ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ

Ανάπτυξη Κρυπτογραφικών Αλγορίθμων Για Ετερογενή Ασύρματα Δίκτυα Αισθητήρων

Μεταπτυχιακο Προγραμμα Επιστημη & Τεχνολογια των Υπολογιστων

Επιβλέπων:
Παύλος Σπυρακης

Συγγραφέας:
Απόστολος Πυργελης

Συνεπιβλέποντες:
Ιωάννης Σταματιου
Ιωάννης Χατζηγιαννακης

Πάτρα, 5 Φεβρουαρίου 2012

Abstract

A wireless sensor network consists of spatially distributed autonomous devices, that use sensors in order to cooperatively monitor natural and environmental conditions. A wireless sensor network node typically consists of a microprocessor, a radio transceiver, an energy source (usually a battery) and various kinds of sensors that measure different environmental conditions (e.g. temperature, humidity, lightness).

Wireless sensor networks are economically viable solutions to various applications. Networks of this type are used in biomedical, military, industrial as well as in applications that monitor the natural environment. Moreover, sensor networks are a key for the creation of *smart spaces* that introduce information technology in everyday environments like those of home and work. Because sensor networks are typically deployed in environments where sensitive information is communicated, security and privacy issues arise. Application characteristics like deployment in adverse environments, the incomplete knowledge of the network topology, the possibility of self-organization and the operation without human surveillance make the maintenance of security a big challenge.

Cryptography is a science field that provides solutions to security problems. Cryptography is a well established science field that has provided protocols and standards of wide acceptance. However, the use of such protocols and standards in resource constrained environments, like those of sensor networks, requires changes and adaptations. This applies due to the special characteristics of such networks and the devices that consist them, e.g. the restricted processing power, the limited storage possibility and energy as well as the wireless nature of communication.

A second problem that arises in wireless sensor networks is that of heterogeneity. The nodes produced by the industry today have different processing possibilities and execute various operating systems. Some nodes consist of 8-bit micro-processors that offer tiny amounts of RAM memory, whereas others are produced with 32-bit processors that can even execute desktop operating systems like Linux. Thus, while it is easy for a developer to implement a platform specific application, it is quite difficult to write some generic platform indepedent code. In order to re-use implemented applications and algorithms in different development environments a programmer has to make adaptations and changes according to the new platform specifications. A solution to this problem is provided by Wiselib [3]. Wiselib offers an development enviroment suitable for the implementation of generic algorithms that can be compiled and executed on heterogenous wireless sensor networks.

The objective of this work is to provide solutions to the aforementioned sensor network problems (security, heterogeneity). For this reason, we develop a cryptographic library on the generic program environment of Wiselib, a generic algorithm library for heterogenous sensor networks. Wiselib is implemented in C++ and employing advanced techniques, like templates and inline functions, it allows writing of generic code that can be allocated and resolved during compilation without producing computation or memory overhead.

Due to the security requirements imposed by the sensor network applications and the resource restrictions by the sensor nodes, our cryptographic library

provides symmetric as well as asymmetric cryptographic algorithms. The public key algorithms are based on elliptic curve cryptography. Elliptic curves consist an ideal system for the development of public key cryptography in resource constrained environments that offer restricted computation power, memory and energy. That is because elliptic curve cryptosystems offer the same level of security with other public key cryptosystems (e.g. RSA), using much smaller key sizes.

Thus, in total our cryptographic library provides the following algorithms:

- the symmetric encryption algorithm AES
- the hash algorithm SHA-1
- the key agreement scheme Diffie Hellman (ECDH)
- the public key encryption algorithm ECIES
- the signature scheme ECDSA

In order to evaluate the performance of our cryptographic library we experimentally test (in terms of execution time, compiled code size, energy consumption) its algorithms on two devices (iSense, TelosB) with different processing possibilities (16 MHz, 8 MHz) that execute different operating systems (iSense OS, Contiki Sky). The fact that we evaluated the cryptographic algorithms on two different devices with unlike capabilities and that execute different operating systems proves that our implementation is generic. Finally, in order to prove the ease of use of the implemented cryptographic algorithms we present three sensor network applications that employ them. More precisely, we show how our cryptographic algorithms can be combined with routing and clustering algorithms (provided by Wiselib), resulting in secure sensor network applications.

Περίληψη

Ένα ασύρματο δίκτυο αισθητήρων αποτελείται από χωρικά κατανεμημένες αυτόνομες συσκευές, οι οποίες χρησιμοποιούν αισθητήρες για την συνεργατική παρακολούθηση φυσικών και περιβαλλοντικών συνθηκών. Ένας κόμβος ενός δικτύου αισθητήρων συνήθως αποτελείται από ένα μικροελεγκτή, ένα ραδιο-πομποδέκτη, μια πηγή ενέργειας, η οποία συχνά είναι μια μπαταρία και διαφόρων ειδών αισθητήρες (π.χ. θερμοκρασίας, υγρασίας, φωτεινότητας).

Τα ασύρματα δίκτυα αισθητήρων προσφέρουν οικονομικά βιώσιμες λύσεις σε ποικίλες εφαρμογές. Δίκτυα τέτοιου τύπου δραστηριοποιούνται σε βιοιατρικές, στρατιωτικές, βιομηχανικές εφαρμογές καθώς και εφαρμογές παρακολούθησης του φυσικού περιβάλλοντος. Επιπλέον, τα δίκτυα αισθητήρων είναι κλειδί για τη δημιουργία *έξυπνων χώρων* που ενθέτουν τεχνολογία πληροφορίας στα καθημερινά περιβάλλοντα εργασίας και κατοικίας. Λόγω της χρήσης τέτοιων δικτύων σε περιβάλλοντα που ανταλλάσσονται ευαίσθητες πληροφορίες, δημιουργούνται θέματα ασφάλειας και μυστικότητας. Χαρακτηριστικά των διαφόρων εφαρμογών όπως η λειτουργία σε αντίξοα περιβάλλοντα, η ελλιπής γνώση της τοπολογίας του δικτύου, οι δυνατότητες αυτο-οργάνωσης και αυτόματης διόρθωσης λειτουργιών και η λειτουργία χωρίς ανθρώπινη επιτήρηση καθιστούν τη διατήρηση της ασφάλειας μια μεγάλη πρόκληση.

Ένας κλάδος που παρέχει λύσεις σε προβλήματα ασφαλείας είναι αυτός της κρυπτογραφίας. Η κρυπτογραφία είναι μια καλά εδραιωμένη επιστημονική περιοχή, με πρωτόκολλα και πρότυπα τα οποία τυγχάνουν ευρείας αναγνώρισης. Παρόλα αυτά, η χρήση τους σε περιβάλλοντα περιορισμένων πόρων όπως αυτά των ασυρμάτων δικτύων αισθητήρων, απαιτεί προσαρμογές. Η απαίτηση αυτή προκύπτει λόγω των ιδιαίτερων χαρακτηριστικών των δικτύων αυτών και των κόμβων που τα απαρτίζουν όπως η χαμηλή υπολογιστική ισχύς, οι περιορισμένες δυνατότητες αποθήκευσης και η περιορισμένη διαθέσιμη ενέργεια καθώς και η επικοινωνία ασύρματης φύσης που υιοθετείται.

Ένα επιπλέον πρόβλημα που παρουσιάζεται στα ασύρματα δίκτυα αισθητήρων, είναι η ετερογένεια. Οι συσκευές αισθητήρων που παράγονται από τη βιομηχανία σήμερα, έχουν διαφορετικές υπολογιστικές δυνατότητες και εκτελούν διαφορετικά λειτουργικά συστήματα. Κάποιες αποτελούνται από μικροεπεξεργαστές 8-bit και έχουν ελάχιστη ποσότητα μνήμης RAM, ενώ άλλες έχουν πολύ μεγάλη υπολογιστική δύναμη και μπορούν να εκτελέσουν desktop λειτουργικά συστήματα όπως Linux. Έτσι, ενώ είναι σχετικά εύκολο να αναπτύξει κανείς μια εφαρμογή για μια συγκεκριμένη πλατφόρμα, είναι πολύ δύσκολο να γράψει γενικό κώδικα ανεξάρτητο της πλατφόρμας μεταγλώτισσης. Έτσι, υπάρχοντες υλοποιημένοι αλγόριθμοι και εφαρμογές πρέπει να τροποποιούνται κατάλληλα για να μπορούν να προσαρμοστούν σε διαφορετικά περιβάλλοντα ανάπτυξης. Μια απάντηση στο παραπάνω πρόβλημα δίνει η βιβλιοθήκη Wiselib [3] η οποία προσφέρει ένα προγραμματιστικό περιβάλλον για την ανάπτυξη γενικών αλγορίθμων που έχουν ως στόχο την εκτέλεσή τους σε ετερογενή δίκτυα αισθητήρων.

Σκοπός της παρούσας διπλωματικής εργασίας είναι να προσφέρει λύσεις στα δύο προαναφερθέντα προβλήματα, ανάπτυσσοντας κρυπτογραφικούς αλγόριθμους για ετερογενή ασύρματα δίκτυα αισθητήρων. Για την επίτευξη του σκοπού αυτού, αναπτύσουμε μια κρυπτογραφική βιβλιοθήκη στο προγραμματιστικό περιβάλλον της Wiselib, μιας γενικής βιβλιοθήκης αλγορίθμων για ετερογενή δίκτυα αισθητήρων.

Η Wiselib είναι υλοποιημένη σε C++ και με χρήση τεχνικών όπως τα πρότυπα και οι inline συναρτήσεις, επιτρέπει τη συγγραφή γενικού κώδικα ο οποίος αναλύεται και δεσμεύεται κατά τη διαδικασία μεταγλώττισης χωρίς να δημιουργείται πλεονασμός μνήμης ή υπολογισμού.

Λόγω των απαιτήσεων ασφαλείας που δημιουργούνται από τις εφαρμογές δικτύων αισθητήρων καθώς και των περιορισμένων υπολογιστικών πόρων, η κρυπτογραφική μας βιβλιοθήκη παρέχει αλγορίθμους τόσο συμμετρικής όσο και ασυμμετρικής κρυπτογραφίας. Οι αλγόριθμοι ασυμμετρικής κρυπτογραφίας βασίζονται στην κρυπτογραφία ελλειπτικών καμπυλών. Οι ελλειπτικές καμπύλες αποτελούν ένα ιδανικό σύστημα για ανάπτυξη κρυπτογραφίας δημοσίου κλειδιού σε ενσωματωμένα περιβάλλοντα τα οποία υστερούν σε επεξεργαστική ισχύ, μνήμη και ενέργεια. Αυτό ισχύει διότι τα συστήματα ελλειπτικών καμπυλών προσφέρουν το ίδιο επίπεδο ασφάλειας με άλλα κρυπτοσυστήματα (π.χ. RSA) με χρήση πολύ μικρότερου μεγέθους κλειδιών.

Έτσι, συνολικά η βιβλιοθήκη μας παρέχει τους εξής αλγορίθμους:

- τον αλγόριθμο συμμετρικής κρυπτογράφησης AES
- τον αλγόριθμο κατακερματισμού SHA-1
- το σχήμα συμφωνίας κλειδιών Diffie Hellman (ECDH)
- τον αλγόριθμο ασυμμετρικής κρυπτογράφησης ECIES
- το σχήμα ψηφιακής υπογραφής ECDSA

Για την ανάλυση της απόδοσης της κρυπτογραφικής μας βιβλιοθήκης γίνεται πειραματική αξιολόγηση (χρόνος εκτέλεσης, ενέργεια, μέγεθος μεταφρασμένου κώδικα) των παραπάνω αλγορίθμων σε δύο συσκευές (iSense, TelosB) με διαφορετικές επεξεργαστικές δυνατότητες (16 MHz, 8 MHz) που τρέχουν διαφορετικά λειτουργικά συστήματα (iSense OS, Contiki Sky). Το γεγονός ότι αξιολογήσαμε τους κρυπτογραφικούς αλγορίθμους σε δύο συσκευές διαφορετικών δυνατοτήτων και περιβαλλόντων ανάπτυξης, αποδεικνύει τη γενικότητα της υλοποίησης μας. Τέλος, για να αποδείξουμε την ευκολία χρήσης των υλοποιημένων αλγορίθμων παρουσιάζουμε τρεις εφαρμογές δικτύων αισθητήρων που τους χρησιμοποιούνε. Πιο συγκεκριμένα, επιδεικνύουμε πως οι κρυπτογραφικοί αλγόριθμοι μπορούν να συνδυαστούν με αλγορίθμους δρομολόγησης και ομαδοποίησης που παρέχει η βιβλιοθήκη Wiselib, με αποτέλεσμα να δημιουργηθούν ασφαλείς εφαρμογές δικτύων αισθητήρων.

Ευχαριστίες

Για την εκπόνηση της μεταπτυχιακής διπλωματικής αυτής εργασίας, θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή κ. Παύλο Σπυράκη καθώς και τους συνεπιβλέποντες Ιωάννη Σταματίου και Ιωάννη Χατζηγιαννάκη για την συστηματική υποστήριξη και καθοδήγησή τους. Η συνδρομή τους ήταν καθοριστική σε όλα τα επίπεδα.

Επίσης, επειδή με την εργασία αυτή, ολοκληρώνονται οι σπουδές μου ως μεταπτυχιακός φοιτητής του Πανεπιστημίου Πατρών, θα ήθελα να ευχαριστήσω όλους όσους με βοήθησαν τα χρόνια αυτά, ιδιαιτέρως την οικογένεια μου, που με υποστήριξε σε όλες μου τις αποφάσεις με κάθε τρόπο.

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Κεφάλαιο 1

Εισαγωγή

Ένα ασύρματο δίκτυο αισθητήρων αποτελείται από χωρικά κατανεμημένες αυτόνομες συσκευές, οι οποίες χρησιμοποιούν αισθητήρες για την συνεργατική παρακολούθηση φυσικών και περιβαλλοντικών συνθηκών. Ένας κόμβος ενός δικτύου αισθητήρων συνήθως αποτελείται από ένα μικροελεγκτή, ένα ραδιο-πομποδέκτη, μια πηγή ενέργειας, η οποία συχνά είναι μια μπαταρία και από διαφόρων ειδών αισθητήρες (π.χ. θερμοκρασίας, υγρασίας, φωτεινότητας).

Τα ασύρματα δίκτυα αισθητήρων προσφέρουν οικονομικά βιώσιμες λύσεις σε ποικίλες εφαρμογές. Δίκτυα τέτοιου τύπου δραστηριοποιούνται σε βιοιατρικές, στρατιωτικές, βιομηχανικές εφαρμογές καθώς και εφαρμογές παρακολούθησης του φυσικού περιβάλλοντος. Επιπλέον, τα δίκτυα αισθητήρων είναι κλειδί για τη δημιουργία *έξυπνων χώρων* που ενθέτουν τεχνολογία πληροφορίας στα καθημερινά περιβάλλοντα εργασίας και κατοικίας. Παρόλα αυτά, αρκετά θέματα ασφάλειας και μυστικότητας δημιουργούνται λόγω της ασύρματης φύσης επικοινωνίας καθώς και των περιορισμένων πόρων (υπολογιστική ισχύς, μνήμη, ενέργεια) που παρέχονται. Η συνεχής βελτίωση του υλικού και του λογισμικού αντιμετωπίζει κάποια από τα θέματα ασφαλείας αλλά πρέπει να δοθεί έμφαση και σε νέες τεχνολογίες που υποστηρίζονται από τις συσκευές.

Η αντιμετώπιση των ζητημάτων ασφαλείας που παρουσιάζονται σε ασύρματα δίκτυα αισθητήρων, ενώ παρουσιάζει ορισμένα κοινά στοιχεία με αυτήν που λαμβάνεται σε άλλα δικτυακά περιβάλλοντα (π.χ. ασφάλεια στο διαδίκτυο) έχει βασικές ποιοτικές διαφορές. Αυτό ισχύει λόγω των ιδιαίτερων χαρακτηριστικών των δικτύων αυτών και των κόμβων που τα απαρτίζουν όπως η χαμηλή υπολογιστική ισχύς, οι περιορισμένες δυνατότητες αποθήκευσης και η περιορισμένη διαθέσιμη ενέργεια. Επιπλέον, κάποια ιδιαίτερα χαρακτηριστικά των διαφόρων εφαρμογών τους όπως η λειτουργία σε αντίξοα περιβάλλοντα, η ελλιπής γνώση της τοπολογίας του δικτύου, οι δυνατότητες αυτο-οργάνωσης και αυτόματης διόρθωσης λειτουργιών και η λειτουργία χωρίς ανθρώπινη επιτήρηση καθιστούν τη διατήρηση της ασφάλειας μια μεγάλη πρόκληση.

Ένας κλάδος που παρέχει λύσεις σε προβλήματα ασφαλείας είναι αυτός της κρυπτογραφίας. Η κρυπτογραφία είναι μια καλά εδραιωμένη επιστημονική περιοχή, με πρωτόκολλα και πρότυπα τα οποία τυγχάνουν ευρείας αναγνώρισης. Παρόλα αυτά, η χρήση τους σε περιβάλλοντα περιορισμένων πόρων όπως αυτά των ασυρμάτων δικτύων αισθητήρων, απαιτεί προσαρμογές. Η απαίτηση αυτή προκύπτει λόγω των ιδιαίτερων χαρακτηριστικών των δικτύων αυτών και των κόμβων που τα απαρτίζουν όπως η χαμηλή υπολογιστική ισχύς, οι περιορισμένες δυνατότητες αποθήκευσης

και η περιορισμένη διαθέσιμη ενέργεια καθώς και η επικοινωνία ασύρματης φύσης που υιοθετείται.

Ένα ακόμα πρόβλημα που παρουσιάζεται στα ασύρματα δίκτυα αισθητήρων, παρά την ευρεία διάδοσή τους, είναι η ετερογένεια. Οι συσκευές αισθητήρων που παράγονται από τη βιομηχανία σήμερα, έχουν διαφορετικές υπολογιστικές δυνατότητες και εκτελούν διαφορετικά λειτουργικά συστήματα. Κάποιες αποτελούνται από μικροεπεξεργαστές 8-bit και έχουν ελάχιστη ποσότητα μνήμης RAM, ενώ άλλες έχουν πολύ μεγάλη υπολογιστική δύναμη και μπορούν να εκτελέσουν desktop λειτουργικά συστήματα όπως Linux. Έτσι, ενώ είναι σχετικά εύκολο να αναπτύξει κανείς μια εφαρμογή για μια συγκεκριμένη πλατφόρμα, είναι πολύ δύσκολο να γράψει γενικό κώδικα ανεξάρτητο της πλατφόρμας μεταγλώτισσης. Έτσι, υπάρχοντες υλοποιημένοι αλγόριθμοι και εφαρμογές πρέπει να τροποποιούνται κατάλληλα για να μπορούν να προσαρμοστούν σε διαφορετικά περιβάλλοντα ανάπτυξης. Μια απάντηση στο παραπάνω πρόβλημα δίνει η βιβλιοθήκη Wiselib η οποία προσφέρει ένα προγραμματιστικό περιβάλλον για την ανάπτυξη γενικών αλγορίθμων που έχουν ως στόχο την εκτέλεσή τους σε ετερογενή δίκτυα αισθητήρων.

1.1 Αντικείμενο της Διπλωματικής Εργασίας

Σκοπός της παρούσας διπλωματικής εργασίας είναι να προσφέρει λύσεις στα προαναφερθέντα προβλήματα, αναπτύσσοντας κρυπτογραφικούς αλγόριθμους για ετερογενή ασύρματα δίκτυα αισθητήρων. Για την επίτευξη του σκοπού αυτού, αναπτύσουμε μια κρυπτογραφική βιβλιοθήκη στο προγραμματιστικό περιβάλλον της Wiselib [3], μιας γενικής βιβλιοθήκης αλγορίθμων για ετερογενή δίκτυα αισθητήρων. Η Wiselib είναι υλοποιημένη σε C++ και με χρήση τεχνικών όπως τα πρότυπα και οι inline συναρτήσεις, επιτρέπει τη συγγραφή γενικού κώδικα ο οποίος αναλύεται και δεσμεύεται κατά τη διαδικασία μεταγλώττισσης χωρίς να δημιουργείται πλεονασμός μνήμης ή υπολογισμού.

Λόγω των απαιτήσεων ασφαλείας που δημιουργούνται από τις εφαρμογές δικτύων αισθητήρων καθώς και των περιορισμένων υπολογιστικών πόρων, η κρυπτογραφική μας βιβλιοθήκη παρέχει αλγορίθμους τόσο συμμετρικής όσο και ασυμμετρικής κρυπτογραφίας. Οι αλγόριθμοι ασυμμετρικής κρυπτογραφίας βασίζονται στην κρυπτογραφία ελλειπτικών καμπυλών. Οι ελλειπτικές καμπύλες αποτελούν ένα ιδανικό σύστημα για ανάπτυξη κρυπτογραφίας δημοσίου κλειδιού σε ενσωματωμένα περιβάλλοντα τα οποία υστερούν σε επεξεργαστική ισχύ, μνήμη και ενέργεια. Αυτό ισχύει διότι τα συστήματα ελλειπτικών καμπυλών προσφέρουν το ίδιο επίπεδο ασφάλειας με άλλα κρυπτοσυστήματα (π.χ. RSA) με χρήση πολύ μικρότερου μεγέθους κλειδιών. Έτσι, συνολικά η βιβλιοθήκη μας παρέχει τους εξής αλγορίθμους:

- τον αλγόριθμο συμμετρικής κρυπτογράφησης AES
- τον αλγόριθμο κατακερματισμού SHA-1
- το σχήμα συμφωνίας κλειδιών Diffie Hellman (ECDH)
- τον αλγόριθμο ασυμμετρικής κρυπτογράφησης ECIES
- το σχήμα ψηφιακής υπογραφής ECDSA

Για την ανάλυση της απόδοσης της κρυπτογραφικής μας βιβλιοθήκης γίνεται πειραματική αξιολόγηση (χρόνος εκτέλεσης, ενέργεια, μέγεθος μεταφρασμένου

κώδικα) των παραπάνω αλγορίθμων σε δύο συσκευές (iSense, TelosB) με διαφορετικές επεξεργαστικές δυνατότητες (16 MHz, 8 MHz) που τρέχουν διαφορετικά λειτουργικά συστήματα (iSense OS, Contiki Sky). Το γεγονός ότι αξιολογήσαμε τους κρυπτογραφικούς αλγορίθμους σε δύο συσκευές διαφορετικών δυνατοτήτων και περιβαλλόντων ανάπτυξης, αποδεικνύει τη γενικότητα της υλοποίησης μας. Τέλος, για να αποδείξουμε την ευκολία χρήσης των υλοποιημένων αλγορίθμων παρουσιάζουμε τρεις εφαρμογές δικτύων αισθητήρων που τους χρησιμοποιούνε. Πιο συγκεκριμένα, παρουσιάζουμε πως οι κρυπτογραφικοί αλγόριθμοι μπορούν να συνδυαστούν με αλγορίθμους δρομολόγησης και ομαδοποίησης που παρέχει η βιβλιοθήκη Wiselib, με αποτέλεσμα να δημιουργηθούν ασφαλείς εφαρμογές δικτύων αισθητήρων.

1.2 Διάρθρωση της Διπλωματικής Εργασίας

Αρχικά, στο **Κεφάλαιο 2** γίνεται μια εισαγωγή στην κρυπτογραφία και ειδικότερα στην κρυπτογραφία ελλειπτικών καμπυλών που αποτελεί ιδανικό σύστημα για υλοποίηση ασυμμετρικής κρυπτογραφίας σε ενσωματωμένα περιβάλλοντα. Το **Κεφάλαιο 3**, αναφέρεται σε υπάρχουσες κρυπτογραφικές βιβλιοθήκες που έχουν ως στόχο την εφαρμογή κρυπτογραφίας ελλειπτικών καμπυλών σε ασύρματα δίκτυα αισθητήρων και στα προβλήματα που παρουσιάζουν. Εν συνεχεία, στο **Κεφάλαιο 4** περιγράφουμε τη βιβλιοθήκη Wiselib στο περιβάλλον της οποίας αναπτύξαμε την κρυπτογραφική μας βιβλιοθήκη και στο **Κεφάλαιο 5** παρουσιάζουμε το υλικό το οποίο χρησιμοποιήθηκε για την πειραματική αξιολόγησή της. Στο **Κεφάλαιο 6** παρουσιάζουμε τους συμμετρικούς και ασυμμετρικούς κρυπτογραφικούς αλγόριθμους τους οποίους παρέχει η βιβλιοθήκη μας. Το **Κεφάλαιο 7** παρουσιάζει τα πειραματικά αποτελέσματα των αλγορίθμων αυτών σε πραγματικές συσκευές. Ακολούθως, το **Κεφάλαιο 8** περιγράφει τρεις εφαρμογές της κρυπτογραφικής μας βιβλιοθήκης οι οποίες αποδεικνύουν την ευκολία χρήσης της. Τέλος, στο **Κεφάλαιο 9** κλείνουμε την εργασία παραθέτοντας τα συμπεράσματα μας και τους στόχους για μελλοντική έρευνα.

Κεφάλαιο 2

Κρυπτογραφία

Η λέξη κρυπτογραφία προέρχεται από τα συνθετικά ΄κρυπτός΄ + ΄γράφω΄ και είναι ένας επιστημονικός κλάδος που ασχολείται με την μελέτη, την ανάπτυξη και την χρήση τεχνικών κρυπτογράφησης και αποκρυπτογράφησης με σκοπό την απόκρυψη του περιεχομένου των μηνυμάτων που ανταλάσσονται μεταξύ οντοτήτων. Ο κύριος στόχος της είναι να παρέχει μηχανισμούς ώστε δύο ή περισσότερα μέλη να επικοινωνήσουν χωρίς κάποιος άλλος να είναι ικανός να διαβάζει την πληροφορία που επικοινωνείται, εκτός από τα μέλη αυτά.

Ιστορικά, η κρυπτογραφία χρησιμοποιήθηκε για την κρυπτογράφηση μηνυμάτων δηλαδή μετατροπή της πληροφορίας από μια κανονική κατανοητή μορφή σε έναν γρίφο, που χωρίς την γνώση του κρυφού μετασχηματισμού θα παρέμενε ακατανόητος. Κύριο χαρακτηριστικό των παλαιότερων μορφών κρυπτογράφησης ήταν ότι η επεξεργασία γινόταν πάνω στην γλωσσική δομή. Στις νεότερες μορφές η κρυπτογραφία κάνει χρήση του αριθμητικού ισοδύναμου, η έμφαση έχει μεταφερθεί σε διάφορα πεδία των μαθηματικών, όπως διακριτά μαθηματικά, θεωρία αριθμών, θεωρία πληροφορίας, υπολογιστική πολυπλοκότητα, στατιστική και συνδυαστική ανάλυση.

Η κρυπτογραφία παρέχει τέσσερις βασικές λειτουργίες (Αντικειμενικοί σκοποί):

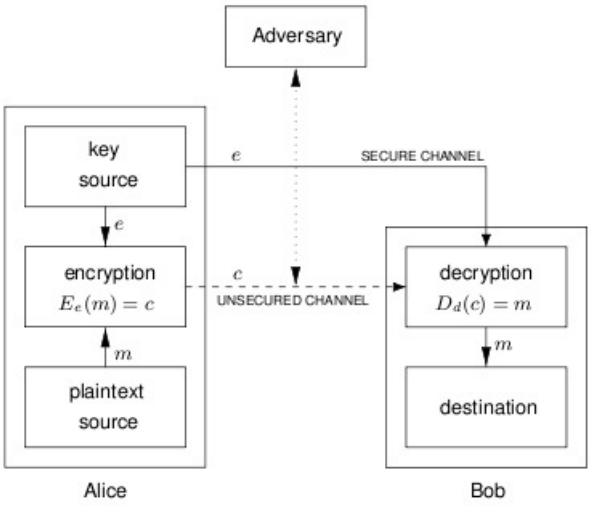
- **Εμπιστευτικότητα:** Η πληροφορία προς μετάδοση είναι προσβάσιμη μόνο στα εξουσιοδοτημένα μέλη. Η πληροφορία είναι ακατανόητη σε κάποιον τρίτο.
- **Ακεραιότητα:** Η πληροφορία μπορεί να αλλοιωθεί μόνο από τα εξουσιοδοτημένα μέλη και δεν μπορεί να αλλοιώνεται χωρίς την ανίχνευση της αλλοίωσης.
- **Μη απάρνηση:** Ο αποστολέας ή ο παραλήπτης της πληροφορίας δεν μπορεί να αρνηθεί την αυθεντικότητα της μετάδοσης ή της δημιουργίας της.
- **Πιστοποίηση:** Οι αποστολέας και παραλήπτης μπορούν να εξακριβώνουν τις ταυτότητές τους καθώς και την πηγή και τον προορισμό της πληροφορίας με διαβεβαίωση ότι οι ταυτότητές τους δεν είναι πλαστές.

