την ασφαλής λειτουργία του δικτύου αλλά είναι εξίσου σημαντική για την αύξηση της διαθεσιμότητας του.

### 2.3.5 ΑΥΤΟ-ΟΡΓΑΝΩΣΗ

Ένα ασύρματο δίκτυο αισθητήρων είναι χαρακτηριστικό ad-hoc δίκτυο, το οποίο προϋποθέτει κάθε κόμβος να είναι ανεξάρτητος και ευέλικτο έτσι ώστε να μπορεί να αυτό-οργανώνεται και "αυτό-θεραπεύεται" σύμφωνα με διαφορετικές καταστάσεις. Δεν υπάρχει καμιά σταθερή υποδομή όσο αφορά την διαχείριση δικτύων ασύρματων αισθητήρων. Αυτό το εκ φύσεως χαρακτηριστικό γνώρισμα προσθέτει ακόμα μια μεγάλη πρόκληση στην ασφάλεια

τέτοιων δικτύων. Παραδείγματος χάριν, η δυναμική ολόκληρου του δικτύου εμποδίζει την ιδέα της προεγκατάστασης ενός κοινού κλειδιού μεταξύ του σταθμού βάσης και των άλλων κόμβων αισθητήρων [5]. Πολλά και διάφορα πρωτόκολλα που έχουν σαν βάση την προ-διανομή των κλειδιών έχουν προταθεί στα πλαίσια της συμμετρικής κρυπτογράφησης [6,5,7,8]. Στα πλαίσια της χρησιμοποίησης κρυπτογράφηση δημοσίου κλειδιού στα δίκτυα αισθητήρων, είναι απαραίτητος επίσης ένας μηχανισμός διανομής του δημόσιου κλειδιού. Με τον ίδιο τρόπο που τα κατανεμημένα δίκτυα αισθητήρων πρέπει να αυτό-οργανώνονται για να υποστηρίζουν multi-hop επικοινωνία, πρέπει επίσης να αυτό-οργανώνονται για να διαχειρίζονται τα κλειδιά και να εδραιώνουν σχέσεις εμπιστοσύνης με άλλους αισθητήρες. Αν δεν υπάρχει η έννοια της αυτό-οργάνωσης σε ένα δίκτυο αισθητήρων, οι συνέπειες από μια επίθεση η και ακόμα το αντίξοο περιβάλλον μπορεί να είναι καταστρεπτικό.

### 2.3.6 ΣΥΓΧΡΟΝΙΣΜΟΣ

Οι περισσότερες εφαρμογές που έχουν αναπτυχθεί για τα δίκτυα αισθητήρων στηρίζονται σε κάποια μορφή συγχρονισμού. Προκειμένου να διατηρείται η ενέργεια, η κεραία ενός αισθητήρα μπορεί να κλείνει για κάποια χρονικά διαστήματα. Επιπλέον, οι αισθητήρες μπορεί να επιθυμούν να υπολογίσουν το χρόνο που μεσολαβεί από την αποστολή έως την λήψη ενός πακέτου μεταξύ δυο αισθητήρων. Ένα πολύ συνεργάσιμο δίκτυο μπορεί να προϋποθέτει ακόμα και τον συγχρονισμό μιας ομάδας κόμβων.

### 2.3.7 ΑΣΦΑΛΗΣ ΕΝΤΟΠΙΣΜΟΣ ΤΩΝ ΚΟΜΒΩΝ

Πολύ συχνά η χρησιμότητα ενός δικτύου αισθητήρων στηρίζεται στη δυνατότητα του να εντοπίζει αυτόματα κάθε αισθητήρα στο δίκτυο. Ένα δίκτυο αισθητήρων που σχεδιάζεται έτσι ώστε να μπορεί να εντοπίζει τυχόν βλάβες κόμβων θα χρειαστεί εξακριβωμένες πληροφορίες θέσης προκειμένου να επισημάνει την θέση του κόμβου που παρουσίασε το πρόβλημα. Δυστυχώς, ένας επιτιθέμενος μπορεί εύκολα να χρησιμοποιήσει γνωστές συντεταγμένες κόμβων με την αποστολή άκυρων μηνυμάτων με την επανάληψη σημάτων, κλπ.

Υπάρχει μια γνωστή τεχνική που ονομάζεται πολυμερή επαλήθευση (verifiable multilateration VM) η οποία υπολογίζει την ακριβή θέση ενός κόμβου χρησιμοποιώντας μια

σειρά από σημεία αναφοράς (κόμβους). Σε αυτή την περίπτωση ένας επιτιθέμενος μπορεί απλά να αυξήσει κακόβουλα την απόσταση ενός κόμβου από ένα σημείο αναφοράς. Παρόλα αυτά, για να εξασφαλιστεί η συνέπεια της θέσης του κόμβου ο επιτιθέμενος θα πρέπει να αποδείξει ότι η απόσταση του από ένα άλλο σημείο αναφοράς είναι μικρότερη. Αν δεν γίνει αυτό, ένας κόμβος που χρησιμοποιεί το πρωτόκολλο εντοπισμού θέσης μπορεί να εντοπίσει την ασυνέπεια.

Σε μεγάλα δίκτυα αισθητήρων χρησιμοποιείται το SPINE (Secure Positioning For Sensor Networks) το οποίο είναι αλγόριθμος με τρεις φάσεις ο οποίος βασίζεται στην πολυμερή επαλήθευση.

### 2.3.8 ΑΥΘΕΝΤΙΚΟΠΟΙΗΣΗ

Ένας επιτιθέμενος δεν περιορίζεται μόνος στην μετατροπή των μηνυμάτων. Μπορεί να αλλάξει ολόκληρη την σειρά των πεδίων ενός πακέτου με την προσθήκη περισσότερων η την διαγραφή κάποιων. Έτσι ο αποδέκτης του μηνύματος πρέπει να σιγουρευτεί ότι οποιοδήποτε δεδομένα που χρησιμοποιούνται για την λήψη αποφάσεων προέρχονται από την αυθεντική πηγή. Αφετέρου, κατά την κατασκευή του δικτύου η αυθεντικοποίηση είναι σημαντική για πολλούς λόγους διαχείρισης (π.χ επανα-προγραμματισμός του δικτύου). Από τα παραπάνω γίνεται κατανοητό ότι η αυθεντικοποίηση είναι ιδιαίτερα σημαντική σε πολλές εφαρμογές στα δίκτυα αισθητήρων. Με την αυθεντικοποίηση των δεδομένων και των μηνυμάτων ένας κόμβος μπορεί να ελέγξει αν τα δεδομένα προέρχονται από τον αποστολέα που ισχυρίζεται ότι τα έχει στείλει. Στην περίπτωση της επικοινωνίας μεταξύ δυο πλευρών (two-party communication) η αυθεντικοποίηση επιτυγχάνεται με έναν απλό συμμετρικό αλγόριθμο: Ο πομπός και ο δέκτης μοιράζονται ένα κοινό κλειδί και παράγουν το κώδικα αυθεντικοποίησης μηνύματος (message authentication code – MAC).

Στον πρωτόκολλο ασφάλειας μ-Tesla, προτείνεται ένα αλυσιδωτό σύστημα ανταλλαγής κλειδιών. Η βασική ιδέα είναι να επιτευχθεί ασύμμετρη κρυπτογράφηση με την καθυστερημένη αποκάλυψη των συμμετρικών κλειδιών. Σε αυτή την περίπτωση ένας αποστολέας προωθεί σε όλους τους κόμβους (broadcast) ένα μήνυμα που έχει δημιουργηθεί με ένα κρυφό κλειδί. Μετά από ένα καθορισμένο χρονικό διάστημα ο αποστολέας θα κοινοποιήσει το μυστικό κλειδί. Ο δέκτης του μηνύματος είναι υπεύθυνος στο να κρατήσει το

πακέτο έως ότου λάβει και το κλειδί. Μετά την κοινοποίηση ο δέκτης μπορεί να αυθεντικοποιήσει το πακέτο, υπό τον όρο ότι το πακέτο έχει ληφθεί πριν την κοινοποίηση του κλειδιού. Ένας περιορισμός του μTESLA είναι ότι πρέπει να σταλεί σε όλους τους κόμβους του δικτύου (με unicast) κάποια αρχική πληροφορία έτσι ώστε να αρχίσει η αυθεντικοποίηση και η αποστολή μηνυμάτων (broadcast). Για τον λόγο αυτό έχει προταθεί μια τροποποίηση του μTESLA που χρησιμοποιεί broadcast των κλειδιών και όχι unicast.

## 2.4 Επιθεσεις

Τα δίκτυα αισθητήρων είναι ιδιαίτερα ευπαθή σε διάφορα είδη επιθέσεων. Οι διάφορες επιθέσεις μπορούν να εκτελεστούν με ποικίλους τρόπους, τις περισσότερες φορές σαν επιθέσεις άρνησης εξυπηρέτησης (denial of servise) αλλά και μέσω ανάλυσης κίνησης (traffic analysis), παραβίαση της μυστικότητας, σαν επιθέσεις στο φυσικό επίπεδο κλπ. Οι επιθέσεις άρνησης εξυπηρέτησης στα δίκτυα αισθητήρων μπορεί να κυμαίνονται από απλές επιθέσεις μπλοκαρίσματος του καναλιού επικοινωνίας, έως πιο πολύπλοκες επιθέσεις, σχεδιασμένες έτσι ώστε να παραβιάσουν το 802.11 MAC πρωτόκολλο ή και οποιοδήποτε επίπεδο του ασύρματου δικτύου.

Η προστασία ενάντια σε μια καλά σχεδιασμένη επίθεση άρνησης εξυπηρέτησης μπορεί να είναι αδύνατη, λόγω των πιθανών περιορισμών στους διάφορους πόρους του δικτύου. Ένας ισχυρότερος κόμβος μπορεί άνετα να επικαλύψει κάποιον άλλον και να τον αποτρέψει από την εκτέλεση των καθηκόντων του.

Αξίζει να σημειωθεί ότι οι επιθέσεις στα δίκτυα αισθητήρων δεν περιορίζονται μόνο σε επιθέσεις άρνησης εξυπηρέτησης αλλά καλύπτουν ποικίλες τεχνικές όπως ανάληψη κόμβων και επιθέσεις στο φυσικό επίπεδο αλλά και επιθέσεις στα πρωτόκολλα δρομολόγησης του δικτύου. Σε αυτήν την υποενότητα εξετάζονται αρχικά ορισμένες επιθέσεις άρνησης εξυπηρέτησης και στην συνέχεια κάποιες επιπλέον επιθέσεις που λαμβάνουν χώρα στο πρωτόκολλο δρομολόγησης του δικτύου αλλά και μια επίθεση στην ταυτότητα των κόμβων γνωστή ως επίθεση Sybil.

### 2.4.1 ΥΠΟΒΑΘΡΟ

Ο Wood και ο Stankovic έχουν ορίσει ένα είδος άρνησης εξυπηρέτησης ως "ένα οποιοδήποτε γεγονός που μικραίνει η μηδενίζει την δυνατότητα ενός δικτύου να εκτελέσει την αναμενόμενη λειτουργία του". Βεβαίως, οι επιθέσεις άρνησης εξυπηρέτησης δεν είναι ένα καινούργιο φαινόμενο που εμφανίστηκε στα δίκτυα αισθητήρων. Στην πραγματικότητα, υπάρχον διάφορες τεχνικές που εφαρμόζονται στα τυπικά δίκτυα έτσι ώστε να αντιμετωπιστούν μερικές από τις πιο κοινές επιθέσεις τέτοιου είδους. Δυστυχώς τα δίκτυα αισθητήρων δεν είναι σε θέση να χειριστούν το επιπλέον υπολογιστικό κόστος μιας υλοποίησης μιας τεχνικής ασφάλειας ενάντια σε επιθέσεις άρνησης εξυπηρέτησης.

Το γεγονός ότι στα δίκτυα αισθητήρων μπορούν να αναπτυχθούν ιδιαίτερα κρίσιμες εφαρμογές καθιστά την ασφάλεια ενάντια σε επιθέσεις άρνησης εξυπηρέτησης επιτακτική. Για παράδειγμα, ένα δίκτυο αισθητήρων που έχει αναπτυχθεί με σκοπό την πυροπροστασία και την ειδοποίηση των ατόμων εντός ενός κτιρίου σε περίπτωση πυρκαγιάς είναι αρκετά κρίσιμο να είναι ασφαλές σε επιθέσεις άρνησης εξυπηρέτησης. Επίσης, μια επίθεση άρνησης εξυπηρέτησης ενάντια σε ένα δίκτυο που έχει σχεδιαστεί για τον έλεγχο της ροής της κυκλοφοράς καθώς και των σηματοδοτών κίνησης μπορεί να προκαλέσει πολλά δυσάρεστα γεγονότα ιδιαίτερα αν έχει αναπτυχθεί σε δρόμους ταχείας κυκλοφορίας.

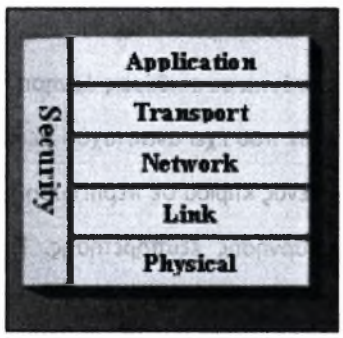
Για τους παραπάνω λόγους, οι ερευνητές έχουν ξοδέψει πολύ χρόνο για να προσδιοριστούν και αντιμετωπιστούν οι διάφοροι τύποι επιθέσεων άρνησης εξυπηρέτησης.

### 2.4.2 ΤΥΠΟΙ ΕΠΙΘΕΣΕΩΝ ΑΡΝΗΣΗΣ ΕΞΥΠΗΡΕΤΗΣΗΣ

Μια τυπική επίθεση άρνησης εξυπηρέτησης στα δίκτυα αισθητήρων είναι η παρεμβολή (jamming) στην εμβέλεια ενός κόμβου η μιας ομάδας κόμβων. Μια τέτοια περίπτωση είναι η εκπομπή ενός σήματος που παρεμποδίζει τις συχνότητες στις οποίες εκπέμπουν οι κόμβοι του δικτύου. Μια επίθεση jamming σε ένα δίκτυο μπορεί να έχει μια σταθερή (constant jamming) ή μια διακοπτόμενη (intermittent jamming) μορφή. Η πρώτη έχει να κάνει με την συνολική και αδιάλειπτη δημιουργία παρασίτων σε ολόκληρο το δίκτυο με αποτέλεσμα να μην είναι δυνατή η αποστολή και η λήψη πακέτων-μηνυμάτων. Στην περίπτωση των

διακοπτόμενων παρεμβολών οι κόμβοι μπορούν να ανταλλάσουν μηνύματα περιοδικά αλλά όχι συνέχεια. Έτσι μπορεί να προκληθούν επίσης δυσάρεστες συνέπειες καθώς τα μηνύματα που ανταλλάσουν οι κόμβοι μεταξύ τους μπορεί να είναι ευαίσθητα στο χρόνο (να μεταφέρουν χρονοσφραγίδες) [9].

Επιθέσεις μπορεί να πραγματοποιηθούν και στο επίπεδο σύνδεσης (link layer). Ένας επιτιθέμενος μπορεί απλά να παραβιάσει σκόπιμα το πρωτόκολλο επικοινωνίας (802.11b Wi-Fi) και να αποστέλλει συνεχώς μηνύματα με αποτέλεσμα την δημιουργία συγκρούσεων. Οι συγκρούσεις θα έχουν σαν αποτέλεσμα την επανα-αποστολή οποιοδήποτε μηνύματος που έχει χαθεί. Χρησιμοποιώντας αυτή την τεχνική ο επιτιθέμενος στοχεύει στην μείωση της ενέργειας των κόμβων με την αποστολή πολλών μηνυμάτων.



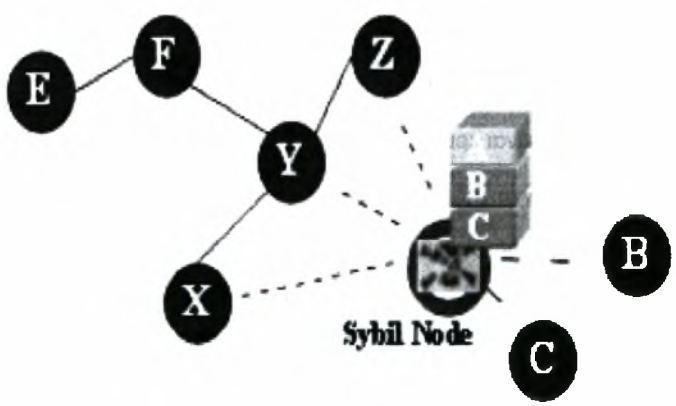

**Εικόνα 1:** επίπεδα OSI

Στο επίπεδο δρομολόγησης, ένας κακόβουλος χρήστης μπορεί να εκμεταλλευτεί ένα multi hop δίκτυο με την άρνηση της δρομολόγησης των μηνυμάτων. Αυτό θα μπορούσε να γίνει περιοδικά ή συνεχώς με αποτέλεσμα οποιοσδήποτε κόμβος που δρομολογεί τα πακέτα μέσω ενός γειτονικού κακόβουλου κόμβου να μην μπορεί να ανταλλάξει μηνύματα με τουλάχιστον ένα μέρος του δικτύου. Αξίζει να τονιστεί ότι υπάρχουν διάφορες επεκτάσεις αυτής της επίθεσης έτσι ώστε τα μηνύματα να καθοδηγούνται σε κόμβους που επηρεάζει ο επιτιθέμενος.

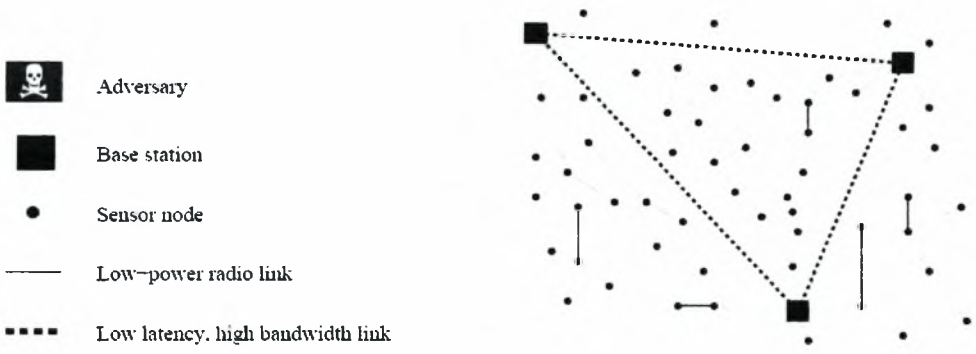
Το επίπεδο μεταφοράς είναι επίσης ευάλωτο σε επιθέσεις, όπως στην περίπτωση της πλημμύρας (flooding attack). Η πλημμύρα μπορεί να πραγματοποιηθεί απλά με την αποστολή πολλών αιτημάτων σύνδεσης σε έναν επιρρεπή κόμβο. Σε αυτή την περίπτωση θα σπαταληθούν πόροι για να εξυπηρετηθεί η αίτηση σύνδεσης. Τελικά οι πόροι του κόμβου θα εξαντληθούν καθιστώντας τον κόμβο άχρηστο.