Τα υπάρχοντα κρυπτοσυστήματα χωρίζονται σε δύο βασικές κατηγορίες : α) τα κρυπτοσυστήματα συμμετρικού κλειδιού στα οποία χρησιμοποιείται το ίδιο κλειδί για κρυπτογράφηση και αποκρυπτογράφηση και β) τα κρυπτοσυστήματα δημοσίου

κλειδιού στα οποία χρησιμοποιείται ένα δημόσιο κλειδί για κρυπτογράφηση και ένα μυστικό κλειδί για την αποκρυπτογράφηση.

2.1 Κρυπτογράφηση Συμμετρικού Κλειδιού

Ας θεωρήσουμε ένα σχήμα κρυπτογράφησης το οποίο αποτελείται από τα σύνολα των μετασχηματισμών κρυπτογράφησης και αποκρυπτογράφησης $\{E_e : e \in K\}$ και $\{D_d : d \in K\}$ αντίστοιχα, όπου K είναι ο χώρος κλειδιού (key space). Το σχήμα κρυπτογράφησης λέγεται *συμμετρικού κλειδιού* εαν για κάθε ζευγάρι κλειδιών (e, d) είναι υπολογιστικά εύκολο να βρεθεί το d γνωρίζοντας μόνο το e και αντιστοίχως να βρεθεί το e γνωρίζοντας το d. Στα πιο γνωστά κρυπτοσυστήματα συμμετρικού κλειδιού ισχύει ότι $d = e$. Άλλοι όροι που χρησιμοποιούνται είναι κρυπτογραφία ιδιωτικού κλειδιού και συμβατική κρυπτογραφία. Το Σχήμα 2.1 αναπαριστά την επικοινωνία δύο οντοτήτων που χρησιμοποιούν κρυπτογράφηση συμμετρικού κλειδιού.



Σχήμα 2.1: Επικοινωνία Δύο Μελών Χρησιμοποιώντας Συμμετρική Κρυπτογράφηση. Το Κλειδί Αποκρυπτογράφησης d Μπορεί να Υπολογιστεί Εύκολα από το Κλειδί Κρυπτογράφησης e.

Ένα μείζον θέμα στα συμμετρικά κρυπτοσυστήματα είναι να βρεθεί μία αποτελεσματική μέθοδος για την ασφαλή ανταλλαγή των κλειδιών μεταξύ των οντοτήτων. Το πρόβλημα αυτό είναι γνωστό ως *πρόβλημα διανομής κλειδιών*. Στην κρυπτογράφηση συμμετρικού κλειδιού υποτίθεται ότι όλα τα μέλη γνωρίζουν το σύνολο των μετασχηματισμών κρυπτογράφησης/αποκρυπτογράφησης. Όπως αναφέραμε και προηγουμένως η μόνη πληροφορία που παραμένει μυστική είναι το κλειδί d καθώς και το e αφού το d προκύπτει από αυτό. Υπάρχουν δύο βασικές κατηγορίες σχημάτων συμμετρικής κρυπτογράφησης: η block cipher και η stream cipher.

2.1.1 Block Cipher

Το block cipher αποτελεί ένα σχήμα κρυπτογράφησης το οποίο κατακερματίζει τα αρχικά μηνύματα και τα μεταδίδει σε μπλοκ δεδομένου μεγέθους t πάνω από ένα αλφάβητο A. Το σχήμα αυτό κρυπτογραφεί κάθε μπλοκ ξεχωριστά. Οι πιο γνωστές τεχνικές κρυπτογράφησης συμμετρικού κλειδιού ανήκουν στην κατηγορία αυτή όπως οι αλγόριθμοι DES, AES και RC5. Οι βασικές υποκατηγορίες της block cipher είναι οι substitution cipher και transposition cipher. Στην πρώτη υποκατηγορία, τα σύμβολα των μπλοκ δεδομένων αντικαθίστανται από άλλα σύμβολα ή ομάδες συμβόλων ενώ στη δεύτερη γίνεται μετάθεση των συμβόλων των μπλοκ δεδομένων.

2.1.2 Stream Cipher

Τα stream cipher αποτελούν μια σημαντική κατηγορία των σχημάτων κρυπτογράφησης συμμετρικού κλειδιού. Υπό μία έννοια, αποτελούν απλά block cipher με μέγεθος μπλοκ ίσο με ένα. Η ιδιότητα που τα κάνει ιδιαίτερα εύχρηστα είναι ότι ο μετασχηματισμός κρυπτογράφησης μπορεί να αλλάζει για κάθε διαφορετικό σύμβολο πληροφορίας που κρυπτογραφείται. Σε περιπτώσεις που τα λάθη μετάδοσης είναι πολύ συχνά, τα stream cipher έχουν πλεονέκτημα καθώς δεν έχουν σφάλμα διάδοσης. Επίσης, μπορούν να χρησιμοποιηθούν αποτελεσματικά όταν τα δεδομένα πρέπει να επεξεργάζονται ένα σύμβολο τη φορά όπως π.χ. σε περιπτώσεις που ο εξοπλισμός δεν διαθέτει επαρκή μνήμη για αποθήκευση δεδομένων. Για τον ορισμό των stream cipher είναι απαραίτητη η έννοια του keystream. Δεδομένου του χώρου κλειδιών K και ενός συνόλου μετασχηματισμών κρυπτογράφησης, μια ακολουθία συμβόλων $e_1e_2...e_i \in K$, ονομάζεται keystream. Ας θεωρήσουμε το A ως ένα αλφάβητο συμβόλων και E_e ένα απλό substitution cipher με μέγεθος μπλοκ ένα, όπου $e \in K$. Η ακολουθία $m_1m_2m_3...$ είναι το ακρυπτογράφητο αλφαριθμητικό και $e_1e_2e_3...$ είναι ένα keystream από το K. Ένα stream cipher μετατρέπει το ακρυπτογράφητο αλφαριθμητικό στο κρυπτογράφημα $c_1c_2c_3....$ όπου $c_i = E_{e_i}(m_i)$. Εάν το d_i αναπαριστά τον αντίστροφο του e_i τότε η συνάρτηση $D_{d_i}(c_i) = m_i$ αποκρυπτογραφεί το κρυπτογραφημένο μήνυμα.

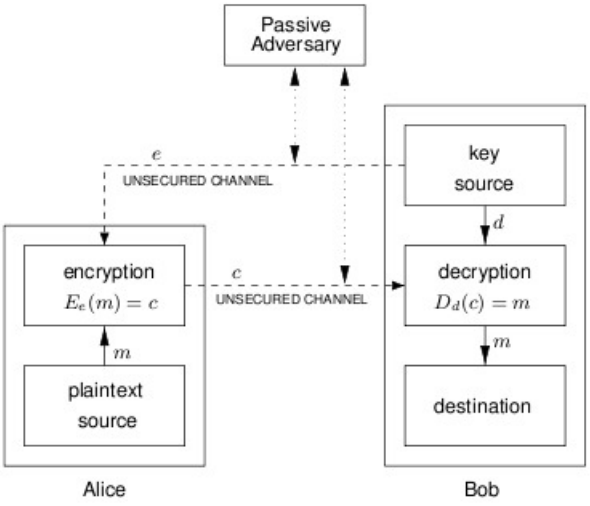
Ένα stream cipher υλοποιεί απλά σχήματα κρυπτογράφησης ανάλογα με το keystream που χρησιμοποιείται. Το keystream μπορεί να δημιουργείται τυχαία, ή από έναν αλγόριθμο ο οποίο γεννά νέα keystreams από ένα αρχικό που ονομάζεται seed. Οι πιο γνωστοί αλγόριθμοι stream cipher είναι οι RC4, FISH και Chameleon.

2.2 Κρυπτογράφηση Δημοσίου Κλειδιού

Ας θεωρήσουμε ένα σχήμα κρυπτογράφησης το οποίο αποτελείται από τα σύνολα των μετασχηματισμών κρυπτογράφησης και αποκρυπτογράφησης $\{E_e : e \in K\}$ και $\{D_d : d \in K\}$ αντίστοιχα, όπου K είναι ο χώρος κλειδιού(key space). Επίσης, ας θεωρήσουμε ένα τυχαίο ζευγάρι μετασχηματισμών κρυπτογράφησης/αποκρυπτογράφησης (E_e, D_d) και ας υποθέσουμε ότι για κάθε ζευγάρι είναι αδύνατο, γνωρίζοντας το E_e και δεδομένου ενός κρυπτογραφήματος $c \in C$, να υπολογίσουμε το μήνυμα $m \in M$ ώστε $E_e(m) = c$. Αυτή η ιδιότητα συνεπάγεται ότι δεδομένου του e είναι αδύνατο να καθορίσουμε το κλειδί αποκρυπτογράφησης d. Η συνάρτηση E_e θεωρείται σαν μία one-way trapdoor συνάρτηση, με το d να αποτελεί την trapdoor

πληροφορία η οποία είναι απαραίτητη για τον υπολογισμό την αντίστροφη συνάρτηση και να επιτραπεί η αποκρυπτογράφηση. Αυτό είναι διαφορετικό σε σχέση με την κρυπτογράφηση συμμετρικού κλειδιού που τα κλειδιά e και d είναι πρακτικά ίδια.

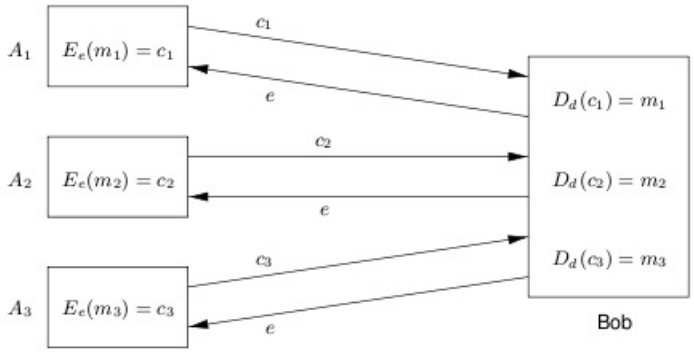
Υπό αυτές τις συνθήκες, ας θεωρήσουμε την επικοινωνία δύο οντοτήτων (Alice-Bob) όπως αυτή φαίνεται στο Σχήμα 2.2. Ο Bob επιλέγει το ζευγάρι κλειδιών (e, d) και στέλνει στην Alice το κλειδί e, που ονομάζεται δημόσιο κλειδί, μέσω ενός οποιοδήποτε καναλιού. Το κλειδί d, που ονομάζεται ιδιωτικό κλειδί, το κρατάει μυστικό και ασφαλές. Η Alice μπορεί να στείλει ένα μήνυμα m στον Bob εφαρμόζοντας το μετασχηματισμό κρυπτογράφησης που καθορίζεται από το δημόσιο κλειδί του Bob, $c = E_e(m)$. Ο Bob αποκρυπτογραφεί το μήνυμα c εφαρμόζοντας τον αντίστροφο μετασχηματισμό D_d που καθορίζεται από το d.



Σχήμα 2.2: Κρυπτογράφηση με Τεχνικές Κρυπτογραφίας Δημοσίου Κλειδιού.

Παρατηρούμε ότι το Σχήμα 2.2 σε σχέση με αυτό συμμετρικής κρυπτογράφησης διαφέρει. Σε αυτήν την περίπτωση, το κλειδί κρυπτογράφησης μεταδίδεται πάνω από ένα ανασφαλές κανάλι. Το κανάλι αυτό μπορεί να είναι το ίδιο με αυτό πάνω από το οποίο θα μεταδοθούν τα κρυπτογραφημένα δεδομένα. Από τη στιγμή που το κλειδί κρυπτογράφησης e δε χρειάζεται να παραμένει μυστικό γίνεται κοινώς γνωστό και οποιαδήποτε οντότητα μπορεί να στείλει κρυπτογραφημένα μηνύματα στον Bob, ο οποίος είναι ο μόνος που μπορεί να τα αποκρυπτογραφήσει. Το Σχήμα 2.3 απεικονίζει αυτήν την ιδέα, θεωρώντας τα $A1, A2$ και $A3$ ως διακριτές οντότητες.

Σαν ένα φυσικό ανάλογο της κρυπτογραφίας δημοσίου κλειδιού, ας θεωρήσουμε ένα μεταλλικό κουτί του οποίου η κλειδαρία προστατεύεται από κάποιον μυστικό συνδυασμό, τον οποίο μόνο ο Bob γνωρίζει. Αν η κλειδαρία αφεθεί δημοσίως ανοιχτή, οποιοσδήποτε μπορεί να τοποθετήσει ένα μήνυμα μέσα και κατόπιν να την κλειδώσει. Μόνο ο Bob μπορεί να λάβει το μήνυμα αφού είναι ο μόνος που ξέρει το μυστικό συνδυασμό της κλειδαριάς και μπορεί να την ανοίξει. Ακόμα και η οντότητα που τοποθέτησε το μήνυμα μέσα στο κουτί δεν μπορεί να το ανακτήσει.



Σχήμα 2.3: Σχηματική Απεικόνιση της Κρυπτογραφίας Δημοσίου Κλειδιού.

Η κρυπτογραφία δημοσίου κλειδιού προυποθέτει τη γνώση ενός δημοσίου κλειδιού e η οποία όμως δεν επιτρέπει τον υπολογισμό του ιδιωτικού κλειδιού d.

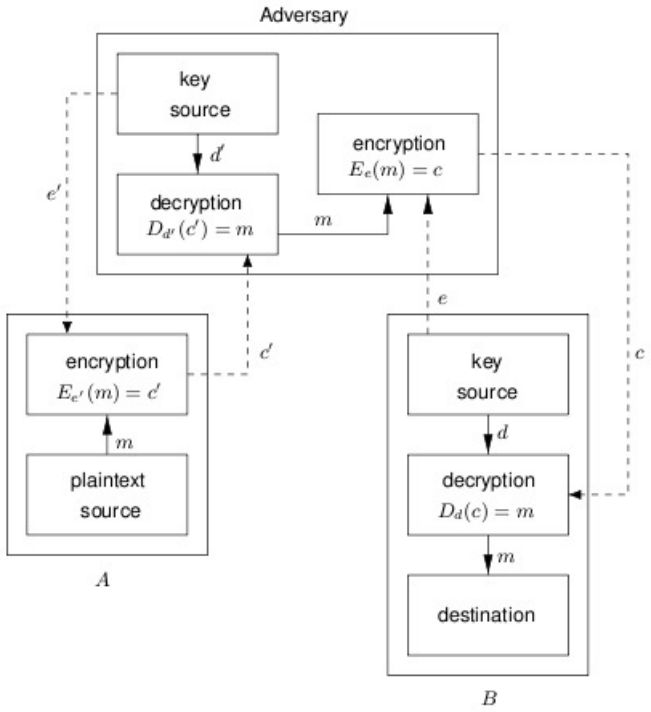
Ας θεωρήσουμε ένα σχήμα κρυπτογράφησης το οποίο αποτελείται από τα σύνολα των μετασχηματισμών κρυπτογράφησης και αποκρυπτογράφησης $\{E_e : e \in K\}$ και $\{D_d : d \in K\}$ αντίστοιχα, όπου K είναι ο χώρος κλειδιού(key space). Η μέθοδος κρυπτογράφησης θεωρείται σχήμα κρυπτογράφησης δημοσίου κλειδιού εάν για κάθε ζευγάρι (e, d) το κλειδί e γίνεται δημοσίως γνωστό (δημόσιο κλειδί) ενώ το d παραμένει μυστικό (ιδιωτικό κλειδί). Για τη διατήρηση της ασφάλειας αυτού του σχήματος, θα πρέπει να είναι υπολογιστικά δύσκολο να υπολογιστεί το d από το e.

Τα πιο γνωστά κρυπτοσυστήματα δημοσίου κλειδιού είναι το RSA, το Diffie-Hellman, το El-Gamal και το DSA.

2.2.1 Η Ανάγκη για Αυθεντικότητα στα Κρυπτοσυστήματα Δημοσίου Κλειδιού

Υπό κάποιες συνθήκες, η κρυπτογραφία δημοσίου κλειδιού θα φαινόταν ως ιδανικό κρυπτοσύστημα καθώς δεν απαιτεί ένα ασφαλές κανάλι για τη μετάδοση του κλειδιού κρυπτογράφησης. Αυτό θα συνεπαγόταν ότι δύο οντότητες θα μπορούσαν να επικοινωνήσουν πάνω από ένα ανασφαλές κανάλι χωρίς να ανταλλάξουν κλειδιά. Δυστυχώς, κάτι τέτοιο δεν είναι αλήθεια. Το Σχήμα 2.4 παρουσιάζει πως ένας ενεργός επιτιθέμενος μπορεί να νικήσει το κρυπτοσύστημα χωρίς *σπάσει* το μηχανισμό κρυπτογράφησης. Η επίθεση που παρουσιάζεται είναι ένας τύπος προσωποποίησης(impersonation) και είναι ένα παράδειγμα αποτυχίας των συστημάτων δημοσίου κλειδιού. Στο σενάριο αυτό, ο επιτιθέμενος προσποιείται στον A ότι είναι η οντότητα B στέλνοντας του ένα κλειδί e' το οποίο ο A υποθέτει ότι είναι το δημόσιο κλειδί του B. Με τον τρόπο αυτό ο επιτιθέμενος λαμβάνει τα μηνύματα που θέλει ο A να στείλει στο B, κρυπτογραφημένα με το δικό του δημόσιο κλειδί. Τα αποκρυπτογραφεί με το ιδιωτικό του κλειδί d' και στη συνέχεια τα κρυπτογραφεί με το δημόσιο κλειδί του B και του τα στέλνει. Το γεγονός αυτό τονίζει την ανάγκη για αυθεντικότητα προέλευσης των δεδομένων στα κρυπτοσυστήματα δημοσίου κλειδιού. Η οντότητα A πρέπει να είναι πεπεισμένη ότι κρυπτογραφεί τα

δεδομένα της με το έντιμο δημόσιο κλειδί της οντότητας B.



Σχήμα 2.4: Μία Επίθεση Προσωποποίησης στην Επικοινωνία δύο Οντοτήτων.

2.3 Κρυπτογραφία Συμμετρικού Κλειδιού Εναντίον Κρυπτογραφίας Δημοσίου Κλειδιού

Τα σχήματα κρυπτογράφησης συμμετρικού κλειδιού και δημοσίου κλειδιού παρουσιάζουν διάφορα πλεονεκτήματα και μειονεκτήματα, κάποια από τα οποία είναι κοινά και στα δύο. Παρακάτω αναφέρονται τα βασικά θετικά και αρνητικά των σχημάτων αυτών.

2.3.1 Πλεονεκτήματα Κρυπτογραφίας Συμμετρικού Κλειδιού

- Τα κρυπτοσυστήματα συμμετρικού κλειδιού μπορούν να σχεδιαστούν έτσι ώστε να έχουν υψηλούς ρυθμούς απόδοσης δεδομένων όπως π.χ. ρυθμούς κρυπτογράφησης δεδομένων της τάξης εκατοντάδων Megabytes το δευτερόλεπτο.
- Τα κλειδιά που χρησιμοποιούνται στη συμμετρική κρυπτογράφηση είναι σχετικά μικρά.

- Τα κρυπτογραφήματα συμμετρικού κλειδιού μπορούν να αποτελέσουν τη βάση για τη δημιουργία διαφόρων κρυπτογραφικών μηχανισμών όπως γεννήτριες ψευδο-τυχαίων αριθμών, συναρτήσεις κατακερματισμού και σχήματα ψηφιακής υπογραφής.
- Τα κρυπτογραφήματα συμμετρικού κλειδιού μπορούν να συνθέσουν δυνατότερα κρυπτογραφήματα με απλούς μετασχηματισμούς.
- Η κρυπτογραφία συμμετρικού κλειδιού έχει εκτεταμένη και πλούσια ιστορία.

2.3.2 Μειονεκτήματα Κρυπτογραφίας Συμμετρικού Κλειδιού

- Κατά την επικοινωνία δύο οντοτήτων, το κλειδί πρέπει να παραμείνει μυστικό και στα δύο άκρα επικοινωνίας.
- Σε δίκτυα μεγάλου μεγέθους, υπάρχουν πολλά ζευγάρια κλειδιών που πρέπει να διαχειριστούν. Αυτό συνεπάγεται ότι η αποτελεσματική διαχείριση κλειδιών απαιτεί τη χρήση έμπιστης τρίτης οντότητας (Trusted Third Party).
- Κατά την επικοινωνία δύο οντοτήτων, το κλειδί πρέπει να αλλάζει πολύ συχνά, ενδεχομένως και πριν από κάθε εκ νέου επικοινωνία.
- Οι μηχανισμοί ψηφιακής υπογραφής που προκύπτουν από την κρυπτογράφηση συμμετρικού κλειδιού, απαιτούν μεγαλύτερα κλειδιά για τη συνάρτηση δημόσιας επαλήθευσης και για τη χρήση έμπιστης τρίτης οντότητας.

2.3.3 Πλεονεκτήματα Κρυπτογραφίας Δημοσίου Κλειδιού

- Μόνο το ιδιωτικό κλειδί πρέπει να διατηρείται μυστικό. Παρόλα αυτά, πρέπει να εξασφαλίζεται η αυθεντικότητα των δημοσίων κλειδιών.
- Η διαχείριση των κλειδιών απαιτεί την παρουσία μιας έμπιστης τρίτης οντότητας, η οποία ανάλογα τον τρόπο λειτουργίας, μπορεί να χρειάζεται μόνο για off-line χρήση.
- Ανάλογα με τον τρόπο λειτουργίας, ένα ζευγάρι ιδιωτικού/δημοσίου κλειδιού μπορεί να παραμείνει χωρίς αλλαγές για αξιοσημείωτο χρονικό διάστημα.
- Πολλά σχήματα δημοσίου κλειδιού συνεπάγονται αποτελεσματικούς μηχανισμούς ψηφιακής υπογραφής. Το κλειδί που χρησιμοποιείται για την συνάρτηση δημόσιας επαλήθευσης είναι πολύ μικρότερο από το αντίστοιχο συμμετρικό κλειδί.
- Σε ένα μεγάλο δίκτυο, ο αριθμός των απαραίτητων κλειδιών είναι σημαντικά μικρότερος από τον αντίστοιχο ενός σενάριου συμμετρικού κλειδιού.

2.3.4 Μειονεκτήματα Κρυπτογραφίας Δημοσίου Κλειδιού

- Η ρυθμαπόδοση των πιο γνωστών σχημάτων κρυπτογραφίας δημοσίου κλειδιού είναι τάξεις μικρότερη από αυτήν των σχημάτων συμμετρικού κλειδιού.

- Τα μεγέθη των κλειδιών είναι αρκετά μεγαλύτερα από τα αντίστοιχα εκείνων που απαιτούνται στην κρυπτογραφία συμμετρικού κλειδιού.
- Δεν έχει βρεθεί σχήμα κρυπτογραφίας δημοσίου κλειδιού που να έχει αποδειχθεί ασφαλές(το ίδιο ισχύει και για τα block ciphers). Τα πιο αποτελεσματικά τέτοια σχήματα κρυπτογράφησης βασίζονται σε προβλήματα της θεωρίας αριθμών.
- Η κρυπτογραφία δημοσίου κλειδιού δεν έχει τόσο εκτενή ιστορία όσο η συμμετρική κρυπτογραφία, αφού ανακαλύφθηκε τη δεκαετία του 1970.

2.3.5 Περίληψη της Σύγκρισης

Η συμμετρική κρυπτογραφία καθώς και η κρυπτογραφία δημοσίου κλειδιού έχουν συμπληρωματικά πλεονεκτήματα. Τα τρέχοντα κρυπτοσυστήματα προσπαθούν να εκμεταλλευτούν τις δυνατότητες της καθεμίας και το σκεπτικό αυτό φαίνεται με το παρακάτω παράδειγμα.

Οι τεχνικές κρυπτογραφίας δημοσίου κλειδιού μπορούν να χρησιμοποιηθούν για τη δημιουργία ενός συμμετρικού κλειδιού που χρησιμοποιείται από τις οντότητες A και B που θέλουν να επικοινωνήσουν. Στο σενάριο αυτό οι οντότητες A και B εκμεταλλεύονται τη μακρόχρονη φύση των δημοσίων/ιδιωτικών κλειδιών του σχήματος κρυπτογραφίας δημοσίου κλειδιού και την αποδοτικότητα του σχήματος συμμετρικού κλειδιού. Από τη στιγμή που η κρυπτογράφηση των δεδομένων είναι συνήθως το πιο χρονοβόρο κομμάτι της συνολικής διαδικασίας, το σχήμα δημοσίου κλειδιού για την δημιουργία κλειδιού αποτελεί ένα μικρό κομμάτι της διαδικασίας κρυπτογράφησης μεταξύ των οντοτήτων A και B.

Για να συνοψίσουμε, τα δύο βασικά συμπεράσματα της σύγκρισης αυτής είναι:

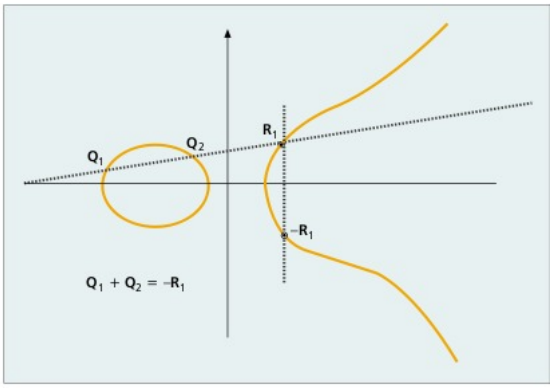
- η κρυπτογραφία δημοσίου κλειδιού διευκολύνει τις διαδικασίες ψηφιακής υπογραφής και διαχείρισης κλειδιών
- η κρυπτογραφία συμμετρικού κλειδιού είναι αποτελεσματική για κρυπτογράφηση δεδομένων και εφαρμογές που στοχεύουν στην ακεραιότητα των δεδομένων.

2.4 Οι Ελλειπτικές Καμπύλες στην Κρυπτογραφία

Οι ελλειπτικές καμπύλες αποτελούν μια διαφορετική προσέγγιση για την κρυπτογραφία δημοσίου κλειδιού και βασίζονται στην αλγεβρική δομή καμπυλών πάνω από πεπερασμένα πεδία. Η χρήση των ελλειπτικών καμπυλών στη κρυπτογραφίαν προτάθηκε ανεξαρτήτως από τους Neal Koblitz και Victor S. Miller το 1985.

Οι ελλειπτικές καμπύλες ως αλγεβρικές/γεωμετρικές οντότητες έχουν μελετηθεί εκτενώς τα τελευταία 150 χρόνια. Από τις μελέτες αυτές έχει προκύψει μια πλούσια θεωρία. Κάθε κρυπτοσύστημα βασίζεται σε αλγεβρικές ομάδες για την εκτέλεση των αριθμητικών του πράξεων. Μια αλγεβρική ομάδα είναι το σύνολο των στοιχείων στα οποία επιτρέπονται ορισμένες αριθμητικές πράξεις. Για τις ομάδες ελλειπτικών καμπυλών, αυτές οι πράξεις ορίζονται γεωμετρικά. Για παράδειγμα, για την πρόσθεση των σημείων Q_1, Q_2 στην καμπύλη που φαίνεται στο Σχήμα 2.5, τραβάμε μια ευθεία γραμμή που τα ενώνει και βρίσκουμε το σημείο R_1 στο οποίο

η ευθεία αυτή τέμνει την καμπύλη. Το σημείο $-R_1$, το οποίο αποτελεί το αντίθετο σημείο του R_1 ως προς τον άξονα x, είναι το αποτέλεσμα της πρόσθεσης των σημείων Q_1, Q_2, δηλαδή $Q_1 + Q_2 = -R_1$. Η εισαγωγή αυστηρών ιδιοτήτων στα στοιχεία μιας ομάδας ελλειπτικών καμπυλών, δημιουργεί ένα υποκείμενο πεδίο. Εν συνεχεία, αναφερόμαστε στη θεωρία ελλειπτικών καμπυλών ορισμένες στο πεδίο F_p, όπου p ένα πρώτος αριθμός καθώς και στο πεδίο F_{2^m}.



Σχήμα 2.5: Γεωμετρική Αναπαράσταση της Πρόσθεσης Δυο Σημείων Πάνω σε Μια Ελλειπτική Καμπλύλη.