### 2.4.3 Η ΕΠΙΘΕΣΗ SYBIL

Η επίθεση Sybil μπορεί απλά να οριστεί ως μια επίθεση που προέρχεται από μια συσκευή (κόμβος) που κατέχει παράνομα πολλαπλές ταυτότητες (identities) [10]. Αρχικά έχει περιγραφεί ως μια επίθεση ικανή να υπερνικήσει τους μηχανισμούς επανάληψης αποστολής των κατανεμημένων δικτύων peer-to-peer [11],. Εκτός από την αποτελεσματικότητά της έναντι των μηχανισμών αποθήκευσης δεδομένων, η επίθεση Sybil είναι επίσης αποτελεσματική έναντι των αλγορίθμων δρομολόγησης, της συσσώρευσης των δεδομένων, των εκλογών, της δίκαιης κατανομής πόρων και της ανίχνευσης βλαβών. Ανεξάρτητα από τον στόχο (εκλογές, δρομολόγηση, συσσώρευση δεδομένων), ο αλγόριθμος της επίθεσης Sybil λειτουργεί πανομοιότυπα. Καθεμία από τις τεχνικές της επίθεσης χρησιμοποιεί τις πολλαπλές ταυτότητες των κόμβων. Για παράδειγμα, σε μια διαδικασία εκλογής ενός δικτύου αισθητήρων μια επίθεση Sybil μπορεί να χρησιμοποιήσει πολλαπλές ταυτότητες ενός κόμβου για να παράγει επιπλέον ψηφοφορίες. Ομοίως, μια τυχόν επίθεση σε ένα πρωτόκολλο δρομολόγησης θα στηριζόταν σε ένα "κακόβουλο" κόμβο με πολλαπλές ταυτότητες, με αποτέλεσμα η διαδικασία προώθησης ενός πακέτου-μηνύματος, τις περισσότερες φορές, να περνά μέσα από αυτόν.




**Εικόνα 2:** Η επίθεση Sybil




**Εικόνα 3:** Παράδειγμα δικτύου αισθητήρων

### 2.4.4 ΕΠΙΘΕΣΕΙΣ ΑΝΑΛΥΣΗΣ ΚΙΝΗΣΗΣ

Τα ασύρματα δίκτυα αισθητήρων αποτελούνται από πολλούς, περιορισμένους σε πόρους, κόμβους αισθητήρων και από μερικούς σχετικά ισχυρούς και με αρκετούς πόρους σταθμούς βάσης (μια τυπική τοπολογία δικτύου αισθητήρων εικονίζεται στην παραπάνω εικόνα). Έτσι, συνηθίζεται οι πληροφορίες που συγκεντρώνονται από τους απλούς κόμβους να καθοδηγούνται τελικά σε έναν από τους σταθμούς βάσης. Συχνά, ένας αντίπαλος μπορεί απλά να θέσει έναν σταθμό βάσης εκτός λειτουργίας με αποτέλεσμα την διάλυση του δικτύου. Υπάρχουν επιθέσεις που μπορούν να προσδιορίσουν τους σταθμούς βάσης σε ένα δίκτυο (με μεγάλη πιθανότητα) χωρίς καν να γνωρίζουν το περιεχόμενο των μηνυμάτων (υποθέτοντας ότι τα πακέτα είναι κρυπτογραφημένα) [12].

Μια επίθεση στον έλεγχο της ροής, χρησιμοποιεί απλά την ιδέα ότι οι κόμβοι που είναι πιο κοντά στον σταθμό βάσης διαβιβάζουν περισσότερα μηνύματα σε σχέση με τους πιο μακρινούς κόμβους. Ένας επιτιθέμενος αρκεί να εντοπίσει ποιοι κόμβοι στέλνουν μηνύματα και να ακολουθήσει τους κόμβους που στέλνουν τα περισσότερα πακέτα. Σε μια επίθεση συσχέτιση χρόνου (time correlation attack), ο επιτιθέμενος απλά δημιουργεί γεγονότα (events) και ελέγχει σε ποιον κόμβο στέλνεται το μήνυμα. Αξίζει να σημειωθεί ότι για την δημιουργία ενός γεγονότος, ένας επιτιθέμενος, μπορεί απλά να αναπαραγάγει ένα φυσικό γεγονός που θα γίνει κατανοητό από τον αισθητήρα (π.χ άνοιγμα ενός φακού πάνω από έναν αισθητήρα ελέγχου φωτός).

### 2.4.5 ΕΠΙΘΕΣΗ ΑΝΤΙΓΡΑΦΗΣ ΚΟΜΒΟΥ

Μια επίθεση αντιγραφής ενός κόμβου είναι ιδιαίτερα απλή. Ένας επιτιθέμενος επιδιώκει να προσθέσει στο υπάρχον δίκτυο αισθητήρων έναν κόμβο με την αντιγραφή της ταυτότητας (ID) ενός κόμβου που υπάρχει ήδη στο δίκτυο. Ο κόμβος που αντιγράφεται με τον παραπάνω τρόπο μπορεί να δημιουργήσει πολλαπλά προβλήματα και να μειώσει αισθητά την απόδοση του δικτύου: τα πακέτα μπορεί να διασπαστούν η και ακόμα να προωθηθούν λανθασμένα. Αυτό μπορεί να οδηγήσει σε ένα αποσυνδεδεμένο δίκτυο όπου θα διακινούνται ψευδή γεγονότα από τους αισθητήρες. Εάν ένας επιτιθέμενος αποκτήσει πρόσβαση σε ολόκληρο το δίκτυο μπορεί να αποθηκεύσει κρυπτογραφικά κλειδιά στον αντιγραμμένο αισθητήρα και επίσης να εισάγει τον κόμβο σε στρατηγικά σημεία εντός του δικτύου (κοντά σε έναν σταθμό βάσης). Αυτό θα έχει σαν αποτέλεσμα τον εύκολο χειρισμό ενός τομέα του δικτύου.

### 2.4.6 ΕΠΙΘΕΣΕΙΣ ΕΝΑΝΤΙΑ ΣΤΗΝ ΕΜΠΙΣΤΕΥΤΙΚΟΤΗΤΑ

Η τεχνολογία των δικτύων αισθητήρων υπόσχεται μια μεγάλη αύξηση της αυτόματης συλλογής δεδομένων μέσω της αποδοτικής επέκτασης μικροσκοπικών συσκευών. Ενώ οι τεχνολογίες που εφαρμόζονται στα δίκτυα αισθητήρων προσφέρουν σημαντικά οφέλη, είναι επίσης εκτεθειμένες σε διάφορες επιθέσεις εναντίον της εμπιστευτικότητας λόγω των αυξημένων δυνατοτήτων στην συλλογή δεδομένων. Οι επιτιθέμενοι μπορεί να χρησιμοποιήσουν φαινομενικά αβλαβή δεδομένα για να παράγουν ευαίσθητες για το δίκτυο πληροφορίες. Παίρνοντας ως παράδειγμα το γνωστό "panda-hunter" [13], πρόβλημα ένας κυνηγός μπορεί να υπονοήσει την θέση των panda απλά ελέγχοντας την ροή της κυκλοφορίας.

Το κύριο πρόβλημα για την εμπιστευτικότητα, εντούτοις, δεν είναι ότι τα δίκτυα αισθητήρων επιτρέπουν την συλλογή πληροφοριών. Στην πραγματικότητα, μπορούν να συλλέγουν πολλές πληροφορίες μέσω της άμεσης επιτήρησης περιοχών. Επίσης, τα δίκτυα αισθητήρων οξύνουν το πρόβλημα της εμπιστευτικότητας των δεδομένων, επειδή μεγάλες ποσότητες δεδομένων μπορούν να προσπελαστούν από απόσταση. Έτσι οι επιτιθέμενοι δεν χρειάζεται να είναι παρόντες στο φυσικό περιβάλλον για να εξαπολύσουν τις επιθέσεις.

Μπορούν να συλλέξουν πληροφορίες με χαμηλό κίνδυνο και ανώνυμα. Η εξ' αποστάσεως πρόσβαση επιτρέπει επίσης στον επιτιθέμενο να ελέγξει πολλαπλές περιοχές του δικτύου ταυτόχρονα. Μερικές από τις πιο κοινές επιθέσεις [14,15] ενάντια στην εμπιστευτικότητα των δεδομένων στα δίκτυα αισθητήρων είναι:

- **Μη εξουσιοδοτημένη ακρόαση (eavesdropping).** Αποτελεί την πιο προφανή επίθεση ενάντια στην εμπιστευτικότητα. Ένας ορισμός της μη εξουσιοδοτημένης ακρόασης μπορεί να είναι η μη με ενεργητικούς τρόπους απόκτηση πληροφοριών για το δίκτυο. Με την μη εξουσιοδοτημένη ακρόαση ο επιτιθέμενος μπορεί εύκολα να αποκαλύψει το περιεχόμενο της επικοινωνίας μεταξύ των κόμβων. Το σημαντικότερο πρόβλημα που προκαλείται από την απειλή αυτή, σχετίζεται με τη δυνατότητα την οποία έχει ο υποκλοπέας να επεξεργάζεται τα μηνύματα που υποκλέπτει και στη συνέχεια, είτε να τα εκμεταλλεύεται ως έχουν, είτε να τα χρησιμοποιεί για την εκτόξευση εναντίων του δικτύου κάποιας άλλου είδους επίθεσης.
- **Ανάλυση κίνησης (traffic analysis).** Η ανάλυση της κίνησης τις περισσότερες φορές συνδυάζεται με μια επίθεση μη εξουσιοδοτημένης ακρόασης. Μια αύξηση στον αριθμό των πακέτων που ανταλλάσσονται μεταξύ συγκεκριμένων κόμβων μπορεί να σημαίνει ότι ένας συγκεκριμένος αισθητήρας έχει αρχίσει μια συγκεκριμένη διαδικασία. Μέσω της ανάλυσης της κυκλοφορίας μπορούν να προσδιοριστούν αποτελεσματικά αισθητήρες με ειδικό ρόλο στο δίκτυο.
- **Συγκάλυψη (camouflage).** Οι επιτιθέμενοι μπορούν να παρεμβάλλουν ή να αναγκάσουν τους κόμβους τους έτσι ώστε να αποκρυφτούν από το δίκτυο. Μετά από αυτό, οι κόμβοι μπορούν να μεταμφιεστούν σε κανονικούς κόμβους του δικτύου και να προσελκύσουν πακέτα με σκοπό την εσφαλμένη προώθησή τους.

Αξίζει να σημειωθεί ότι τα υπάρχοντα πρωτόκολλα και η υπάρχουσα βιβλιογραφία δεν καλύπτει πλήρως το θέμα της ασφάλειας σε επιθέσεις εναντίον της εμπιστευτικότητας για τα ασύρματα δίκτυα αισθητήρων. Γίνεται έτσι κατανοητό ότι πρόκειται για ένα ανοικτό πεδίο έρευνας.

### 2.4.7 ΕΠΙΘΕΣΕΙΣ ΣΤΟ ΦΥΣΙΚΟ ΕΠΙΠΕΔΟ

Τα δίκτυα αισθητήρων, όπως τονίσαμε και σε προηγούμενες ενότητες της εργασίας, πολλές φορές αναπτύσσονται σε εχθρικά υπαίθρια περιβάλλοντα. Σε τέτοια περιβάλλοντα, το μικρό μέγεθος των αισθητήρων συνδυασμένο με ευάλωτη ασύρματη και κατανεμημένη φύση της ανάπτυξής τους, τους καθιστά ιδιαίτερα ευάλωτους σε επιθέσεις στο φυσικό επίπεδο, π.χ φυσική καταστροφή των αισθητήρων λόγω πυρκαγιάς ή άσκηση βίας [16]. Αντίθετα με τα προαναφερθέντα είδη επιθέσεων, τυχόν επιθέσεις στην φυσική υποδομή του δικτύου έχουν σαν αποτέλεσμα την μόνιμη καταστροφή του με τα αποτελέσματα να είναι αμετάκλητα. Για παράδειγμα, οι επιτιθέμενοι μπορούν να αποσπάσουν την μνήμη των αισθητήρων, με αποτέλεσμα να αποκτήσουν πρόσβαση στα διάφορα κλειδιά κρυπτογράφησης, να επανα-προγραμματίσουν τους αισθητήρες ή και ακόμα να αντικαταστήσουν κόμβους του δικτύου με αισθητήρες που θα έχουν οι ίδιοι υπό την κατοχή τους [17]. Η πρόσφατη έρευνα έχει καταδείξει ότι ένας επιτιθέμενος μπορεί να αποκτήσει πρόσβαση στους πιο κοινούς τύπους ασύρματων αισθητήρων, όπως οι MICA2, σε λιγότερο από ένα λεπτό. Τα αποτελέσματα των ερευνών δεν είναι μη αναμενόμενα, αφού οι αισθητήρες MICA2 δεν παρέχουν ανθεκτική προστασία υλικού. Τέλος εάν ένας επιτιθέμενος καταφέρει να αποκτήσει πρόσβαση σε έναν κόμβο ενός δικτύου μπορεί να τροποποιήσει τον κώδικα που βρίσκεται στην μνήμη του κόμβου.

## 2.5 ΜΕΤΡΑ ΓΙΑ ΠΡΟΣΤΑΣΙΑ ΕΝΑΝΤΙΑ ΣΕ ΕΠΙΘΕΣΕΙΣ

Στο σημείο αυτό βρισκόμαστε σε θέση να περιγράψουμε τα μέτρα που πρέπει να επιτευχθούν έτσι ώστε τα διάφορα δίκτυα αισθητήρων να είναι ασφαλή (σε σχέση με τις απαιτήσεις ασφάλειας που περιγράψαμε παραπάνω). Αρχίζουμε με τον τρόπο που εγκαθίστανται και διανέμονται τα κλειδιά κρυπτογράφησης στα δίκτυα αισθητήρων, τα οποία είναι ουσιαστικά ο θεμέλιος λίθος για την δημιουργία ενός πραγματικά ασφαλούς συστήματος. Στην συνέχεια ασχολούμαστε με τα αντίμετρα ενάντια σε επιθέσεις άρνησης εξυπηρέτησης καθώς και με το ασφαλές broadcasting και multicasting. Ακολουθούν τα αντίμετρα σε επιθέσεις ενάντια σε πρωτόκολλα δρομολόγησης, σε επιθέσεις ανάλυσης κίνησης, σε επιθέσεις ενάντια στην εμπιστευτικότητα καθώς και σε επιθέσεις στο φυσικό επίπεδο.

### 2.5.1 ΔΙΑΧΕΙΡΙΣΗ ΚΑΙ ΕΓΚΑΤΑΣΤΑΣΗ ΚΛΕΙΔΙΩΝ ΚΡΥΠΤΟΓΡΑΦΗΣΗΣ

Μια πτυχή ασφάλειας που πρέπει να ληφθεί σοβαρά υπόψη για την καθολική ασφάλεια στα δίκτυα αισθητήρων είναι ο τρόπος εγκατάστασης, διαχείρισης αλλά και αποστολής των διάφορων κλειδιών κρυπτογράφησης. Όπως έχουμε τονίσει αρκετές φορές παραπάνω, τα ασύρματα δίκτυα αισθητήρων είναι μοναδικά ( σε σχέση άλλα είδη ασύρματων δικτύων) λόγω του μεγέθους τους και της υπολογιστικής και ενεργειακής ισχύς. Επίσης η σπουδαιότητα και η κρισιμότητα των δεδομένων που θα μεταφέρονται ανάμεσα στους κόμβους των δικτύων (σε περίπτωση ανάπτυξης ενός δικτύου σε εχθρικό και εκ φύσεως ανασφαλούς περιβάλλοντος), καθιστά τον τρόπο που γίνεται η διαχείριση των κλειδιών κρυπτογράφησης ένα από τα σημαντικότερα θέματα για την ασφάλεια. Αφού η κρυπτογράφηση και η διαχείριση των κρυπτογραφικών κλειδιών είναι τόσο κρίσιμα για την ασφάλεια των δικτύων αισθητήρων αρχίζουμε με την επισκόπηση του μοναδικού κλειδιού (unique key) και περιληπτική περιγραφή ζητημάτων κρυπτογράφησης που αφορούν τα δίκτυα αισθητήρων πριν προχωρήσουμε σε πιο συγκεκριμένες περιπτώσεις που αφορούν την ασφάλειά τους.

#### 2.5.1.1 ΥΠΟΒΑΘΡΟ

Τα ζητήματα διαχείρισης των κρυπτογραφικών κλειδιών δεν είναι μοναδικά για τα δίκτυα αισθητήρων. Τα θέματα της εγκατάστασης και διαχείρισης των κλειδιών Οι εγκατάσταση και η διαχείριση των κλειδιών στα παραδοσιακά δίκτυα πραγματοποιούνται χρησιμοποιώντας ένα από τα πολλά πρωτόκολλα δημοσίου κλειδιού. Ένα από τα πιο συνηθισμένα πρωτόκολλα είναι το Diffle-Hellman public key protocol, αλλά υπάρχουν πολλά περισσότερα.

Οι περισσότερες από τις παραδοσιακές τεχνικές, όμως, δεν ταιριάζουν στους εκ φύσεως ενεργειακούς περιορισμούς των δικτύων αισθητήρων. Αυτό οφείλεται κατά ένα μεγάλο μέρος στο γεγονός ότι οι παραδοσιακές τεχνικές χρησιμοποιούν το μοντέλο της ασύμμετρης κρυπτογράφησης ή κρυπτογράφηση δημόσιου κλειδιού. Σε αυτή την περίπτωση είναι απαραίτητο να διατηρηθούν δυο κλειδιά που έχουν μια μαθηματική σχέση μεταξύ τους, ένα από τα οποία δημοσιοποιείται (δημόσιο κλειδί) και το άλλο μένει ιδιωτικό. Τα στοιχεία που κρυπτογραφούνται με το δημόσιο κλειδί μπορούν να αποκρυπτογραφηθούν μόνο με το ιδιωτικό κλειδί κάθε κόμβου. Το πρόβλημα με την ασύμμετρη κρυπτογραφία, σε ένα ασύρματο δίκτυο αισθητήρων, είναι ότι απαιτείται μεγάλη υπολογιστική ισχύ από τους μεμονωμένους κόμβους του δικτύου. Αυτό βέβαια ισχύει στην γενική περίπτωση αλλά σύμφωνα με τα [18,19,20,21] αποδεικνύεται ότι είναι εφικτό με την κατάλληλη επιλογή των αλγορίθμων.