2.4.1 Ελλειπτικές Καμπύλες Ορισμένες στο F_p

Ας θεωρήσουμε ένα πεπερασμένο πεδίο F_p, όπου p είναι ένας περιττός πρώτος αριθμός. Επιπλέον, ας θεωρήσουμε τις ποσότητες $a, b \in F_p$ οι οποίες ικανοποιούν τη σχέση $4 \cdot a^3 + 27 \cdot b^2 \neq 0$. Μια ελλειπτική καμπύλη $E(F_p)$ που ορίζεται από τις παραμέτρους $a, b \in F_p$, αποτελείται απο το σύνολο των σημείων $P = (x, y)$ για $x, y \in F_p$ τα οποία ικανοποιούν την εξίσωση:

$$y^2 = x^3 + a \cdot x + b \, (mod \, p) \tag{2.1}$$

μαζί με ένα σημείο O που ονομάζεται σημείο απειρίας. Η εξίσωση 2.1 ονομάζεται η καθοριστική εξίσωση της καμπύλης $E(F_p)$. Για ένα δεδομένο σημείο $P = (x_P, y_P)$, η ποσότητα x_P ονομάζεται x-συντεταγμένη και η ποσότητα y_P ονομάζεται y-συντεταγμένη του σημείου P. Ο αριθμός σημείων πάνω στη καμπύλη $E(F_p)$ συμβολίζεται με $\sharp E(F_p)$. Σύμφωνα με το θεώρημα του Hasse ισχύει ότι

$$p + 1 - 2 \cdot \sqrt{p} \leq \sharp E(F_p) \leq p + 1 + 2 \cdot \sqrt{p}.$$

Ο ορισμός της πράξης πρόσθεσης σημείων πάνω στην καμπύλη E αποτελείται από τους εξής κανόνες:

- Κανόνας για την πρόσθεση του σημείο απειρίας με τον εαυτό του:

$$O + O = O.$$

- Κανόνας για την πρόσθεση του σημείου απειρίας με οποιοδήποτε άλλο σημείο (x, y):

$$(x, y) + O = O + (x, y) = (x, y)$$

για όλα τα $(x, y) \in E(F_p)$.

- Κανόνας για την πρόσθεση δύο σημείων με την ίδια x-συντεταγμένη που είναι όμως διαφορετικά ή έχουν μηδενική y-συντεταγμένη:

$$(x, y) + (x, -y) = O$$

για όλα τα $(x, y) \in E(F_p)$. Ο κανόνας αυτός υποδεικνύει ότι το αρνητικό ενός σημείου (x, y) είναι το $-(x, y) = (x, -y)$.

- Κανόνας για την πρόσθεση δύο σημείων με διαφορετική x-συντεταγμένη: Ας θεωρήσουμε τα σημεία $(x_1, y_1) \in F_p$ και $(x_2, y_2) \in F_p$ έτσι ώστε $x_1 \neq x_2$. Τότε, $(x_1, y_1) + (x_2, y_2) = (x_3, y_3)$ όπου:

$$x_3 = \lambda^2 - x_1 - x_2 \,(mod\, p),\ y_3 = \lambda \cdot (x_1 - x_3) - y_1 \,(mod\, p),\ \lambda = \tfrac{y_2 - y_1}{x_2 - x_1} \,(mod\, p)$$

- Κανόνας για το διπλασιασμό ενός σημείου: Ας θεωρήσουμε ένα σημείο $(x_1, y_1) \in E(F_p)$ με $y_1 \neq 0$. Τότε, $(x_1, y_1) + (x_1, y_1) = (x_3, y_3)$ όπου:

$$x_3 = \lambda^2 - 2 \cdot x_1 \,(mod\, p),\ y_3 = \lambda \cdot (x_1 - x_3) - y_1 \,(mod\, p),\ \lambda = \tfrac{3 \cdot x_1^2 + a}{2 \cdot y_1} \,(mod\, p)$$

Το σύνολο σημείων πάνω στην καμπύλη $E(F_p)$ σχηματίζουν μια ομάδα με βάση την πράξη πρόσθεσης. Επιπλέον, η ομάδα αυτή είναι αβελιανή (abelian) καθώς $P_1 + P_2 = P_2 + P_1$ για όλα τα σημεία $P_1, P_2 \in E(F_p)$.

Τα κρυπτογραφικά σχήματα που βασίζονται στις ελλειπτικές καμπύλες στηρίζονται στο βαθμωτό πολλαπλασιασμό των σημείων της καμπύλης. Δεδομένου ενός ακεραίου k και ενός σημείου $P \in E(F_p)$, ο βαθμωτός πολλαπλασιασμός είναι η διαδικασία πρόσθεσης του σημείου P στον εαυτό του k φορές. Το αποτέλεσμα αυτής της πράξης συμβολίζεται ως $k \cdot P$. Ο βαθμωτός πολλαπλασιασμός των σημείων μιας ελλειπτικής καμπύλης μπορεί να υπολογιστεί αποδοτικά χρησιμοποιώντας το κανόνα πρόσθεσης σε συνδυασμό με τον κανόνα διπλασιασμού ενός σημείου.

2.4.2 Ελλειπτικές Καμπύλες Ορισμένες στο F_{2^m}

Ας θεωρήσουμε ένα πεπερασμένο πεδίο F_{2^m} το οποίο αποτελείται από 2^m στοιχεία. Επιπλέον, ας θεωρήσουμε τις ποσότητες $a, b \in F_{2^m}$ για τις οποίες ισχύει ότι $b \neq 0$. Μια ελλειπτική καμπύλη που ορίζεται από τις παραμέτρους $a, b \in F_{2^m}$, αποτελείται από το σύνολο των σημείων $P = (x, y)$ για $x, y \in F_{2^m}$ τα οποία ικανοποιούν την εξίσωση:

$$y^2 + x \cdot y = x^3 + a \cdot x^2 + b \tag{2.2}$$

μαζί με ένα σημείο O που ονομάζεται σημείο απειρίας. Ο αριθμός των σημείων πάνω στην καμπύλη $E(F_{2^m})$ συμβολίζεται ως $\sharp E(F_{2^m})$. Σύμφωνα με το θεώρημα του Hasse ισχύει η σχέση :

$$2^m + 1 - 2 \cdot \sqrt{2^m} \leq \sharp E(F_{2^m}) \leq 2^m + 1 + 2 \cdot \sqrt{2^m}.$$

Ο ορισμός της πράξης πρόσθεσης σημείων πάνω στην καμπύλη E αποτελείται από τους εξής κανόνες:

- Κανόνας για την πρόσθεση του σημείο απειρίας με τον εαυτό του:

$$O + O = O.$$

- Κανόνας για την πρόσθεση του σημείου απειρίας με οποιοδήποτε άλλο σημείο (x, y):

$$(x, y) + O = O + (x, y) = (x, y)$$

για όλα τα $(x, y) \in E(F_{2^m})$.

- Κανόνας για την πρόσθεση δύο σημείων με την ίδια x-συντεταγμένη που είναι όμως διαφορετικά ή έχουν μηδενική x-συντεταγμένη:

$$(x, y) + (x, x + y) = O$$

για όλα τα $(x, y) \in E(F_{2^m})$. Ο κανόνας αυτός υποδεικνύει ότι το αρνητικό ενός σημείου (x, y) είναι το $-(x, y) = (x, x + y)$.

- Κανόνας για την πρόσθεση δύο σημείων με διαφορετική x-συντεταγμένη: Ας θεωρήσουμε τα σημεία $(x_1, y_1) \in E(F_{2^m})$ και $(x_2, y_2) \in E(F_{2^m})$ έτσι ώστε να ισχύει $x_1 \neq x_2$. Τότε, $(x_1, y_1) + (x_2, y_2) = (x_3, y_3)$, όπου:

$$x_3 = \lambda^2 + \lambda + x_1 + x_2 + a \in F_{2^m},\ y_3 = \lambda \cdot (x_1 + x_3) + x_3 + y_1 \in F_{2^m},\ \lambda = \frac{y_1 + y_2}{x_1 + x_2} \in F_{2^m}$$

- Κανόνας για το διπλασιασμό ενός σημείου: Ας θεωρήσουμε ένα σημείο $(x_1, y_1) \in E(F_{2^m})$ με $x_1 \neq 0$. Τότε, $(x_1, y_1) + (x_1, y_1) = (x_3, y_3)$ όπου:

$$x_3 = \lambda^2 + \lambda + a \in F_{2^m},\ y_3 = x_1^2 + (\lambda + 1) \cdot x_3 \in F_{2^m},\ \lambda = x_1 + \frac{y_1}{x_1} \in F_{2^m}$$

Το σύνολο σημείων πάνω στην καμπύλη $E(F_{2^m})$ σχηματίζουν μια ομάδα με βάση την πράξη πρόσθεσης. Επιπλέον, η ομάδα αυτή είναι αβελιανή (abelian) καθώς $P_1 + P_2 = P_2 + P_1$ για όλα τα σημεία $P_1, P_2 \in E(F_p)$.

2.4.3 Ασφάλεια των Ελλειπτικών Καμπυλών

Πολλαπλασιασμός Σημείου Ελλειπτικής Καμπύλης

Η κυρίαρχη πράξη στα κρυπτοσυστήματα ελλειπτικών καμπυλών είναι ο βασθμωτός πολλαπλασιασμός σημείου. Αυτή η πράξη αποτελεί το κλειδί για τη χρήση των ελλειπτικών καμπυλών στην ασυμμετρική κρυπτογραφία. Είναι η πράξη η οποία είναι εύκολη να υπολογιστεί αλλά η αντίστροφή της (το πρόβλημα του διακριτού λογαρίθμου πάνω από ελλειπτικές καμπύλες) είναι υπολογιστικά δύσκολη.

Ο πολλαπλασιασμός σημείου είναι ο υπολογισμός της ποσότητας $k \cdot P$, όπου k είναι ένας ακέραιος στο υποκείμενο πεδίο και P είναι ένα προκαθορισμένο σημείο της ελλειπτικής καμπύλης. Χρησιμοποιώντας τους κανόνες πρόσθεσης και διπλασιασμού σημείου του πεδίου πάνω από το οποίο είναι ορισμένη μια ελλειπτική καμπύλη, ο πολλαπλασιασμός σημείου μπορεί να υπολογιστεί αποδοτικά.

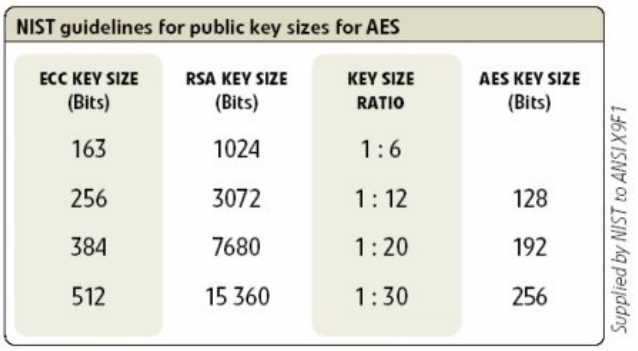
Το Πρόβλημα του Διακριτού Λογαρίθμου Πάνω από Ελλειπτικές Καμπύλες

Το πρόβλημα του διακριτού λογαρίθμου ορίζεται πάνω από τυχαίες κυκλικές πεπερασμένες ομάδες. Ένα παράδειγμα τέτοιων ομάδων είναι η πολλαπλασιαστική ομάδα Z_n^*, τάξης n όπου n πρώτος αριθμός. Η βασική πράξη σε αυτήν την ομάδα είναι ο πολλαπλασιασμός με υπόλοιπο n. Σε μια τέτοια ομάδα, το πρόβλημα του διακριτού λογαρίθμου ορίζεται ως εξής: δεδομένου ενός πρώτου αριθμού n, ενός γεννήτορα $g \in Z_n^*$ και ενός στοιχείου $b \in Z_n^*$, να βρεθεί ένας ακέραιος $x, 0 \leq x \leq n-2$ έτσι ώστε να ισχύει ότι $g^x = b \, (mod \, n)$.

Ένα άλλο παράδειγμα κυκλικών ομάδων είναι οι ομάδες ελλειπτικών καμπυλών ορισμένες πάνω από μια αθροιστική ομάδα F τάξεως n, ο οποίος δεν είναι απαραίτητα πρώτος. Το πρόβλημα του διακριτού λογαρίθμου πάνω από ελλειπτικές ομάδες ορίζεται ώς εξής: δεδομένης μιας ελλειπτικής καμπύλης E πάνω από ένα πεδίο F_n, ενός σημείου γεννήτορα $G \in E/F_n$ πάνω στη καμπύλη και ενός σημείου $B \in F_n$, να βρεθεί ένας ακέραιος x έτσι ώστε να ισχύει $B = x \cdot G$.

2.4.4 Ελλειπτικές Καμπύλες και Συσκευές Περιορισμένων Πόρων

Σε σχέση με άλλα κρυπτοσυστήματα όπως το Diffie-Hellman και το RSA, το βασικό πλεονέκτημα των ελλειπτικών καμπυλών είναι το εξής: η αντίστροφη πράξη γίνεται πιο δύσκολη, πιο γρήγορα, καθώς αυξάνεται το μήκος του κλειδίου. Αυτό σημαίνει ότι καθώς οι απαιτήσεις ασφαλείας γίνονται πιο αυστηρές, οι ελλειπτικές καμπύλες αποτελούν το πιο πρακτικό σύστημα. Δηλαδή, συστήματα βασισμένα στις ελλειπτικές καμπύλες μπορούν να χρησιμοποιήσουν μικρότερα κλειδιά και να προσφέρουν το ίδιο επίπεδο ασφάλειας με άλλα κρυπτοσυστήματα. Επιπλέον, το χάσμα μεταξύ των συστημάτων ελλειπτικών καμπυλών και των ανταγωνιστών τους σχετικά με το μήκος κλειδιού, γίνεται δραματικά μεγαλύτερο όσο το επίπεδο ασφάλειας αυξάνεται. Το Σχήμα 2.6, δείχνει τα μήκη κλειδιών για την επίτευξη ίδιου επιπέδου ασφάλειας, μεταξύ ελλειπτικών καμπυλών και RSA σύμφωνα με τον οργανισμό NIST (National Institute of Standards and Technology). Σαν ένα παράδειγμα, αναφέρουμε ότι το προτεινόμενο μέγεθος κλειδιού για εφαρμογές του συστήματος RSA είναι 2048 bits ενώ χρησιμοποιώντας ελλειπτικές καμπύλες απαιτείται κλειδί μεγέθους μόλις 224 bits.



NIST guidelines for public key sizes for AES

ECC KEY SIZE (Bits)	RSA KEY SIZE (Bits)	KEY SIZE RATIO	AES KEY SIZE (Bits)
163	1024	1 : 6	
256	3072	1 : 12	128
384	7680	1 : 20	192
512	15 360	1 : 30	256

Supplied by NIST to ANSI X9F1

Σχήμα 2.6: Μήκη Κλειδιών για την Επίτευξη Ίδιου Επιπέδου Ασφάλειας Μεταξύ Ελλειπτικών Καμπυλών και RSA.

Το παραπάνω πλεονέκτημα, είναι ο λόγος για τον οποίο οι ελλειπτικές καμπύλες αποτελούν μια ιδανική επιλογή για την υλοποίηση ασσυμετρικής κρυπτογραφίας σε ενσωματωμένα συστήματα όπου οι υπολογιστικοί πόροι είναι περιορισμένοι. Οι μικρότερες παράμετροι που απαιτούνται για τα συστήματα ελλειπτικών καμπυλών, υποδεικνύουν ότι οι κρυπτογραφικές πράξεις μπορούν να εκτελεστούν σε μικρότερο υλικό. Επιπλέον, οι κρυπτογραφικές πράξεις μπορούν να εκτελεστούν σε λιγότερους κύκλους ενός επεξεργαστή καθώς και με λιγότερες απαιτήσεις μνήμης. Τα γεγονότα αυτά συνεπάγονται λιγότερη παραγόμενη θερμότητα και συνεπώς λιγότερη κατανάλωση ενέργειας στα ολοκληρωμένα κυκλώματα του υλικού που θα χρησιμοποιηθεί. Τέλος, οι εφαρμογές λογισμικού που βασίζονται σε συστήματα ελλειπτικών καμπυλών έχουν λίγες απαιτήσεις υπολογιστικής δύναμης, μνήμης και ενέργειας. Οι λόγοι αυτοί καθιστούν τις ελλειπτικές καμπύλες το πλέον κατάλληλο σύστημα για υλοποίηση ασυμμετρικής κρυπτογραφίας σε συσκευές όπου οι βασικοί πόροι όπως υπολογιστική ισχύς, μνήμη και ενέργεια είναι περιορισμένοι και όπου χρησιμοποιούνται πρωτόκολλα ασύρματης επικοινωνίας χαμηλής ισχύος [13].

Κεφάλαιο 3

Κρυπτογραφικές Βιβλιοθήκες

Μέχρι σήμερα έχουν παρουσιαστεί πολλές βιβλιοθήκες λογισμικού ανοιχτού κώδικα που παρέχουν έτοιμους κρυπτογραφικούς αλγόριθμους. Αρκετές από αυτές παρέχουν και αλγόριθμους βασισμένους στις ελλειπτικές καμπύλες. Σαν παράδειγμα, αναφέρουμε τις ECC-LIB [14] και Crypto++ [8]. Ένα κοινό χαρακτηριστικό αυτών των βιβλιοθηκών είναι ότι για την επίτευξη αριθμητικών πράξεων με μεγάλους ακεραίους αριθμούς χρησιμοποιούν υπάρχουσες βιβλιοθήκες κατάλληλες για αριθμητική απεριόριστης ακρίβειας (π.χ. τη GnuMP [12]). Προφανώς αυτού του είδους βιβλιοθήκες δεν έχουν ως στόχο την εφαρμογή τους σε ενσωματωμένα περιβάλλοντα όπου οι διαθέσιμοι πόροι υπολογιστικής ισχύος και μνήμης είναι περιορισμένοι. Αναφέρουμε π.χ. ότι η βιβλιοθήκη GnuMP για την αναπαράσταση μεγάλων ακεραίων αριθμών καθώς και για τις πράξεις μεταξύ τους κάνει συχνή χρήση δυναμικής ανάθεσης μνήμης, μια τεχνική που στα περισσότερα ενσωματωμένα περιβάλλοντα δεν είναι διαθέσιμη. Επιπλέον, για την ταχύτερη εκτέλεση συγκεκριμένων πράξεων χρησιμοποιεί κώδικα γραμμένο σε γλώσσα μηχανής. Η χρήση κώδικα μηχανής ενώ προσφέρει καλύτερη απόδοση περιορίζει την εκτέλεσή του σε συγκεκριμένες πλατφόρμες. Ένας αναγνώστης καταλαβαίνει ότι η μεταφορά τέτοιων βιβλιοθηκών σε ενσωματωμένα περιβάλλοντα είναι πολύ δύσκολη, εάν όχι αδύνατη.

Όπως αναφέραμε και προηγουμένως, για την επίτευξη κρυπτογραφίας δημοσίου κλειδιού σε συσκευές περιορισμένων πόρων όπως αυτές που αποτελούν τα δίκτυα αισθητήρων, προτείνεται η χρήση κρυπτογραφίας με ελλειπτικές καμπύλες [13], [21]. Στη συνέχεια, παρουσιάζουμε τις πιο γνωστές υλοποιήσεις κρυπτογραφίας με βάση τις ελλειπτικές καμπύλες για ενσωματωμένα συστήματα.

3.1 Η Βιβλιοθήκη ECC-LIB

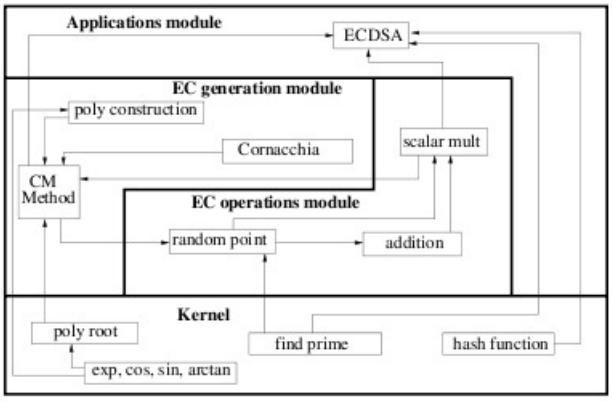
Η ECC-LIB αποτελεί μια πλήρως εξοπλισμένη, μεταφερόμενη βιβλιοθήκη λογισμικού που στοχεύει στην υλοποίηση κρυπτογραφίας ελλειπτικών καμπυλών. Η βιβλιοθήκη αυτή επιτρέπει την εύκολη ανάπτυξη κρυπτογραφικών πρωτοκόλλων με βάση τις ελλειπτικές καμπύλες καθώς παρέχει βασικές αλγεβρικές πράξεις και κατάλληλες μεθόδους για τη γέννηση ασφαλών ελλειπτικών καμπυλών. Η ECC-LIB είναι υλοποιημένη σε ANSI C και βασίζεται στην GMP (GNU Multiple Precission

Library) για αριθμητική αυθαίρετης ακρίβειας πάνω σε ακέραιους και αριθμούς κινητής υποδιαστολής. Ο πηγαίος κώδικας της βιβλιοθήκης παρέχεται και διανέμεται υπό την άδεια λογισμικού GPL (General Public License).

Όπως αναφέρθηκε, βασικός στόχος της βιβλιοθήκης ECC-LIB είναι η μεταφερσιμότητα και η ευκολία χρήσης. Κατά την ανάπτυξή της έπρεπε να παρθούν αποφάσεις σχετικά με το πεδίο των ελλειπτικών καμπυλών που θα χρησιμοποιηθεί, το μέγεθος του, οι μέθοδοι γέννησης ελλειπτικών καμπυλών καθώς και οι βιβλιοθήκες που θα χρησιμοποιηθούν για αριθμητική πάνω σε μεγάλους αριθμούς.

Για την αντιμετώπιση της μεταφερσιμότητας, η βιβλιοθήκη γράφτηκε σε ANSI C χρησιμοποιώντας τη GnuMP [12] για αριθμητική μεγάλης ακρίβειας. Όσον αφορά στην υλοποίηση των ελλειπτικών καμπυλών, επιλέχτηκε το πεδίο πρώτων αριθμών F_p λόγω της απλότητας του στην αναπαράσταση και στις αλγεβρικές πράξεις. Για την αναπαράσταση των αριθμών στο πεδίο πρώτων αριθμών F_p χρησιμοποιήθηκαν οι μεγάλοι αριθμοί που προσφέρει η GnuMP. Η βιβλιοθήκη αυτή αναπαριστά τους ακέραιους και τους αριθμούς κινητής υποδιαστολής χρησιμοποιώντας μια μονάδα που ονομάζεται limb και αποτελείται από 32-bits. Εκτός από τις συναρτήσεις που προσφέρει η GMP υλοποιήθηκαν βασικές αλγεβρικές πράξεις των μιγαδικών αριθμών (πρόσθεση, πολλαπλασιασμός, δύναμη) καθώς και κάποιες συναρτήσεις για τους αριθμούς κινητής υποδιαστολής όπως $\cos(x)$, $\sin(x)$, $\ln(x)$, $\sqrt{x}$ και $\arctan(x)$. Οι συναρτήσεις αυτές είναι απαραίτητες για μεθόδους όπως η Complex Multiplication η οποία γεννά μια καμπύλη κατάλληλης τάξης και υπολογίζει τις μεταβλητές a και b.

Η αρχιτεκτονική της βιβλιοθήκης ECC-LIB αποτελείται από τέσσερα βασικά επίπεδα: τον πυρήνα (kernel), το επίπεδο πράξεων ελλειπτικών καμπυλών (EC operations module), το επίπεδο γέννησης ελλειπτικών καμπυλών (EC generation module) και το επίπεδο εφαρμογών (applications module). Τα βασικά στοιχεία της αρχιτεκτονικής της βιβλιοθήκης φαίνονται στο Σχήμα 3.1.



Σχήμα 3.1: Η Αρχιτεκτονική της Βιβλιοθήκης ECC-LIB.

Ο πυρήνας αποτελείται από διάφορα στοιχεία που υλοποιούν βασικές αλγεβρικές και τριγωνομετρικές πράξεις πάνω σε ακέραιους και αριθμούς κινητής υποδιαστολής. Επιπλέον, παρέχει και κάποιες πιο εξειδικευμένες συναρτήσεις που υλοποιήθηκαν, όπως χειρισμός ακεραίων συντελεστών πολυωνύμων και εύρεση των ριζών ενός πολυωνύμου modulo ενός πρώτου αριθμού. Όλα τα στοιχεία που αποτελούν τον πυρήνα είναι ανεξάρτητα από τα στοιχεία των ανωτέρων επιπέδων που τα χρησιμο-

ποιούνε έτσι ώστε να υπάρχει η δυνατότητα ανεξάρτητης βελτίωσης τους για την καλύτερη απόδοση της βιβλιοθήκης.

Το EC operations module αποτελείται από στοιχεία που υλοποιούν τις βασικές αλγεβρικές πράξεις πάνω σε ελλειπτικές καμπύλες. Ένα στοιχείο ορίζει τον τύπο δεδομένων της ελλειπτικής καμπύλης (ένας πίνακας που αποτελείται απο δύο αριθμούς άπειρης ακρίβειας οι οποίοι αναπαριστούν τις μεταβλητές a και b της καμπύλης) και μία δομή αναπαριστά ένα σημείο πάνω σε μια καμπύλη ως ένα ζευγάρι ακεραίων άπειρης ακρίβειας. Τέλος, υπάρχουν στοιχεία τα οποία δημιουργούν τυχαία σημεία πάνω σε μια καμπύλη, εκτελούν την πρόσθεση δύο σημείων πάνω στην ελλειπτική καμπύλη, τον βαθμωτό πολλαπλασιασμό ενός σημείου με έναν ακέραιο καθώς και τη γέννηση ενός σημείου γεννήτορα πάνω στην καμπύλη.

Το EC generation module είναι το πιο σημαντικό στοιχείο της βιβλιοθήκης ECC-LIB. Αποτελείται από διάφορα στοιχεία που υλοποιούν μεθόδους όπως τνο αλγόριθμο του Cornacchia για την επίλυση διοφαντικών εξισώσεων και την Complex Multiplication για τη γέννηση ασφαλών καμπυλών με χρήση πολυωνύμων Weber και Hilbert.

Τέλος, το applications module περιέχει διάφορα κρυπτογραφικά πρωτόκολλα υψηλού επιπέδου και μεθόδους που βασίζονται στις ελλειπτικές καμπύλες. Μερικά από αυτά είναι το πρωτόκολλο ανταλλαγής κλειδιού Diffie-Hellman, οι μέθοδοι γέννησης ιδιωτικού και δημόσιου κλειδιού, κρυπτογράφηση/αποκρυπτογράφηση δεδομένων και ο αλγόριθμος ψηφιακών υπογραφών ECDSA. Ένας προγραμματιστής έχει τη δυνατότητα δημιουργίας πλουσιότερων κρυπτογραφικών πρωτοκόλλων βασιζόμενος στα στοιχεία που του παρέχει η βιβλιοθήκη ECC-LIB.

3.1.1 Πειραματικά Αποτελέσματα

Για την εξέταση της απόδοσης της βιβλιοθήκης ECC-LIB έγινε πειραματική μελέτη πάνω στα βασικά της στοιχεία. Τα πειράματα εκτελέστηκαν σε υπολογιστή Pentium III (933 MHz) με κύρια μνήμη 256 Mb. Όπως προαναφέρθηκε χρησιμοποιήθηκε η βιβλιοθήκη GnuMP για αριθμητική αυθαίρετης ακρίβειας και ο μεταφραστής της ANSI-C gcc-2.95.2. Στον Πίνακα 3.1 παρουσιάζονται οι επεξεργαστικοί χρόνοι του βαθμωτού πολλαπλασιασμού και των βασικών κρυπτογραφικών πρωτοκόλλων που υλοποιήθηκαν στη βιβλιοθήκη. Η μεταβλητή p αποτελεί έναν πρώτο αριθμό, η $|p|$ αποτελεί το μέγεθος του και η $p_{|p|}$ αναπαριστά έναν πρώτο αριθμό μεγέθους $|p|$. Τα πειράματα έγιναν σε τρία διαφορετικά πεδία με διαφορετικά μεγέθη: $F_{p_{175}}$, $F_{p_{192}}$ και $F_{p_{224}}$.

$\|p\|$	175 bits	192 bits	224 bits
Scalar Multiplication	13.6	15.7	19.5
Key Generation	19.6	23.9	30.8
ECDH protocol	27.2	31.4	39
ECES encryption	28.8	36.5	46
ECES decryption	13.5	16.3	19.1
ECDSA Signature	19.1	22.7	30.6
ECDSA Verify	24.5	28.3	36.8

Πίνακας 3.1: Επεξεργαστικός Χρόνος σε msec των Διαφόρων Στοιχείων της Βιβλιοθήκης ECC-LIB.

3.1.2 Η Βιβλιοθήκη ECC-LIB για Ενσωματωμένα Περιβάλλοντα

Στο σημείο αυτό, αναφέρουμε ότι στα πλαίσια της εργασίας [17] έγινε μεταφορά κομματιών της βιβλιοθήκης ECC-LIB στο ενσωματωμένο περιβάλλον ενός ATMEL AT76C520 802.11 WLAN Access Point. Στόχος της μεταφοράς αυτής ήταν η χρήση κρυπτογραφίας ελλειπτικών καμπυλών σε ενσωματωμένα συστήματα. Βασικό χαρακτηριστικό της μεταφοράς αυτής, είναι ότι παρέχει τις κατάλληλες συναρτήσεις για την υπολογιστικά βαριά διαδικασία γέννησης ελλειπτικών καμπυλών με τη μέθοδο σύνθετου πολλαπλασιασμού (complex multiplication).

Η συσκευή AT76C520, αποτελείται από ένα μικροεπεξεργαστή ARM946 ο οποίος τρέχει στα 100 MHz. Επιπλέον, ένας ARM7TDMI ο οποίος τρέχει στα 80 MHz χρησιμοποιείται για την υλοποίηση των συναρτήσεων επιπέδου MAC του πρωτοκόλλου 802.11 a/b/g. Η συσκευή AT76C520 έχει εσωτερική SRAM 32 KB τα οποία μπορούν να χρησιμοποιηθούν από τον επεξεργαστή για την αποφυγή δοσοληψίας με την εξωτερική μνήμη. Επίσης, υποστηρίζει πρόσβαση σε εξωτερική μνήμη 32-bit SDRAM μεγέθους 256MB, 16 MB εξωτερικής SRAM και 16 MB εξωτερικής Flash. Ο επεξεργαστής ARM7 χρησιμοποιεί SRAM 32 KB για εντολές και δεδομένα. Ο ARM946 χρησιμοποιεί 8 KB κρυφής μνήμης για εντολές (ICache) και 8 KB κρυφής μνήμης για δεδομένα (DCache) ενώ ταυτόχρονα χρησιμοποιεί SDRAM για εντολές και δεδομένα.

Το λειτουργικό σύστημα που τρέχει η συσκευή AT76C520 είναι το μCLinux το οποίο μεταφράζεται ως Microcontroller Linux. Το μCLinux αποτελεί μια έκδοση του πυρήνα Linux χωρίς διαχειριστή μνήμης (Memory Management Unit), κατάλληλη για μικροελεγκτές. Εκτός από τον πυρήνα, το μCLinux περιλαμβάνει μια πρότυπη βιβλιοθήκη C (uClibc) και επιπλέον περιέχει εφαρμογές, βιβλιοθήκες, εργαλεία καθώς και ένα σύστημα αρχείων.

Καθώς η βιβλιοθήκη ECC-LIB χρησιμοποιεί τη GnuMP για την εκτέλεση πράξεων με αριθμούς μεγάλης ακρίβειας, κατά τη μεταφορά της βιβλιοθήκης στο ενσωματωμένο περιβάλλον του ATMEL AT76C520 802.11 WLAN Access Point χρησιμοποιηθήκαν μόνο εκείνα τα κομμάτια των δύο βιβλιοθηκών που είναι απαραίτητα για τη γέννηση ελλειπτικών καμπυλών με τη μέθοδο του σύνθετου πολλαπλασιασμού. Το μέγεθος του τελικού κώδικα μετρήθηκε στα 250 KB, το οποίο είναι αποδεκτό για τη συσκευή AT76C520.

Τέλος, έγιναν πειραματικές μετρήσεις για την αξιολόγηση της απόδοσης της υλοποίησης αυτής. Ο Πίνακας 3.2 δείχνει τους χρόνους που απαιτούνται για την κατασκευή μιας ελλειπτικής καμπύλης με τη μέθοδο σύνθετου πολλαπλασιασμού χρησιμοποιώντας πολυώνυμα Hilbert και Weber με διακρίνουσα $D = 39$ και βαθμό πολυωνύμου $h = 4$. Επιπλέον, ο Πίνακας 3.3 παρουσιάζει τους χρόνους που απαιτούνται για τη γέννηση ενός σημείου βάσης T_b και ενός ζευγαριού κλειδιών (ιδιωτικό, δημόσιο) T_{pp} ελλεπτικής καμπύλης με βάση το μέγεθος του πεδίου που χρησιμοποιείται.

3.2 Η Υλοποίηση EccM-2.0

Το 2004 οι David Malan, Matt Welsh και Michael Smith στην εργασία [9] παρουσίασαν την πρώτη υλοποίηση κρυπτογραφίας με ελλειπτικές καμπύλες για ασύρματα δίκτυα αισθητήρων. Συγκεκριμένα, η υλοποίηση αυτή έγινε για τις συσκευές MICA 2 οι οποίες περιέχουν έναν 8-bit επεξεργαστή που τρέχει στα 7.3828 MHz.

	Hilbert	Weber
Size of p	T_{EC}	T_{EC}
175 bits	122.89 sec	111.71 sec
192 bits	129.98 sec	128.13 sec

Πίνακας 3.2: Επεξεργαστικός Χρόνος για την Κατασκευή Ελλειπτικής Καμπύλης (T_{EC}) με Διακρίνουσα $D = 39$ και Πολυώνυμα Hilbert και Weber βαθμού $h = 4$ στη Συσκευή AT76C520.

Size of p	T_b	T_{pp}
161 bits	10.41 sec	0.77 sec
175 bits	11.43 sec	0.91 sec
192 bits	13.67 sec	1.03 sec
224 bits	15.59 sec	1.37 sec

Πίνακας 3.3: Επεξεργαστικός Χρόνος για τη Γέννηση Σημείου Βάσης T_b και Ζευγαριού Κλειδιών T_{pp} Ελλειπτικών Καμπυλών στη Συσκευή AT76C520.

Ο βασικός στόχος της εργασίας αυτής ήταν να καλύψει την ανάγκη ύπαρξης ενός ασφαλούς μηχανισμού για τη διανομή κλειδιών ανάμεσα στους κόμβους, παρά το γεγονός ότι η υποδομή δημόσιου κλειδιού δε θεωρούνταν πρακτική. Αποδείξανε ότι η ασυμμετρική κρυπτογραφία είναι βιώσιμη στις συσκευές MICA 2 και υποστηρίξανε με την υλοποίηση πολλαπλασιασμού σημείων πάνω σε μια ελλειπτική καμπύλη ότι η υποδομή δημοσίου κλειδιού είναι ικανοποιητική για τη διανομή των μυστικών κλειδιών στους κόμβους.

Το TinyOS προσφέρει στις συσκευές MICA 2 έλεγχο πρόσβασης, πιστοποίηση, ακεραιότητα και εμπιστευτικότητα μέσω του TinySec. Το TinySec αποτελεί ένα μηχανισμό ασφάλειας στο επίπεδο συνδέσμου ο οποίος βασίζεται στο Skipjack. Ο Skipjack είναι ένας αλγόριθμος κρυπτογράφησης που αναπτύχθηκε από το NIST : National Institute for Standards and Technology και χρησιμοποιεί ένα κλειδί μήκους 80-bit για την κρυπτογράφηση 64-bit μπλοκ δεδομένων. Για τον κατάλληλο διαμοιρασμό των 80-bit κλειδιών του TinySec χρειάζεται ένας μηχανισμός αντίστοιχης ασφάλειας. Έχουμε αναφέρει και σε προηγούμενα κεφάλαια ότι με το πρωτόκολλο Diffie-Hellman δύο κόμβοι μπορούν να συμφωνήσουν σε ένα κοινό μυστικό. Σύμφωνα με το NIST για τον ασφαλή διαμοιρασμό κλειδιών 80-bit μέσω του πρωτοκόλλου Diffie-Hellman απαιτείται ένας πρώτος αριθμός τουλάχιστον 1024-bits και ένας εκθέτης μεγέθους 160-bits. Όπως καταλαβαίνουμε σε μια αρχιτεκτονική 8-bit όπως το MICA 2 οι υπολογισμοί αριθμών μήκους 160 και 1024 bit είναι αρκετά επίπονοι.

Οι συγγραφείς κατάφεραν να λύσουν το παραπάνω πρόβλημα με χρήση των ελλειπτικών καμπυλών. Η ασφαλής διανομή 80-bit κλειδιών μπορεί να γίνει με κλειδιά που βασίζονται στις ελλειπτικές καμπύλες μήκους μόλις 163-bit. Οι ελλειπτικές καμπύλες προσφέρουν μια διαφορετική βάση για την ανταλλαγή κοινών μυστικών ανάμεσα σε επιτιθέμενους με απόλυτη προς τα εμπρός ασφάλεια. Το πρόβλημα ECDLP: Elliptic Curve Discrete Logarithm Problem στο οποίο βασίζεται η κρυπτογραφία με ελλειπτικές καμπύλες ECC, περιλαμβάνει την επαναφορά ενός αριθμού k πάνω από ένα πεπερασμένο πεδίο Galois F, δεδομένων του γινομένου $k \cdot G$, του σημείου G και της εξίσωσης E της ελλειπτικής καμπύλης σε μορφή Weierstrass η οποία φαίνεται παρακάτω:

$$y^2 + a_1 \cdot x \cdot y + a_3 \cdot y = x^3 + a_2 \cdot x^2 + a_4 \cdot x + a_6$$

όπου $a_i \in F$.

Οι κρυπτογράφοι δείχνουν ενδιαφέρον στις ελλειπτικές καμπύλες που είναι ορισμένες πάνω από τα πεδία F_p και F_{2^p}, όπου το p είναι ένας πρώτος αριθμός. Ειδικότερα, οι ελλειπτικές καμπύλες ορισμένες πάνω από τα δυαδικά πεδία είναι ιδιαίτερα δημοφιλείς καθώς προσφέρουν αποδοτικούς αλγόριθμους.

3.2.1 Κρυπτογραφία Ελλειπτικών Καμπυλών στο F_{2^p}

Η κρυπτογραφία ελλειπτικών καμπυλών πάνω από το πεδίο F_{2^p} αρχικά απαιτεί την επιλογή μιας βάσης για την αναπαράσταση των σημείων έτσι ώστε κάθε $a \in F_{2^p}$ να μπορεί να γραφτεί στη μορφή

$$a = \sum_{i=0}^{m-1} a_i \cdot a_i$$

όπου $a_i \in \{0, 1\}$. Ορισμένο με τον τρόπο αυτό, το a μπορεί να αναπαραστεί ως ένα δυαδικό διάνυσμα, $\{a_0, a_1, ..., a_{p-1}\}$, όπου το $\{a_0, a_1, ..., a_{p-1}\}$ αποτελεί τη βάση πάνω από το πεδίο F_2. Οι πιο κοινές βάσεις είναι οι πολυωνυμικές και οι κανονικές. Όταν κάθε $a_i \in F_{2^p}$ αναπαρίσταται με πολυωνυμική βάση, αντιστοιχεί σε ένα δυαδικό πολυώνυμο βαθμού μικρότερου του p, όπως:

$$a = a_{p-1} \cdot x^{p-1} + a_{p-2} \cdot x^{p-2} + ... + a_0 \cdot x^0$$

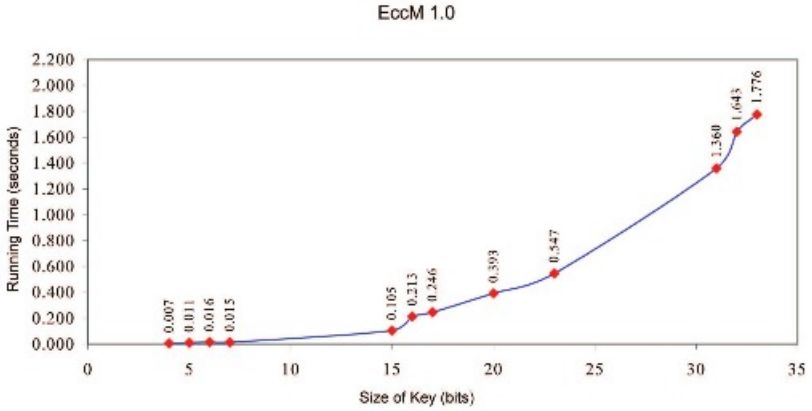
όπου και πάλι $a_i \in \{0, 1\}$. Με τον τρόπο αυτό κάθε $a \in F_{2^p}$ μπορεί να απεικονιστεί στη μνήμη της συσκευής MICA 2 σαν μια ακολουθία από bit, $a_{p-1}a_{p-2}...a_0$. Όλες οι πράξεις των στοιχείων αυτών γίνονται modulo ενός απλοποιημένου πολυωνύμου, f , βαθμού p πάνω από το πεδίο F_2, έτσι ώστε $f(x) = x^p + \sum_{i=0}^{p-1} f_i \cdot x_i$, με $f_i \in \{0, 1\}$ και $i \in \{0, 1, ..., p-1\}$. Τυπικά, εαν υπάρχει στο F_2 ένα μη μειούμενο τριώνυμο $x^p + x^k + 1$, το $f(x)$ επιλέγεται έτσι ώστε να είναι αυτό με το μικρότερο k. Στην περίπτωση που τέτοιο πολυώνυμο δεν υπάρχει, η $f(x)$ επιλέγεται έτσι ώστε να είναι πενταώνυμο, $x^p + x^{k3} + x^{k2} + x^{k1} + 1$, έτσι ώστε το $k1$ να είναι ελάχιστο, το $k2$ να είναι ελάχιστο ως προς το $k1$ και το $k3$ να είναι ελάχιστο ως προς τα $k1$ και $k2$.

Σε μια πολυωνυμική βάση, η πρόσθεση δύο στοιχείων a, b ορίζεται ως $a+b = c$, όπου $c_i = a_i + b_i (mod 2)$. Ο πολλαπλασιασμός των a, b ορίζεται ως $a * b = c$, όπου $c(x) = (\sum_{i=0}^{p-1} a_i x^i) \cdot (\sum_{i=0}^{p-1} b_i x^i) (mod f(x))$. Οι συγγραφείς επέλεξαν μια πολυωνυμική βάση για την υλοποίηση του βαθμωτού πολλαπλασιασμού καθώς προσφέρει πιο αποδοτικές υλοποιήσεις.

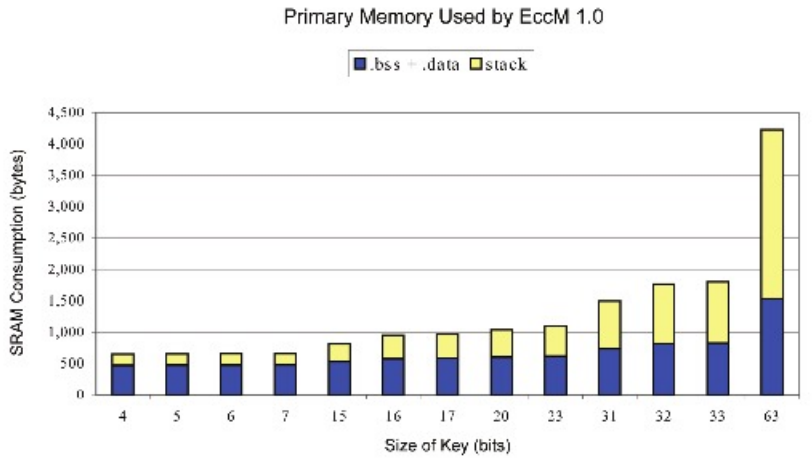
3.2.2 Πρώτη Υλοποίηση EccM-1.0

Η πρώτη υλοποίηση που επιχείρησαν οι συγγραφείς για τις συσκευές MICA 2 ονομαζόταν EccM-1.0. Το EccM-1.0 αποτελεί ένα module για το λειτουργικό σύστημα TinyOS και αρχικά επιλέγει μια τυχαία καμπύλη στη μορφή της εξίσωσης που αναφέραμε παραπάνω έτσι ώστε $a = 0$ και $b \in F_{2^p}$. Εν συνεχεία, υπολογίζει ένα τυχαίο σημείο $G \in F_{2^p} \times F_{2^p}$ πάνω στην καμπύλη καθώς και ένα τυχαίο αριθμό $k \in F_{2^p}$ που αποτελεί το ιδιωτικό κλειδί του κόμβου. Τέλος, υπολογίζει το γινόμενο $k \cdot G$ που είναι το δημόσιο κλειδί του κόμβου. Τα πρώτα αποτελέσματα της υλοποίησης αυτής ήταν ενθαρυντικά καθώς η δημιουργία ενός κλειδιού μήκους 33-bit απαιτούσε μόλις 1.776 δευτερόλεπτα. Παρά το γεγονός αυτό, το πρωτόκολλο

απέτυχε να δημιουργήσει κλειδιά μεγαλύτερου μήκους (π.χ. 63 bit) αναγκάζοντας τον κόμβο να επανεκκινήσει λόγω υπερχείλισης της στοίβας. Τα αποτελέσματα της υλοποίησης αυτής φαίνονται στις παρακάτω Σχήματα (Σχήμα 3.2, Σχήμα 3.5).



Σχήμα 3.2: Χρόνοι Δημιουργίας Κλειδιού στο EccM-1.0. Για Κλειδιά Μήκους 63-bit το Πρωτόκολλο Δεν Απέφερε Αποτελέσματα.



Σχήμα 3.3: Κατανάλωση Μνήμης στο EccM-1.0. Κλειδιά Μήκους 63-bit Εξαντλούν τη RAM των Συσκευών MICA 2.

3.2.3 Δεύτερη Υλοποίηση EccM-2.0

Λόγω της αποτυχίας της πρώτης υλοποίησης στη δημιουργία κλειδιών έστω και 63-bit, οι συγγραφείς προχώρησαν σε μια δεύτερη που ονομάζεται EccM-2.0. Η EccM-2.0 διαλέγει έναν κόμβο A και δημιουργεί το ιδιωτικό του κλειδί k_A χρησιμοποιώντας μια πολυωνυμική βάση στο F_{2^p}. Στη συνέχεια, αφού βρει ένα σημείο βάσης G πάνω σε μια καμπύλη Koblitz, υπολογίζει το δημόσιο κλειδί του κόμβου

$T_A = k_A \cdot G$. Το δημόσιο κλειδί του Α γίνεται γνωστό σε κάθε κόμβο Β με τον οποίο επιθυμείται ασφαλής επικοινωνία. Ο Α λαμβάνει με αντίστοιχο τρόπο το δημόσιο κλειδί του Β και έτσι ο καθένας τους μπορεί να υπολογίσει το κοινό μυστικό $k_A \cdot k_B G$, όπου k_B είναι το ιδιωτικό κλειδί του Β.

Η υλοποίηση EccM-2.0 δεν κατάφερε μόνο τη δημιουργία δημοσίων κλειδιών 163-bit αλλά έδωσε λύσεις και σε ένα άλλο σημαντικό πρόβλημα. Στην EccM-1.0, οι ελλειπτικές καμπύλες επιλέγονταν τυχαία με κίνδυνο να είναι ευάλωτες σε επιθέσεις MOV. Αντίθετα, στην EccM-2.0 επιλέγεται μια καμπύλη που ικανοποιεί τις απαιτήσεις του NIST για ECC πάνω από το πεδίο F_{2^p}. Η εξίσωση της καμπύλης αυτής είναι η:

$$y^2 + x \cdot y = x^3 + x^2 + 1$$

, σαν απλοποιημένο πολυώνυμο χρησιμοποιείται το:

$$f(x) = x^{163} + x^7 + x^6 + x^3 + 1$$

, η τάξη της καμπύλης (ο αριθμός των σημείων πάνω σε αυτή) είναι:

$$0x4000000000000000000020108a2e0cc0d99f8a5ef$$

και το σημείο βάσης είναι το $G = (G_x, G_y)$ με :

$$G_x = 0x2fe13c0537bbc11acaa07d793de4e6d5e5c94eee8$$

και

$$G_y = 0x289070fb05d38ff58321f2e800536d538ccdaa3d9$$

Αυτό που κατάφεραν οι συγγραφείς με την υλοποίηση EccM-2.0 ήταν η δημιουργία μεγαλύτερων και πιο ασφαλών κλειδιών σε σχέση με αυτά της EccM-1.0 καθώς και η πολύ μικρότερη κατανάλωση μνήμης. Ο βαθμωτός πολλαπλασιασμός σημείου της ελλειπτικής καμπύλης χρονομετρήθηκε στα 34 δευτερόλεπτα με απόκλιση 0.9 δευτερόλεπτα. Στον Πίνακα 3.4 φαίνεται η σύγκριση της κατανάλωσης μνήμης ανάμεσα στις δύο υλοποιήσεις και στον Πίνακα 3.5 παρουσιάζονται τα αποτελέσματα των πειραμάτων μέτρησης της απόδοσης του EccM-2.0 στις συσκευές MICA 2.

	EccM-1.0 (32-bit key)	EccM-2.0 (163-bit key)
.bss	826B	1,055B
.data	6B	4B
.text	17,544B	34,342B
stack	976B	81B

Πίνακας 3.4: Σύγκριση της Κατανάλωσης Μνήμης Ανάμεσα στις Υλοποιήσεις EccM-1.0 και EccM-2.0.

	Private Key Generation	Public Key Generation
Total Time	0.229 sec	34.161 sec
CPU Utilization	1.690×10^6 cycles	2.512×10^8 cycles
Total Energy	0.00549 Joules	0.816 Joules

Πίνακας 3.5: Απόδοση της Υλοποίησης EccM-2.0.

3.3 Η Βιβλιοθήκη TinyECC

Το 2007, οι An Liu και Peng Ning εξέδωσαν την πρώτη έκδοση της εργασίας τους με όνομα TinyECC. Το TinyECC είναι ένα πακέτο λογισμικού το οποίο παρέχει λειτουργίες κρυπτογράφησης με ελλειπτικές καμπύλες οι οποίες μπορούν εύκολα να ενσωματωθούν σε εφαρμογές ασυρμάτων δικτύων αισθητήρων. Οι λειτουργίες που παρέχει είναι οι εξής: ένα αλγόριθμο ψηφιακής υπογραφής (ECDSA), ένα σχήμα ανταλλαγής κλειδιών (ECDH) και ένα αλγόριθμο κρυπτογράφησης δημοσίου κλειδιού (ECIES). Επιπλέον, παρέχει τη δυνατότητα ενεργοποίησης διαφόρων τεχνικών βελτιστοποίησης οι οποίες μπορούν να χρησιμοποιηθούν ανάλογα με τις ανάγκες της εκάστοτε εφαρμογής. Το TinyECC προορίζεται για πλατφόρμες αισθητήρων που τρέχουν το λειτουργικό σύστημα TinyOS. Είναι υλοποιημένο στη γλώσσα nesC και κομμάτια του κώδικα είναι γραμμένα σε γλώσσα μηχανής για την παροχή βελτιστοποιήσεων ανάλογα με την πλατφόρμα για την οποία μεταφράζεται. Η αξιολόγηση της απόδοσης της TinyECC έγινε στις συσκευές MICAz, TelosB, Tmote Sky και Imote2 οι οποίες τρέχουν το λειτουργικό σύστημα TinyOS.

3.3.1 Αρχές Σχεδίασης της Βιβλιοθήκης TinyECC

Όπως προαναφέραμε, βασικός στόχος της βιβλιοθήκης TinyECC είναι να παρέχει ένα **έτοιμο για χρήση** πακέτο λογισμικού το οποίο υλοποιεί λειτουργίες κρυπτογραφίας δημοσίου κλειδιού με ελλειπτικές καμπύλες. Οι βασικές αρχές σχεδίασης της TinyECC φαίνονται παρακάτω.

- **Ασφάλεια :** Η TinyECC παρέχει σχήματα κρυπτογράφησης με ελλειπτικές καμπύλες τα οποία έχουν αποδειχθεί ασφαλή. Αυτά είναι: ο αλγόριθμος ψηφιακής υπογραφής (ECDSA), το σχήμα συμφωνίας κλειδιών (ECDH) και ο αλγόριθμος κρυπτογράφησης δημοσίου κλειδιού (ECIES). Επιπλέον, οι παράμετροι των ελλειπτικών που χρησιμοποιεί, προτείνονται από το SECG: Standars for Efficient Cryptography Group.

- **Μεταφερσιμότητα :** Για τη δυνατότητα εκτέλεσης της TinyECC σε διάφορες πλατφόρμες, υλοποιήθηκε στη γλώσσα nesC του λειτουργικού συστήματος TinyOS. Κάποια κομμάτια του κώδικα γράφτηκαν σε γλώσσα μηχανής για βελτιστοποιήσεις ανάλογα με την πλατφόρμα εκτέλεσης. Η λειτουργικότητα της TinyECC έχει εξεταστεί με επιτυχία στις συσκευές MICAz, TelosB, Tmote Sky και Imote2.

- **Αποτελεσματικότητα :** Οι δημιουργοί της TinyECC πήραν κάποιες σχεδιαστικές αποφάσεις έτσι ώστε οι κρυπτογραφικές του λειτουργίες να εκτελούνται σωστά, γρήγορα και να καταναλώνουν λιγότερη ενέργεια. Για το στόχο αυτό υλοποίησαν την κρυπτογραφία με ελλειπτικές καμπύλες πάνω από τα πρώτα πεδία F_p, θεωρώντας ότι οι αριθμητικές πράξεις πάνω από τα δυαδικά πεδία F_{2^p} δεν εκτελούνται αποδοτικά στους μικροελεγκτές. Επίσης, όπως αναφέραμε και προηγουμένως κάποια κρίσιμα κομμάτια του κώδικα υλοποιήθηκαν σε γλώσσα μηχανής για τις διάφορες πλατφόρμες. Με τον τρόπο αυτό κάποιες υπολογιστικά απαιτητικές αριθμητικές πράξεις εκτελούνται πολύ πιο γρήγορα.

- **Λειτουργικότητα :** Με τα κρυπτογραφικά σχήματα που παρέχει το TinyECC, δηλαδή τα ECSDA, ECIES και ECDH, καλύπτει τις βασικές ανάγκες για κρυπτογράφηση με ελλειπτικές καμπύλες.

3.3.2 Τεχνικές Βελτιστοποίησης της Βιβλιοθήκης TinyECC

Για την επίτευξη καλύτερης απόδοσης, οι δημιουργοί της TinyECC χρησιμοποίησαν κάποιες τεχνικές βελτιστοποίησης οι οποίες παρουσιάζονται στη συνέχεια.

Τεχνικές Βελτιστοποίησης για Πράξεις με Μεγάλους Ακεραίους Αριθμούς

- Μείωση Κατά Barrett (Barrett Reduction) : Ένας απλός τρόπος για την εκτέλεση modulo αναγωγής μεγάλων ακεραίων αριθμών είναι η χρήση της διαίρεσης. Η τεχνική Barrett Reduction είναι ένας εναλλακτικός τρόπος για εκτέλεση modulo αναγωγής. Η τεχνική αυτή μετατρέπει την αναγωγή ενός τυχαίου ακεραίου σε δύο πολλαπλασιασμούς και μερικές αναγωγές modulo ακεραίων της μορφής 2^n. Όταν χρησιμοποιείται για την αναγωγή μεγάλων ακεραίων modulo του ίδιου αριθμού πολλές φορές είναι αρκετά πιο γρήγορη από τις αναγωγές με χρήση διαίρεσης.

- Υβριδικός Πολλαπλασιασμός (Hybrid Multiplication) : Οι πράξεις πολλαπλασιασμού μεγάλων ακεραίων αποθηκεύουν τους τελεστές και τα αποτελέσματα σε πίνακες. Όταν τέτοιες πράξεις υλοποιούνται σε γλώσσες υψηλού επιπέδου, ο μεταγλωττιστής δε χρησιμοποιεί αποδοτικά τους καταχωρητές του μικρο-ελεγκτή. Για το λόγο αυτό χρησιμοποιήθηκε η τεχνική Hybrid Multiplication η οποία υλοποιήθηκε σε γλώσσα μηχανής. Η τεχνική αυτή μεγιστοποιεί τη χρήση καταχωρητών και μειώνει τον αριθμό των λειτουργιών της μνήμης.

Τεχνικές Βελτιστοποίησης για Πράξεις με Ελλειπτικές Καμπύλες

- Προβολικό Σύστημα Συντεταγμένων (Projective Coordinate System) : Ο μικρο-ελεγκτής ATmega128 δεν έχει εντολή για διαίρεση με αποτέλεσμα η διαδικασία αντιστροφής ενός ακεραίου στο πεδίο της ελλειπτικής καμπύλης να είναι πιο *ακριβή* από τον πολλαπλασιασμό. Για το λόγο αυτό, η εκτέλεση των πράξεων των ελλειπτικών καμπυλών σε συντεταγμένες προβολής αντί για τις σχετικές συντεταγμένες είναι πιο αποδοτική. Οι δημιουργοί της TinyECC χρησιμοποίησαν την Jacobian αναπαράσταση για την επιτάχυνση της πρόσθεσης σημείου, του διπλασιασμού σημείου και του βαθμωτού πολλαπλασιασμού σημείου ελλειπτικής καμπύλης.