Παρόλα αυτά η συμμετρική κρυπτογραφία είναι η καταλληλότερη επιλογή για εφαρμογές που δεν μπορούν να ανταπεξέλθουν στις απαιτήσεις σε υπολογιστική ισχύ (που συνεπάγεται και ενεργειακή ισχύ) που επιβάλλει η ασύμμετρη κρυπτογραφία. Οι συμμετρικές τεχνικές χρησιμοποιούν ένα μοναδικό κλειδί που είναι γνωστό μόνο μεταξύ των επικοινωνούντων κόμβων. Το κλειδί αυτό χρησιμοποιείται τόσο για την κρυπτογράφηση όσο και για την αποκρυπτογράφηση των δεδομένων. Το παραδοσιακό παράδειγμα για την συμμετρική κρυπτογράφηση είναι το DES (Data Encryption Standard). Η χρήση του DES, εντούτοις είναι περιορισμένη καθώς μπορεί να σπάσει σχετικά εύκολα. Λαμβάνοντας υπόψη τις ανεπάρκειες του DES έχουν αναπτυχθεί άλλα συμμετρικά πρωτόκολλα κρυπτογράφησης όπως το 3DES (triple DES), RC5, AES κλπ [22]. Στο σημείο αυτό αξίζει να επισημανθεί ότι ο αλγόριθμος που αναπτύσσεται στο μέρος 2 (tiny encryption algorithm) της εργασίας είναι και

αυτός ένας μηχανισμός που βασίζεται στην συμμετρική κρυπτογράφηση καθώς τα επικοινωνούντα μέρη κατέχουν ένα κοινό κλειδί και κρυπτογραφούν – αποκρυπτογραφούν τα δεδομένα χρησιμοποιώντας το ιδιωτικό κλειδί.

| *By key setup:* | | | | | | |
|---|---|---|---|---|---|---|
| Rank | Size Optimized | | | Speed Optimized | | |
| | Code mem. | Data mem. | Speed | Code mem. | Data mem. | Speed |
| 1 | RC5-32 | MISTY1 | MISTY1 | RC6-32 | MISTY1 | MISTY1 |
| 2 | KASUMI | Rijndael | Rijndael | KASUMI | Rijndael | Rinjdael |
| 3 | RC6-32 | KASUMI | KASUMI | RC5-32 | KASUMI | KASUMI |
| 4 | MISTY1 | RC6-32 | Camellia | MISTY1 | RC6-32 | Camellia |
| 5 | Rijndael | RC5-32 | RC5-32 | Rijndael | Camellia | RC5-32 |
| 6 | Camellia | Camellia | RC6-32 | Camellia | RC5-32 | RC6-32 |
| *By encryption (CBC/CFB/OFB/CTR)* | | | | | | |
| Rank | Size Optimized | | | Speed Optimized | | |
| | Code mem. | Data mem. | Speed | Code mem. | Data mem. | Speed |
| 1 | RC5-32 | RC5-32 | Rijndael | RC6-32 | RC5-32 | Rijndael |
| 2 | RC6-32 | MISTY1 | MISTY1 | RC5-32 | MISTY1 | Camellia |
| 3 | MISTY1 | KASUMI | KASUMI | MISTY1 | KASUMI | MISTY1 |
| 4 | KASUMI | RC6-32 | Camellia | KASUMI | RC6-32 | RC5-32 |
| 5 | Rijndael | Rijndael | RC6-32 | Rijndael | Rijndael | KASUMI |
| 6 | Camellia | Camellia | RC5-32 | Camellia | Camellia | RC6-32 |

**Πίνακας 1:** Περίληψη ανάλυσης κρυπτογραφημάτων

Στο [23] παρουσιάζεται μια ανάλυση με τα διάφορα κρυπτογραφήματα και μια περίληψή της παρουσιάζεται στον πίνακα 1. Ο πίνακας παρουσιάζει δυο διαφορετικές ταξινομήσεις – μια με το κλειδί που χρησιμοποιείται και μια με τον τρόπο κρυπτογράφησης. Σημειώνεται ότι και στις δυο ταξινομήσεις οι αλγόριθμοι είναι βελτιστοποιημένοι τόσο για την ταχύτητα όσο και για το μέγεθος και είναι ταξινομημένοι σύμφωνα με την ταχύτητα, με το μέγεθος του κώδικα και τω δεδομένων.

Ένα βασικό πρόβλημα του συμμετρικού τρόπου κρυπτογράφησης είναι ο η ανταλλαγή των κλειδιών. Το πρόβλημα ανταλλαγής των κλειδιών προκύπτει από το γεγονός ότι οι δυο επικοινωνούντες κόμβοι πρέπει με κάποιο τρόπο να γνωρίζουν το κοινό κλειδί πριν αρχίσουν να επικοινωνούν με ασφάλεια. Έτσι το πρόβλημα που προκύπτει είναι το πώς θα εξασφαλιστεί ότι το κλειδί θα μοιραστεί μόνο στους κόμβους που θέλουν να επικοινωνήσουν και ότι κανένας άλλος (κακόβουλος) κόμβος, που επιθυμεί την μη εξουσιοδοτημένη ακρόαση, δεν θα αποκτήσει πρόσβαση στο κλειδί.

### 2.5.1.2 ΕΓΚΑΤΑΣΤΑΣΗ ΚΛΕΙΔΙΩΝ ΚΑΙ ΣΧΕΤΙΚΑ ΠΡΩΤΟΚΟΛΛΑ

Ορισμένες τεχνικές προ-διανομής κλειδιών παρουσιάζονται στα [5,6,7,8]. Ο Eschenauer και ο Glidor προτείνουν μια τεχνική προ-διανομής κλειδιών που στηρίζεται στην πιθανολογική διανομή των κλειδιών στο δίκτυο αισθητήρων. Το σύστημα τους λειτουργεί με την διανομή ενός αρχικού σχήματος κλειδιών (key ring) σε κάθε συμμετέχοντα κόμβο στο δίκτυο πριν την επέκταση του δικτύου. Κάθε αρχικό κλειδί πρέπει να αποτελείται από έναν αριθμό τυχαία επιλεγμένων κλειδιών από μια μεγαλύτερη ομάδα που παράγεται off-line.

Χρησιμοποιώντας αυτή την τεχνική δεν είναι απαραίτητο ότι κάθε ζεύγος κόμβων μοιράζεται ένα κλειδί. Οποιοιδήποτε δυο κόμβοι που μοιράζονται ένα κλειδί μπορούν να το χρησιμοποιήσουν για να εγκαταστήσουν μια σύνδεση το ένα με το άλλο. Ο Eschenauer και ο Glidor αποδεικνύουν ότι, αν και όχι τέλεια, η παραπάνω τεχνική μπορεί να εφαρμοστεί σε δίκτυα αισθητήρων μεγάλης κλίμακας.

Το πρωτόκολλο LEAP που περιγράφεται από τον Zhu [93], υιοθετεί μια μέθοδο που χρησιμοποιεί πολλαπλούς μηχανισμούς κλειδιών. Η παρατήρηση τους είναι ότι η μια μοναδική μέθοδος διαχείρισης κλειδιών δεν καλύπτει όλα τα είδη επικοινωνίας ανάμεσα στους κόμβους ενός δικτύου αισθητήρων. Επομένως χρησιμοποιούνται τέσσερα διαφορετικά ανάλογα με το ποιον επικοινωνεί κάθε κόμβος. Στους αισθητήρες φορτώνεται εκ των προτέρων ένα αρχικό κλειδί από το οποίο μπορούν να αναπτυχθούν τα επιπλέον κλειδιά. Για λόγους ασφάλειας το αρχικό κλειδί μπορεί να διαγραφεί μετά από την χρήση για να διασφαλιστεί ότι κανένας "κακόβουλος" κόμβος δεν μπορεί να επηρεάσει άλλους κόμβους του δικτύου.

Στο PIKE [24], ο Chan και ο Perrig περιγράφουν ένα μηχανισμό για την εγκατάσταση ενός κλειδιού ανάμεσα σε δυο κόμβους που βασίζεται σε σχέσεις εμπιστοσύνης με έναν άλλο τρίτο κόμβο του δικτύου. Οι κόμβοι και τα διαμοιραζόμενα κλειδιά τους είναι εξαπλωμένα μέσα στο δίκτυο αισθητήρων έτσι ώστε για οποιουσδήποτε δυο κόμβους Α και Β υπάρχει ένας κόμβος Γ που μοιράζεται ένα κλειδί με τον Α και με τον Β. Επομένως το πρωτόκολλο εγκατάστασης κλειδιών ανάμεσα στους κόμβους Α και Β μπορεί να δρομολογηθεί με ασφάλεια μέσω του Γ.

Ο Huang [25] προτείνει ένα υβριδικό σχέδιο διανομής και εγκατάστασης κλειδιών που χρησιμοποιεί την διαφορά στην υπολογιστική ισχύ και την ενέργεια ανάμεσα σε έναν κόμβο και τον σταθμό βάσης. Προϋποθέτει ότι ένας μεμονωμένος αισθητήρας έχει πολύ μικρότερη υπολογιστική και ενεργειακή ισχύ σε σχέση με έναν σταθμό βάσης. Λαμβάνοντας υπόψη το παραπάνω γεγονός προτείνουν ο μεγαλύτερος φόρτος της κρυπτογραφικής διαδικασίας να τοποθετηθεί πάνω στον σταθμό βάσης, Από την πλευρά των αισθητήρων οι συμμετρικές διαδικασίες χρησιμοποιούνται αντί των ασύμμετρων εναλλακτικών λύσεων. Ο εκάστοτε αισθητήρας και ο σταθμός βάσης αυθεντικοποιούνται χρησιμοποιώντας ένα μηχανισμό κρυπτογραφίας που ονομάζεται elliptic curve cryptography. Η τεχνική αυτή χρησιμοποιείται συχνά στους αισθητήρες λόγω του γεγονότος ότι μικρά σε μέγεθος κλειδιά είναι ικανά να επιτύχουν ένα δεδομένο μέγεθος ασφάλειας.

Ο Huang επίσης χρησιμοποιεί επίσης τα πιστοποιητικά για να καθιερώσει την νομιμότητα ενός δημόσιου κλειδιού. Τα πιστοποιητικά είναι βασισμένα σε ένα elliptic curve σχήμα πιστοποιητικών [25]. Τέτοιου είδους πιστοποιητικά είναι χρήσιμα και μπορούν να εξασφαλίσουν τόσο ότι το κλειδί ανήκει σε μια συσκευή αλλά και το ότι η συσκευή είναι ένα νόμιμο μέλος του δικτύου. Κάθε κόμβος λαμβάνει ένα πιστοποιητικό πριν ενσωματωθεί στο δίκτυο με την χρήση μιας out-of-band (εκτός εύρους του δικτύου) διεπαφής.

### 2.5.2 ΠΡΟΣΤΑΣΙΑ ΕΝΑΝΤΙΑ ΣΕ ΕΠΙΘΕΣΕΙΣ ΑΡΝΗΣΗΣ ΕΞΥΠΗΡΕΤΗΣΗΣ (DoS attacks)

Στον πίνακα 2 συνοψίζονται τα επίπεδα από ένα τυπικό ασύρματο δίκτυο μαζί με τις επιθέσεις και τον τρόπο υπεράσπισή τους ενάντια σε αυτές. Από την στιγμή που οι επιθέσεις άρνησης εξυπηρέτησης είναι τόσο κοινές (όπως περιγράφονται στο υποκεφάλαιο 2.4.2), πρέπει να αναπτυχθούν αντίμετρα ικανά να εξασφαλίσουν την ασφάλεια του δικτύου ενάντια σε αυτά. Μια στρατηγική για την άμυνα ενάντια σε κλασσικές επιθέσεις jamming (παρεμβολής) είναι να εντοπιστεί το τμήμα του δικτύου που δέχεται την επίθεση και να αποτραπεί αποτελεσματικά η δρομολόγηση της κίνησης του δικτύου μέσω αυτού του τμήματος. Ο Wood και ο Stankovic [9] περιγράφουν μια προσέγγιση που αποτελείται από δυο φάσεις κατά τις οποίες οι κόμβοι που βρίσκονται στην περίμετρο του τμήματος του δικτύου που δέχεται την επίθεση στέλνουν αναφορές στους γειτονικούς κόμβους τους και στην

συνέχεια συνεργάζονται για να προσδιορίσουν το τμήμα που δέχεται επίθεση. Το αποτέλεσμα του εντοπισμού είναι όπως αναφέρθηκε η μη δρομολόγηση της κίνησης του δικτύου μέσω αυτού του τμήματος.

| Network Layer | Attacks | Defenses |
|---|---|---|
| Physical | Jamming | Spread-spectrum, priority messages, lower duty cycle, region mapping, mode change |
| | Tampering | Tamper-proof, hiding |
| Link | Collision | Error correcting code |
| | Exhaustion | Rate limitation |
| | Unfairness | Small frames |
| Network and routing | Neglect and greed | Redundancy, probing |
| | Homing | Encryption |
| | Misdirection | Egress filtering, authorization monitoring |
| | Black holes | Authorization, monitoring, redundancy |
| Transport | Flooding | Client Puzzles |
| | Desynchronization | Authentication |

**Πίνακας 2:** Κατηγοριοποίηση επιθέσεων και τρόποι άμυνας

Για να αντιμετωπιστούν επιθέσεις jamming στο επίπεδο MAC, οι κόμβοι μπορούν να χρησιμοποιήσουν έλεγχο ροής στο επίπεδο MAC. Αυτό μπορεί να επιτρέψει στο δίκτυο να αγνοήσει τυχόν αιτήσεις που έχουν σαν σκοπό την εξάντληση των αποθεμάτων ενέργειας των αισθητήρων. Παρόλα αυτά το παραπάνω μπορεί να έχει σαν αποτέλεσμα την μη αποτελεσματική διαχείριση μεγάλων όγκων κυκλοφορίας από το δίκτυο.

Επίσης μπορεί να επιτευχθεί η ασφάλεια ενάντια σε "κακόβουλους" κόμβους που δρομολογούν (σκόπιμα λανθασμένα) μηνύματα. Σε αυτή την περίπτωση ο αποστολέας μπορεί να στείλει ένα μήνυμα σε πολλαπλά μονοπάτια με σκοπό να αυξηθεί η πιθανότητα ότι το μήνυμα τελικά θα φτάσει στον προορισμό του. Αυτό βέβαια έχει σαν αποτέλεσμα την σπατάλη πόρων του δικτύου.

Για να ξεπεραστούν και οι επιθέσεις άρνησης εξυπηρέτησης στο επίπεδο μεταφοράς (transport layer), ο Aura, ο Nikander και ο Leiwo συνιστούν να χρησιμοποιηθούν οι γρίφοι πελατών (client puzzles), που έχουν προταθεί από τον Juels και τον Brainard, σε μια

προσπάθεια να αναγκαστεί ο κόμβος να δημιουργήσει την σύνδεση χρησιμοποιώντας του προσωπικούς του πόρους. Ο Aura τονίζει ότι ο εξυπηρετητής πρέπει να αναγκάσει τον πελάτη χρησιμοποιήσει τους πόρους του αρχικά. Περαιτέρω, προτείνουν ότι ένας εξυπηρετητής πρέπει πάντα να αναγκάζει τον πελάτη να δεσμεύει παραπάνω πόρους από αυτόν. Αυτή η στρατηγική μπορεί να είναι αποτελεσματική αν οι πελάτες έχουν συγκρίσιμη υπολογιστική ισχύ σε σχέση με αυτήν των εξυπηρετητών.

### 2.5.3 ΠΡΟΣΤΑΣΙΑ ΕΝΑΝΤΙΑ ΣΕ ΕΠΙΘΕΣΕΙΣ ΣΤΑ ΠΡΩΤΟΚΟΛΛΑ ΔΡΟΜΟΛΟΓΗΣΗΣ

Η δρομολόγηση στα δίκτυα αισθητήρων έχει ερευνηθεί και μελετηθεί σε μεγάλο βαθμό έως ένα σημείο. Παρόλα αυτά οι τρέχουσες έρευνες εστιάζουν στην παροχή περισσότερης αποδοτικής δρομολόγησης στο τομέα της ενέργειας. Είναι πολύ σημαντικό τα πρωτόκολλα δρομολόγησης να είναι ασφαλή αλλά και να ελαχιστοποιούν την κατανάλωση ενέργειας. Δεδομένου ότι τα δίκτυα αισθητήρων συνεχίζουν να αυξάνονται σε μέγεθος η ασφαλής δρομολόγηση πακέτων πρέπει να ληφθεί υπόψη στον αρχικό σχεδιασμό του δικτύου. Αυτή η υποενότητα περιγράφει την κατάσταση που επικρατεί αυτή την στιγμή όσο αφορά την προστασία της δρομολόγησης του δικτύου από διάφορες επιθέσεις.

#### 2.5.3.1 Υπόβαθρο

Τα ασύρματα δίκτυα αισθητήρων σχεδιάζονται έτσι ώστε η υπολογιστική και ενεργειακή ισχύ να είναι κατανεμημένη κατά μήκος του δικτύου. Αυτό έχει σαν αποτέλεσμα να πρέπει να χρησιμοποιούνται αποδοτικά πρωτόκολλα δρομολόγησης προκειμένου να μεγιστοποιηθεί η διάρκεια ζωής κάθε κόμβου. Υπάρχει μια ποικιλία από πρωτόκολλα δρομολόγησης που χρησιμοποιούνται στα δίκτυα αισθητήρων με αποτέλεσμα να μην είναι δυνατόν να παραχθεί ένα μοναδικό πρωτόκολλο ασφάλειας που να καλύπτει όλα τα είδη δρομολόγησης μαζί. Πριν αναφέρουμε τις διάφορες τεχνικές που χρησιμοποιούνται για την ασφαλής δρομολόγηση στα δίκτυα αισθητήρων θα αρχίσουμε με μια γενική επισκόπηση των πρωτοκόλλων δρομολόγησης.

Γενικά, οι αλγόριθμοι δρομολόγησης χρησιμοποιούνται για την ανταλλαγή πακέτων ανάμεσα σε κόμβους που βρίσκονται εκτός μιας συγκεκριμένης εμβέλειας. Αυτό είναι διαφορετικό σε σχέση με τους κόμβους που βρίσκονται εντός εμβέλειας καθώς μπορεί να χρησιμοποιηθεί ένα μόνο hop. Ακόμη και σε τέτοιου είδους δίκτυα (single hop networks) η ασφάλεια είναι ένα σημαντικό ζήτημα.

Ο πρώτος αλγόριθμος δρομολόγησης πακέτων βασίζεται στα προσδιοριστικά των κόμβων όπως και στην παραδοσιακή δρομολόγηση. Σε αυτήν την περίπτωση, ο κάθε κόμβος προσδιορίζεται από μια διεύθυνση και η δρομολόγηση από και προς έναν κόμβο γίνεται μέσω αυτής. Αυτό βέβαια θεωρείται ανεπαρκές για τα δίκτυα αισθητήρων αφού οι κόμβοι προσδιορίζονται από την θέση τους και όχι από ένα αναγνωριστικό.

Σαν συνέπεια του παραπάνω έχουν εισαχθεί τα λεγόμενα γεωγραφικά πρωτόκολλα δρομολόγησης (geographic routing protocols) [26, 27]. Ένα καθιερωμένο πρωτόκολλο δρομολόγησης, το GPSR, επιτρέπει στους κόμβος να στέλνουν ένα πακέτο σε μια περιοχή και όχι μόνο σε έναν συγκεκριμένο κόμβο. Αυτού του είδους τα πρωτόκολλα είναι αρκετά κατάλληλα για τα "δεδομενο-κεντρικά" (data-centric) δίκτυα. Σε ένα τέτοιο δίκτυο τα δεδομένα αποθηκεύονται με βάση το όνομα. Τα δεδομένα με το ίδιο όνομα αποθηκεύονται στον ίδιο κόμβο. Οι αναζητήσεις πραγματοποιούνται στο δίκτυο σύμφωνα με το όνομα και όχι το προσδιοριστικό του κόμβου στον οποίο έχουν αποθηκευτεί τα δεδομένα.

### 2.5.3.2 Τεχνικές για την ασφάλεια των αλγορίθμων δρομολόγησης

Οι Deng, Han και Mishra περιγράφουν ένα ανθεκτικό πρωτόκολλο δρομολόγησης (intrusion tolerant routing protocol – INTENS) το οποίο έχει ως σκοπό να περιορίσει το μέγεθος της καταστροφής που μπορεί να προκαλέσει ένας εισβολέας ο οποίος έχει αποκτήσει πρόσβαση στο δίκτυο χωρίς να έχει αναγνωριστεί.