- Μέθοδος Ολισθαίνοντος Παραθύρου (Sliding Window Method) : Η μέθοδος αυτή υλοποιήθηκε για την επιτάχυνση του βαθμωτού πολλαπλασιασμού σημείου ελλειπτικής καμπύλης. Η κλασική μέθοδος υπολογισμού του βαθμωτού πολλαπλασιασμού είναι η δυαδική. Η μέθοδος αυτή σαρώνει τα bits του αριθμού n από αριστερά προς τα δεξιά, ένα bit κάθε φορά. Σε κάθε βήμα εκτελείται ένας διπλασιασμός σημείου και αναλόγως του εκάστοτε bit εκτελείται και μια πρόσθεση σημείου. Η μέθοδος Sliding Window σαρώνει k bits τη φορά. Ο διπλασιασμός σημείου γίνεται k φορές σε κάθε βήμα και αναλόγως τα k bits εκτελείται και μια πρόσθεση σημείου.Η μέθοδος αυτή επιτυγχάνει την επιτάχυνση του βαθμωτού πολλαπλασιασμού μειώνοντας τον αριθμό των συνολικών προσθέσεων σημείου. Βέβαια, έχει μεγαλύτερες απαιτήσεις σε μνήμη.

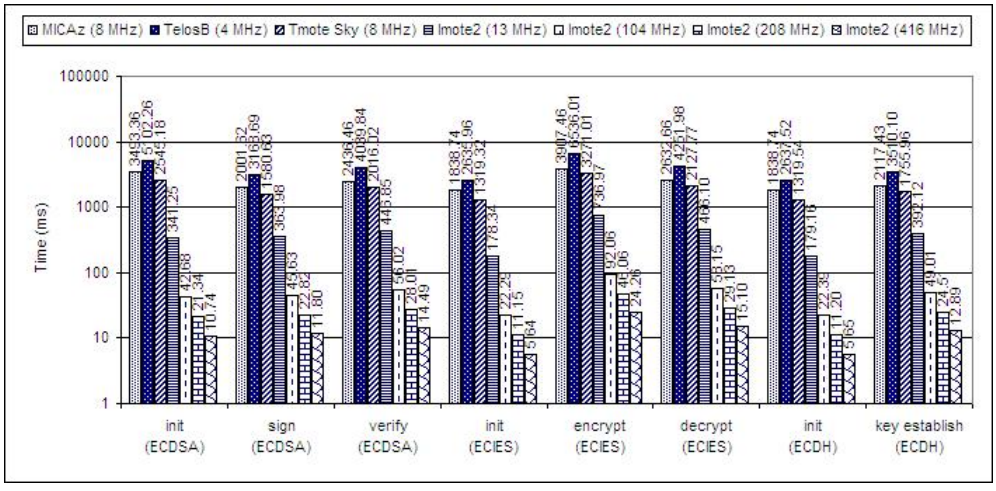
- Το Τέχνασμα του Shamir : Η μέθοδος αυτή χρησιμοποιείται για τη βελτιστοποίηση της ταχύτητας του σχήματος ψηφιακής υπογραφής ECDSA.
- Βελτιστοποιήσεις Συγκεκριμένων Καμπυλών : Οι περισσότερες ελλειπτικές καμπύλες που προτείνονται από το NIST και το SECG χρησιμοποιούν τους ψευδο-Mersenne πρώτους αριθμούς. Ένας ψευδο-Mersenne πρώτος αριθμός είναι της μορφής $p = 2^n - c$, όπου $c \ll 2^n$. Η πράξη της αναγωγής modulo ενός ψευδο-Mersenne πρώτου αριθμού πραγματοποιείται με λίγους πολλαπλασιασμούς και προσθέσεις, χωρίς να γίνει χρήση διαίρεσης. Με τον τρόπο αυτό η αναγωγή γίνεται πιο γρήγορα και η χρήση τέτοιων ελλειπτικών καμπυλών βελτιώνει τη γενικότερη απόδοση του συστήματος.

3.3.3 Απόδοση της TinyECC

Η απόδοση της υλοποίησης TinyECC εξετάστηκε στις συσκευές MICAz (8 MHz), TelosB (4 MHz), Tmote Sky (8 MHz) και Imote2 (13, 104, 208, 416 Mhz) οι οποίες εκτελούν το λειτουργικό σύστημα TinyOS. Χρησιμοποιείται το πρότυπο secp160r1 το οποίο προτείνεται από το SECG και ορίζει παραμέτρους για κρυπτογράφηση με ελλειπτικές καμπύλες πάνω από τα πρώτα πεδία F_p. Παρακάτω φαίνονται τα αποτελέσματα της απόδοσης της βιβλιοθήκης TinyECC σε δύο περιπτώσεις: όταν όλες οι τεχνικές βελτιστοποίησης είναι ενεργοποιημένες και όταν είναι απενεργοποιημένες.

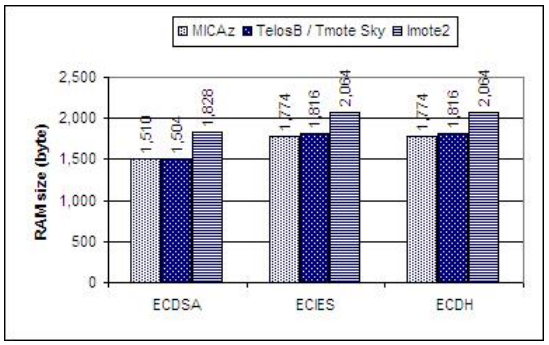
Ενεργοποιημένες Τεχνικές Βελτιστοποίησης

Στο Σχήμα 3.4 φαίνεται η απόδοση της TinyECC όταν όλες οι τεχνικές βελτιστοποίησης είναι ενεργοποιημένες. Όπως φαίνεται σε αυτό το σχήμα οι συσκευές TelosB (4 MHz) είναι οι πιο αργές και οι συσκευές Imote2 (416 Mhz) είναι οι πιο γρήγορες.

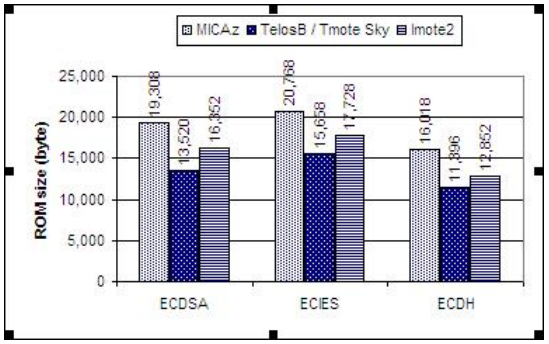


Σχήμα 3.4: Χρόνοι Εκτέλεσης των Λειτουργιών της Βιβλιοθήκης TinyECC στις Διάφορες Πλατφόρμες Όταν Όλες οι Τεχνικές Βελτιστοποίησης Είναι Ενεργοποιημένες.

Στο Σχήματα 3.5 και 3.6 φαίνεται η κατανάλωση μνήμης του κωδικα της TinyECC. Η συσκευή Imote2 έχει τις μεγαλύτερες απαιτήσεις σε RAM λόγω των 32-bit words που χρησιμοποιεί. Η συσκευή MICAz έχει τις μεγαλύτερες απαιτήσεις σε ROM λόγω του κώδικα σε γλώσσα μηχανής που χρησιμοποιείται για βελτιστοποίηση των λειτουργιών με ελλειπτικές καμπύλες.

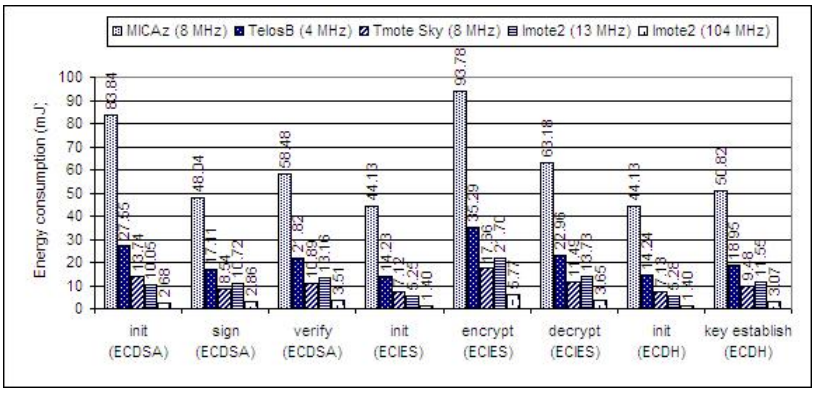


Σχήμα 3.5: Κατανάλωση Μνήμης RAM της Βιβλιοθήκης TinyECC Όταν Όλες οι Τεχνικές Βελτιστοποίησης Είναι Ενεργοποιημένες.



Σχήμα 3.6: Κατανάλωση Μνήμης ROM της Βιβλιοθήκης TinyECC Όταν Όλες οι Τεχνικές Βελτιστοποίησης Είναι Ενεργοποιημένες.

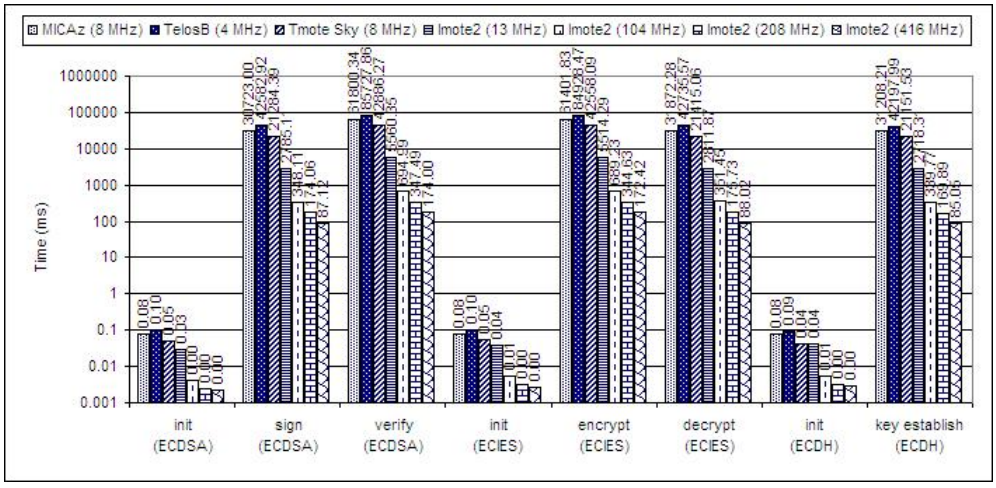
Τέλος, στο Σχήμα 3.7 φαίνεται η κατανάλωση ενέργειας των βασικών αλγορίθμων της TinyECC. Από αυτό παρατηρούμε ότι η συσκευή MicaZ απαιτεί την περισσότερη ενέργεια για την εκτέλεση των αλγορίθμων και ότι η συσκευή Imote2 (104 MHz) είναι η πιο αποτελεσματική.



Σχήμα 3.7: Κατανάλωση Ενέργειας (mJ) της Βιβλιοθήκης TinyECC Όταν Όλες οι Τεχνικές Βελτιστοποίησης Είναι Ενεργοποιημένες.

Απενεργοποιημένες Τεχνικές Βελτιστοποίησης

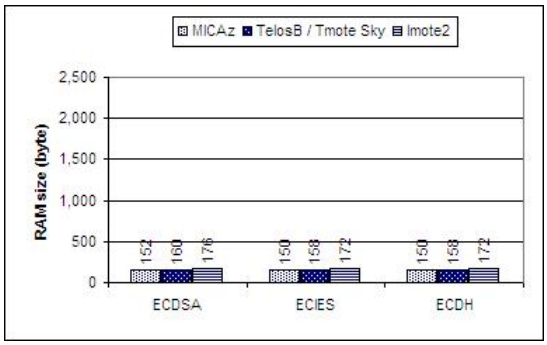
Για την ενσωμάτωση της TinyECC σε άλλες εφαρμογές του TinyOS, χρειάζεται μείωση του μεγέθους του κώδικα του. Αυτό ισχύει ειδικά στις συσκευές MICAz και TelosB που είναι χαμηλότερων πόρων σε σχέση με τις άλλες. Για το λόγο αυτό μετρήθηκε η απόδοση της TinyECC όταν όλες οι τεχνικές βελτιστοποίησης είναι απενεργοποιημένες. Τα αποτελέσματα φαίνονται στα επόμενα Σχήματα (3.8, 3.9 και 3.10).



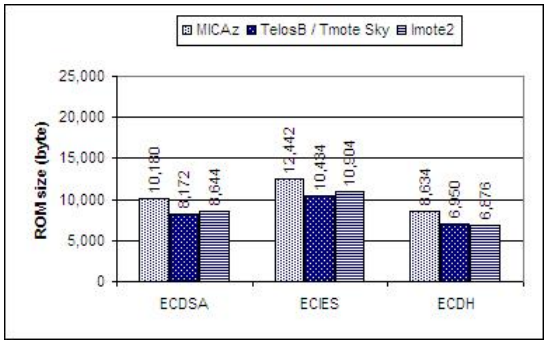
Σχήμα 3.8: Χρόνοι Εκτέλεσης των Λειτουργιών της Βιβλιοθήκης TinyECC στις Διάφορες Πλατφόρμες Όταν Όλες οι Τεχνικές Βελτιστοποίησης Είναι Απενεργοποιημένες.

Όπως παρατηρούμε από την παραπάνω γραφική παράσταση οι χρόνοι εκτέλεσης των κρυπτογραφικών λειτουργιών της TinyECC αυξάνονται αρκετά όταν οι τεχνικές βελτιστοποίησης είναι απενεργοποιημένες. Το μέγεθος του κώδικα, σε αντίθε-

ση, μειώνεται δραματικά όπως φαίνεται στις παρακάτω γραφικές παραστάσεις. Για παράδειγμα, οι απαιτήσεις σε RAM για τις συσκευές MICAz είναι περίπου μόνο 150 bytes και το μέγεθος ROM του ECIES μειώνεται από τα 20,768 bytes στα 12,442 bytes.

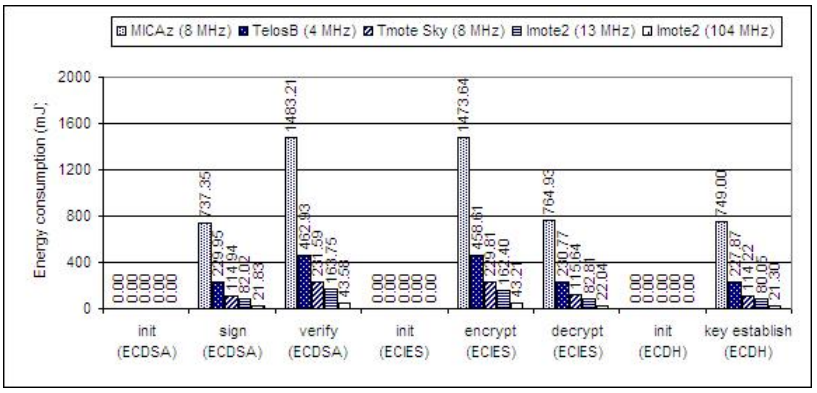


Σχήμα 3.9: Κατανάλωση Μνήμης RAM της Βιβλιοθήκης TinyECC Όταν Όλες οι Τεχνικές Βελτιστοποίησης Είναι Απενεργοποιημένες.



Σχήμα 3.10: Κατανάλωση Μνήμης ROM της Βιβλιοθήκης TinyECC Όταν Όλες οι Τεχνικές Βελτιστοποίησης Είναι Απενεργοποιημένες.

Τέλος, στο Σχήμα 3.11 φαίνεται η κατανάλωση ενέργειας των βασικών αλγορίθμων της βιβλιοθήκης TinyECC όταν όλες οι τεχνικές βελτιστοποίσης είναι απενεργοποιημένες. Από αυτό παρατηρούμε ότι καθώς όλοι οι χρόνοι εκτέλεσης έχουν αυξηθεί, έχει αυξηθεί και η απαιτούμενη ενέργεια. Για παράδειγμα, η συσκευή MicaZ απαιτεί 15 και 25 φορές περισσότερη ενέργεια για τη δημιουργία και επαλήθευση υπογραφής ECDSA σε σχέση με την περίπτωση που οι βελτιστοποιήσεις είναι ενεργοποιημένες.



Σχήμα 3.11: Κατανάλωση Ενέργειας (mJ) της Βιβλιοθήκης TinyECC Όταν Όλες οι Τεχνικές Βελτιστοποίησης Είναι Απενεργοποιημένες.

3.4 Το Πρόβλημα με τις Υπάρχουσες Κρυπτογραφικές Βιβλιοθήκες

Οι εργασίες που παρουσιάστηκαν παραπάνω αποτελούν τις πιο δημοφιλείς υλοποιήσεις κρυπτογραφίας με ελλειπτικές καμπύλες σε ενσωματωμένα περιβάλλοντα.

Όπως αναφέρθηκε, η ECC-LIB αποτελεί μια πλήρη βιβλιοθήκη για κρυπτογραφία με ελλειπτικές καμπύλες καθώς παρέχει υλοποιήσεις συναρτήσεων για γέννηση ελλειπτικών καμπυλών αλλά και βασικά πρωτόκολλα με ελλειπτικές καμπύλες. Το βασικό πρόβλημα της βιβλιοθήκης ECC-LIB είναι ότι για τις κρυπτογραφικές πράξεις με μεγάλους αρίθμους βασίζεται στη βιβλιοθήκη GnuMP. Η GnuMP κάνει χρήση δυναμικής ανάθεσης μνήμης μια τεχνική που στα περισσότερα συστήματα περιορισμένων πόρων δεν είναι διαθέσιμη. Επιπλέον, για την ταχύτερη εκτέλεση συγκεκριμένων πράξεων χρησιμοποιεί κομμάτια κώδικα γραμμένα σε γλώσσα μηχανής. Η υλοποίηση σε κώδικα μηχανής ενώ επιτυγχάνει καλύτερη απόδοση περιορίζει την εκτέλεση του σε συγκεκριμένες πλατφόρμες. Με βάση τα γεγονότα αυτά, κάποιος καταλαβαίνει ότι η ECC-LIB είναι ένα ακατάλληλο σύστημα για μια γενική υλοποίηση σε συσκευές περιορισμένων πόρων όπως αυτές που συνθέτουν τα ασύρματα δίκτυα αισθητήρων. Όμως, στην εργασία [17] παρατηρούμε ότι κομμάτια της βιβλιοθήκης ECC-LIB μεταφέρθηκαν στο ενσωματωμένο σύστημα ATMEL AT76C520 802.11 WLAN Access Point. Παρά το γεγονός αυτό, αναφέρουμε ότι το ενσωματωμένο σύστημα AT76C520 παρέχει περισσότερες δυνατότητες από ένα τυπικό κόμβο ενός δικτύου αισθητήρων. Χαρακτηριστικά, επισημάνουμε ότι η συσκευή AT76C520 έχει έναν μικροεπεξεργαστή ARM που τρέχει στα 100 MHz, παρέχει μνήμη μεγέθους 256 MB και μπορεί να εκτελέσει το λειτουργικό σύστημα μCLinux.

Η εργασία EccM-2.0 αποτελεί την πρώτη προσπάθεια εφαρμογής ελλειπτικών καμπυλών σε συσκευές περιορισμένων πόρων. Ο κώδικας αυτής της εργασίας είναι γραμμένος σε γλώσσα nesC και στην ουσία υλοποιεί το σχήμα συμφωνίας κλειδιών ECDH μεταξύ δύο συσκευών. Έτσι, καταλαβαίνει κανείς ότι ο κώδικας αυτής της εργασίας αφορά συγκεκριμένες συσκευές (MICA 2) και δεν είναι εύκολο να επαναχρησιμοποιηθεί από ένα προγραμματιστή που θέλει να κάνει χρήση κρυπτογραφίας

ελλειπτικών καμπυλών στην εφαρμογή του. Συμπερασματικά, η εργασία EccM-2.0 δεν είναι εύκολο να αποτελέσει μια γενική λύση για εφαρμογές κρυπτογραφίας ελλειπτικών καμπυλών σε ασύρματα δίκτυα αισθητήρων.

Ο κώδικας του συτήματος TinyECC παρέχει έτοιμους κρυπτογραφικούς αλγόριθμους με ελλειπτικές καμπύλες οι οποίοι μπορούν να εκτελεστούν σε συσκευές που τρέχουν το λειτουργικό σύστημα TinyOS. Μάλιστα, οι αλγόριθμοι αυτοί αξιολογούνται πειραματικά σε διάφορες συσκευές όπως οι TelosB, MicaZ, Tmote Sky, Imote2. Το κοινό χαρακτηριστικό αυτών των συσκευών είναι ότι εκτελούν το λειτουργικό σύστημα TinyOS. Έτσι, η εργασία TinyECC παρέχει κώδικα αλγορίθμων που μπορεί σχετικά εύκολα να επαναχρησιμοποιηθεί αλλά μόνο για συσκευές που εκτελούν συγκεκριμένο λειτουργικό σύστημα (TinyOS). Επιπλέον, οι βελτιστοποιήσεις που παρέχονται σε γλώσσα μηχανής μπορούν να εφαρμοστούν μόνο στις πλατφόρμες για τις οποίες προορίζονται οπότε δεν έχουν νόημα όταν κάποιος στοχεύει σε γενικές υλοποιήσεις.

Σκοπός της διπλωματικής αυτής εργασίας είναι η δημιουργία μιας κρυπτογραφικής βιβλιοθήκης η οποία να είναι γενική, να μπορεί να μεταφραστεί εύκολα στα διάφορα λειτουργικά συστήματα που συχνά χρησιμοποιούνται σε ετερογενή ασύρματα δίκτυα αισθητήρων και κατόπιν να εκτελείται στο διαφορετικό υλικό που τα απαρτίζουν. Για το λόγο αυτό, αναπτύξαμε τον κώδικά μας χρησιμοποιώντας το προγραμματιστικό περιβάλλον της βιβλιοθήκης Wiselib (περιγράφεται στη συνέχεια), η οποία αποτελεί μια γενική βιβλιοθήκη αλγορίθμων για ετερογενή δίκτυα αισθητήρων. Η βιβλιοθήκη μας παρέχει ένα συμμετρικό αλγόριθμο κρυπτογράφησης (AES), ένα αλγόριθμο κατακερματισμού (SHA-1) και τους αλγορίθμους με ελλειπτικές καμπύλες που παρέχει η βιβλιοθήκη TinyECC. Κατά τη μεταφορά του κώδικα της εργασίας TinyECC στο περιβάλλον της Wiselib αποφασίσαμε να μη χρησιμοποιήσουμε τις διάφορες βελτιστοποιήσεις λογισμικού (προβολικές συντεταγμένες, μέθοδος ολισθαίνοντος παραθύρου, υβριδικό πολλαπλασιασμό κτλ) που παρέχει. Η απόφαση αυτή λήφθηκε με στόχο την επίτευξη μικρότερου μεταφρασμένου κώδικα ο οποίος να μην επιβαρύνει τις διάφορες εφαρμογές που θα χρησιμοποιούν τη βιβλιοθήκη μας. Παρά το γεγονός αυτό, θεωρήσαμε σκόπιμο να παρέχουμε στον εκάστοτε χρήστη τη δυνατότητα να επιλέγει την ψηφιολέξη που υποστηρίζει η πλατφόρμα για την οποία θα μεταφραστεί ο κώδικας. Με τον τρόπο αυτό είναι δυνατό να επιτυγχάνονται γρηγορότερες κρυπτογραφικές πράξεις.

Κεφάλαιο 4

Wiselib: Μια Γενική Βιβλιοθήκη Αλγορίθμων για Ετερογενή Δίκτυα Αισθητήρων

4.1 Γενική Περιγραφή της Wiselib

Παρά την επιτυχία και τη διάδοση των ασυρμάτων δικτύων αισθητήρων σε διάφορες ερευνητικές κοινότητες, υπάρχει ένα μεγάλο πρόβλημα - το χάσμα μεταξύ θεωρίας και πράξης. Ενώ υπάρχει ένας μεγάλος αριθμός αλγοριθμικών μεθόδων, η πλειοψηφία τους ποτέ δεν έχει δοκιμαστεί στην πράξη. Το γεγονός αυτό οφείλεται κυρίως στο ότι ο προγραμματισμός των κόμβων αισθητήρων γίνεται ακόμα σε πολύ τεχνικό επίπεδο.

Επιπλέον, οι διάφοροι κόμβοι αισθητήρων που παράγονται από τη σημερινή βιομηχανία έχουν διαφορετικές δυνατότητες. Κάποιοι κόμβοι αποτελούνται από μικροεπεξεργαστές 8-bit και έχουν ελάχιστη ποσότητα μνήμης RAM, ενώ άλλοι έχουν πολύ μεγάλη υπολογιστική δύναμη και μπορούν να εκτελέσουν desktop λειτουργικά συστήματα όπως Linux. Έτσι λοιπόν, ενώ είναι σχετικά εύκολο να γράψει κανείς κώδικα για μια συγκεκριμένη πλατφόρμα είναι πολύ δύσκολο να γράψει γενικό κώδικα ανεξάρτητο της πλατφόρμας. Ακόμα χειρότερα, τα λειτουργικά συστήματα που τρέχουν σε μικροσκοπικούς κόμβους παρέχουν μικρή λειτουργικότητα για την υλοποίηση απλών αλγορίθμων, γεγονός που σημαίνει ότι ένας προγραμματιστής πρέπει να δώσει προσοχή σε λεπτομέρειες χαμηλού επιπέδου.

Η βιβλιοθήκη Wiselib [3] δίνει μια λύση στα παραπάνω προβλήματα. Η Wiselib αποτελεί μια γενική βιβλιοθήκη αλγορίθμων η οποία επιτρέπει την υλοποίηση και ανάπτυξή τους σε μια ποικιλία υλικού και λογισμικού. Αυτό επιτυγχάνεται με τη χρήση προηγμένων τεχνικών της γλώσσας C++ όπως τα πρότυπα (templates) και οι inline συναρτήσεις. Οι τεχνικές αυτές επιτρέπουν τη συγγραφή γενικού κώδικα ο οποίος αναλύεται και δεσμεύεται κατά τη διαδικασία μεταγλώτισσης με αποτέλεσμα να μην υπάρχει πλεονασμός (overhead) μνήμης ή υπολογισμού κατά την εκτέλεσή του.

Η Wiselib μπορεί να εκτελεστεί σε διαφορετικά λειτουργικά συστήματα όπως

το Contiki, το iSense OS, το TinyOS καθώς και το ScatterWeb. Επιπλέον, μπορεί να εκτελεστεί σε εικονικούς κόμβους οι οποίοι προσομοιώνονται από το εργαλείο Shawn [15]. Για οποιοδήποτε αλγόριθμο, η Wiselib παρέχει κατάλληλες δομές δεδομένων οι οποίες ταιριάζουν με τις ιδιότητες της στοχευόμενης πλατφόρμας. Ο κώδικας του αλγορίθμου δεν περιέχει ειδικεύσεις (specializations) συγκεκριμένης πλατφόρμας, επιτρέποντας έτσι την υλοποίηση να τρέχει σε ετερογενή δίκτυα. Κάθε υλοποιημένος αλγόριθμος της Wiselib μπορεί να μεταγλωττιστεί για οποιοδήποτε υποστηριζόμενο σύστημα χωρίς καμία αλλαγή στον κωδικά του. Επιπλέον, παρέχεται στον προγραμματιστή μια ενοποιημένη διεπαφή (API - Application Programming Interface).

Εν συνεχεία, συνοψίζονται τα ζητήματα τα οποία αντιμετωπίζει η βιβλιοθήκη Wiselib.

Ανεξαρτησία από την Πλατφόρμα : Ο κώδικας της Wiselib μπορεί να μεταγλωττιστεί για διάφορες πλατφόρμες υλικού, χωρίς ρυθμίσεις εξαρτώμενες από την πλατφόρμα, π.χ. χωρίς ♯ ifdef δηλώσεις στον κώδικα.

Ανεξαρτησία από το Λειτουργικό Σύστημα: Ο κώδικας της Wiselib μπορεί να μεταγλωττιστεί για διάφορα λειτουργικά συστήματα. Σαν παραδείγματα αναφέρουμε το Contiki το οποίο είναι ένα C-like λειτουργικό, το υλοποιημένο σε C++ iSense καθώς και το TinyOS που υποστηρίζει τη γλώσσα nesC.

Ανταλλαξιμότητα : Οι αλγόριθμοι και οι εφαρμογές μπορούν να αποτελούνται από διαφορετικά δομικά στοιχεία τα οποία επικοινωνούν χρησιμοποιώντας καλώς ορισμένες διεπαφές που ονομάζονται concepts. Τα δομικά στοιχεία μπορούν να ανταλλάζονται με αυτά άλλων υλοποιήσεων χωρίς να επηρεάζεται ο υπόλοιπος κώδικας. Επιπλέον, γενικά δομικά στοιχεία καθώς και βελτιστοποιημένα με βάση την πλατφόρμα δομικά στοιχεία μπορούν να χρησιμοποιούνται ταυτόχρονα.

Κάλυψη Ποικιλίας Αλγορίθμων: Η Wiselib καλύπτει μια μεγάλη ποικιλία αλγορίθμων όπως αλγόριθμους δρομολόγησης (routing algorithms), αλγόριθμους ομαδοποίησης (clustering algorithms), αλγόριθμους συγχρονισμού (time-synchronization algorithms), αλγόριθμους εντοπισμού (localization algorithms), αλγόριθμου κρυπτογραφίας (cryptographic algorithms) καθώς και αλγόριθμους διάδοσης δεδομένων (data dissemination algorithms).

Αλγόριθμοι Πολλαπλών Επιπέδων : Στη Wiselib ένας αλγόριθμος μπορεί να σχεδιαστεί έτσι ώστε να χρησιμοποιεί το concept ενός άλλου αλγορίθμου με αποτέλεσμα να δημιουργείται ένας πιο σύνθετος και πολύπλοκος αλγόριθμος. Επιπλέον, αλγόριθμοι μπορούν να στοιβαχτούν ο ένας πάνω από τον άλλο έτσι ώστε να επεκτείνεται η λειτουργία τους.

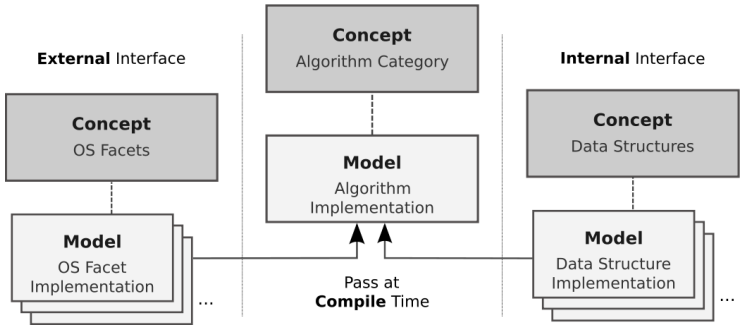
Συμβατότητα με Πρότυπα : Η Wiselib είναι γραμμένη σε ένα καλώς ορισμένο υποσύνολο της γλώσσας ISO C++. Το γεγονός αυτό έχει ένα σημαντικό οφέλος σε σχέση με γλώσσες όπως η nesC: οι μεταγλωττιστές είναι πιο ώριμοι και καλύτερα υποστηριζόμενοι. Επιπλέον, ενά μεγάλο ποσοστό χρηστών γνωρίζει τη γλώσσα C++ καθώς είναι πολύ διαδεδομένη.

Κλιμακωσιμότητα και Αποτελεσματικότητα: Η Wiselib μπορεί να εκτελεστεί σε μια ποικιλία υλικού, το οποίο μπορεί να αποτελείται από μικροεπεξεργαστές των 8-bit εώς επεξεργαστές RISC των 32-bit και με μνήμη από μερικά kilobytes μέχρι κάποια megabytes. Οι αλγόριθμοι είναι φιλικοί ως προς τους πόρους για τις περιορισμένες πλατφόρμες αλλά ταυτόχρονα μπορούν να χρησιμοποιούν περισσότερους πόρους εφόσον υποστηρίζονται από το υλικό.

4.2 Αρχιτεκτονική της Wiselib

Όπως έχουμε ήδη αναφέρει, ο πυρήνας της βιβλιοθήκης Wiselib αποτελείται από γενικές προγραμματιστικές τεχνικές οι οποίες έχουν υλοποιηθεί χρησιμοποιώντας τα πρότυπα (templates) της γλώσσας C++. Η βασική ιδέα είναι να περνάται σαν παράμετρος προτύπου σε έναν αλγόριθμο, η βασική του λειτουργία (υλοποίηση λειτουργικού συστήματος, δομές δεδομένων). Έτσι, είναι δυνατό να μεταφραστεί ένας αλγόριθμος με βάση τις τρέχουσες ανάγκες.

Η σχεδιαστική αρχή της Wiselib βασίζεται στις έννοιες των concepts και models. Η αρχιτεκτονική της αποτελείται από τρια βασικά κομμάτια όπως φαίνεται στο Σχήμα 4.1: τους αλγορίθμους, τις όψεις (facets) λειτουργικών συστημάτων και τις δομές δεδομένων.



Σχήμα 4.1: Η Αρχιτεκτονική της Βιβλιοθήκης Wiselib.

Πρώτα από όλα, υπάρχουν τα concepts για τους αλγορίθμους. Κάθε κατηγορία αλγορίθμων ομαδοποιεί τους αλγόριθμους με βάση τη λειτουργικότητά τους, π.χ. αλγόριθμοι δρομολόγησης, αλγόριθμοι εντοπισμού και υπάρχει ένα concept για κάθε τέτοια κατηγορία. Οποιοδήποτε μοντέλο αλγορίθμου, υλοποιεί ένα ή περισσότερα concepts και ουσιαστικά αποτελεί ένα πρότυπο το οποίο δέχεται διάφορες παραμέτρους. Οι παράμετροι αυτοί μπορεί να είναι όψεις λειτουργικών συστημάτων και δομές δεδομένων.

Οι όψεις λειτουργικού συστήματος (OS facets) αντιπροσωπεύουν τη σύνδεση με το υποκείμενο λειτουργικό σύστημα ή firmware, π.χ. σύνδεση με το radio ή τον timer του λειτουργικού. Έτσι, οι όψεις παρέχουν ένα ελαφρύ επίπεδο αφαίρεσης του λειτουργικού συστήματος. Στο σημείο αυτό, επισημάνουμε ότι οι όψεις είναι απλώς ορισμοί τύπων και wrapper συναρτήσεις οπότε δεν κάνουν αναπαραγωγή της λειτουργικότητας του λειτουργικού συστήματος.

Με τη βοήθεια των δομών δεδομένων, ένας αλγόριθμος μπορεί να ταιριάξει στις ανάγκες της πλατφόρμας για την οποία μεταφράζεται. Παραδείγματος χάριν, στατικές δομές δεδομένων μπορούν να χρησιμοποιηθούν για πλατφόρμες που δεν προσφέρουν δυναμική δέσμευση μνήμης ενώ δυναμικές και αποτελεσματικές δομές μπορούν να χρησιμοποιηθούν σε δυνατούς μικροελεγκτές και desktop PCs.

4.2.1 External Interface Εξωτερική Διεπαφή)

Η εξωτερική διεπαφή η οποία αποτελείται από όψεις λειτουργικού συστήματος, αναπαριστά τη σύνδεση με το υποκείμενο λειτουργικό σύστημα. Οι υλοποιήσεις

αυτών των όψεων περνώνται στον αλγόριθμο σαν παράμετροι προτύπου. Ο μεταγλωττιστής είναι ικανός να αναλύσει άμεσα τέτοιες κλήσεις στο λειτουργικό σύστημα. Παραδείγματος χάριν, για την εγγραφή στον χρονομετρητή (timer) απαιτείται μια γραμμή κώδικα και υλοποιείται σαν inline συνάρτηση στο κατάλληλο μοντέλο του χρονομετρητή. Με τον τρόπο αυτό, γίνεται μια άμεση κλήση στη συνάρτηση του λειτουργικού συστήματος η οποία δεν παράγει πλεονασμό τόσο στο χρόνο εκτέλεσης όσο και στο μέγεθος του μεταφρασμένου κώδικα. Για τα λειτουργικά συστήματα που βασίζονται στη γλώσσα C, οι όψεις λειτουργικού συστήματος πρέπει να παρέχουν μια μετάφραση μεταξύ των C++ μελών κλήσεων συναρτήσεων και των C κλήσεων συναρτήσεων καθώς και να μετατρέπουν τα C++ μέλη σε C δείκτες. Αυτό είναι ένα τίμημα της γενικότητας το οποίο πρέπει να πληρωθεί.

Ένα παράδειγμα ενός μοντέλου για την όψη λειτουργικού συστήματος radio (RadioFacet) αναφέρεται παρακάτω. Βασίζεται στο iSense firmware το οποίο βασίζεται στη γλωσσα C++.

```
1 concept RadioFacet
2 {
3   [...]
4
5   int send( node_id_t receiver, size_t len, block_data_t *data );
6   node_id_t id();
7   template<class T, void (T::*TMethod)(node_id_t, size_t, block_data_t*)>
8     int reg_recv_callback( T *obj_pnt );
9 }
```

Το παράδειγμα αυτό υποδεικνύει την υλοποίησης μιας μεθόδου send() την οποία προσφέρει το μοντέλο του radio για την αποστολή μηνυμάτων, μια μέθοδο για την επανάκληση της κατάλληλης συνάρτησης κατά την λήψη μηνυμάτων και μια μέθοδο για την επιστροφή της ταυτότητας του κόμβου.

Κληρονομία Εννοιών: Το παραπάνω παράδειγμα της μεθόδου send() του μοντέλου radio με παραμέτρους τη διεύθυνση προορισμού και το φορτίο (payload) ορίζεται στο concept του radio. Για παράδειγμα, οι αλγόριθμοι δρομολόγησης που στέλνουν και λαμβάνουν μηνύματα χωρίς επιπλέον πληροφορίες (όπως π.χ. τιμές RSSI (Received Signal Strength Indication)) μπορούν να χρησιμοποιήσουν υλοιποιήσεις αυτού του concept. Επιπλέον, αυτό το βασικό radio concept μπορεί εύκολα να επεκταθεί. Εάν ένας αλγόριθμος απαιτεί πρόσβαση σε τιμές RSSI, ένα παραγόμενο concept μπορεί να χρησιμοοιηθεί. Το concept αυτό επεκτείνει το βασικό και η μέθοδος receive() παρέχει τις επιπλέον πληροφορίες.

Stackability: Μια βασική σχεδιαστική αρχή του radio concept είναι η έννοια του stackability, δηλαδή η δυνατότητα να χτιστεί μια πολυεπίπεδη δομή από πολλαπλά radio. Το επίπεδο που βρίσκεται στην κορυφή δε γνωρίζει σε ποια και σε πόσα επίπεδα είναι συνδεδεμένο. Το πλεονέκτημα αυτής της προσέγγισης είναι ότι μπορεί να υλοποιηθεί ένα εικονικό radio, το οποίο εκτελείται πάνω από το βασικό μοντέλο και περνιέται σαν παράμετρος πρότυπο σε ένα αλγόριθμο. Με τον τρόπο αυτό, μπορεί εύκολα να υλοποιηθεί ένας αλγόριθμος για ετερογενή δίκτυα. Επιπλέον, είναι δυνατή η επικοινωνία μεταξύ κόμβων που χρησιμοποιούν διαφορετικού είδους ταυτότητες (node IDs), καθώς το εικονικό radio *κρύβει* τις πραγματικές ταυτότητες των κόμβων χρησιμοποιώντας γενικές διευθύνσεις (π.χ. 128-bit).

Μια άλλη δυνατότητα είναι να *κρυφτεί* τελείως ένας αλγόριθμος δρομολόγησης πίσω από μια όψη λειτουργικού συστήματος. Για παράδειγμα, η διαδικασία εκτύπωσης debug μηνυμάτων γίνεται μέσω της θύρας UART. Χρησιμοποιώντας ένα μοντέλο, είναι δυνατό τα debug μηνύματα να προωθηθούν σε ένα gateway. Ο αλγόριθμος δρομολόγησης δε χρειάζεται να γνωρίζει το μοντέλο πάνω από το οποίο λειτουργεί, παρά μόνο να χρησιμοποιεί το κατάλληλο concept.

Παράδοση Μηνυμάτων σε Ετερογενή Συστήματα : Ένα άλλο θέμα που αντιμετωπίζει ο σχεδιασμός λογισμικού της Wiselib είναι η παράδοση μηνυμάτων σε ετερογενή δίκτυα. Τα προβλήματα που υπάρχουν είναι δύο: η διάταξη της ψηφιολέξης (byte order) και διαφορές στο χειρισμό ευθυγράμμισης (alignment handling). Τα θέματα διάταξης της ψηφιολέξης επιλύονται, βάζοντας επιπλέον πληροφορία στα μηνύματα που κυκλοφορούν στο δίκτυο. Τα θέματα ευθυγράμμισης επιλύονται με ειδίκευση των προτύπων. Προσφέρεται μια κλάση serialization η οποία προσφέρει γενικές μεθόδους read και write για όλους τους τύπους δεδομένων.

4.2.2 Internal Interface Εσωτερική Διεπαφή)

Η εσωτερική διεπαφή παρέχει δομές δεδομένων οι οποίες μπορούν να χρησιμοποιηθούν από έναν αλγόριθμο. Παραδείγματος χάριν, αναφέρουμε ένα πίνακα δρομολόγησης ο οποίος χρειάζεται για αλγορίθμους δρομολόγησης ή μια λίστα γειτόνων που απαιτείται από έναν αλγόριθμο ανακάλυψης γειτονιάς. Ο λόγος για τον οποίο τέτοιες δομές δεδομένων *κρύβονται* πίσω από concepts, είναι ότι η Wiselib στοχεύει στην εκτέλεση του κώδικα της σε διαφορετικές πλατφόρμες. Από μικροσκοπικούς μικροελεγκτές όπως ο MSP430 ο οποίος τρέχει firmware το οποίο δεν υποστηρίζει δυναμική ανάθεση μνήμης και δυνατούς κόμβους όπως ο iMote2 ο οποίος μπορεί να τρέξει Linux μέχρι και εξομοιωτές που τρέχουν σε κανονικούς desktop υπολογιστές. Στην πρώτη περίπτωση απαιτούνται στατικές δομές δεδομένων των οποίων το μέγεθος είναι γνωστό, ενώ στην περίπτωση του desktop υπολογιστή προσφέρεται STL. Η προσέγγιση της εσωτερικής διεπαφής επιτρέπει το πέρασμα της ακριβής δομής δεδομένων η οποία είναι κατάλληλη για την προοριζόμενη πλατφόρμα.

pSTL

Τα περισσότερα συστήματα δικτύων αισθητήρων δεν παρέχουν τη δυνατότητα δυναμικής δέσμευσης μνήμης. Δεδομένου ότι καμία παραλλαγή της STL (Standard Template Library) δεν ικανοποιεί τις απαιτήσεις τέτοιων συστημάτων (π.χ. μη χρήση: της libstdc++, των κανόνων new/delete καθώς και εξαιρέσεων (exceptions» η Wiselib παρέχει την pSTL (pico STL), μια υλοποίηση κομματιών της STL χωρίς χρήση δυναμικής μνήμης και εξαιρέσεων. Τη δεδομένη χρονική στιγμή, υπάρχουν υλοποιήσεις των δομών map, vector και list.

pMP

Σε διάφορες εφαρμογές ενσωματωμένων συστημάτων, όπως στην κρυπτογραφία και στη συνάθροιση δεδομένων απαιτείται η χρήση αριθμητικής πολλαπλής ακρίβειας (multi-precision arithmetic). Υπάρχουν διάφορες βιβλιοθήκες λογισμικού οι οποίες υλοποιούν πράξεις με μεγάλους αριθμούς, όπως η GnuMP [12]. Τέτοιες βιβλιοθήκες όμως, βασίζονται κατά πολύ στη δυναμική δέσμευση μνήμης για την αναπαράσταση των μεγάλων αριθμών και για την υλοποίηση των αριθμητικών πράξεων. Επιπλέον, για την επίτευξη καλύτερης απόδοσης χρησιμοποιούν βελτιστοποιημένο κώδικα μηχανής (assembly code) ο οποίος εκμεταλλεύεται συγκεκριμένες εντολές του υλικού. Δυστυχώς, το υλικό που χρησιμοποιείται στις πλατφόρες δικτύων αισθητήρων, δεν υποστηρίζει δυνάμικη δέσμευση μνήμης, γεγονός που καθιστά αδύνατη τη μεταφορά των βιβλιοθηκών αυτών σε τέτοιες πλατφόρμες. Για το λόγο

αυτό, η Wiselib παρέχει την pMP (pico Multi-Precision), μια υλοποίηση βασικών αριθμητικών πράξεων με μεγάλους ακεραίους χωρίς να γίνεται χρήση δυναμικής δέσμευσης μνήμης. Είναι υλοποιημένη σε C και παρέχει βασικές αριθμητικές πράξεις και πράξεις με υπόλοιπο (modulo operations) που χρησιμοποιούνται για την ανάπτυξη κρυπτογραφικών συστημάτων. Όπως είναι φυσικό η pMP δε μπορεί να συγκριθεί ως προς την αποδοτικότητα με βιβλιοθήκες όπως η GnuMP αλλά αποτελεί μια γενική λύση για πλατφόρμες περιορισμένων πόρων.

4.2.3 Υποστήριξη Αλγορίθμων

Το κεντρικό κομμάτι της Wiselib είναι οι αλγόριθμοι. Οι αλγόριθμοι χωρίζονται σε κατηγορίες όπως αναφέρθηκε προηγουμένως. Οι υλοιποιήσεις τους, μπορεί να ανήκουν σε διάφορες κατηγορίες, κάτι το οποίο είναι κοινό για αλγορίθμους διαφόρων επιπέδων.

Κάθε κλάση αλγορίθμου αποτελείται από ένα concept το οποίο περιγράφει τη λειτουργικότητα του αλγορίθμου καθώς και από concepts για τις δομές δεδομένων που μπορεί να απαιτεί αυτός. Το γεγονός αυτό αποσυνδέει τη λογική του κάθε αλγορίθμου (η οποία είναι ίδια για όλες τις πλατφόρμες) από την αποθήκευση των δεδομένων (η οποία αλλάζει όταν ένας αλγόριθμος μεταφέρεται σε μια πλατφόρμα με διαφορετικά χαρακτηριστικά).

Ένα παράδειγμα ενός concept αλγορίθμου φαίνεται παρακάτω:

```
1  template<typename Os, typename Radio = typename Os::Radio>
2  class MyAlgorithm
3  {
4  public:
5    int init(Radio& radio) { radio_ = &radio; }
6    void run()
7    {
8      Radio::block_data_t buffer[10];
9      radio_->send( Radio::BROADCAST_ADDRESS, sizeof(buffer), buffer );
10   }
11 private:
12   Radio *radio_;
13 }
```

Στο παράδειγμα αυτό, ο αλγόριθμος MyAlgorithm αναμένει σαν πρότυπες παραμέτρους το λειτουργικό σύστημα για το οποίο θα μεταφραστεί και ένα μοντέλο ραδίου. Εν συνεχεία, αρχικοποιείται το ράδιο με τη μέθοδο init() και χρησιμοποιείται με την κλήση της μεθόδου run() στέλνοντας ένα μήνυμα μετάδοσης. Κάθε αλγόριθμος της βιβλιοθήκης Wiselib ακολουθεί αυτήν την προσέγγιση.

Ένα κέρδος του να υπάρχει μια καλώς ορισμένη διεπαφή για κάθε αλγόριθμο, είναι ότι οι αλγόριθμοι μπορούν εύκολα να εναλλαχθούν για δοκιμαστικούς σκοπούς - ιδανικά αλλάζοντας το όνομα της κλάσης στον κώδικα αρχικοποίησης κάποιος μπορεί να ελέγξει ένα διαφορετικό αλγόριθμο. Ένα δεύτερο πλεονέκτημα είναι ότι ένας προγραμματιστής μπορεί να ξεκινήσει μια νέα υλοποίηση ενός αλγορίθμου αντιγράφοντας υπάρχοντα κομμάτια κώδικα, αντί να ξεκινάει από μια φάση σχεδιασμού η οποία μπορεί να πάρει πολύ χρόνο στοχεύοντας στη μέγιστη μεταφερσιμότητα.

Η παροχή μιας ποικίλης ομάδας από υλοποιήσεις δομών δεδομένων βοηθάει πάρα πολύ στην επίτευξη κλιμακωσιμότητας. Για κάθε δομή δεδομένων π.χ. πίνακας δρομολόγησης (routing table), χάρτες γειτονικών ομάδων (neighborhood cluster maps), παρέχεται μια υλοποίηση που ταιριάζει στο εύρος των πλατφόρμων στόχων. Για αρχιτεκτονικές περιορισμένων πόρων, όπως αυτή του MSP430, απαιτούνται δομές που χρησιμοποιούν στατική αποθήκευση χώρου το μέγεθος του

οποίου είναι γνωστό κατά τη διαδικασία μεταγλώτισσης. Αναπόφευκτα, τέτοιες δομές είναι αναποτελεσματικές όσον αφορά το χρόνο εκτέλεσης. Για αρχιτεκτονικές μεγαλύτερης κλίμακας, μπορούν να χρησιμοποιηθούν βελτιστοποιημένες δομές με δυναμική διαχείριση μνήμης οι οποίες είναι πολύ πιο αποδοτικές. Η επιλογή των δομών δεδομένων μπορεί να γίνει κατά την αρχικοποίηση του αλγορίθμου και δεν επηρεάζει τον κώδικα του. Το αποτέλεσμα αυτού του γεγονότος είναι αλγόριθμοι που μπορούν να εκτελεστούν τόσο σε κόμβους περιορισμένων δυνατοτήτων αλλά και σε κόμβους με πολλούς διαθέσιμους πόρους.

4.2.4 User Applications Εφαρμογές Χρηστών)

Ο βασικός στόχος των αλγορίθμων που παρέχει η βιβλιοθήκη Wiselib είναι η ενσωμάτωση τους στις εφαρμογές των χρηστών. Η Wiselib αποτελεί αλγοριθμική βιβλιοθήκη οπότε δεν παρέχει δυνατότητες μεταγλώτισσης. Η εφαρμογή στην οποία περιλαμβάνονται οι αλγόριθμοι, είναι αυτή που αναλαμβάνει την μεταγλώτισση του κώδικα.

Η Wiselib παρέχει έτοιμα παραδείγματα για την ενσωμάτωση αλγορίθμων σε εφαρμογές χρηστών. Ιδιαίτερα, παρέχει παραδείγματα για την απευθείας ενσωμάτωση των αλγορίθμων σε εφαρμογές iSense και Shawn.

Παρόλα αυτά είναι εύκολο να δημιουργηθεί μια γενική εφαρμογή (Generic Wiselib Application), η οποία έχει τη δυνατότητα να μεταγλωτιστεί για διάφορες πλατφόρμες χωρίς να αλλάζει καμία γραμμή του κώδικα. Αυτό γίνεται με την επιλογή της πλατφόρμας στόχου (target) κατά τη μεταγλώτισση.

Αρχικά, ο προγραμματιστής πρέπει να υλοποιήσει μια μέθοδο application_main(), η οποία δέχεται μια παράμετρο του τύπου Os::AppMainParameter&. Εν συνεχεία, πρέπει να γράψει μια κλάση η οποία θα παρέχει τουλάχιστον μια μέθοδο αρχικοποίησης (init()), η οποία θα δέχεται την AppMainParameter σαν παράμετρο. Μέσα στη μέθοδο init(), πρέπει να δημιουργηθούν στιγμιότυπα των όψεων του λειτουργικού συστήματος. Τέλος, η κλάση αυτή πρέπει να περαστεί σαν πρότυπη παράμετρος σε μια κλάση WiselibApplication, η οποία καλείται από την application_main μέθοδο. Συνολικά, μια γενική εφαρμογή Wiselib έχει την ακόλουθη μορφή:

```
1  #include "external_interface/external_interface.h"
2  #include "algorithms/routing/tree/tree_routing.h"
3
4  // OSMODEL is defined by the provided makefiles, and
5  // thus adapts automatically to current platform
6
7  typedef wiselib::OSMODEL Os;
8  typedef wiselib::TreeRouting<Os, Os::Radio, Os::Timer, Os::Debug> TreeRouting;
9
10 class TreeRoutingApplication
11 {
12    public:
13       void init( Os::AppMainParameter& value )
14       {
15          Os::Radio& radio = wiselib::FacetProvider<Os, Os::Radio>::get_facet( value );
16          Os::Timer& timer = wiselib::FacetProvider<Os, Os::Timer>::get_facet( value );
17          Os::Debug& debug = wiselib::FacetProvider<Os, Os::Debug>::get_facet( value );
18
19          tree_routing.init( radio, timer, debug );
20          timer.set_timer<TreeRoutingApplication,
21           &TreeRoutingApplication::start>( 500, this, 0 );
22       }
23       // -----------------------------------------------------------------
24       void start( void* )
25       {
26          tree_routing.send( 1, 0, 0 );
```

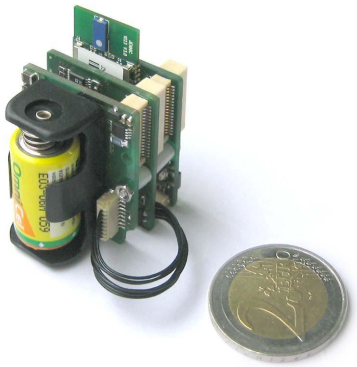
```
27         }
28      private:
29         TreeRouting tree_routing;
30  };
31  // --------------------------------------------------------------------
32  wiselib::WiselibApplication<Os, TreeRoutingApplication> routing_app;
33
34  void application_main( Os::AppMainParameter& value )
35  {
36      routing_app.init( value );
37  }
```

Κεφάλαιο 5

Υλικό

5.1 Η Πλατφόρμα iSense

Το iSense αποτελεί μια πλατφόρμα υλικού και λογισμικού για ασύρματα δίκτυα αισθητήρων. Η βασιζόμενη στα πρότυπα προσέγγιση που υποστηρίζει, επιτρέπει στο υλικό και στο λογισμικό να συμμορφώνονται πλήρως με τις απαιτήσεις της εκάστοτε εφαρμογής. Ο σχεδιασμός της πλατφόρμας που στηρίζεται στην χαμηλή κατανάλωση, επιτρέπει την εκτέλεση μιας μεγάλης αυτόνομης λειτουργίας, η οποία σε συνδυασμό με την ικανότητα για ασύρματο επαναπρογραμματισμό, καταλήγει στην εύκολη διαχείριση του δικτύου. Ο πυρήνας είναι συμβατός με το πρότυπο IEEE 802.15.4 και υποστηρίζει ZigBee ράδιο, κρυπτογράφηση στο επίπεδο του υλικού καθώς και υψηλούς ρυθμούς μετάδοσης δεδομένων. Η διεπαφή του λογισμικού είναι αρκετά ευέλικτη και παρέχει πλούσια ποικιλία προτύπων και γνωστών εργαλειών για την γρήγορη ανάπτυξη εφαρμογών.



Σχήμα 5.1: Μια Συσκευή iSense.

5.1.1 Επισκόπηση του Υλικού

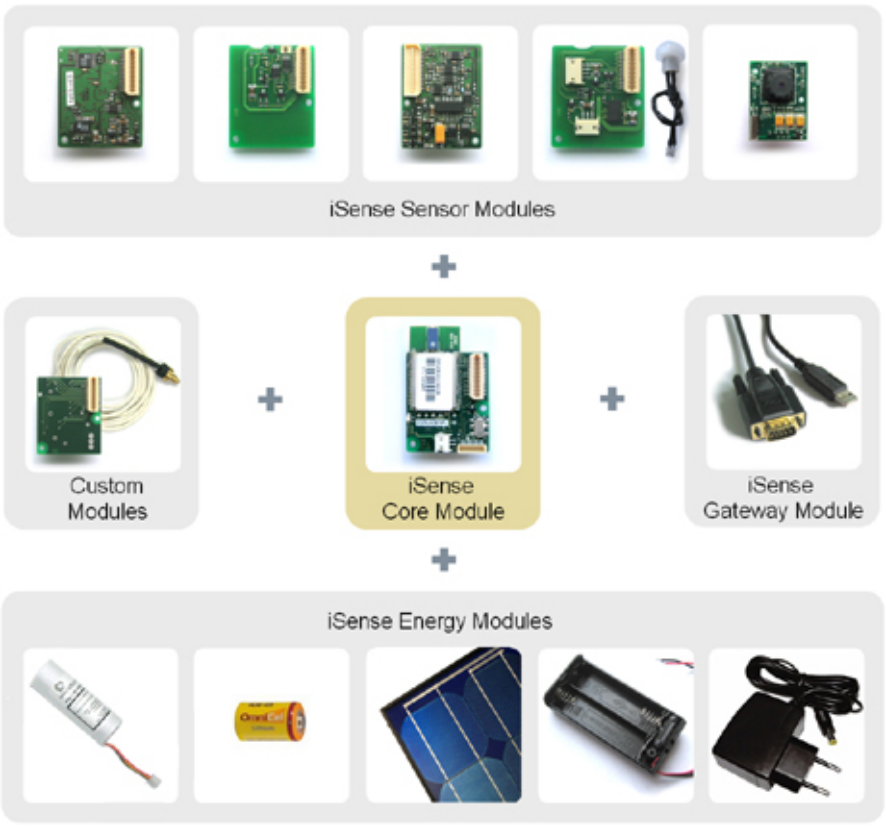
Η πλατφόρμα του υλικού αποτελείται από διάφορα στοιχεία (modules) τα οποία μπορούν να συνδυαστούν με διάφορους τρόπους ανάλογα με τις απαιτήσεις των εφαρμογών. Με τον τρόπο αυτό η λειτουργικότητα της κάθε εφαρμογής μπορεί

να επαναπροσδιοριστεί καθώς νέα χαρακτηριστικά μπορούν να προστεθούν προσάπτοντας νέα στοιχεία υλικού.