Σημειώνουν ότι ένας εισβολέας δεν είναι ανάγκη να έχει παρεισφρήσει στο δίκτυο αλλά μπορεί απλά ένας κόμβος να δυσλειτουργεί χωρίς κάποιο κακόβουλο λόγο. Ο προσδιορισμός ενός πραγματικού εισβολέα σε σχέση με μια δυσλειτουργία είναι εξαιρετικά δύσκολος. Για το λόγο αυτό ο Deng κλπ κάνουν μια διάκριση μεταξύ των δυο περιπτώσεων. Η πρώτη τεχνική που χρησιμοποιούν για να μετριάσουν την ζημιά που μπορεί να προκληθεί από έναν πιθανό εισβολέα είναι απλά η χρήση της επαναληπτικής αποστολής (redundancy).

Σε αυτή την περίπτωση, όπως περιγράφηκε προηγουμένως για το denial of servise, μπορεί να δρομολογηθούν πολλαπλά μηνύματα αναγνώρισης ανάμεσα σε μια πηγή και ένα προορισμό. Ένα μήνυμα στέλνεται πολλές φορές κατά μήκος πολλών διαφορετικών μονοπατιών. Αυτό μειώνει την πιθανότητα να μην ληφθεί από κανέναν κόμβο. Για να διευκρινιστεί ποιος κόμβος έλαβε τελικά το μήνυμα μπορεί να υιοθετηθεί μια τεχνική αυθεντικοποίησης για να επιβεβαιωθεί η ακεραιότητα και η εγκυρότητα του μηνύματος.

Ο Deng κλπ χρησιμοποιούν επίσης μια υποτιθέμενη ασυμμετρία μεταξύ των σταθμών βάσεων και των ασύρματων κόμβων αισθητήρων. Υποθέτουν ότι η σταθμοί βάσης είναι λιγότερο περιορισμένοι σε πόρους σε σχέση με τους κόμβους αισθητήρων. Για αυτό τον λόγο προτείνουν ο υπολογισμός των πινάκων δρομολόγησης να γίνεται στους σταθμούς βάσης για λογαριασμό των κόμβων αισθητήρων. Αυτό γίνεται σε τρεις φάσεις. Στην πρώτη φάση ο σταθμός βάσης μεταδίδει ένα μήνυμα (με την μορφή αιτήματος) σε κάθε γείτονά του το οποίο διαδίδεται στην συνέχεια σε όλο το δίκτυο. Στην δεύτερη φάση ο σταθμός συλλέγει τοπικά τις πληροφορίες σύνδεσης από κάθε κόμβο. Τέλος, ο σταθμός βάσης κατασκευάζει ένα πίνακα δρομολόγησης (routing table) για κάθε κόμβο. Οι πίνακες δρομολόγησης θα συμπεριλαμβάνουν και πληροφορία σχετικά με την επαναληπτική αποστολή (redundancy) που περιγράφηκε παραπάνω.

Υπάρχουν πολλές πιθανές επιθέσεις που μπορεί να έχουν στόχο το πρωτόκολλο δρομολόγησης σε καθένα από τα παραπάνω τρία στάδια. Στην πρώτη φάση ένας κόμβος μπορεί να εξαπατήσει τον σταθμό βάσης με την αποστολή ενός πλαστού μηνύματος αίτησης. Ένας κακόβουλος κόμβος μπορεί επίσης να συμπεριλάβει ένα πλαστό μονοπάτι ή και περισσότερα όταν προωθεί το μήνυμα αίτησης στους γειτονικούς κόμβους. Μπορεί επίσης να μην προωθήσει το μήνυμα αίτησης.

Για να αντιμετωπιστεί το παραπάνω, ο Deng κλπ, χρησιμοποιούν ένα μηχανισμό παρόμοιο με το μTESLA όπου χρησιμοποιούνται αλυσίδες one-way κλειδιών για να προσδιορίσουν εάν το μήνυμα όντως προέρχεται από τον σταθμό βάσης.

Ο Tanachaiwiwat κλπ, παρουσιάζει μια νέα τεχνική που ονομάζεται TRANS (Trust Routing for Location Aware Sensor Networks) [28]. Το πρωτόκολλο TRANS είναι σχεδιασμένο έτσι ώστε η δρομολόγηση να γίνεται σε δεδομενο-κεντρικά δίκτυα (data-centric networks). Επίσης χρησιμοποιείται ένα ασύμμετρο κρυπτογραφικό σύστημα για να

εξασφαλιστεί η ακεραιότητα του μηνύματος. Στην υλοποίηση τους, χρησιμοποιείται το μTESLA για να εξασφαλιστεί η αυθεντικότητα και η εμπιστευτικότητα των μηνυμάτων. Χρησιμοποιώντας το μTESLA, το TRANS μπορεί να εξασφαλίσει ότι ένα μήνυμα στέλνεται κατά μήκος ενός μονοπατιού που αποτελείται από αυθεντικοποιημένους κόμβους. Η στρατηγική για τον σταθμό βάσης είναι να μεταδώσει σε όλους τους κόμβους ένα κρυπτογραφημένο μήνυμα σε όλους τους γειτονικούς κόμβους τους. Μόνο οι γειτονικοί κόμβοι οι οποίοι είναι εμπιστευμένοι μπορούν να αποκρυπτογραφήσουν το μήνυμα με το δημόσιο κλειδί. Στην συνέχεια ο κάθε κόμβος εισάγει στο μήνυμα την θέση του, κρυπτογραφεί το νέο μήνυα με το κοινό κλειδί και διαβιβάζει το μήνυμα στον γείτονα που βρίσκεται πλησιέστερα στον προορισμό. Όταν το μήνυμα φτάσει στον προορισμό ο παραλήπτης είναι σε θέση να επικυρώσει την πηγή (σταθμός βάσης) χρησιμοποιώντας την MAC που αντιστοιχεί στον σταθμό βάσης. Για να βεβαιωθεί η λήψη ή να απαντηθεί το μήνυμα ο παραλήπτης μπορεί να στείλει ένα νέο μήνυμα (acknowledgement) κατά μήκος του εμπιστευμένου μονοπατιού από το οποίο παραλήφθηκε.

Μια ιδιαίτερη πρόκληση για την ασφαλή δρομολόγηση στα ασύρματα δίκτυα αισθητήρων είναι ότι είναι πολύ απλό για έναν μεμονωμένο κόμβο να αναστατώσει ολόκληρο το πρωτόκολλο δρομολόγησης με την απλή τροποποίηση των πινάκων δρομολόγησης. Ο Papadimitratos και ο Haas πρότειναν ένα ασφαλές πρωτόκολλο για την ανακάλυψη των μονοπατιών και την δημιουργία των πινάκων δρομολόγησης που εγγυάται ότι θα ληφθούν από τον σταθμό βάσης οι σωστές πληροφορίες δρομολόγησης. Αυτό το σενάριο είναι παρόμοιο με το προαναφερθέν πρωτόκολλο TRANS. Η ασφάλεια στηρίζεται στο MAC (message authentication code) και σε μια συσσώρευση από τις ταυτότητες των κόμβων που βρίσκονται κατά μήκος του μονοπατιού που ταξίδεψε το μήνυμα. Με την παραπάνω τεχνική μια πηγή μπορεί να ανακαλύψει την τοπολογία του δικτύου καθώς κάθε κόμβος που παραλαμβάνει το μήνυμα προσθέτει την ταυτότητα του και το επαναπροωθεί. Για να εξασφαλιστεί ότι το μήνυμα δεν έχει τροποποιηθεί κατασκευάζεται μια MAC και έτσι μπορεί να ελεγχθεί τόσο στον προορισμό όσο και στην πηγή (για το μήνυμα που επιστρέφεται από τον παραλήπτη).

Ένα παρόμοιο πρόβλημα με τα παραπάνω είναι τα wormholes σε ένα δίκτυο αισθητήρων. Σε μια wormhole επίθεση ο επιτιθέμενος ακούει μη εξουσιοδοτημένα ένα η και

παραπάνω πακέτα, τα προωθεί κρυφά σε άλλους κακόβουλους κόμβους και στην συνέχεια επαναπροωθεί τα πακέτα. Αυτό μπορεί να γίνει για να τροποποιηθεί η απόσταση ανάμεσα στους δυο συνεργαζόμενους κόμβους. Μπορεί επίσης να χρησιμοποιηθεί σε μια πιο γενική αναστάτωση του πρωτοκόλλου δρομολόγησης με την παραπλάνηση της διαδικασίας εντοπισμού των γειτονικών κόμβων.

### 2.5.4 ΑΣΦΑΛΕΙΑ ΕΝΑΝΤΙΟΝ ΤΗΣ ΕΠΙΘΕΣΗΣ SYBIL

Για την υπεράσπιση ενάντια στην επίθεση Sybil, που περιγράφεται σε προηγούμενη υποενότητα (2.4.3), το δίκτυο χρειάζεται ένα μηχανισμό που θα πιστοποιήσει ότι κάθε κόμβος κατέχει και χρησιμοποιεί μόνο ένα προσδιοριστικό. Ο Newsome κλπ περιγράφουν δυο μεθόδους για να επικυρώσουν τα διάφορα προσδιοριστικά των κόμβων αισθητήρων: την άμεση επικύρωση και την έμμεση επικύρωση. Στην άμεση επικύρωση ένας εμπιστευμένος κόμβος εξετάζει άμεσα κατά πόσο το προσδιοριστικό του κόμβου που θέλει να εισαχθεί στο δίκτυο ισχύει. Στην έμμεση επικύρωση ένας άλλος εμπιστευμένος κόμβος επιτρέπεται να αποδείξει την εγκυρότητα ενός κόμβου. Ο Newsome κλπ, επίσης, περιγράφουν τις τεχνικές για άμεση επικύρωση, συμπεριλαμβανομένου μιας ασύρματης δοκιμής των πόρων (radio resource test). Σε αυτή την, κάθε κόμβος ορίζει ένα ξεχωριστό κανάλι επικοινωνίας για κάθε γείτονά του. Έπειτα επιλέγει ένα κανάλι και ακούει. Εάν ο κόμβος ανιχνεύσει μια μετάδοση πάνω στο κανάλι υποτίθεται ότι ο κόμβος που διαβιβάζει στο κανάλι είναι ένας φυσικός κόμβος και γείτονάς του. Ομοίως εάν ο κόμβος δεν ανιχνεύσει μια μετάδοση στο συγκεκριμένο κανάλι, ο κόμβος υποθέτει ότι το προσδιοριστικό που έχει οριστεί για το κανάλι δεν είναι φυσικό.

Μια άλλη τεχνική για την υπεράσπιση ενάντια στην επίθεση Sybil είναι να χρησιμοποιηθεί τεχνικές τυχαίας προ-διανομής κλειδιών. Η ιδέα πίσω από αυτήν την τεχνική είναι ότι με ένα συγκεκριμένο αριθμών κλειδιών, ένας κόμβος που παράγει τυχαία προσδιοριστικά κόμβων δεν κατέχει αρκετά κλειδιά για να υπερισχύσει και έτσι θα είναι αδύνατον να ανταλλάξει πολλά μηνύματα καθώς ο κόμβος με την μη έγκυρη ταυτότητα δεν θα μπορεί να κρυπτογραφεί και αποκρυπτογραφεί τα δεδομένα.

### 2.5.5 ΑΣΦΑΛΕΙΑ ΕΝΑΝΤΙΑ ΤΩΝ ΕΠΙΘΕΣΕΩΝ ΑΝΑΛΥΣΗΣ ΚΙΝΗΣΗΣ

Οι επιθέσεις ανάλυσης κυκλοφορίας που περιγράφονται σε προηγούμενη υποενότητα είναι δυνατόν να καταπολεμηθούν χρησιμοποιώντας ορισμένες τεχνικές. Ο Deng κλπ, προτείνουν την χρησιμοποίηση μιας τεχνικής κατά την οποία κάθε κόμβος διαβιβάζει περιστασιακά ένα πακέτο σε έναν κόμβο εκτός του γονέα του [12]. Αυτό θα καθιστούσε δύσκολο να διακριθεί ένα σαφές μονοπάτι από τον κόμβο στον σταθμό βάσης με αποτέλεσμα να μετριαστεί η επίθεση ελέγχου της ροής. Παρόλα αυτά το σύστημα θα ήταν ακόμα ευάλωτο στην επίθεση χρονικού συσχετισμού. Για το λόγο αυτό ο Deng κλπ προτείνουν μια στρατηγική διάδοσης κατά την οποία ένας κόμβος (με μια ορισμένη πιθανότητα) παράγει ένα πλαστό πακέτο όταν ένας γείτονας του προωθεί ένα πακέτο στον σταθμό βάσης. Το πλαστό πακέτο στέλνεται τυχαία σε έναν άλλο κόμβο ο οποίος μπορεί επίσης να παράγει ένα παρόμοιο πλαστό πακέτο. Τα πακέτα αυτά χρησιμοποιούν ουσιαστικά ένα χρονικό όριο για αποστολή (time to live – TTL) για να αποφασιστεί πότε η αποστολή πρέπει να σταματήσει. Αυτό αποκρύπτει αποτελεσματικά τον σταθμό βάσης από επιθέσεις χρονικού συσχετισμού.

### 2.5.6 ΑΣΦΑΛΕΙΑ ΣΤΟ ΦΥΣΙΚΟ ΕΠΙΠΕΔΟ

Οι επιθέσεις στο φυσικό επίπεδο, όπως υποστηρίξαμε στην αρχή του κεφαλαίου, θέτουν μια από τις μεγαλύτερες απειλές στα ασύρματα δίκτυα αισθητήρων. Οι κόμβοι αισθητήρων μπορεί να είναι εξοπλισμένοι με hardware για να προστατευτούν οι διάφορες επιθέσεις. Για παράδειγμα, για την προστασία από τις φυσικές παρεμβάσεις ένα είδος άμυνας είναι να υπάρχει κάποιο υλικό (tamper – proofing) το οποίο θα αποδεικνύει αν έχει όντως παραβιαστεί ο αισθητήρας [9]. Η βιβλιογραφία εστιάζει στην δημιουργία hardware το οποίο θα είναι ανθεκτικό σε προσπάθειες παραβίασης και έτσι ώστε να περιεχόμενο της μνήμης να είναι απρόσιτο σε κάθε είδους επίθεσης. Ένας άλλος τρόπος είναι να σχεδιαστεί ειδικό software το οποίο θα εντοπίζει τις όποιες παραβιάσεις.

Δεδομένου του ότι το κόστος του hardware γίνεται φθηνότερο, υλικό το οποίο θα μπορεί να αποτρέψει τις παραβιάσεις μπορεί να χρησιμοποιηθεί σε επεκτάσεις δικτύων ευκολότερα. Μια πιθανή προσέγγιση για να προστατευτούν οι αισθητήρες είναι ένα είδος

αυτό-τερματισμού. Η βασική ιδέα είναι ο κόμβος να βγαίνει εκτός λειτουργίας από μόνος του, αφού έχει καταστρέψει τα κλειδιά και τα άλλα ευαίσθητα δεδομένα, όταν υπάρχει περίπτωση για κάποια επίθεση. Αυτό είναι ιδιαίτερα εφικτό στα μεγάλα δίκτυα με επαν-αποστολή και σε περιπτώσεις τις οποίες το κόστος του αισθητήρα είναι μικρότερο από το κόστος των συνεπειών από μια επίθεση με αρνητικά αποτελέσματα. Το κλειδί για αυτή την τεχνική είναι να ανιχνευτεί αποτελεσματικά η μελλοντική επίθεση. Μια απλή λύση είναι να ελέγχονται περιοδικά οι γειτονικοί κόμβοι έχοντας υπόψη μια στατιστική μελέτη επέκτασης. Το παραπάνω όπως γίνεται φανερό αποτελεί ένα πρόβλημα για δίκτυα στα οποία οι κόμβοι μετακινούνται στο φυσικό επίπεδο.

Στα [3, 4, 43], οι συντάκτες περιγράφουν τεχνικές για την προστασία του λογισμικού οι οποίες εξάγονται από επεξεργαστές με έξυπνες κάρτες. Αυτές περιλαμβάνουν χειρισμό δεσμών ιόντων, κοπή με λέιζερ, ανάλυση ισχύς, έλεγχο για απότομη μεταβολή τάσης οι περισσότερες από τις οποίες αποτελούν επιθέσεις στο φυσικό επίπεδο. Με βάση μια ανάλυση των παραπάνω επιθέσεων, ο Andersen κλπ παραθέτει κάποια παραδείγματα χαμηλού κόστους μέτρων που κάνουν τέτοιου είδους επιθέσεις ακόμα δυσκολότερες [29]:

- Τυχαίο σήμα ρολογιών (randomized clock signal): Εισάγονται τυχαίες καθυστερήσεις μεταξύ οποιονδήποτε αντιδράσεων των αισθητήρων και ανάμεσα σε δυο διαδοχικές κρίσιμες διαδικασίες.
- Τυχαία χρήση πολλών νημάτων (randomized multithreading): Σχεδιασμός ενός multithread επεξεργαστή σύμφωνα με την αρχιτεκτονική του οποίου γίνεται τυχαία εκτέλεση των νημάτων σε αυτόν.
- Ισχυρός αισθητήρας χαμηλής συχνότητας: Οικοδόμηση ενός συστήματος για αυτό-έλεγχο. Κάθε προσπάθεια παραβίασης του αισθητήρα θα έχει σαν αποτέλεσμα την δυσλειτουργία ολόκληρου του επεξεργαστή.
- Καταστροφή του κυκλώματος δοκιμής: Καταστροφή η τερματισμός ενός ειδικού κυκλώματος δοκιμής με αποτέλεσμα να αποτραπεί η παραβίαση από τους επιτιθέμενους.
- Περιορισμός μετρητή προγράμματος: Αποφυγή χρησιμοποίησης ενός μετρητή προγράμματος ο οποίος θα μπορεί να τρέξει σε ολόκληρο το διάστημα διευθύνσεων.

- Πλέγματα αισθητήρων σε υψηλό επίπεδο: Εισάγονται πρόσθετα στρώματα μετάλλων διαμορφώνοντας ένα πλέγμα πάνω από το πραγματικό κύκλωμα στο οποίο δεν διακινούνται κρίσιμα σήματα με αποτέλεσα να απωθούνται οι επιτιθέμενοι.