Κατά την τρέχουσα περίοδο, τα εξής στοιχεία υλικού είναι διαθέσιμα:

- core module (υπεύθυνο για υπολογισμούς και ασύρματη επικοινωνία)
- power module
- gateway module (για σύνδεση με υπολογιστές)
- GPS module (για εύρεση της θέσης των συσκευών)
- vehicle detection module (για ανίχνευση μεγάλων μεταλλικών αντικειμένων)
- solar power system (για self powered δίκτυα)

Τα στοιχεία αυτά φαίνονται στην παρακάτω εικόνα.



Σχήμα 5.2: Τα Στοιχεία Υλικού της Πλατφόρμας iSense.

Η καρδία της πλατφόρμας υλικού είναι το core module. Το στοιχείο αυτό διαχειρίζεται τον μικρο-ελεγκτή Jennic JN5139, ένα chip στο οποίο περιέχονται ο ελεγκτής και ο πομποδέκτης ασύρματης επικοινωνίας. Ο ελεγκτής υποστηρίζει 32-bit RISC υπολογισμούς και τρέχει στα 16 MHz. Αποτελείται από 96 Kb μνήμης τα οποία μοιράζονται μεταξύ του κώδικα του προγράμματος και των δεδομένων.

Το πλεονέκτημα αυτής της ιδέας είναι ότι η κατανάλωση μνήμης μπορεί να εξισορροπείται ανάμεσα στα δεδομένα και τον κώδικα του προγράμματος σε αντίθεση με άλλους ελεγκτές που ο χρήστης περιορίζεται σε συγκεκριμένο μέγεθος μνήμης για δεδομένα και συγκεκριμένο για κώδικα.

Ο πομποδέκτης ασύρματης επικοινωνίας είναι συμβατός με το πρότυπο ασύρματης επικοινωνίας IEEE 802.15.4. Υποστηρίζει ρυθμούς μετάδοσης δεδομένων μέχρι 250 KBits/s καθώς και κρυπτογράφηση στο επίπεδο του υλικού με το βάση πρότυπο AES : Advanced Encryption Standard. Το εύρος επικοινωνίας φτάνει μέχρι τα 500m, δεδομένων της ευαισθησίας λήψης περίπου -97 dBm και της ισχύος μετάδοσης ανάμεσα σε -60 dBm και +3 dBm. Εκτός από τη βασική έκδοση η οποία είναι εξοπλισμένη με SMA σύνδεσμο κεραίας, υπάρχουν συσκευές με εσωτερική κεραία για πιο συμπαγή συστήματα και συσκευές με επιπρόσθετο ενισχυτή σήματος για εύρη επικοινωνίας μέχρι 2 Km.

Ένα σύνηθες δίλημμα κατά το σχεδιασμό είναι η χρήση ή όχι ενός ρυθμιστή τάσης. Η χρήση ενός ρυθμιστή τάσης έχει το πλεονέκτημα της σωστής λειτουργίας μιας συσκευής με τάση χαμηλότερη από την απαιτούμενη. Το μειονέκτημα είναι ότι ο ρυθμιστής σπαταλάει ενέργεια. Ειδικότερα στην περίπτωση που η τάση είναι μεγαλύτερη από την απαιτούμενη, ο ρυθμιστής σπαταλάει ενέργεια χωρίς η χρήση του να είναι απαραίτητη. Για την αντιμετώπιση αυτού του προβλήματος, δίνεται η δυνατότητα ενεργοποίησης του ρυθμιστή τάσης μέσω του λογισμικού. Με τον τρόπο αυτό ο ρυθμιστής τάσης αποφεύγεται όταν δεν χρειάζεται και μπορεί να ενεργοποιείται όταν η τάση είναι χαμηλή.

Για την υποστήριξη συγχρονισμού, το στοιχείο αυτό είναι εξοπλισμένο με ένα ρολόι μεγάλης ακρίβειας. Ένας σύνδεσμος 34-pin βρίσκεται και στις δύο πλευρές του core module έτσι ώστε άλλα στοιχεία του υλικού να μπορούν να συνδεθούν με αυτό. Το core module μπορεί να παρέχει μέχρι 500 mA στα στοιχεία υλικού που συνδέονται με αυτό.

Ο ελεγκτής μπορεί να προγραμματιστεί με διάφορους τρόπους. Ο βασικός είναι ο OTAP : Over The Air Programming, αλλά επίσης το πρόγραμμα μπορεί να μεταφερθεί μέσω του gateway module(που παρουσιάζεται παρακάτω) ή μέσω ενός ειδικού προγραμματιστικού adapter που συνδέεται με το core module.

Το core module σε πλήρη λειτουργία καταναλώνει περίπου 9mA και το ράδιο του περίπου 29mA. Στην περίπτωση που μια συσκευή βρίσκεται σε κατάσταση αδράνειας η κατανάλωση μπορεί να πέσει μέχρι 10μA. Το στοιχείο αυτό μπορεί να δέχεται τροφοδοσία από την πρίζα, από κλασικές μπαταρίες, από τα power modules και από τη USB διεπαφή του gateway module.

Διάφορα power modules είναι διαθέσιμα στην πλατφόρμα iSense. Το lithium-ion module είναι ένας συνδυασμός μιας επαναφορτιζόμενης μπαταρίας μεγάλης χωρητικότητας, με ένα ελεγκτή κατανάλωσης και ένα ψηφιακό όργανο παρακολούθησης της τάσης. Το module αυτό επιτρέπει την φόρτιση της μπαταρίας των συσκευών με την σύνδεση του συστήματος στην πρίζα με ειδικό επαφέα καθώς και την παροχή της πληροφορίας της τρέχουσας ενέργειας που έχει παραμείνει στην μπαταρία. Το coin cell module είναι σχεδιασμένο για συγκεκριμένα συμπαγή συστήματα. Αποτελείται από μία μπαταρία CR2477 και ένα όργανο παρακολούθησης της ακριβής κατανάλωσης της μπαταρίας. Τέλος, το module iSense 1/2AA Battery αποτελείται από μία βάση μπαταρίας 1/2AA και μια μικρή οθόνη για την παροχή πληροφοριών όπως ενέργεια που έχει καταναλωθεί και ενέργεια που έχει απομείνει.

Το gateway module επιτρέπει τη σύνδεση σε υπολογιστές καθώς και σε άλλα δίκτυα. Εξουσιοδοτεί την ανταλλαγή δεδομένων καθώς και τον προγραμματισμό

Σχήμα 5.3: iSense Core Module.

Σχήμα 5.4: Το Power Module 1/2AA Battery της Πλατφόρμας iSense.

των συνδεδεμένων core modules. Εκτός από διάφορα LEDS, κουμπιά και ένα ποτενσιόμετρο παρέχει διεπαφές USB και RS232.

5.1.2 Επισκόπηση του Λογισμικού

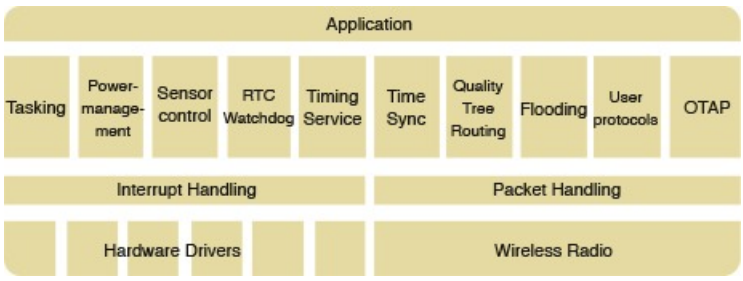
Το λογισμικό που εκτελείται σε κάθε κόμβο του δικτύου απαιτεί την ίδια ευελιξία σχεδιασμού αντίστοιχη με αυτή του υλικού της πλατφόρμας iSense. Ένας από τους βασικούς στόχους σχεδιασμού λογισμικού είναι η χρήση προγραμματιστικών μεθόδων που είναι ευρέως διαδεδομένες και αντιληπτές από την ερευνητική κοινότητα. Προηγμένες τεχνικές, όπως ο αντικειμενοστραφής C++ προγραμματισμός και η δυναμική ανάθεση μνήμης, επιτρέπουν τη γρήγορη και σωστή ανάπτυξη εφαρμογών. Τέτοιες τεχνικές συνήθως δεν είναι διαθέσιμες σε δίκτυα αισθητήρων. Η πλατφόρμα iSense υποστηρίζει μια παρόμοια με STL υλοποίηση η οποία παρέχει μεθόδους για χρήση lists, sets και maps. Έτσι η ανάπτυξη εφαρμογών στην πλατφόρμα iSense βασίζεται πλήρως σε γνωστές τεχνολογίες.

Όπως και το υλικό, αντίστοιχα και το λογισμικό της πλατφόρμας iSense οργανώνεται σε ένα σύνολο από στοιχεία (modules) καθένα από τα οποία προσφέρει

Σχήμα 5.5: Τα Power Modules της Πλατφόρμας iSense.

Σχήμα 5.6: iSense Gateway Module με USB Καλώδιο.

μια εξειδικευμένη υπηρεσία στην εφαρμογή. Όταν ένας χρήστης αναπτύσσει μια εφαρμογή, συναθροίζει τα στοιχεία σε ένα λειτουργικό σύστημα που παρέχει τις επιθυμητές λειτουργίες. Η επιλογή αυτών των λειτουργιών όπως η υποστήριξη συγκεκριμένων modules, υποστήριξη διαφόρων πρωτοκόλλων και αριθμών κινητής υποδιαστολής μπορεί να γίνει από τον κάθε χρήστη κατά το web-configuration. Η παρακάτω εικόνα παρουσιάζει την αρχιτεκτονική του λογισμικού της πλατφόρμας iSense η οποία αποτελείται από τέσσερα διακριτά επίπεδα: το αφαιρετικό επίπεδο του υλικού, το επίπεδο του λειτουργικού συστήματος, το επίπεδο δικτυακής υποστήριξης και το επίπεδο των εφαρμογών των χρηστών.



Σχήμα 5.7: Η Δομή του Λογισμικού της Πλατφόρμας iSense.

Το αφαιρετικό επίπεδο του υλικού (HAL: Hardware Abstraction Layer) συμπεριλαμβάνει τις λειτουργίες του και αποκρύπτει από τους προγραμματιστές τις

περίπλοκες λεπτομέρειες του, παρέχοντας τους μια διεπαφή για τα παραπάνω επίπεδα. Το επίπεδο αυτό παρέχει αφαιρέσεις για αλληλεπίδραση με μετατροπείς A/D και D/A, με I/O διεπαφές(π.χ. UARTs,SPI), με timers καθώς και με τον πομποδέκτη ασύρματης επικοινωνίας. Ένα τυπικό σενάριο χρήσης του HAL περιλαμβάνει την ενσωμάτωση των στοιχείων υλικού, τα οποία είναι συνήθως συνδεδεμένα σε ένα από τα I/O pins, όπως και την χρήση τους μέσω των λειτουργιών αυτού του επιπέδου. Με την αρχιτεκτονική αυτή, όλα τα modules πάνω από το HAL είναι ανεξάρτητα της συγκεκριμένης πλατφόρμας υλικού. Ο κώδικας εφαρμογής που αναπτύσσεται σε αυτό το πλαίσιο είναι έτοιμος να τρέξει σε οποιαδήποτε πλατφόρμα παρέχει υλοποίηση της iSense διεπαφής. Τη δεδομένη χρονική στιγμή, ένας προγραμματιστής μπορεί να τρέξει την εφαρμογή του στην πλατφόρμα υλικού iSense ή στο εργαλείο εξομοίωσης Shawn. Με τον τρόπο αυτό μπορεί να εξετάστεί η απόδοση και η λειτουργικότητα της εφαρμογής σε επίπεδο εξομοίωσης πριν την ανάπτυξη της σε πραγματικές συσκευές.

Πάνω από το αφαιρετικό επίπεδο του υλικού βρίσκεται το λειτουργικό σύστημα το οποίο διευκολύνει την ανάπτυξη εφαρμογών μέσω ενός προσανατολισμένου στα γεγονότα μοντέλου. Μια εφαρμογή ειδοποιείται καταλλήλως όταν λαμβάνει χώρα ένα γεγονός για το οποίο ενδιαφέρεται. Ένα γεγονός μπορεί να συμβεί μέσω της εφαρμογής (π.χ. παρέρχεται ένας timer) ή μέσω του υλικού (π.χ. λήψη δεδομένων στα I/O συστήματα, αλλαγή του σήματος στην είσοδο ενός A/D μετατροπέα). Για τα γεγονότα που λαμβάνουν χώρα μέσω της εφαρμογής το λειτουργικό σύστημα παρέχει δύο επιλογές: σε λειτουργίες που απαιτείται υψηλή ακρίβεια χρόνου, μια χρονό-υπηρεσία επιτρέπει στις ειδοποιήσεις να γίνουν αμέσως ενώ σε λειτουργίες που δεν απαιτείται ακρίβεια χρόνου οι ειδοποιήσεις μπορούν να γίνουν και αργότερα. Τέλος, το λειτουργικό σύστημα είναι υπεύθυνο για την διατήρηση των ενεργειακών πόρων των κόμβων του δικτύου, όταν αυτό είναι δυνατό. Εαν είναι επιθυμητό από τον χρήστη, η υποδομή για την διαχείριση της ενέργειας μπορεί να θέσει τις συσκευές σε λειτουργία χαμηλής κατανάλωσης.

Εκτός από την λειτουργικότητα του κάθε κόμβου, ένα βασικό συστατικό των δικτύων αισθητήρων είναι η ασύρματη επικοινωνία. Το επίπεδο HAL παρέχει κατάλληλες αφαιρέσεις της ασύρματης διεπαφής πάνω από την οποία, το επίπεδο δικτυακής υποστήριξης παρέχει δυναμικές υπηρεσίες όπως δρομολόγηση(routing), συγχρονισμό (time synchronization) και over the air προγραμματιστικά modules. Η πλατφόρμα iSense παρέχει δύο υλοποιήσεις δρομολόγησης που καλύπτουν το μεγαλύτερο κομμάτι του χώρου σχεδίασης εφαρμογών για δίκτυα αισθητήρων. Η πρώτη είναι η υλοποίηση ελεγχόμενης πλημμύρας (flooding), η οποία αποτελεί μια σταθερή και ανθεκτική στα λάθη μέθοδο για τη μεταβίβαση δεδομένων, σε ένα σύνολο κόμβων, οι οποίοι βρίσκονται σε γειτονιά n βημάτων από τον κόμβο-αποστολέα. Η δεύτερη αποτελεί μια υλοποίηση ενός δέντρου δρομολόγησης το οποίο επιτρέπει τη μεταφορά των δεδομένων του δικτύου σε έναν ή περισσότερους σταθμούς βάσης. Στο σχήμα αυτό το βασικό μέτρο για την επιλογή των συνδέσμων είναι ο ρυθμός απώλειας πακέτων έτσι ώστε να διατηρούνται μονοπάτια με υψηλούς ρυθμούς παράδοσης μηνυμάτων και να αυξάνεται η αξιοπιστία του δικτύου.

Στις εφαρμογές των δικτύων αισθητήρων ο συγχρονισμός των ρολογιών των κόμβων είναι ζωτικής σημασίας έτσι ώστε λειτουργίες όπως συνάθροιση δεδομένων (data aggregation) να εκτελούνται ομαλά. Το χαρακτηριστικό αυτό ενσωματώνεται σαν module στο iSense και παρέχεται στους προγραμματιστές που χρησιμοποιούν αυτή τη λειτουργία, ακριβής συγχρονισμός των ρολογιών των κόμβων με απόκλιση λιγότερη από 1ms ανά δέκα βήματα. Ακόμα ένα πολύ σημαντικό

module παρέχει τη δυνατότητα του ασύρματου επαναπρογραμματισμού ενός ήδη ανεπτυγμένου δικτύου. Αυτή η διαδικασία που ονομάζεται OTAP (Over the Air Programming) διασφαλίζει την ευέλικτη ανάπτυξη και λειτουργία των δικτύων αισθητήρων καθώς περιττεύουν οι ενσύρματες συνδέσεις και δεν απαιτείται μαζικός προγραμματισμός των συσκευών.

Εκτός από τα χαρακτηριστικά που παρέχει η πλατφόρμα iSense ένα διαδεδομένο και ευρέως αποδεκτό περιβάλλον ανάπτυξης είναι απαραίτητο για την ανέγερση επιτυχών εφαρμογών.Το λογισμικό του iSense χρησιμοποιεί διάσημα εργαλεία ανάπτυξης όπως ο GCC: GNU Compiler Collection και το εργαλείο ανάπτυξης Eclipse. Επιπλέον, το iSense παρέχει το iShell, ένα κατάλληλο μέσο για την αλληλεπίδραση με το δίκτυο αισθητήρων. Το εργαλείο αυτό συνδυάζει τη λειτουργικότητα ενός σειριακού τερματικού με προγραμματισμό OTAP (Over the Air Programming) των αισθητήρων. Επιπροσθέτως, παρέχει ευέλικτα plug-in συστήματα για την ενσωμάτωση λειτουργιών ορισμένων από τον χρήστη όπως ανάλυση δεδομένων και παρακολούθηση της ασύρματης επικοινωνίας στο δίκτυο.Τέλος, το iSense και το iShell παρέχουν προαιρετικά πολυπλεγμένες υπηρεσίες έτσι ώστε οι εφαρμογές να μπορούν να χρησιμοποιούν ανεξάρτητες ροές δεδομένων.

5.1.3 Βασικά Χαρακτηριστικά της Πλατφόρμας iSense

Με βάση όσα αναφέραμε παραπάνω συνοψίζουμε τα πλεονεκτήματα της πλατφόρμας iSense σε σχέση με άλλες πλατφόρμες ασυρμάτων δικτύων αισθητήρων.

- Η πλατφόρμα iSense υποστηρίζει μια πρότυπη προγραμματιστική γλώσσα (C++ like) η οποία είναι προσανατολισμένη στα αντικείμενα.
- Η πλατφόρμα iSense υποστηρίζει δυναμική ανάθεση μνήμης.
- Το λειτουργικό σύστημα της πλατφόρμας iSense παρέχει πλούσιες λειτουργικότητες.
- Υπάρχουν αρκετά έτοιμα υλοποιημένα πρωτόκολλα, όπως πρωτόκολλα δρομολόγησης, μεταφοράς, συγχρονισμού και εντοπισμού.
- Υπάρχει διαθέσιμο λογισμικό για τα διάφορα modules του υλικού.
- Ο κώδικας των εφαρμογών της πλατφόρμας iSense μπορεί να εκτελεστεί στον εξομοιωτή δικτύων αισθητήρων Shawn. Αυτό βοηθάει σημαντικά στην εξέταση της λειτουργικότητας των εφαρμογών και την αποσφαλμάτωση πριν την ανάπτυξη τους σε πραγματικές συσκευές.

5.2 Η Πλατφόρμα Crossbow TelosB

Η πλατφόρμα Crossbow TelosB αποτελεί μια πλατφόρμα ανοιχτού κώδικα σχεδιασμένη έτσι ώστε να διευκολύνει τους πειραματισμούς της ερευνητικής κοινότητας πάνω στα ασύρματα δίκτυα αισθητήρων. Ένας κόμβος TelosB παρέχει όλα τα απαραίτητα στοιχεία για εργαστηριακή έρευνα : δυνατότητα προγραμματισμού μέσω θύρας USB, ενσωματωμένη κεραία με παροχή ράδιου συμβατού με το πρότυπο IEEE 802.15.4, μικρο-επεξεργαστή χαμηλής κατανάλωσης και μια ποικιλία από αισθητήρες όπως αισθητήρες φωτός, υπέρυθρης ακτινοβολίας, υγρασίας και θερμοκρασίας. Η πλατφόρμα TelosB αναπτύχθηκε και δημοσιεύτηκε στις ερευνητικές

κοινότητες από το πανεπιστήμιο Berkeley της Καλιφόρνια. Είναι αρκετά διαδεδομένη στις εφαρμογές δικτύων αισθητήρων καθώς προσφέρει μικρή κατανάλωση ρεύματος και ταχύτατη επαναφορά από την κατάσταση αδράνειας.

Σχήμα 5.8: Μια Συσκευή TelosB.

5.2.1 Επισκόπηση του Υλικού

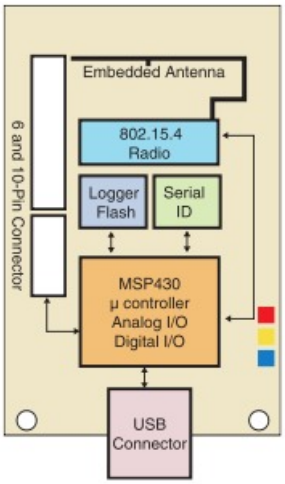
Η χαμηλής κατανάλωσης λειτουργία μιας συσκευής TelosB οφείλεται κυρίως στον μικροελεγκτή MSP430 της εταιρείας Texas Instruments. Ο μικροελεγκτής αυτός αποτελείται από έναν 16-bit RISC επεξεργαστή ο οποίος παρέχει 10Kb μνήμης RAM, 48Kb μνήμης flash καθώς και 1 Mb για αποθήκευση πληροφορίας. Ο MSP430 έχει έναν ψηφιακά ελεγχόμενο ταλαντωτή, ο οποίος λειτουργεί στα 8 MHz. Ο ταλαντωτής μπορεί να ενεργοποιηθεί ξανά από την κατάσταση αδράνειας μόλις σε 6 μs. Εκτός από τον ταλαντωτή, ο μικροελεγκτής MPS430 παρέχει 8 εσωτερικές και 8 εξωτερικές Analog to Digital Converter (ADC) θύρες, οι οποίες μπορούν να χρησιμοποιηθούν για τον έλεγχο της τάσης που παρέχουν οι μπαταρίες.

Μια συσκευή TelosB σε πλήρη λειτουργία καταναλώνει περίπου 1.8mA και το ράδιο του περίπου 23mA. Στην περίπτωση που μια συσκευή βρίσκεται σε κατάσταση αδράνειας η κατανάλωση μπορεί να πέσει μέχρι 5.1μA. Στην κατάσταση αδράνειας το ράδιο καταναλώνει μέχρι και 21 μA.

Για την επίτευξη ασύρματης επικοινωνίας, η συσκευή TelosB έχει ενσωματωμένο το chip CC2420 της εταιρείας Chipcon. Το chip αυτό επιτυγχάνει αξιόπιστη ασύρματη επικοινωνία καθώς παρέχει τις λειτουργίες του φυσικού επιπέδου (PHY) και του επιπέδου ελέγχου του μέσου (MAC) όπως αυτές περιγράφονται στο πρότυπο IEEE 802.15.4. Το CC2420 ελέγχεται από το μικροελεγκτή MSP430, μέσω της σειριακής περιφερειακής διεπαφής (SPI) και μια σειρά από γραμμές και διακοπές εισόδου/ εξόδου. Το ράδιο μπορεί να απενεργοποιηθεί από τον μικροελεγκτή στις περιπτώσεις που απαιτείται λειτουργία χαμηλής κατανάλωσης ενέργειας. Υποστηρίζει ρυθμούς μετάδοσης δεδομένων μέχρι 250 KBits/s. Το εύρος επικοινωνίας φτάνει μέχρι τα 100m, δεδομένων της ευαισθησίας λήψης περίπου -94 dBm και της ισχύς μετάδοσης ανάμεσα σε -24 dBm και +0 dBm.

Μια συσκευή TelosB μπορεί να τροφοδοτείται από δύο μπαταρίες AA. Είναι σχεδιασμένη έτσι ώστε να παρέχει ένα πακέτο υποδοχής για τις μπαταρίες. Οι AA μπαταρίες μπορούν να χρησιμοποιηθούν στο λειτουργικό εύρος των 2.1 εώς 3.6 V. Παρόλα αυτά, η τάση τροφοδοσίας πρέπει να είναι τουλάχιστον 2.7 V όταν ο χρήστης προγραμματίζει τη μνήμη flash του μικροεπεξεργαστή. Εάν η συσκευή

είναι συνδεδεμένη σε έναν υπολογιστή για προγραμματισμό ή μεταφορά δεδομένων, τότε η τροφοδοσία γίνεται από τον υπολογιστή. Εφόσον η συσκευή είναι πάντα συνδεδεμένη με έναν υπολογιστή, τότε δεν απαιτείται το πακέτο υποδοχής των μπαταριών.



Σχήμα 5.9: Το Μπλοκ Διάγραμμα μιας Συσκευής TelosB.

5.2.2 Επισκόπηση του Λογισμικού

Μια συσκευή TelosB μπορεί να εκτελέσει είτε το λειτουργικό σύστημα TinyOS είναι το λειτουργικό σύστημα Contiki. Εν συνεχεία, γίνεται μια σύντομη περιγραφή των λειτουργικών συστημάτων αυτών.

Το TinyOS αποτελεί ένα ελεύθερο ανοιχτού κώδικα λειτουργικό σύστημα, το οποίο βρίσκει εφαρμογή σε πλατφόρμες ασυρμάτων δικτύων αισθητήρων. Το TinyOS είναι ένα ενσωματωμένο λειτουργικό σύστημα γραμμένο στη γλώσσα προγραμματισμού nesC σαν ένα σύνολο από συνεργατικά έργα και διεργασίες. Το TinyOS ξεκίνησε από τη συνεργασία του πανεπιστημίου Berkeley της Καλιφόρνια με τις εταιρείες Crossbow Technology και Intel Research και από τότε έχει εξελιχθεί σε μια διεθνής κοινοπραξία με όνομα TinyOS Alliance.

Όπως αναφέρθηκε, το TinyOS είναι γραμμένο στη γλώσσα προγραμματισμού nesC, μια διάλεκτο της γλώσσας C βελτιστοποιημένη για τις περιορισμένης μνήμης εφαρμογές των δικτύων αισθητήρων. Οι συνεργαζόμενες βιβλιοθήκες και τα εργαλεία, όπως ο μεταγλωττιστής NesC και τα Atmel AVR binutils, είναι γραμμένα σε C.

Ένα πρόγραμμα TinyOS αποτελείται από δομικά συστατικά λογισμικού τα οποία παρέχουν αφαιρετικότητα ως προς το υλικό. Τα συστατικά αυτά συνδέονται μέσω διεπαφών. Το TinyOS παρέχει διεπαφές και συστατικά για λειτουργίες όπως επικοινωνία πακέτων, δρομολόγηση, αίσθηση και αποθήκευση πληροφορίας.

Το TinyOS δεν μπλοκάρει διεργασίες, οπότε οποιαδήποτε διεργασία εισόδου-/εξόδου η οποία διαρκεί περισσότερο από μερικές εκατοντάδες microseconds είναι ασύγχρονη και έχει μια συνάρτηση επανάκλησης (callback). Για να μπορεί ο με-

ταφραστής να βελτιστοποιήσει τις συναρτήσεις επανάκλησης, χρησιμοποιούνται κάποια χαρακτηριστικά της nesC για τη στατική σύνδεση των συναρτήσεων που ονομάζονται γεγονότα (events). Καθώς το TinyOS δε μπλοκάρει διεργασίες επιτυγχάνει υψηλό συγχρονισμό, αλλά αναγκάζει τους προγραμματιστές να γράψουν πολύπλοκης λογικής προγράμματα επικολλώντας μαζί πολλούς χειριστές γεγονότων. Για μεγαλύτερους υπολογισμούς, το TinyOS παρέχει εργασίες (tasks), οι οποίες είναι παρόμοια με τους χειριστές διακοπών (interrupt handlers). Ένα συστατικό TinyOS μπορεί να καλέσει μια εργασία, την οποία το λειτουργικό θα προγραμματίσει να εκτελέσει αργότερα. Οι εργασίες είναι μη-προαγοραστικές και εκτελούνται με σειρά First In First Out. Το μοντέλο συγχρονισμού αυτό είναι ικανοποιητικό για κεντρικοποιημένες εφαρμογές. Η δυσκολία του όμως με βαρειές υπολογιστικά εφαρμογές, οδήγησε στην ανάπτυξη μιας βιβλιοθήκης για νήματα (threads), με όνομα TOSThreads. Ο κώδικας του TinyOS συνδέεται στατικά με τον κώδικα του προγράμματος και μεταφράζεται σε δυαδική μορφή χρησιμοποιώντας GNU εργαλεία.