Για την επέκταση των εξαρτημάτων έξω από τον αισθητήρα έχουν προταθεί διάφορες προσεγγίσεις και συνοψίζονται στο [30]. Ο Sastry κλπ [35], [εισάγει την έννοια της ασφαλούς επαλήθευσης και προτείνετε ένα ασφαλές σχέδιο εντοπισμού, το πρωτόκολλο ECHO. Στην εργασία τους η ασφάλεια στηρίζεται στις ιδιότητες του ήχου και στη διάδοση του σήματος RF. Ένας αντίπαλος δεν μπορεί να εξαπατήσει και να απαιτήσει μια πιο σύντομη διαδρομή στέλνοντας την απάντηση νωρίτερα επειδή δεν θα έχει το nonce. Ο Hu κλπ, [33] εισάγει τις ομοιοκατευθυντικές κεραίες για την άμυνα ενάντια σε επιθέσεις wormhole. Στο [17] οι συντάκτες μελετούν την διαμόρφωση και την υπεράσπιση των δικτύων αισθητήρων ενάντια σε επιθέσεις στο φυσικό επίπεδο που βασίζονται στην αναζήτηση. Καθορίζουν μια επίθεση βασισμένη στην αναζήτηση στο φυσικό επίπεδο, όπου ο επιτιθέμενος κινείται ανάμεσα στους αισθητήρες και ανιχνεύουν τον εξοπλισμό για να εντοπιστούν οι ενεργοί αισθητήρες με αποτέλεσμα την καταστροφή τους. Σε μια προγενέστερη εργασία έχουν προσδιορίσει και διαμορφώσει τυφλές επιθέσεις στο φυσικό επίπεδο [31]. Ο αλγόριθμος για την ασφάλεια εκτελείται ατομικά από κάθε αισθητήρα σε δυο φάσεις. Στην πρώτη φάση, οι αισθητήρες ανιχνεύουν τον επιτιθέμενο και στέλνουν τα μηνύματα ανακοίνωσης σε άλλους αισθητήρες. Στην δεύτερη φάση, οι κόμβοι που λαμβάνουν τα μηνύματα ανακοίνωσης προσδιορίζουν την κατάστασή τους ως προς αλλαγή. Ένας μηχανισμός, που ονομάζεται SWATT, ελέγχει πότε η μνήμη ενός αισθητήρα έχει αλλάξει [32] προτείνεται από τον Seshadri κλπ.

## 2.6 ΣΥΜΠΕΡΑΣΜΑΤΑ

Στα παραπάνω κεφάλαια μελετήσαμε τα τέσσερα βασικότερα σημεία που αφορούν την ασφάλεια των δικτύων αισθητήρων: τα εμπόδια, οι προϋποθέσεις, οι επιθέσεις και η αμυντική προστασία. Ο στόχος μας για αυτό το μέρος είναι να παρέχουμε μια γενική επισκόπηση του ευρύ τομέα της ασφάλειας των ασύρματων δικτύων αισθητήρων και παραθέτουμε τις κύριες παραπομπές για την παραπάνω μελέτη αυτών των θεμάτων.

Δεδομένου ότι τα ασύρματα δίκτυα αισθητήρων συνεχίζουν να αυξάνονται και να γίνονται πιο κοινά, αναμένουμε ότι οι προσδοκίες για ασφάλεια θα γίνουν πραγματικότητα σε εφαρμογές των δικτύων. Ειδικότερα η εισαγωγή του δημόσιου κλειδιού και η ασύμετρη κρυπτογραφία που περιγράψαμε παραπάνω θα καταστήσουν την ασφάλεια στα δίκτυα αισθητήρων μια ρεαλιστικότερη προσδοκία. Επίσης αναμένουμε ότι η τρέχουσα και η μελλοντική έρευνα όσο αφορά την εμπιστευτικότητα και την εμπιστοσύνη θα καταστήσει τα δίκτυα αισθητήρων μια ελκυστικότερη επιλογή.

Στην επόμενη ενότητα της εργασίας ασχολούμαστε με ένα αλγόριθμο κρυπτογράφησης, τον tiny encryption algorithm (TEA). Αρχικά αναλύουμε τον αλγόριθμο και στην συνέχεια παραθέτουμε τα κυριότερα σημεία της υλοποίησης του.

# 3 TINY ENCRYPTION ALGORITHM

## 3.1 ΕΙΣΑΓΩΓΗ

Η επιλογή ενός αλγόριθμου κρυπτογράφησης για τα ασύρματα δίκτυα αισθητήρων εξαρτάται, κατά κύριο λόγο, από την ενέργεια που καταναλώνει το υποσύστημα επεξεργασίας του κόμβου στην ενεργή περιοχή λειτουργίας του, τη συχνότητας λειτουργίας του, και τον αριθμό των κύκλων που απαιτούνται από αυτό για τον υπολογισμό του αλγορίθμου. Για το λόγο αυτό, επιλέχθηκε ο αλγόριθμος Tiny Encryption Algorithm, ο οποίος διακρίνεται για τη χαμηλή κατανάλωση ενέργειας που τον χαρακτηρίζει λόγω απλότητας.

## 3.2 ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ

Ο αλγόριθμος κρυπτογράφησης Tiny Encryption Algorithm (TEA) σχεδιάστηκε το 1994 από τους David Wheeler και Roger Needham στο Cambridge [34]. Χαρακτηρίζεται από την απλότητα του. Χρησιμοποιεί ένα μεγάλο αριθμό επαναλήψεων παρά ένα μεγάλο και περίπλοκο πρόγραμμα. Μπορεί εύκολα να μεταφραστεί και υλοποιηθεί στις περισσότερες γλώσσες προγραμματισμού. Το αρχικό πρόγραμμα φαίνεται παρακάτω. Χρησιμοποιείται μια απλή και γρήγορη αρχικοποίηση και κάνει μια αδύναμη μη γραμμική επανάληψη για αρκετούς γύρους έτσι ώστε να γίνει ασφαλές. Δεν υπάρχουν προεγκατεστημένοι πίνακες ούτε μεγάλοι χρόνοι αρχικοποίησης. Χρησιμοποιεί 32 bit ψηφιολέξεις.

## 3.3 ΡΟΥΤΙΝΑ ΚΡΥΠΤΟΓΡΑΦΗΣΗΣ

Παρακάτω παραθέτουμε τον βασικό αλγόριθμο κρυπτογράφησης για την κρυπτογράφηση των δεδομένων στα διανύσματα v[0] και v[1] με την χρήση των κλειδιών k[0] έως k[3].

## 3.4 ΒΑΣΙΚΟΤΕΡΑ ΣΗΜΕΙΑ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ

Είναι μια ρουτίνα τύπου Feistel αν και χρησιμοποιούνται οι πράξεις πρόσθεσης και αφαίρεσης για την αντιστροφή παρά η πράξη XOR. Η ρουτίνα στηρίζεται στην εναλλακτική των πράξεων XOR και ADD για να παρέχει μη γραμμικότητα. Μια διπλή μετατόπιση των δεδομένων (dual shift) έχει σαν αποτέλεσμα την επανειλημμένη ανάμειξη τους.

Ο αριθμός των κύκλων πριν η αλλαγή ενός bit των δεδομένων ή του κλειδιού φτάσει στο 32 είναι το περισσότερο έξι, έτσι οι δεκαέξι κύκλοι επαρκούν και προτείνονται 32.

Το κλειδί αρχικοποιείται στα 128 bits καθώς αυτό το μέγεθος είναι αρκετό για να επιτευχθεί η ασφάλεια του εναντίον των απλών επιθέσεων αναζήτησης.

```
void code(long* v, long* k)
{
unsigned long y=v[0],z=v[1], sum=0,                    /* set up */
delta=0x9e3779b9,                    /* a key schedule constant */
n=32 ;
while (n-->0)
{                                                  /* basic cycle start */
sum += delta ;
y += ((z<<4)+k[0]) ^ (z+sum) ^ ((z>>5)+k[1]) ;
z += ((y<<4)+k[2]) ^ (y+sum) ^ ((y>>5)+k[3]) ;
}                                                      /* end cycle */
v[0]=y ; v[1]=z ;
}
```

**Πίνακας 3:** Ρουτίνα κρυπτογράφησης αλγορίθμου TEA

Τα πέντε περισσότερα σημαντικά και τα τέσσερα λιγότερο σημαντικά bits είναι ελαφρώς πιο ασθενή από τα bit που βρίσκονται στην μέση. Τα bit αυτά παράγονται από μόνο δυο εκδόσεις του z (η του y) συν το άλλο y ή το z. Κατά συνέπεια ο ρυθμός σύγκλισης ακόμα και της διάχυσης είναι πιο αργό. Εντούτοις, η μετατόπιση εξισώνει το παραπάνω με πιθανώς μια καθυστέρηση της τάξης του ενός ή των δυο παραπάνω κύκλων.

Όσο αφορά τα κλειδιά χρησιμοποιείται πρόσθεση και εφαρμόζεται στο μη μετατοπισμένο z. Σε κάποια άλλα παραδείγματα τα διανύσματα με τα κλειδιά k[0] κλπ,

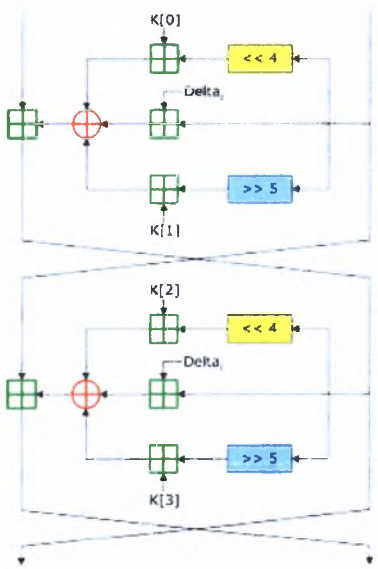
αλλάζουν από την πρόσθεση, αλλά η παραπάνω τεχνική είναι απλούστερη και το ίδιο αποτελεσματική. Ο αριθμός delta, παράγεται από τον "χρυσό" αριθμό όπου:

$$\mathrm{delta} = (\sqrt{5} - 1)2^{31}$$

Κάθε φορά χρησιμοποιείται ένα διαφορετικό πολλαπλάσιο του delta έτσι ώστε κανένα bit του δεν θα αλλάζει συχνά. Η αποτελεσματικότητα του αλγορίθμου δεν έχει να κάνει με τον αριθμό delta αλλά χρειάζεται να αποφεύγονται οι κακές τιμές.

Η χρήση του πολλαπλασιασμού αναμειγνύει τα δεδομένα αποτελεσματικά αλλά χρειάζονται επίσης και οι μετατοπίσεις.

Ο αλγόριθμος μπορεί εύκολα να μεταφραστεί σε κώδικα assembly από την στιγμή που το XOR είναι μια πράξη. Η κατασκευή του hardware δεν είναι δύσκολη και έχει την ίδια πολυπλοκότητα σε σχέση με τον αλγόριθμο DES, αφού και οι δύο χρησιμοποιούν ένα κλειδί μεγέθους 128 bits (double length).




**Εικόνα 4:** Δυο κύκλοι Feistel (one cycle) του TEA

```
void decode(long* v, long* k)
{
unsigned long n=32, sum, y=v[0], z=v[1],
delta=0x9e3779b9 ;
sum=delta<<5 ;
                                                    /* start cycle */
while (n-->0)
{
z-= ((y<<4)+k[2]) ^ (y+sum) ^ ((y>>5)+k[3]) ;
y-= ((z<<4)+k[0]) ^ (z+sum) ^ ((z>>5)+k[1]) ;
sum-=delta ;
}                                                    /* end cycle */
v[0]=y ; v[1]=z ; }
```

**Πίνακας 4:** Ρουτίνα αποκρυπτογράφησης αλγορίθμου TEA

## 3.5 ΣΗΜΕΙΩΣΕΙΣ ΓΙΑ ΤΗΝ ΥΛΟΠΟΙΗΣΗ

Υπάρχει η δυνατότητα ο κώδικας να γίνει μικρότερος αλλά και ταχύτερος, αλλά η παραπάνω ρουτίνα είναι ευκολότερη στην υλοποίηση και στην απομνημόνευση.

Μια απλή βελτίωση είναι να αντιγραφούν τα k[0] – k[3] στα a, b, c, d πριν την επανάληψη έτσι ώστε η τοποθέτηση των δεικτών να γίνεται εκτός του πεδίου των επαναλήψεων. Σε μια υλοποίηση αυτό έχει σαν αποτέλεσμα την μείωση του χρόνου στο 1/6.

## 3.6 ΣΥΜΠΕΡΑΣΜΑΤΑ

Ο Tiny Encryption Algorithm (TEA) είναι ένας αλγόριθμος ο οποίος μπορεί να υλοποιηθεί εύκολα στις περισσότερες γλώσσες προγραμματισμού. Στην δικιά μας περίπτωση η απλότητα του αλγορίθμου μας οδηγεί στον συμπέρασμα ότι είναι ιδιαίτερα κατάλληλος για τα ασύρματα δίκτυα αισθητήρων καθώς είναι αρκετά μικρός αλγόριθμος και δεν έχει υπερβολικές απαιτήσεις σε μνήμη ή επεξεργαστική ισχύ. Στο τομέα της ασφάλειας φαίνεται να είναι αρκετά ασφαλής λόγω του μεγάλου αριθμού των επαναλήψεων κατά την κρυπτογράφηση αλλά και του μεγάλου μεγέθους του κλειδιού.

Λευκή σελίδα

# ΜΕΡΟΣ 2 – ΕΦΑΡΜΟΓΗ

# 4 ΥΛΟΠΟΙΗΣΗ

## 4.1 ΕΡΓΑΛΕΙΑ ΠΡΟΣΟΜΟΙΩΣΗΣ

### 4.1.1 TINYOS ΚΑΙ Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ NESC

Το TinyOS [37] είναι ένα λειτουργικό σύστημα που σχεδιάστηκε για δίκτυα του τύπου των ασύρματων δικτύων αισθητήρων. Χρησιμοποιεί το μοντέλο του component based προγραμματισμού αφού είναι γραμμένο σε NesC [36,38]. Η NesC είναι μια διάλεκτος της γλώσσας προγραμματισμού C βελτιστοποιημένη για τις απαιτήσεις μνήμης και υπολογιστικής ισχύς των δικτύων αισθητήρων. Υπάρχουν πολλά συμπληρωματικά προγράμματα και εργαλεία τα οποία είναι υλοποιημένα κυρίως σε Java και shell scripts. Επίσης ο NesC compiler είναι γραμμένος σε γλώσσα C.

Ένα πρόγραμμα TinyOS αποτελείται από ένα γράφο από components, καθένα από τα οποία είναι μια ανεξάρτητη υπολογιστική οντότητα. Ο πρωταρχικός στόχος κατά την δημιουργία του λειτουργικού ήταν να επιτραπεί στους σχεδιαστές και προγραμματιστές εφαρμογών η εύκολη χρήση μικρών κομματιών κώδικα (component) για την δημιουργία ολοκληρωμένων και σύγχρονων συστημάτων. Η χρήση του μοντέλου εξαρτημάτων (component model) έχει σαν αποτέλεσμα την επίτευξη του πρωταρχικού στόχου.

Τα components περιέχουν τρία είδη συναρτήσεων· εντολές (commands), γεγονότα (events), και εργασίες (tasks). Οι εντολές και τα γεγονότα είναι μηχανισμοί επικοινωνίας μεταξύ των components, ενώ τα tasks χρησιμοποιούνται για το συγχρονισμό κώδικα που εκτελείται στο εσωτερικό των components.

Κάθε εφαρμογή σε NesC προϋποθέτει μια διασύνδεση μεταξύ των components που χρησιμοποιούνται έτσι ώστε να γίνεται η επικοινωνία μεταξύ τους και να φαίνεται η ροή των γεγονότων (events). Η διασύνδεση αυτή είναι το λεγόμενο wiring specification και δεν έχει σχέση με τα components που χρησιμοποιούνται κάθε φορά. Επίσης υπάρχουν διπλής κατεύθυνσης συνδέσεις μεταξύ των components που ονομάζονται interfaces.

Οι εντολές (commands) χρησιμοποιούνται για την εκτέλεση κάποιας λειτουργίας από ένα component, όπως ο τερματισμός της λειτουργίας ενός αισθητήρα. Η ολοκλήρωση της

εντολής μπορεί να δηλώνεται από ένα event. Επίσης τα event μπορούν να δηλώνουν κάποια άφιξη ενός μηνύματος, γεγονότα δηλαδή που προκαλούνται ασύγχρονα.

Ένα task, που ουσιαστικά είναι μια ρουτίνα που εκτελείται από τον scheduler αργότερα, μπορεί επίσης να εκτελείται από κάποια εντολή ή ακόμα και από κάποιο event. Αυτό έχει σαν αποτέλεσμα στις περισσότερες των περιπτώσεων να χρησιμοποιούνται τα events και τα commands για την εκτέλεση σύντομων λειτουργιών και τα tasks για πιο εκτεταμένους υπολογισμούς. Τα tasks, όπως αναφέραμε, εκτελούνται από τον scheduler ο οποίος είναι μια FIFO στοίβα και δεν υπάρχει περίπτωση κάποιο να λιμοκτονήσει καθώς δεν επιτρέπεται το προσπέρασμα από κάποιο άλλο που είναι στην στοίβα. Ο μόνος τρόπος να σταματήσει να εκτελείται ένα task είναι ένα event.

### 4.1.2 AVRORA

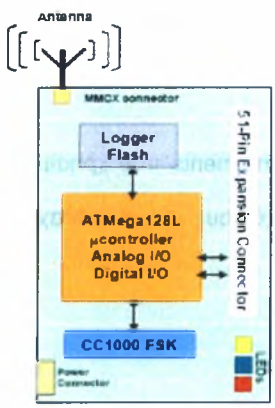
Το Avrora [38] είναι ένα λογισμικό ανοιχτού κώδικα και χρησιμοποιήθηκε για τις μετρήσεις κατανάλωσης ενέργειας. Κατασκευάστηκε στο πανεπιστήμιο UCLA και παρέχει τη δυνατότητα προσομοίωσης και ανάλυσης προγραμμάτων, γραμμένων για AVR μικροεπεξεργαστές της Atmel και αισθητήρες Mica2.

Η ακρίβεια των αποτελεσμάτων, η ευκολία χρήσης και η επεκτασιμότητα το καθιστούν ως ένα από τα πιο πολύ χρησιμοποιούμενα προγράμματα προσομοίωσης. Τα παραπάνω επιτυγχάνονται με την προσομοίωση πραγματικού κώδικα επεξεργαστή σε επίπεδο κύκλων CPU (κώδικα μηχανής) έναντι της προσομοίωσης μοντέλων δικτύου που χρησιμοποιείται από παρόμοια λογισμικά (TOSSIM, ATEMU). Η υλοποίηση του AVRORA έχει γίνει σε γλώσσα Java, πράγμα που ενισχύει τη φορητότητα (λειτουργικού συστήματος και γλώσσας υλοποίησης) και την ευελιξία σε σχέση με τα προαναφερόμενα λογισμικά που είναι γραμμένα σε κώδικα C.

## 4.2 ΠΛΑΤΦΟΡΜΑ Crossbow MICA2

Ο αλγόριθμος έχει προσομοιωθεί σε συσκευές hardware, και συγκεκριμένα με την πλατφόρμα mica/mica2dot της εταιρίας Crossbow (εικόνα 5.1, 5.2). Η πλατφόρμα mica/mica2dot χρησιμοποιείται από πολλά ιδρύματα και ερευνητικές ομάδες σε ολόκληρο τον κόσμο. Τα βασικότερα χαρακτηριστικά τους είναι:

- 4Kb RAM
- 128 Kb program flash memory
- Atmel ATmega128L μικροεπεξεργαστής
- 10 bit ADC
- 3 LEDs
- 4Kb EEPROM
- 18 γρ. συνολικό βάρος
- 512 Kb serial flash για τις μετρήσεις
- 2 A4 μπαταρίες




**Εικόνα 5.1** Διάγραμμα MPR400CB.

**Εικόνα 5.1** Mica2 Mote.

Επίσης υπάρχει ενσωματωμένο ένα 51-pin connector για την σύνδεση με ένα πλήθος περιφερειακών. Περισσότερα χαρακτηριστικά της πλατφόρμας αναγράφονται στο παράρτημα Α.

## 4.3 ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ ΚΑΙ ΜΕΘΟΔΟΛΟΓΙΑ

### 4.3.1 ΥΠΟΘΕΣΗ ΕΡΓΑΣΙΑΣ

Το πρωτόκολλο υλοποιήθηκε για ένα δίκτυο το οποίο αποτελείται από ένα σταθμό βάσης (BS) συνδεδεμένο με την σειριακή θύρα (UART) ενός Η/Υ και 2 πελάτες οι οποίοι συλλέγουν τιμές τις οποίες στην συνέχεια κρυπτογραφούν και τις στέλνουν στον σταθμό βάσης.


**Εικόνα 6:** Υπόθεση εργασίας

Στην συνέχεια ο σταθμός βάσης αποκρυπτογραφεί τις μετρήσεις και με την βοήθεια μιας JAVA εφαρμογής (Oscilloscope) προβάλει τις τιμές που έχουν ληφθεί από τον ADC στην οθόνη συναρτήσει του χρόνου.

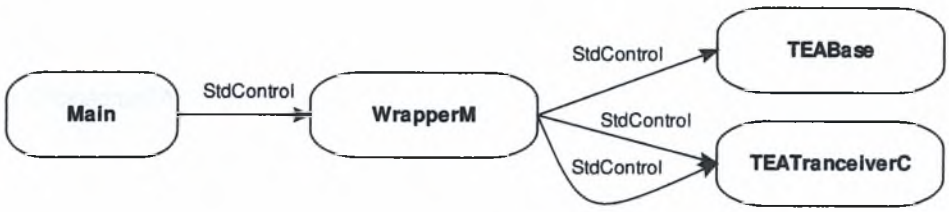
### 4.3.2 ΠΕΡΙΓΡΑΦΗ ΤΩΝ COMPONENTS

Παρακάτω γίνεται μια σύντομη περιγραφή των components που χρησιμοποιήθηκαν αλλά και ο τρόπος που έγινε η διασύνδεση (wiring) μεταξύ τους για να παραχθεί η τελική εφαρμογή.

#### 4.3.2.1 WrapperC

Το αρχείο *WrapperC (configuration)*, συνδέει τις εφαρμογές *TEABase* και *TEATransceiver*. Χρησιμοποιείται μόνο για τις ανάγκες της προσομοίωσης σε περιβάλλον TOSSIM και Avrora και παράγεται ένα αυτόνομο εκτελέσιμο main.exe. Ειδικότερα, στο αρχείο συνδέεται το Main.StdControl της εφαρμογής με το WrapperM (module). Όπως φαίνεται και στο εικόνα 7, το *WrapperM* με τη σειρά του συνδέει τα `StdControl` του σταθμού βάσης

*TEABase* και του αρχείου κώδικα των clients *TEATranceiverC*, τα οποία περιγράφονται στις επόμενες παραγράφους.

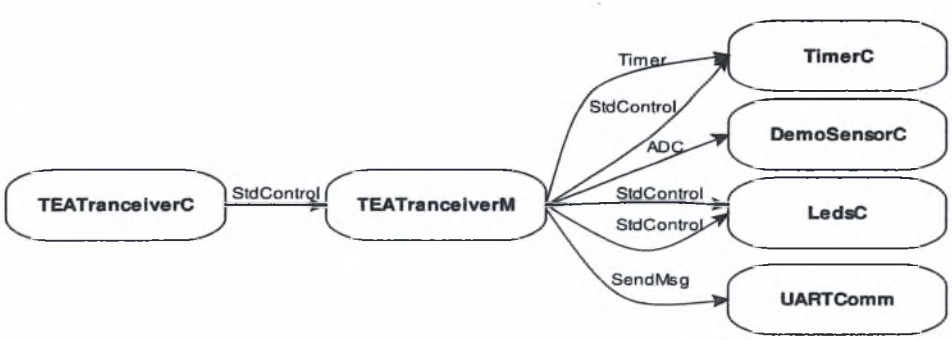



**Εικόνα 7:** Διάγραμμα ροής για τη σύνδεση των εφαρμογών TEABase και TEATransceiverC.

4.3.2.2 TEATranceiverC

Στο configuration αρχείο TEATranceiverC γίνεται το wiring των components που χρησιμοποιούνται από τους clients. Στην εικόνα 8 παρατηρούμε ότι στο module TEATranceiverM υλοποιείται το interface StdControl που απαιτείται για αρχικοποίηση, τερματισμό και έναρξη της εφαρμογής.. Τα κυριότερα components που χρησιμοποιήθηκαν είναι τα εξής:

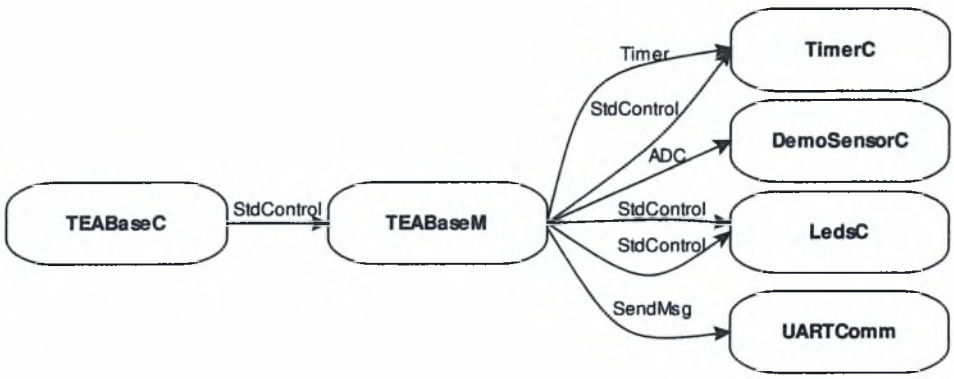
- TimerC: Ο χρονιστής του προγράμματος, σύμφωνα με τον οποίο λαμβάνονται οι τιμές από τον ADC
- DemoSensorC: Παράδειγμα αισθητήρα ο οποίος λαμβάνει τιμές για διάφορα επίπεδα θερμοκρασίας.
- LedsC: Ρυθμίζει τα τρία LEDS που είναι ενσωματωμένα στον αισθητήρα (κόκκινο, πράσινο, κίτρινο).
- UARTComm: χρησιμοποιείται για την αποστολή των κρυπτογραφημένων μηνυμάτων στον σταθμό βάσης.




**Εικόνα 8:** Διάγραμμα ροής για τον client TEATransceiver

#### 4.3.2.3 TEABase

Τέλος, το αρχείο TEABase υλοποιεί το wiring των components που χρησιμοποιεί το module TEABaseM για τον κώδικα του σταθμού βάσης. Τα κύρια Modules Που χρησιμοποιούνται είναι τα ίδια με αυτά που χρησιμοποιούνται και στο TEATranceiverC και περιγράφηκαν παραπάνω.




**Εικόνα 9:** Διάγραμμα ροής για τον client TEABase

# 5. ΕΚΤΙΜΗΣΗ ΑΠΟΔΟΣΗΣ ΠΡΩΤΟΚΟΛΛΟΥ

## 5.1 ΣΗΜΕΙΩΣΕΙΣ

Στο κεφάλαιο αυτό γίνεται η σύγκριση του αλγορίθμου κρυπτογράφησης TEA με άλλες υλοποιημένες εφαρμογές. Η σύγκριση επικεντρώνεται στο μέγεθος του κώδικα καθώς η μνήμη είναι ένας σημαντικός παράγοντας για τα ασύρματα δίκτυα αισθητήρων. Επίσης συγκρίνονται οι χρόνοι εκτέλεσης της κάθε εφαρμογής αφού μεγαλύτεροι χρόνοι εκτέλεσης έχουν σαν αποτέλεσμα την μεγαλύτερη κατανάλωση ενέργειας.

Οι συγκρίσεις των αποτελεσμάτων του αλγορίθμου γίνονται με τα πρωτόκολλα XTEA, SEA, AES, DES και HIGHT. Η παρουσίαση και η ανάλυση των πρωτοκόλλων αυτών είναι εκτός του στόχου της πτυχιακής εργασίας και ο ενδιαφερόμενος αναγνώστης παραπέμπεται στην βιβλιογραφία [40]. Συνοπτικά να αναφέρουμε ότι το XTEA είναι μια άλλη υλοποίηση του TEA με μέγεθος κλειδιού 126 bits και όχι 128 bits. Στον πίνακα 5 παραθέτονται χαρακτηριστικά μεγέθη των παραπάνω αλγορίθμων.

| **Αλγόριθμος** | **AES** | **DES** | **HIGHT** | **SEA** | **TEA** | **XTEA** |
|---|---|---|---|---|---|---|
| **Μέγεθος Block** | 128 | 64 | 64 | 96 | 64 | 64 |
| **Μέγεθος κλειδιού** | 128 | 56 | 128 | 96 | 128 | 126 |
| **Κύκλοι μηχανής** | 10 | 16 | 32 | 141 | 32 | 32 |

**Πίνακας 5:** Χαρακτηριστικά μεγέθη των αλγορίθμων.

Παρατηρούμε ότι το μέγεθος του κρυπτογραφήματος του αλγορίθμου TEA είναι αρκετά μεγάλο σε σχέση με το πρωτόκολλο DESL που δεν προσφέρει σημαντική ασφάλεια.

## 5.2 ΑΠΟΤΕΛΕΣΜΑΤΑ

Στην υποενότητα αυτή παρουσιάζονται τα αποτελέσματα στις μετρήσεις των υλοποιήσεων συμπεριλαμβανομένου και του πρωτοκόλλου TEA που ασχολούμαστε.

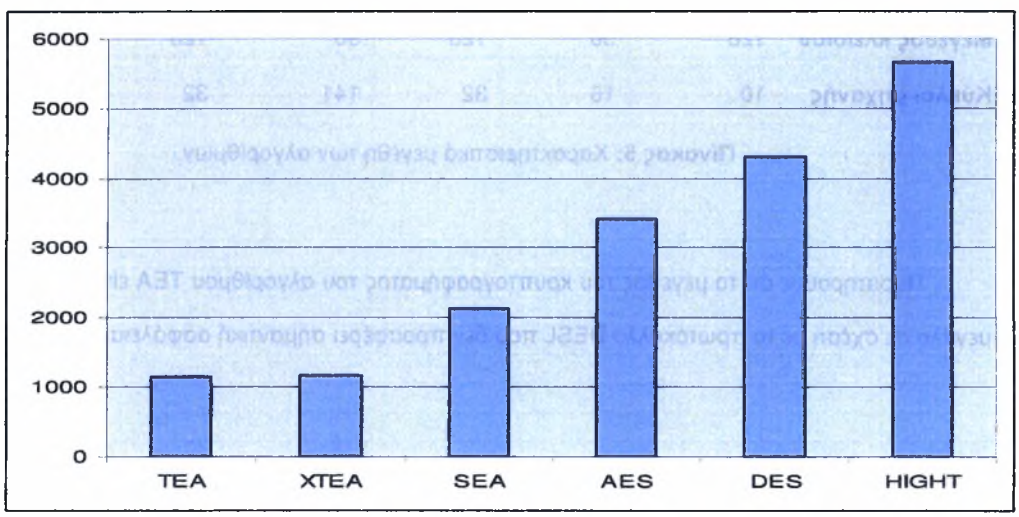
### 5.2.1 ΧΡΗΣΗ ΜΝΗΜΗΣ

Όπως έχουμε αναφέρει και σε προηγούμενες ενότητες τα δίκτυα αισθητήρων και οι αισθητήρες ειδικότερα είναι αρκετά ευαίσθητοι σε μνήμη (Flash memory). Στον παρακάτω πίνακα συνοψίζονται τα αποτελέσματα όσον αφορά την χρησιμοποίηση μνήμης και η εικόνα 10 παρουσιάζει τις τιμές αυτές σε ένα διάγραμμα διατεταγμένες ως προς το μέγεθος.

| **Αλγόριθμος** | **TEA** | **XTEA** | **SEA** | **AES** | **DES** | **HIGHT** |
|---|---|---|---|---|---|---|
| **Μέγεθος κώδικα** | 1140 | 1160 | 2132 | 3410 | 4314 | 5672 |

**Πίνακας 6:** Κατειλημμένος αποθηκευτικός χώρος από τον κώδικα του προγράμματος

Όπως φαίνεται στην παρακάτω εικόνα ο αλγόριθμος TEA παράγει το μικρότερο κρυπτογράφημα ακολουθούμενο από τον XTEA και τον SEA.




**Εικόνα 10:** Μέγεθος κρυπτογραφημάτων

### 5.2.2 ΑΠΟΔΟΣΗ

Στην υποενότητα αυτή παραθέτονται τα αποτελέσματα σχετικά με την απόδοση στην κρυπτογράφηση και την αποκρυπτογράφηση block δεδομένων με τα αντίστοιχα κλειδιά των αλγορίθμων.

Ο πίνακας 7 δείχνει τον αριθμό των κύκλων που χρειάζονται για την κρυπτογράφηση και αποκρυπτογράφηση του κάθε κρυπτογραφήματος.

| **Αλγόριθμος** | **HIGHT** | **AES** | **TEA** | **XTEA** | **DES** | **SEA** |
|---|---|---|---|---|---|---|
| **Κρυπτογράφηση** | 2449 | 3766 | 6271 | 6718 | 8633 | 9654 |
| **Αποκρυπτογράφηση** | 2449 | 4558 | 6299 | 6718 | 8154 | 9654 |

**Πίνακας 7:** Απόδοση της κρυπτογράφηση και αποκρυπτογράφησης σε κύκλους μηχανής.

Παρατηρούμε ότι η υλοποίηση του HIGHT είναι αυτή που χρησιμοποιεί την περισσότερη μνήμη. Παρόλα αυτά σε αυτές τις μετρήσεις φαίνεται ότι επιτυγχάνει τους λιγότερους κύκλους μηχανής για την κρυπτογράφηση και αποκρυπτογράφηση.

Ο πίνακας 8 και ο πίνακας 9 εστιάζουν στην ταχύτητα της κρυπτογράφησης και αποκρυπτογράφησης. Η στήλη 2 στον πίνακα 8 και στον πίνακα 9 εμφανίζουν το μέγεθος κάθε block δεδομένων σε bytes, η στήλη 3 αντιγράφει τον αριθμό κύκλων από τον παραπάνω πίνακα. Η στήλη 4 είναι το πηλίκο των στηλών 3 και 2 και η στήλη 5 δείχνει την ταχύτητα της κρυπτογράφησης και αποκρυπτογράφησης σε κύκλους ανά byte.

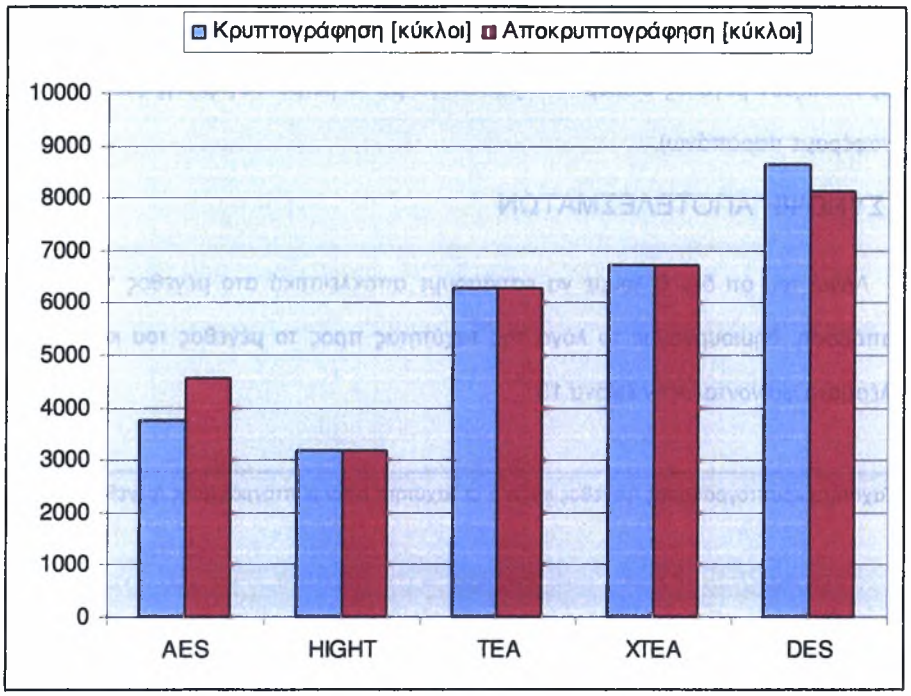
Οι εικόνες 11 και 12 αναπαριστούν τα αποτελέσματα των πινάκων 7, 8 και 9 διατεταγμένα σε κύκλους μηχανής και σε ταχύτητα αντίστοιχα.

| Αλγόριθμος | Μέγεθος block [bit] | Κρυπτογράφηση [κύκλοι] | Κρυπτογράφηση [κύκλοι/bit] | Ταχύτητα [bit/sec] |
|---|---|---|---|---|
| **AES** | 128 | 3766 | 29,42 | 135953 |
| **HIGHT** | 64 | 3188 | 49,81 | 80301 |
| **TEA** | 64 | 6271 | 97,98 | 40823 |
| **XTEA** | 64 | 6718 | 104,97 | 38107 |
| **DES** | 64 | 8633 | 134,89 | 29654 |

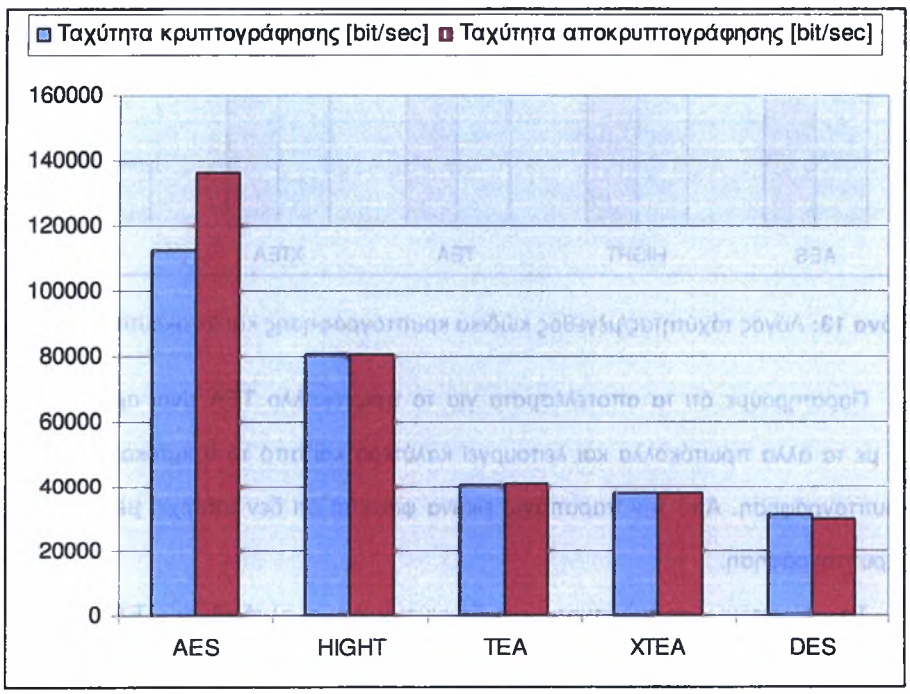
**Πίνακας 8**: Αποτελέσματα κρυπτογράφησης

| Αλγόριθμος | Μέγεθος block [bit] | Αποκρυπτογράφηση [κύκλοι] | Αποκρυπτογράφηση [κύκλοι/bit] | Ταχύτητα [bit/sec] |
|---|---|---|---|---|
| **AES** | 128 | 4558 | 35,61 | 112330 |
| **HIGHT** | 64 | 3188 | 49,81 | 80301 |
| **TEA** | 64 | 6299 | 98,42 | 40641 |
| **XTEA** | 64 | 6718 | 104,97 | 38107 |
| **DES** | 64 | 8154 | 127,41 | 31396 |

**Πίνακας 9**: Αποτελέσματα αποκρυπτογράφησης




**Εικόνα 11:** Αριθμός κύκλων μηχανής για κρυπτογράφηση και αποκρυπτογράφηση

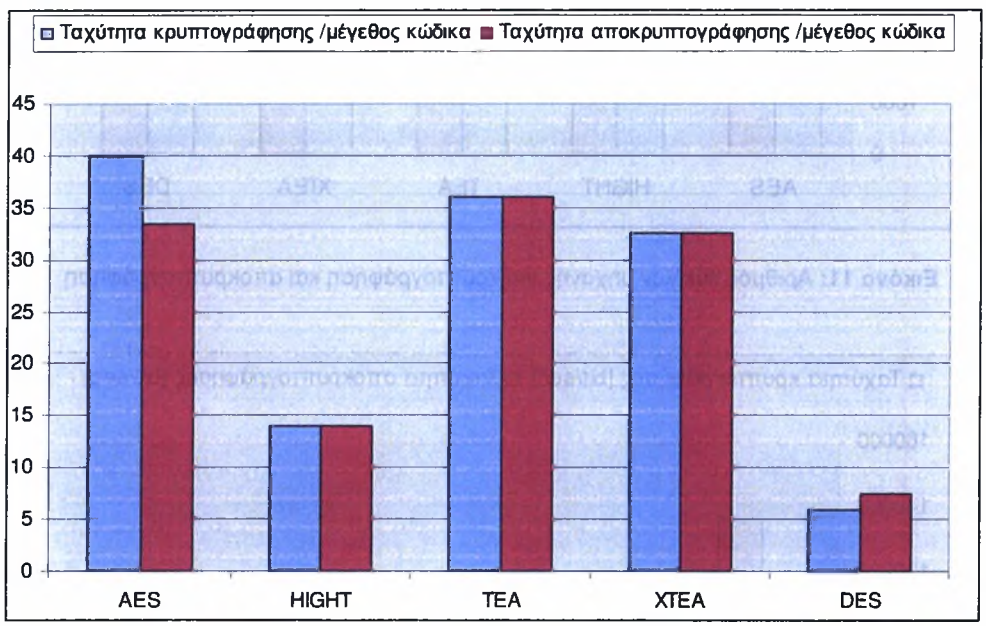



**Εικόνα 12:** Ταχύτητα κρυπτογράφησης και αποκρυπτογράφησης

Παρατηρούμε ότι τα αποτελέσματα για τον αλγόριθμο TEA είναι αρκετά ελκυστικά και δεν παρουσιάζουν μεγάλες διακυμάνσεις ανάλογα με το μέτρο σύγκρισης (όπως ο HIGHT που αναφέραμε παραπάνω).

### 5.2.3 ΣΥΝΟΨΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ

Λόγω του ότι δεν θέλουμε να εστιάσουμε αποκλειστικά στο μέγεθος του κώδικα ή στην απόδοση, δημιουργούμε το λόγο της ταχύτητας προς το μέγεθος του κώδικα και τα αποτελέσματα φαίνονται στην εικόνα 13.




**Εικόνα 13:** Λόγος ταχύτητας/μέγεθος κώδικα κρυπτογράφησης και αποκρυπτογράφησης

Παρατηρούμε ότι τα αποτελέσματα για το πρωτόκολλο TEA είναι αρκετά καλά σε σχέση με τα άλλα πρωτόκολλα και λειτουργεί καλύτερα και από το πρωτόκολλο AES στην αποκρυπτογράφηση. Από την παραπάνω εικόνα φαίνεται ότι δεν υπάρχει μεγάλη διαφορά στην κρυπτογράφηση.

Τα παραπάνω αποτελέσματα αποδεικνύουν ότι ο αλγόριθμος TEA παρουσιάζει ιδιαίτερα ελκυστική απόδοση τόσο σε χρήση αποθηκευτικού χώρου όσο και σε ταχύτητα αποκρυπτογράφησης και κρυπτογράφησης. Το γεγονός ότι ο κώδικας δεν καταλαμβάνει μεγάλο αποθηκευτικό χώρο το καθιστά ιδιαίτερα κατάλληλο για την χρήση του σε δίκτυα

αισθητήρων και ειδικά σε εφαρμογές που χρειάζεται τόσο αποτελεσματική κρυπτογράφηση όσο και ικανοποιητικές αποδόσεις σε χρόνο και ταχύτητα.

# 6. ΕΠΙΛΟΓΟΣ

## 6.1 ΠΑΡΑΤΗΡΗΣΕΙΣ – ΣΥΜΠΕΡΑΣΜΑΤΑ

Τα δίκτυα αισθητήρων χρησιμοποιούνται σε πολλές εφαρμογές σε όλο τον κόσμο. Υπάρχουν ήδη πολλές επιθέσεις που έχουν σαν στόχο την ορθή λειτουργία τέτοιων δικτύων. Οι περιορισμένοι πόροι αλλά και τα εχθρικά περιβάλλοντα στα οποία λειτουργούν μετατρέπουν την ασφάλεια των δικτύων αυτών ένα αρκετά δύσκολο εγχείρημα.

Στα δίκτυα αισθητήρων χρειάζονται βέλτιστες εφαρμογές ασφάλειας που θα έχουν σαν σκοπό τόσο την αποτροπή κακόβουλων χρηστών όσο και την ελαχιστοποίηση κατανάλωσης ενέργειας και αποθηκευτικού χώρου.

Ο αλγόριθμος TEA όπως φαίνεται από τα παραπάνω αποτελέσματα είναι ικανός να χρησιμοποιηθεί ευρέως αφού οι πράξεις που γίνονται κατά την κρυπτογράφηση και αποκρυπτογράφηση δεν επιβαρύνουν τον επεξεργαστή των αισθητήρων με υπερβολικά κόστη. Βασίζεται σε ένα μεγάλο αριθμό επαναλήψεων πράξεων XOR και πρόσθεσης παρά σε προετοιμασμένους και προ-εγκατεστημένους πίνακες κλειδιών. Έτσι επιτυγχάνεται καλύτερη απόδοση και λιγότερη πολυπλοκότητα από άλλες εφαρμογές κρυπτογράφησης.

## 6.2 ΜΕΛΛΟΝΤΙΚΗ ΕΡΕΥΝΑ

Η υλοποίηση της εφαρμογής έχει σαν στόχο την απλή παρουσίαση του αλγορίθμου. Έτσι χειρίζεται την περίπτωση ενός απλού δικτύου που αποτελείται από έναν σταθμό βάσης και δύο πελάτες. Στο δίκτυο που έχει υλοποιηθεί δεν λαμβάνουν χώρα περίπλοκα σχήματα επικοινωνίας των κόμβων μεταξύ τους, ούτε υπάρχει βελτιστοποίηση στην δρομολόγηση. Εάν θέλουμε να ελέγξουμε την απόδοση του αλγορίθμου σε μια πραγματική εφαρμογή χρειάζεται να δημιουργήσουμε ένα μεγάλο δίκτυο αισθητήρων με αυτό-οργάνωση.

Επιπλέον στο πείραμα μας χρησιμοποιούμε ένα κοινό κλειδί για όλους τους κόμβους και τον σταθμό βάσης το οποίο είναι προ-εγκατεστημένο στην μνήμη των αισθητήρων. Μια τυχόν παραβίαση της ασφάλειας κάποιου κόμβου έχει σαν αποτέλεσμα την παραβίαση της ασφάλειας ολόκληρου του δικτύου, αφού κάποιος κακόβουλος χρήστης μπορεί να αποκτήσει

πρόσβαση στα κλειδιά κρυπτογράφησης. Χρειάζεται έτσι η δημιουργία μιας στρατηγικής για την διαχείριση του κλειδιού κατά την οποία θα παράγονται νέα κλειδιά (σε περίπτωση κάποιας υποκλοπής) από μια ψευδοτυχαία συνάρτηση και θα διαμοιράζονται με ασφάλεια στους κόμβους του δικτύου.

Λευκή σελίδα

# ΠΑΡΑΡΤΗΜΑ

# Α. ΠΑΡΑΡΤΗΜΑ - ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ MICA2/MICA2DOT

| Processor/Radio Board | MPR400CB | MPR410CB | MPR420CB | Remarks |
|---|---|---|---|---|
| **Processor Performance** | | | | |
| Program Flash Memory | 128K bytes | 128K bytes | 128K bytes | |
| Measurement (Serial) Flash | 512K bytes | 512K bytes | 512K bytes | > 100,000 measurements |
| Configuration EEPROM | 4K bytes | 4K bytes | 4K bytes | |
| Serial Communications | UART | UART | UART | 0-3V transmission levels |
| ADC | 10-bit ADC | 10-bit ADC | 10-bit ADC | 8 channel, 0-3V input |
| Other Interfaces | DIO, I2C, SPI | DIO, I2C, SPI | DIO, I2C, SPI | |
| Current Draw | 8 mA | 8 mA | 8 mA | Active mode |
| | < 15 μA | < 15 μA | < 15 μA | Sleep mode |
| **Multi-Channel Radio** | | | | |
| Center Frequency | 868/916 MHz | 433 MHz | 315 MHz | ISM bands |
| Number of Channels | 4/ 50 | 4 | 5 | Programmable, country specific |
| Data Rate | 38.4 KBaud | 38.4 KBaud | 38.4 KBaud | Manchester encoded |
| RF Power | -20 to +5 dBm | -20 to +10 dBm | -20 to +10 dBm | Programmable typical |
| Receive Sensitivity | -98 dBm | -101 dBm | -101 dBm | Typical, analog RSSI at AD Ch.0 |
| Outdoor Range | 500 ft | 1000 ft | 1000 ft | 1/4 Wave dipole, line of sight |
| Currnent Draw | 27 mA | 25 mA | 25 mA | Transmit with maximum power |
| | 10 mA | 8 mA | 8 mA | Receive |
| | < 1 mA | < 1 mA | < 1 mA | Sleep |
| **Electromechanical** | | | | |
| Battery | 2X AA battaries | 2X AA battaries | 2X AA battaries | Attached pack |
| External Power | 2.7 - 3.3 V | 2.7 - 3.3 V | 2.7 - 3.3 V | Connector provided |
| User Interface | 3 LEDs | 3 LEDs | 3 LEDs | User programmable |
| Size (mm) | 58 x 32 x 7 | 58 x 32 x 7 | 58 x 32 x 7 | Excluding battery pack |
| Weight (grams) | 18 | 18 | 18 | Excluding batteries |
| Expansion Connector | 51-pin | 51-pin | 51-pin | All major I/O signals |

# Β. ΠΑΡΑΡΤΗΜΑ - ΚΩΔΙΚΑΣ NESC

### Β.1.1 Wrapper.nc

Με το configuration αρχείο Wrapper γίνεται η σύνδεση του interface StdControl του TEATranceiver και του TEABase. Παρατηρούμε ότι δημιουργούνται δυο στιγμιότυπα (SndControl, TrdControl) του ίδιου αρχείου.

```
configuration Wrapper {}
implementation
{

      components Main, WrapperM, TEATranceiverC, TEABaseC as
BaseStation;

      Main.StdControl->WrapperM.StdControl;

      WrapperM.FstControl->BaseStation.StdControl;
      WrapperM.SndControl->TEATranceiverC.StdControl;
      WrapperM.TrdControl->TEATranceiverC.StdControl;
}
```

### B.1.2 WrapperM.nc

Το αρχείο WrapperM.nc είναι το module του Wrapper.nc. Υλοποιούνται τα interface StdControl των δυο εφαρμογών με τις εντολές που ορίζονται στο StdControl.nc (init(), start(), stop() ). Βάση του συγκεκριμένου wiring ο σταθμός βάσης λαμβάνει την διεύθυνση 0, ενώ οι πελάτες τις διευθύνσεις 1 και 2.

```
module WrapperM
{
      provides {
interface StdControl;}
      uses {
            interface StdControl as FstControl;
            interface StdControl as SndControl;
            interface StdControl as TrdControl;
                        }
}
implementation {
      command result_t StdControl.init() {
            if (TOS_LOCAL_ADDRESS==0)  {
                  dbg(DBG_BOOT, "WrapperM: initialized.");
                  return (call FstControl.init());
            }
            else if (TOS_LOCAL_ADDRESS==1) {
                  dbg(DBG_BOOT, "WrapperM: initialized ");
                  return (call SndControl.init());
            }
            else if (TOS_LOCAL_ADDRESS==2) {
                  return (call TrdControl.init());
                  return SUCCESS;
            }
            return FALSE;
      }
      command result_t StdControl.start() {
            if (TOS_LOCAL_ADDRESS==0) {
                  dbg(DBG_BOOT, "WrapperM: started.");
                  return (call FstControl.start());
            }
            else if (TOS_LOCAL_ADDRESS==1) {
                  dbg(DBG_BOOT, "WrapperM: started.");
                  return (call SndControl.start());
            }
            else if (TOS_LOCAL_ADDRESS==2) {
                  return (call TrdControl.start());
                  return SUCCESS;
            }
            return FALSE;
      }
      command result_t StdControl.stop() {
            if (TOS_LOCAL_ADDRESS==0) {
                  return (call FstControl.stop());
            }
            else if (TOS_LOCAL_ADDRESS==1) {
return (call SndControl.stop());
            }
            else if (TOS_LOCAL_ADDRESS==2) {
                  return (call TrdControl.stop());
                  return SUCCESS;
            }
```

### B.2.1 TEATranceiverC.nc

Το αρχείο TEATranceiveC.nc είναι το configuration file των πελατών.

```
configuration TEATranceiverC {
      provides interface StdControl;
}
Implementation{

 components
                      TEATranceiverM,
                      TimerC,
                      LedsC,
                      DemoSensorC as Sensor,
                      UARTComm as Comm;

  StdControl = TEATranceiverM.StdControl;

  TEATranceiverM.TimerControl->TimerC.StdControl;
  //Main.StdControl->TimerC;

  TEATranceiverM.Timer->TimerC.Timer[unique("Timer")];
  //TEATranceiverM.TimerControl->TimerC.StdControl;
  TEATranceiverM.Leds -> LedsC;
  TEATranceiverM.SensorControl->Sensor;
  TEATranceiverM.ADC->Sensor;
  TEATranceiverM.CommControl->Comm;
  TEATranceiverM.ResetCounterMsg->Comm.ReceiveMsg[AM_OSCOPERESETMSG];
  TEATranceiverM.DataMsg -> Comm.SendMsg[AM_OSCOPEMSG];
}
```

### B.2.2 TEATranceiverM.nc

Το αρχείο TEATranceiverM.nc υλοποιεί την λειτουργικότητα των πελατών. Παρακάτω περιγράφονται συνοπτικά οι κυριότεροι μέθοδοι του module.

#### B.2.2.1 Async event result_t ADC.dataReady(uint16_t)

Το event αυτό καλείται επαναληπτικά από τον Timer του προγράμματος, με περίοδο που καθορίζεται runtime. Κάθε φορά που εκτελείται επιστρέφεται μια τιμή που αποθηκεύεται σε έναν πίνακα. Στην συνέχεια γίνεται casting της τιμής για να είναι συμβατή με το μέγεθος των ορισμάτων που πρέπει να δέχεται ο αλγόριθμος κρυπτογράφησης. Επίσης καλείται η συνάρτηση κρυπτογράφησης για την κρυπτογράφηση της μέτρησης και την εισαγωγή της στην συνέχεια στο μήνυμα με το dataTask( ).

### B.2..2.2 Task void dataTask( )

Στο task αυτό τοποθετούνται τα απαραίτητα πεδία του πακέτου προς αποστολή και στην συνέχεια γίνεται η αποστολή του με την κλήση της dataMsg.send( ). Η κρυπτογραφημένη μέτρηση εισάγεται στο πακέτο υπό την μορφή πίνακα 2x2 μεγέθους 32bit.

### B.2.2.3 `Void encrypt(uint16_t *, uint32_t key* )`

Η συνάρτηση κρυπτογράφηση καλείται από το event ADC.dataReady. Δέχεται σαν ορίσματα την μέτρηση από τον ADC και τον πίνακα με τα κλειδιά που είναι μεγέθους 32bit το καθένα. Λόγω της ρουτίνας του αλγορίθμου χρειάζεται μετατροπή της τιμής της μέτρησης από 16bit σε 32bit.

```
/*
 *TEATranceiverM.nc
 *University Of Thessaly, 2007-2008
 *Noutsis Giorgos
 */

#include "OscopeMsg.h"

module TEATranceiverM
{
  provides interface StdControl;
  uses {
    interface Timer;
    interface Leds;
    interface StdControl as SensorControl;
    interface StdControl as TimerControl;
    interface ADC;
    interface StdControl as CommControl;
    interface SendMsg as DataMsg;
    interface ReceiveMsg as ResetCounterMsg;
  }
}
implementation{

  uint8_t packetReadingNumber;
  uint16_t readingNumber;
  uint32_t data_p[2];
  TOS_Msg msg[2];
  uint8_t currentMsg;
  uint32_t key[4];

  void encrypt(uint16_t );

  command result_t StdControl.init() {

    result_t ok1,ok2,ok3,ok;

    call Leds.init();
    call Leds.yellowOff(); call Leds.redOff(); call Leds.greenOff();
    ok1 = call TimerControl.init();
    ok2 = call SensorControl.init();
    ok3 = call CommControl.init();
```

```
atomic {
      currentMsg = 0;
      packetReadingNumber = 0;
      readingNumber = 0;

      key[0]=1234;       //  *
      key[1]=2345;       //  *   keys
      key[2]=3456;       //  *
      key[3]=4567;       //  *
   }

   if((ok= rcombine3(ok1,ok2,ok3)))
         dbg(DBG_BOOT, "MOTE.%d
initialized\n",TOS_LOCAL_ADDRESS);
      return ok;
  }
  /**
     * Starts the SensorControl and CommControl components.
     * @return Always returns SUCCESS.
   */
  command result_t StdControl.start() {
      result_t ok1,ok2,ok3;

      ok1= call SensorControl.start();
      ok2 =call Timer.start(TIMER_REPEAT, 125);
      ok3=call CommControl.start();

      return rcombine3(ok1,ok2,ok3);
  }
   /**
     * Stops the SensorControl and CommControl components.
     * @return Always returns SUCCESS.
     */
    command result_t StdControl.stop() {
      result_t ok1,ok2,ok3;

      ok1 = call SensorControl.stop();
      ok2 = call Timer.stop();
      ok3 = call CommControl.stop();

      return rcombine3(ok1,ok2,ok3);
    }
    task void dataTask() {

    struct OscopeMsg *pack;

    atomic {
      pack = (struct OscopeMsg *)msg[currentMsg].data;
      packetReadingNumber = 0;
      pack->lastSampleNumber = readingNumber;
    }

    pack->channel = 1;
    pack->sourceMoteID = TOS_LOCAL_ADDRESS;

    /* Try to send the packet. Note that this will return
     * failure immediately if the packet could not be queued for
     * transmission.
     */
```

```
if (call DataMsg.send(TOS_BCAST_ADDR, sizeof(struct OscopeMsg),
                      &msg[currentMsg]))
      {
      atomic {
        currentMsg ^= 0x1;
      }
      call Leds.yellowToggle();
      }
  }

  /**
   * Signalled when data is ready from the ADC. Stuffs the sensor
   * reading into the current packet, and sends off the packet
when
   * BUFFER_SIZE readings have been taken.
   * @return Always returns SUCCESS.
   */
  async event result_t ADC.dataReady(uint16_t data) {

    struct OscopeMsg *pack;
    atomic {

      dbg(DBG_USR1,"\n Data to be encrypted: %d\n", data);

      encrypt(data);

      pack = (struct OscopeMsg *)msg[currentMsg].data;

      pack->data[packetReadingNumber][0] = data_p[0];
      pack->data[packetReadingNumber][1] = data_p[1];

      packetReadingNumber++;
      readingNumber++;
      dbg(DBG_USR1, "data_event\n");
      if (packetReadingNumber == BUFFER_SIZE) {

      post dataTask();
      }
    }
    if (data > 0x0300)
      call Leds.redOn();
    else
      call Leds.redOff();

    return SUCCESS;
  }

  /**
   * Signalled when the previous packet has been sent.
   * @return Always returns SUCCESS.
   */
  event result_t DataMsg.sendDone(TOS_MsgPtr sent, result_t
success) {

  dbg(DBG_USR1, "SEND DONE\n");

    return SUCCESS;
  }
```

```
/**
   * Signalled when the clock ticks.
   * @return The result of calling ADC.getData().
   */

  event result_t Timer.fired() {
    return call ADC.getData();
  }
  /**
   * Signalled when the reset message counter AM is received.
   * @return The free TOS_MsgPtr.
   */
  event TOS_MsgPtr ResetCounterMsg.receive(TOS_MsgPtr m) {
    atomic {
      readingNumber = 0;
    }
    return m;
  }
   void encrypt(uint16_t data/*, uint32_t key2*/)  {
      uint8_t *p;
      uint32_t y, z , a , b, c, d, n, delta,sum;

      p=&data;
      data_p[0]=*p;
      data_p[1]=*(p+1);

      y = data_p[0];
      z = data_p[1];

      sum = 0;
      delta = 0x9e377b9;
        n=32;

      a = key[0];
      b = key[1];
      c = key[2];
      d = key[3];

        while(n!=0)
      {
           sum += delta;
           y += (z << 4)+(a ^ z)+(sum ^ (z >> 5))+b;
           z += (y << 4)+(c ^ y)+(sum ^ (y >> 5))+d;
           n--;
      }

      data_p[0] = y;
      data_p[1] = z;

dbg(DBG_USR1,"Data encrypted: data_p[0]= %d and data_p[1]= %d",
data_p[0],data_p[1]);
      }
 }
```

### B.3.1 TEABase.nc

Το configuration αρχείο του σταθμού βάσης.

```
/*
 * TEABase.nc
 * University of Thessaly, 2007-2008
 * Noutsis Giorgos
 */

configuration TEABaseC {
      provides interface StdControl;
}
implementation {

 components
             TEABaseM,
             RadioCRCPacket as Comm,
             FramerM,
             UART,
             LedsC;

  StdControl = TEABaseM.StdControl;

  TEABaseM.UARTControl -> FramerM;
  TEABaseM.UARTSend -> FramerM;
  TEABaseM.UARTReceive -> FramerM;
  TEABaseM.UARTTokenReceive -> FramerM;

  TEABaseM.RadioControl -> Comm;
  TEABaseM.RadioSend -> Comm;
  TEABaseM.RadioReceive -> Comm;

  TEABaseM.Leds -> LedsC;

  FramerM.ByteControl -> UART;
  FramerM.ByteComm -> UART;
}
```

### B.3.2 TEABaseM.nc

Ακολουθεί η περιγραφή των κυριότερων μεθόδων του module, το οποίο υλοποιεί την λειτουργικότητα του σταθμού βάσης.

#### B.3.2.1 `event TOS_MsgPtr RadioReceive.receive(TOS_MsgPtr)`

Το event αυτό προκαλείται από κάθε λήψη μηνύματος στον σταθμό βάσης από τους κόμβους. Στο event αυτό καλείται και η συνάρτηση αποκρυπτογράφησης.

### B.3.2.2 void decrypt( )

Η συνάρτηση αποκρυπτογράφησης δέχεται σαν ορίσματα την κρυπτογραφημένη μέτρηση που έχει σταλεί με το μήνυμα και έχουν αποθηκευτεί σε έναν πίνακα. Μετά την ρουτίνα του αλγορίθμου αποκρυπτογράφησης γίνεται ξανά casting στα αποτελέσματα του αλγορίθμου για να παρθεί η αρχική μέτρηση.

### B.3.2.3 task void UARTSendTask ( )

Αφού καθοριστεί κατάλληλα το πακέτο, αποστέλλεται στη διεύθυνση της UART (0x007e) με την κλήση της παρακάτω εντολής:

```
Call UARTSend.send(&uartQueueBufs[uartOut]
```

```
/*
 *TEABaseM.nc
 *University Of Thessaly, 2007-2008
 *Noutsis Giorgos
 */


#ifndef TEABASE_BLINK_ON_DROP
#define TEABASE_BLINK_ON_DROP
#endif

module TEABaseM {
  provides interface StdControl;
  uses {
    interface StdControl as UARTControl;
    interface BareSendMsg as UARTSend;
    interface ReceiveMsg as UARTReceive;
    interface TokenReceiveMsg as UARTTokenReceive;

    interface StdControl as RadioControl;
    interface BareSendMsg as RadioSend;
    interface ReceiveMsg as RadioReceive;

    interface Leds;
  }
}

implementation
{

  enum {
    UART_QUEUE_LEN = 12,
    RADIO_QUEUE_LEN = 12,
  };

  uint32_t key[4];
  void decrypt(uint32_t*);
  TOS_Msg msg2;
```

```
  TOS_Msg    uartQueueBufs[UART_QUEUE_LEN];
  uint8_t    uartIn, uartOut;
  bool       uartBusy, uartCount;

  TOS_Msg    radioQueueBufs[RADIO_QUEUE_LEN];
  uint8_t    radioIn, radioOut;
  bool       radioBusy, radioCount;

  task void UARTSendTask();
  task void RadioSendTask();

  void failBlink();
  void dropBlink();
  void processUartPacket(TOS_MsgPtr Msg, bool wantsAck, uint8_t
Token);

  command result_t StdControl.init() {
    result_t ok1, ok2, ok3;

    uartIn = uartOut = uartCount = 0;
    uartBusy = FALSE;

    radioIn = radioOut = radioCount = 0;
    radioBusy = FALSE;

    ok1 = call UARTControl.init();
    ok2 = call RadioControl.init();
    ok3 = call Leds.init();

    key[0]=1234;
    key[1]=2345;
    key[2]=3456;
    key[3]=4567;
    return rcombine3(ok1, ok2, ok3);
  }
  command result_t StdControl.start() {
    result_t ok1, ok2;
    ok1 = call UARTControl.start();
    ok2 = call RadioControl.start();
    return rcombine(ok1,ok2);
  }

  command result_t StdControl.stop() {
    result_t ok1, ok2;

    ok1 = call UARTControl.stop();
    ok2 = call RadioControl.stop();
    return rcombine(ok1, ok2);
  }
  event TOS_MsgPtr RadioReceive.receive(TOS_MsgPtr Msg) {

    msg2 = *Msg;
    decrypt (msg2.data/*,key*/);

    if ((!Msg->crc) || (Msg->group != TOS_AM_GROUP))
      return Msg;
```

```
if (uartCount < UART_QUEUE_LEN) {

      memcpy(&uartQueueBufs[uartIn], Msg, sizeof(TOS_Msg));
      uartCount++;

      if( ++uartIn >= UART_QUEUE_LEN ) uartIn = 0;

      if (!uartBusy) {
      if (post UARTSendTask()) {
        uartBusy = TRUE;
      }
      }
    } else {
      dropBlink();
    }

    return Msg;
  }

  task void UARTSendTask() {
    dbg (DBG_USR1, "TEABase forwarding Radio packet to UART\n");

    if (uartCount == 0) {

      uartBusy = FALSE;

    } else {

      if (call UARTSend.send(&uartQueueBufs[uartOut]) == SUCCESS) {
      call Leds.greenToggle();
      } else {
      failBlink();
      post UARTSendTask();
      }
    }
  }

  event result_t UARTSend.sendDone(TOS_MsgPtr msg, result_t
success) {

    if (!success) {
      failBlink();
    } else {
      uartCount--;
      if( ++uartOut >= UART_QUEUE_LEN ) uartOut = 0;
    }
    post UARTSendTask();
    return SUCCESS;
  }

  event TOS_MsgPtr UARTReceive.receive(TOS_MsgPtr Msg) {
    processUartPacket(Msg, FALSE, 0);
    return Msg;
  }
  event TOS_MsgPtr UARTTokenReceive.receive(TOS_MsgPtr Msg, uint8_t
Token) {
    processUartPacket(Msg, TRUE, Token);
    return Msg;
  }
```

```
void processUartPacket(TOS_MsgPtr Msg, bool wantsAck, uint8_t
Token) {
    bool reflectToken = FALSE;

    dbg(DBG_USR1, "TEABase received UART token packet.\n");

    if (radioCount < RADIO_QUEUE_LEN) {
      reflectToken = TRUE;

      memcpy(&radioQueueBufs[radioIn], Msg, sizeof(TOS_Msg));

      radioCount++;

      if( ++radioIn >= RADIO_QUEUE_LEN ) radioIn = 0;

      if (!radioBusy) {
      if (post RadioSendTask()) {
        radioBusy = TRUE;
      }
      }
    } else {
      dropBlink();
    }

    if (wantsAck && reflectToken) {
      call UARTTokenReceive.ReflectToken(Token);
    }
  }

  task void RadioSendTask() {

    dbg(DBG_USR1, "TEABase forwarding UART packet to Radio\n");

    if (radioCount == 0) {

      radioBusy = FALSE;

    } else {
      radioQueueBufs[radioOut].group = TOS_AM_GROUP;
      if (call RadioSend.send(&radioQueueBufs[radioOut]) ==
SUCCESS) {
      call Leds.redToggle();
      } else {
      failBlink();
      post RadioSendTask();
      }
    }
  }
event result_t RadioSend.sendDone(TOS_MsgPtr msg, result_t success)
{
    if (!success) {
      failBlink();
    } else {
      radioCount--;
      if( ++radioOut >= RADIO_QUEUE_LEN ) radioOut = 0;
    }
    post RadioSendTask();
    return SUCCESS;
  }
```

```
void dropBlink() {
#ifdef TOSBASE_BLINK_ON_DROP
   call Leds.yellowToggle();
#endif
 }
 void failBlink() {
#ifdef TOSBASE_BLINK_ON_FAIL
   call Leds.yellowToggle();
#endif
 }

 void decrypt (uint32_t* data_p)
 {
        uint32_t y, z, sum,a,b,c,d, delta;
        uint32_t data_n[2];
        uint8_t n;
        uint16_t reading;

    y = data_p[0];
    z = data_p[1];

    delta = 0x9e377b9;


    sum = delta<<5;
      //data decrypted

    a = key[0];
    b = key[1];
    c = key[2];
    d = key[3];
    n=32;

    while(n!=0)
    {
      z -= (y << 4)+(c ^ y) + (sum ^ (y >> 5)) + d;
      y -= (z << 4)+(a ^ z) + (sum ^ (z >> 5)) + b;

      sum -= delta;
      n--;
    }

    data_n[0] = y;
    data_n[1] = z;

    dbg(DBG_USR1,"Data decrypted: data_n[0]= %d and data_n[1]= %d \n\n",
data_n[0],data_n[1]);

    //shift to the reading

    reading = (uint8_t)data_n[1];
    reading = reading<<8;
    reading = reading + (uint8_t)data_n[0];

    dbg(DBG_USR1,"Data decrypted after the shift: reading= %d \n\n", reading);

 }

}
```

Λευκή σελίδα

# ΒΙΒΛΙΟΓΡΑΦΙΑ


[1] http://www.xbow.com/wireless home.aspx.

[2] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. E. Culler, and K. Pister. System architecture directions for networked sensors. In *Architectural Support for Programming Languages and Operating Systems*, pages 93–104, 2000.

[3] D. W. Carman, P. S. Krus, and B. J. Matt. Constraints and approaches for distributed sensor network security. Technical Report 00-010, NAI Labs, Network Associates, Inc., Glenwood, MD, 2000.

[4] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler. Spins: security protocols for sensor networks. *Wireless Networking*, 8(5):521–534, 2002.

[5] L. Eschenauer and V. D. Gligor. A key-management scheme for distributed sensor networks. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 41–47. ACM Press, 2002.

[6] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *Proceedings of the 2003 IEEE Symposium on Security and Privacy*, page 197. IEEE Computer Society, 2003.

[7] J. Hwang and Y. Kim. Revisiting random key pre-distribution schemes for wireless sensor networks. In *Proceedings of the 2nd ACM workshop on Security of Ad hoc and Sensor Networks (SASN '04)*, pages 43–52, New York, NY, USA, 2004. ACM Press.

[8] D. Liu, P. Ning, and R. Li. Establishing pairwise keys in distributed sensor networks. *ACM Trans. Inf. Syst. Secur.*, 8(1):41–77, 2005

[9] D. Wood and J. A. Stankovic. Denial of service in sensor networks. *Computer*, 35(10):54–62, 2002.

[10] J. Newsome, E. Shi, D. Song, and A. Perrig. The sybil attack in sensor networks: analysis & defenses. In *Proceedings of the third international symposium on Information processing in sensor networks*, pages 259–268. ACM Press, 2004.

[11] J. Douceur. The sybil attack. In *Proc. of the 1st International Workshop on Peer-to-Peer Systems (IPTPS'02)*, February 2002.

[12] J. Deng, R. Han, and S. Mishra. Countermeasuers against traffic analysis in wireless sensor networks. Technical Report CU-CS-987-04, University of Colorado at Boulder, 2004.

[13] C. Ozturk, Y. Zhang, and W. Trappe. Source-location privacy in energyconstrained sensor network routing. In *Proceedings of the 2nd ACM workshop on Security of Ad hoc and Sensor Networks*, 2004.

[14] M. Gruteser, G. Schelle, A. Jain, R. Han, and D. Grunwald. Privacy-aware location sensor networks. In *9th USENIX Workshop on Hot Topics in Operating Systems (HotOS IX)*, 2003.

[15] H. Chan and A. Perrig. Security and privacy in sensor networks. *IEEE Computer Magazine*, pages 103–105, 2003 2003.

[16] X. Wang, W. Gu, K. Schosek, S. Chellappan, and D. Xuan. Sensor network configuration under physical attacks. Technical Report Technical Report (OSU-CISRC-7/ 04-TR45), Dept. of Computer Science and Engineering, The Ohio-State University, July 2004.

[17] X. Wang, W. Gu, S. Chellappan, Dong Xuan, and Ten H. Laii. Search-based physical attacks in sensor networks: Modeling and defense. Technical report, Dept. of Computer Science and Engineering, The Ohio-State University, February 2005.

[18] G. Gaubatz, J.P. Kaps, and B. Sunar. Public key cryptography in sensor networks - revisited. In *1st European Workshop on Security in Ad-Hoc and Sensor Networks (ESAS 2004)*, 2004.

[19] N. Gura, A. Patel, A. Wander, H. Eberle, and S. Shantz. Comparing elliptic curve cryptography and rsa on 8-bit cpus. In *In 2004 workshop on Cryptographic Hardware and Embedded Systems*, August 2004.

[20] J. Deng, R. Han, and S. Mishra. *Security, privacy, and fault tolerance in wireless sensor networks.* Artech House, August 2005.

[21] R. Watro, D. Kong, S. Cuti, C. Gardiner, C. Lynn, and P. Kruus. Tinypk: securing sensor networks with public key technology. In *Proceedings of the 2nd ACM workshop on Security of Ad hoc and Sensor Networks (SASN '04)*, pages 59–64, New York, NY, USA, 2004. ACM Press.

[22] B. Schneier. *Applied Cryptography*. Second Edition, John Wiley & Sons, 1996.

[23] Y. Law, J. Doumen, and P. Hartel. Survey and benchmark of block ciphers for wireless sensor networks. Technical Report TR-CTIT-04-07, Centre for Telematics and Information Technology, University of Twente, The Netherlands, 2004.

[24] H. Chan and A. Perrig. Pike: Peer intermediaries for key establishment in sensor networks. In *IEEE Infocom 2005*, 2005.

[25] Q. Huang, J. Cukier, H. Kobayashi, B. Liu, and J. Zhang. Fast authenticated key establishment protocols for self-organizing sensor networks. In *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, pages 141–150. ACM Press, 2003.

[26] B. Karp and H. T. Kung. GPSR: greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 243–254. ACM Press, 2000.

[27] P. Bose, P. Morin, I. Stojmenovi´c;, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *Wirel. Netw.*, 7(6):609–616, 2001.

[28] S. Tanachaiwiwat, P. Dave, R. Bhindwale, and A. Helmy. Poster abstract secure locations: routing on trust and isolating compromised sensors in locationaware sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 324–325. ACM Press, 2003.

[29] R. Anderson and M. Kuhn. Low cost attacks on tamper resistant devices. In *IWSP: International Workshop on Security Protocols, LNCS*, 1997.

[30] C. Hartung, J. Balasalle, and R. Han. Node compromise in sensor networks: The need for secure systems. Technical Report Technical Report CU-CS-988-04, Department of Computer Science, University of Colorado at Boulder, 2004.

[31] X. Wang, W. Gu, S. Chellappan, K.t Schoseck, and Dong Xuan. Lifetime optimization of sensor networks under physical attacks. In *Proc. of IEEE Internationl Conference on Communications*, May 2005.

[32] A. Seshadri, A. Perrig, L. van Doorn, and P. Khosla. Swatt: Software-based attestation for embedded devices. In *In Proceedings of the IEEE Symposium on Security and Privacy*, May 2004.

[33] L. Hu and D. Evans. Using directional antennas to prevent wormhole attacks. In *In 11th Annual Network and Distributed System Security Symposium*, February 2004.

[34] http://www.ftp.cl.cam.ac.uk/ftp/papers/djw-rmn/djw-rmn-tea.html

[35] N. Sastry, U. Shankar, and D. Wagner. Secure verification of location claims. In *ACM Workshop on Wireless Security*, September 2003.

[36] nesC 1.2 Language Reference Manual, David Gay, Philip Levis, David Culler, Eric Brewer,August 2005

[37] http://www.tinyos.net/

[38] B. Titzer, D. Lee, and J. Palsberg, "*Avrora: Scalable Sensor Network Simulation with Precise Timing*". In Proceedings of IPSN'05, Fourth International Conference on Information Processing in Sensor Networks, Los Angeles, (2005).

[39] http://www.ceid.upatras.gr/courses/katanemhmena/wiki/index.php/TinyOS:nesC_Programming_Language

[40] Performance Analysis of Contemporary Light-Weight Block Ciphers on 8-bit Microcontrollers, S¨oren Rinne, Thomas Eisenbarth, and Christof Paar