Το Contiki είναι ένα μικρό, ανοιχτού κώδικα, μεταφέρσιμο λειτουργικό σύστημα το οποίο αναπτύχθηκε για τη χρήση σε περιορισμένης μνήμης δικτυακά συστήματα όπως ενσωματωμένα συστήματα και συσκευές αισθητήρων. Παρά το γεγονός ότι το Contiki παρέχει μια ενσωματωμένη TCP/IP στοίβα, απαιτεί μερικά kilobytes κώδικα και λίγες εκατοντάδες bytes μνήμης RAM. Ένα πλήρες σύστημα, με γραφικό περιβάλλον απαιτεί περίπου 30 kilobytes μνήμης RAM ενώ μια τυπική εφαρμογή καταναλώνει 2 kilobytes RAM και 40 kilobytes ROM. Ο πυρήνας του λειτουργικού αναπτύχθηκε από τον Adam Dunkels στα πλαίσια της ερευνητικής ομάδας Network Embedded Systems στο Ινστιτούτο πληροφορικής της Σουηδίας.

Το Contiki αποτελείται από ένα οδηγούμενο από τα γεγονότα πυρήνα πάνω από τον οποίο τα προγράμματα εφαρμογών φορτώνονται και ξεφορτώνονται δυναμικά κατά το χρόνο εκτέλεσης. Οι διεργασίες του Contiki χρησιμοποιούν ελαφριά protothreads τα οποία προσφέρουν thread-like προγραμματισμό πάνω από το πυρήνα. Το Contiki μπορεί να τρέξει σε διάφορες πλατφόρμες με ποικιλία από ενσωματωμένους μικροελεγκτές όπως ο TI MSP430 και ο Atmel AVR έως παλιούς προσωπικούς υπολογιστές (pc).

Το λειτουργικό σύστημα Contiki υποστηρίζει πολλαπλά νήματα (multi-threading) καθώς και επικοινωνία μεταξύ διεργασιών με τη μεταφορά των μηνυμάτων να γίνεται μέσω των γεγονότων. Επιπλέον, υποστηρίζει προαιρετικά ένα γραφικό περιβάλλον με υποστήριξη τοπικών τερματικών ή διαδικτυακή σύνδεση μέσω τερματικού πάνω από το πρωτόκολλο Telnet. Μια πλήρης εγκατάσταση του Contiki παρέχει τα εξής χαρακτηριστικά: πολυ- επεξεργαστικό πυρήνα, μη-προαγοραστικό πολυνηματισμό, protothreads, διασύνδεση TCP/IP, γραφική διεπαφή χρήστη, φυλλομετρητή ιστού, προσωπικό εξυπηρετητή ιστού και ένα τερματικό Telnet.

5.2.3 Βασικά Χαρακτηριστικά της Πλατφόρμας TelosB

Συνοψίζοντας, τα βασικά χαρακτηριστικά που παρέχει η πλατφόρμα TelosB είναι τα εξής:

- Πομποδέκτης ραδιοσυχνοτήτων συμβατός με το πρότυπο IEEE 802.15.4
- Ραδιοσυχνότητες από 2.4 εώς 2.4835 GHz, που ανήκουν σε παγκοσμίως συμβατή ISM μπάντα
- Ρυθμούς μετάδοσης δεδομένων εως και 250 Kbps

- Ενσωματωμένη κεραία στην πλακέτα
- 16-bit μικροεπεξεργαστή TI MPS430 που τρέχει στα 8 MHz και παρέχει 10 KB μνήμης RAM
- Χαμηλή κατανάλωση ενέργειας
- Εξωτερική μνήμη flash του 1 MB για την αποθήκευση δεδομένων
- Προγραμματισμός και συλλογή δεδομένων μέσω θύρας USB
- Αισθητήρες φωτός, υπέρυθρης ακτινοβολίας, υγρασίας και θερμοκρασίας
- Εκτέλεση των λειτουργικών συστημάτων TinyOS και Contiki

Κεφάλαιο 6

Κρυπτογραφικοί Αλγόριθμοι

Λόγω της συμπληρωματικότηταςτων πλεονεκτημάτων των αλγορίθμων συμμετρικής και ασυμμετρικής κρυπτογραφίας η βιβλιοθήκη που υλοποιήσαμε παρέχει συμμετρικούς αλλά και ασυμμετρικούς αλγόριθμους κρυπτογράφησης. Όπως αναφέραμε σε προηγούμενα κεφάλαια, για την ανάπτυξη ασυμμετρικής κρυπτογραφίας σε συσκευές περιορισμένων πόρων προτείνεται η χρήση ελλειπτικών καμπυλών. Αυτό συμβαίνει διότι οι ελλειπτικές καμπύλες προσφέρουν το ίδιο επίπεδο ασφάλειας με άλλα κρυπτοσυστήματα (π.χ. RSA) με χρήση μικρότερου μεγέθους κλειδιών. Έτσι, οι αλγόριθμοι που παρέχει η βιβλιοθήκη μας είναι οι εξής:

- Ο αλγόριθμος συμμετρικής κρυπτογράφησης AES
- Ο αλγόριθμος κατακερματισμού SHA-1
- Το σχήμα συμφωνίας κλειδιών Elliptic Curve Diffie Hellman
- Ο αλγόριθμος κρυπτογράφησης Elliptic Curve Integrated Encryption Scheme
- Ο αλγόριθμος ψηφιακής υπογραφής Elliptic Curve Digital Signature Algorithm

Στη συνέχεια παρουσιάζουμε αναλυτικά τους αλγόριθμους που παρέχει η κρυπτογραφική μας βιβλιοθήκη.

6.1 Ο Αλγόριθμος Συμμετρικής Κρυπτογράφησης AES

Ο αλγόριθμος κρυπτογράφησης AES (Advanced Encryption Standard) αποτελεί ένα πρότυπο κρυπτογράφησης συμμετρικού κλειδιού. Το πρότυπο αυτό προτάθηκε από την Αμερικανική Κυβέρνηση και αποτελείται από τρία block κρυπτογραφήματα (ciphers) : AES-128, AES-192, AES-256 τα οποία υιοθετήθηκαν από μια συλλογή block κρυπτογραφημάτων που δημοσιεύτηκαν ως Rijndael. Καθένα από τα κρυπτογραφήματα έχει μέγεθος block 128-bits, ενώ το μέγεθος των κλειδιών που

χρησιμοποιούνται είναι 128-bits, 192-bits και 256-bits αντίστοιχα. Ο αλγόριθμος AES έχει αναλυθεί εκτενώς και πλέον χρησιμοποιείται παγκοσμίως έχοντας αντικαταστήσει τον προκάτοχό του DES (Data Encryption Standard).

Ο διεθνής οργανισμός NIST (National Institude of Standards and Technology) μετά από ένα διαγωνισμό ο οποίος κράτησε 5 χρόνια και στον οποίο συμμετείχαν 15 σχεδιασμοί, ανακοίνωσε το πρότυπο AES. Το πρότυπο AES αποτελεί το πρώτο κρυπτογράφημα το οποίο είναι δημοσίως προσβάσιμο και έχει γίνει αποδεκτό από τον οργανισμό NSA (National Security Agency). Το κρυπτογράφημα Rijndael αναπτύχθηκε από δύο Βέλγους κρυπτογράφους, τους Joan Daemen και Vincent Rijmen.

6.1.1 Περιγραφή του Αλγορίθμου

Ο αλγόριθμος AES βασίζεται σε μια σχεδιαστική αρχή γνωστή ώς δίκτυο αντικατάστασης-μετάθεσης. Σε αντίθεση με τον προκάτοχό του DES, ο AES δε χρησιμοποιεί ένα δίκτυο Feistel. Όπως προαναφέρθηκε, ο AES χρησιμοποιεί σταθερό μέγεθος μπλοκ 128-bits ενώ το μέγεθος κλειδιού μπορεί να είναι 128, 192 ή 256-bits. Εκτελείται πάνω σε ένα 4x4 πίνακα ο οποίος ονομάζεται State και οι αριθμητικές πράξεις γίνονται σε ένα ειδικό πεπερασμένο πεδίο. Ο AES λειτουργεί ως ένας αριθμός επαναλήψεων γύρων μετασχηματισμού οι οποίοι μετατρέπουν την είσοδο στην κρυπτογραφημένη έξοδο. Κάθε γύρος αποτελείται από επεξεργαστικά βήματα, συμπεριλαμβανομένου ενός το οποίο περιέχει το κλειδί κρυπτογράφησης. Για την αποκρυπτογράφηση ενός κρυπτογραφημένου μπλοκ, ενα σύνολο από αντίστροφους γύρους εφαρμόζονται χρησιμοποιώντας το ίδιο κλειδί με αυτό της κρυπτογράφησης.

Εν συνεχεία, παρουσιάζεται μια υψηλού επιπέδου περιγραφή των βημάτων του αλγορίθμου AES:

- **Key Expansion** : τα κλειδιά του γύρου δημιουργούνται από το κλειδί κρυπτογράφησης με βάση το πρόγραμμα παραγωγής κλειδιών Rijndael
- **Initial Round**

  AddRoundKey: κάθε byte του πίνακα State συνδυάζεται με το κλειδί του γύρου χρησιμοποιώντας την πράξη bitwise XOR
- **Rounds**

  SubBytes : ένα βήμα μη γραμμικής αντικατάστασης, όπου κάθε byte εναλλάσεται με κάποιο άλλο βάσει ενός πίνακα αναζήτησης (lookup table)

  ShiftRows : ένα βήμα μετάθεσης, όπου κάθε γραμμή του πίνακα State ολισθαίνεται κυκλικά κατά ένα συγκεκριμένο αριθμό βημάτων

  MixColumns : μια πράξη μίξης πάνω στις στήλες του πίνακα State, η οποία συνδυάζει τα 4 bytes σε κάθε στήλη

  AddRoundKey
- **Final Round**

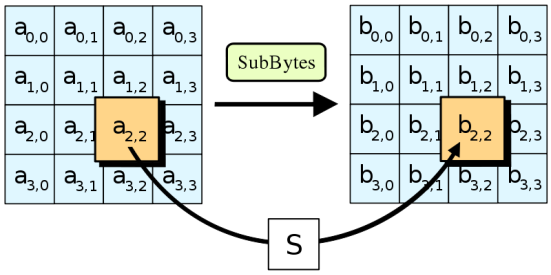
  SubBytes : όπως προηγουμένως

  ShiftRows : όπως προηγουμένως

  AddRoundKey : όπως προηγουμένως

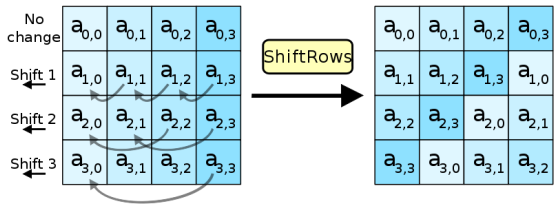
Ακολούθως παρουσιάζονται πιο αναλυτικά τα επιμέρους βήματα του αλγορίθμου AES.

Το Βήμα SubBytes : Στο βήμα αυτό, κάθε byte στον πίνακα ενημερώνεται χρησιμοποιώντας ένα κουτί αντικατάστασης (substitution box) 8-bit, το οποίο ονομάζεται Rijndael S-Box. Η πράξη αυτή εξασφαλίζει τη μη γραμμικότητα στο κρυπτογράφημα. Το S-box που χρησιμοποιείται παράγεται από τους πολλαπλασιαστικούς αντίστροφους του πεδίου $GF(2^8)$, το οποίο είναι γνωστό για τις μη γραμμικές ιδιότητες. Για την αποφυγή επιθέσεων βασιζόμενες σε απλές αλγεβρικές ιδιότητες, το S-box κατασκευάζεται συνδυάζοντας τη συνάρτηση αντιστροφής με έναν αντιστρεφόμενο συνδεόμενο μετασχηματισμό (affine transformation). Επιπλέον, το S-box είναι διαλεγμένο έτσι ώστε να αποφεύγει σταθερά σημεία καθώς και αντίθετα σταθερά σημεία. Το Σχήμα 6.1 περιγράφει το βήμα αυτό.



Σχήμα 6.1: Στο Βήμα SubBytes, Κάθε Byte του State Αντικαθίσταται με την Εγγραφή του σε Ένα 8-bit Σταθερό Πίνακα Αναζήτησης ($b_{ij} = S(a_{ij})$).

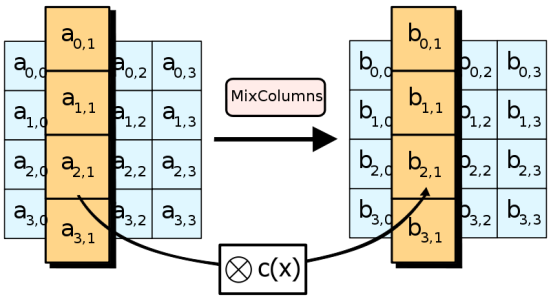
Το Βήμα ShiftRows : Το βήμα αυτό εκτελείται στις γραμμές του πίνακα State. Τα bytes κάθε γραμμής ολισθαίνονται κυκλικά με βάση ένα γνωστό αριθμό θέσεων. Για το πρότυπο AES η πρώτη γραμμή παραμένει ώς έχει. Κάθε byte της δεύτερης γραμμής ολισθαίνεται κατά μία θέση στα αριστερά. Παρομοίως, η τρίτη και η τέταρτη γραμμή ολισθαίνονται κατά δύο και τρεις θέσεις αντιστοίχως. Για τα μπλοκ μεγέθους 128 και 192 bits, η πατέντα ολίσθησης είναι η ίδια. Για τα μπλοκ 256-bit, η πρώτη γραμμή παραμένει απαράλλαχτη ενώ η ολίσθηση για την δεύτερη, τρίτη και τέταρτη γραμμή είναι 1, 3 και 4 bytes αντίστοιχα. Το Σχήμα 6.2 έχει μια περιγραφή του βήματος. Οι ποσότητες A_{ij} προέρχονται από το κρυπτογραφημένο μπλοκ ενώ οι ποσότητες B_{ij} από το κλειδί κρυπτογράφησης.



Σχήμα 6.2: Στο Βήμα ShiftRows, τα Bytes της Κάθε Γραμμής του Πίνακα State Ολισθαίνονται Κυκλικά Προς τα Αριστερά. Ο Αριθμός των Θέσεων Ολίσθησης Διαφέρει για Κάθε Γραμμή.

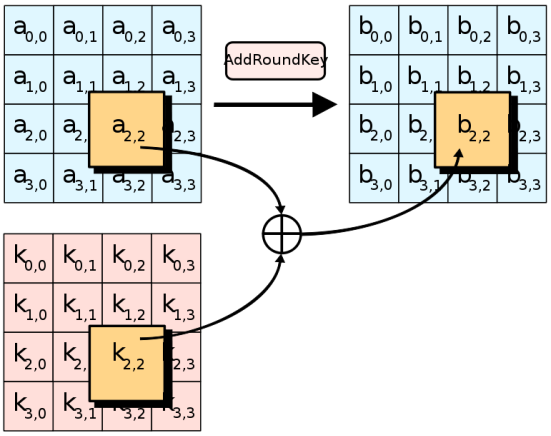
Το Βήμα MixColumns : Στο βήμα αυτό, τα τέσσερα bytes της κάθε

στήλης του πίνακα State συνδυάζονται χρησιμοποιώντας ένα αντιστρεφόμενο γραμμικό μετασχηματισμό. Η συνάρτηση MixColumns δέχεται σαν είσοδο τέσσερα bytes και παράγει στην έξοδο τέσσερα bytes, έτσι ώστε κάθε byte της εισόδου να επηρεάζει και τα τέσσερα bytes της εξόδου. Σε συνδυασμό με το βήμα ShiftRows, το βήμα MixColumns προσφέρει διάχυση (diffusion) στο κρυπτογράφημα. Κατά το βήμα MixColumns, κάθε στήλη πολλαπλασιάζεται με ένα γνωστό πίνακα. Γενικά, κάθε στήλη αντιμετωπίζεται σαν ένα πολυώνυμο πάνω από το πεδίο $GF(2^8)$ το οποίο πολλαπλασιάζεται modulo $x^4 + 1$ με ένα σταθερό πολυώνυμο $c(x) = 0x03 \cdot x^3 + x^2 + x + 0x02$. Οι συντελεστές αναπαρίστανται σε δεκαεξαδική μορφή ισοδύναμη με τη δυαδική αναπαράσταση των πολυωνύμων από το GF(2)[x]. Το Σχήμα 6.3 παρουσιάζει τη διαδικασία του βήματος αυτού.



Σχήμα 6.3: Στο Βήμα MixColumns, Κάθε Στήλη του Πίνακα State Πολλαπλασιάζεται με Ένα Σταθερό Πολυώνυμο $c(x)$.

Το Βήμα AddRounKey : Στο βήμα αυτό, το υποκλειδί του γύρου συνδυάζεται με τον πίνακα State. Για κάθε γύρο κατασκευάζεται ένα υποκλειδί από το βασικό κλειδί χρησιμοποιώντας το πρόγραμμα παραγωγής κλειδιών Rijndael. Κάθε υποκλειδί έχει ίδιο μέγεθος με τον πίνακα State. Το υποκλειδί προστίθεται συνδυάζοντας κάθε byte του πίνακα State με το αντίστοιχο byte του υποκλειδιού χρησιμοποιώντας την πράξη XOR. Το βήμα φαίνεται στο Σχήμα 6.4.



Σχήμα 6.4: Στο Βήμα AddRoundKey, Κάθε Byte του Πίνακα State Συνδυάζεται με Ένα Byte του Υποκλειδιού του Γύρου Εκτελώντας την Πράξη XOR.

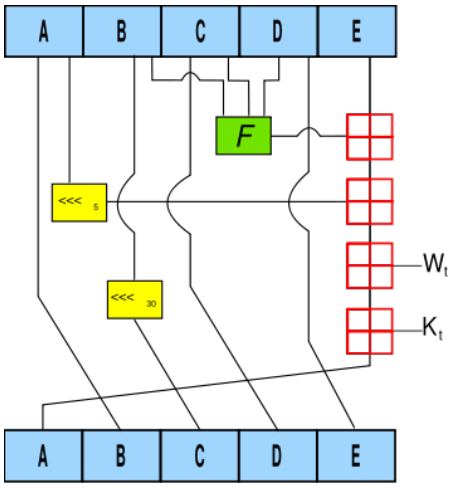
Ο AES ολοκληρώνεται σε 10 γύρους για 128-bit κλειδιά, σε 12 γύρους για 192-bit κλειδιά και σε 14 γύρους για 256-bit κλειδιά.

6.2 Ο Αλγόριθμος Κατακερματισμού SHA-1

Στην κρυπτογραφία, το SHA-1 (Secure Hash Algorithm) [11] αναφέρεται σε μια κρυπτογραφική συνάρτηση κατακερματισμού η οποία σχεδιάστηκε από την οργάνωση NSA (National Security Agency) και δημοσιεύτηκε από το Ινστιτούτο Προτύπων και Τεχνολογίας (NIST). Ενώ μέχρι στιγμής έχουν παρουσιαστεί τρεις αλγόριθμοι προτύπου SHA, ο SHA-1 αποτελεί την πιο συχνά χρησιμοποιημένη συνάρτηση κατακερματισμού. Έχει βρει εφαρμογή τόσο σε εφαρμογές ασφάλειας όσο και σε πρωτόκολλα δικτύων. Κατά την τρέχουσα περίοδο βρίσκεται σε εξέλιξη ένας διαγωνισμός NIST, ο οποίος θα ανακηρύξει μέχρι τα τέλη του 2012 ένα νέο πρότυπο κρυπτογραφικής συνάρτησης κατακερματισμού με όνομα SHA-3.

6.2.1 Περιγραφή του Αλγορίθμου

Η συνάρτηση κατακερματισμού SHA-1 παράγει στην έξοδο μια 160-bit σύνοψη ενός μηνύματος, βασιζόμενη σε αρχές που είχαν προταθεί από τον Ronald Rivest κατά το σχεδιασμό των συναρτήσεων MD-4 και MD-5. Οι αρχικές προδιαγραφές του αλγορίθμου δημοσιεύτηκαν το 1993 ως το πρότυπο Secure Hash Standard, FIPS PUB 180 από τον οργανισμό NIST. Η έκδοση αυτή αναφέρεται συχνά ώς ο αλγόριθμος SHA-0. Ο SHA-0 αποσύρθηκε από την NSA και τη θέση του πήρε η νέα εκδοσή του FIPS PUB 180-1 το 1995. Η έκδοση αυτή έγινε γνωστή ως SHA-1. Ο SHA-1 διαφέρει από το SHA-0 σε μια περιστροφή κατά bit στο πρόγραμμα μηνύματος της συνάρτησης συμπίεσης. Σύμφωνα με την NSA η αλλαγή αυτή έγινε για τη διόρθωση ενός ελαττώματος το οποίο μείωνε την κρυπτογραφική ασφάλεια του αρχικού αλγορίθμου. Ο αλγόριθμος SHA-1 χρειάζεται 80 γύρους για την ολοκληρωσή του. Η πιο γνωστή επίθεση στον αλγόριθμο παρουσιάστηκε το 2008 από τον Manuel. Η επίθεση αυτή σπάει τη συνάρτηση κατακερματισμού, καθώς μπορεί να παράγει συγκρούσεις συνόψεων με πολυπλοκότητα 2^{51} αριθμητικών πράξεων. Η αναλύτική περιγραφή του αλγορίθμου SHA-1 φαίνεται στο Σχήμα 6.5. Στο σχήμα αυτό, τα A, B, C, D, E αποτελούν 32-bit λέξεις του πίνακα State και η F αποτελεί μια μη γραμμική συνάρτηση. Το σύμβολο $\lll_n$ αναπαριστά την αριστερή περιστροφή κατά n θέσεις και το σύμβολο $\boxplus$ αναπαριστά την πρόσθεση με υπόλοιπο 2^{32}. Το W_t αποτελεί την εκτεταμένη λέξη του μηνύματος στο γύρο t και το K_t αποτελεί τη σταθερά γύρου που χρησιμοποιείται κατά το γύρο t.



Σχήμα 6.5: Ένας Γύρος Εκτέλεσης της Συνάρτησης Συμπίεσης του Αλγορίθμου SHA-1.

6.3 Κρυπτογραφικοί Αλγόριθμοι με Βάση τις Ελλειπτικές Καμπύλες

6.3.1 Παραγωγή Ζευγαριού Κλειδιών με Ελλειπτικές Καμπύλες

Όλα τα σχήματα δημοσίου κλειδιού με βάση τις ελλειπτικές καμπύλες χρησιμοποιούν ζευγάρια κλειδιών τα οποία ονομάζονται ζευγάρια κλειδιών ελλειπτικών καμπυλών. Δεδομένων των παραμέτρων μιας ελλειπτικής καμπύλης, ένα ζευγάρι κλειδιών (d, Q) αποτελείται από ένα μυστικό κλειδί d το οποίο είναι ένας ακέραιος στο διάστημα $[1, n-1]$ και ένα δημόσιο κλειδί $Q = (x_Q, y_Q)$ το οποίο είναι ένα σημείο πάνω στην ορισμένη ελλειπτική καμπύλη και δίνεται από τη σχέση $Q = d \cdot G$, όπου G ένα σημείο γεννήτορας της καμπύλης.

Διαδικασία Παραγωγής Ζευγαριού Κλειδιών

Μια οντότητα U, παράγει ένα ζευγάρι κλειδιών με ελλειπτικές καμπύλες ως εξής:

Είσοδος : Οι παράμετροι της ελλειπτικής καμπύλης που θα χρησιμοποιηθεί.

Έξοδος : Ένα ζευγάρι κλειδιών (d, Q) με βάση τις παραμέτρους που έχουν οριστεί.

- Η οντότητα U διαλέγει τυχαία έναν ακέραιο d στο διάστημα $[1, n-1]$.
- Υπολογίζει το σημείο της καμπύλης $Q = d \cdot G$.
- Παράγει ως έξοδο το ζευγάρι μυστικού και δημοσίου κλειδιού (d, Q).

6.3.2 Το Σχήμα Συμφωνίας Κλειδιών ECDH

Το σχήμα ECDH (Elliptic Curve Diffie Hellman) αποτελεί ένα σχήμα συμφωνίας κλειδιών με βάση τις ελλειπτικές καμπύλες. Ο αλγόριθμος αυτός λαμβάνει χώρα

μεταξύ δύο οντοτήτων τις οποίες για συντομία ονομάζουμε U και V. Οι οντότητες αυτές, έχουν προσυμφωνήσει στις παραμέτρους της ελλειπτικής καμπύλης που θα χρησιμοποιθεί κατά τη διάρκεια του σχήματος καθώς και σε μιά συνάρτηση παραγωγής κλειδιού (KDF (Key Derivation Function).

Προετοιμασία Κλειδιών

- Η οντότητα U παράγει ένα ζευγάρι κλειδιών (d_U, Q_U) με βάση τις παραμέτρους της ελλειπτικής καμπύλης που έχει προσυμφωνηθεί.
- Η οντότητα V παράγει ένα ζευγάρι κλειδιών (d_V, Q_V) με βάση τις παραμέτρους της ελλειπτικής καμπύλης που έχει προσυμφωνηθεί.
- Οι οντότητες U και V ανταλλάσσουν τα δημόσια κλειδιά τους, αντίστοιχα Q_U και Q_V.

Διαδικασία Συμφωνίας Κλειδιού

Για τη συμφωνία σε ένα κοινό διαμοιραζόμενο κλειδί, οι δύο οντότητες πολλαπλασιάζουν το μυστικό τους κλειδί με το δημόσιο κλειδί της άλλης οντότητας. Χάριν συντομίας, περιγράφονται μόνο οι ενέργειες της οντότητας U, καθώς οι ενέργειες της οντότητας V είναι αντίστοιχες.

Είσοδος: Ένας ακέραιος $keydatalen$ ο οποίος είναι το μήκος (σε αριθμό οκτάδων (bytes)) του κοινού κλειδιού που θα δημιουργηθεί και το δημόσιο κλειδί Q_V της οντότητας V.

Έξοδος: Ένα αλφαριθμητικό K το οποίο αποτελεί το διαμοιραζόμενο κλειδί μήκους $keydatalen$, ή μήνυμα λάθους.

- Πολλαπλασιάζει το μυστικό της κλειδί d_U με το δημόσιο κλειδί Q_V της οντότητας V και υπολογίζει το σημείο $P = d_U \cdot Q_V = (x_P, y_P)$ της ελλειπτικής καμπύλης.
- Ελέγχει εαν το σημείο P είναι ίσο με το σημείο απειρίας. Εάν ναι, βγάζει μήνυμα λάθους στην έξοδο αλλιώς χρησιμοποιεί τη συντεταγμένη $z = x_P$ σαν το μυστικό στοιχείο του πεδίου της ελλειπτικής καμπύλης.
- Μετατρέπει το μυστικό κλειδί z σε ένα αλφαριθμητικό Z, αποτελούμενο από οκτάδες (bytes).
- Χρησιμοποιεί τη συνάρτηση παραγωγής κλειδιού KDF για να παράγει ένα κλειδί K με μήκος $keydatalen$.
- Παράγει ως έξοδο το αλφαριθμητικό K το οποίο αποτελέι το κοινό διαμοιραζόμενο κλειδί.

6.3.3 Ο Αλγόριθμος Κρυπτογράφησης ECIES

Ο αλγόριθμος ECIES (Elliptic Curve Integrated Encryption Scheme) αποτελεί ένα σχήμα κρυπτογράφησης δημοσίου κλειδιού με βάση τις ελλειπτικές καμπύλες. Ο αλγόριθμος αυτός λαμβάνει χώρα μεταξύ δύο οντοτήτων, U και V. Οι οντότητες αυτές, έχουν προσυμφωνήσει στις παραμέτρους της ελλειπτικής καμπύλης που θα χρησιμοποιθεί κατά τη διάρκεια του αλγορίθμου, σε μιά συνάρτηση παραγωγής κλειδιού (KDF (Key Derivation Function) καθώς και σε ένα MAC (Message Authentication Code) σχήμα.

Προετοιμασία Κλειδιών

Η οντότητα V παράγει ένα ζευγάρι κλειδιών (d_V, Q_V) με βάση της παραμέτρους της ελλειπτικής καμπύλης που έχει προσυμφωνηθεί. Το d_V αποτελεί το μυστικό της κλειδί ενώ το Q_V το δημόσιο κλειδί της. Το δημόσιο κλειδί Q_V της οντότητας V γίνεται γνωστό στην οντότητα U με κάποιο τρόπο αυθεντικοποίησης.

Διαδικασία Κρυπτογράφησης

Για την κρυπτογράφηση ενός μηνύματος M, η οντότητα U χρησιμοποιεί το δημόσιο κλειδί Q_V της οντότητας V και σε συνδυασμό με το μυστικό της κλειδί, παράγει ένα κοινό διαμοιραζόμενο κλειδί. Εν συνεχεία, με χρήση ενός σχήματος συμμετρικής κρυπτογράφησης κωδικοποιεί το μήνυμα M στο κρυπτογραφημένο μήνυμα C.

Είσοδος: Ένα αλφαριθμητικό M το οποίο αποτελεί το μήνυμα για κρυπτογράφηση και το δημόσιο κλειδί Q_V της οντότητας V.

Έξοδος: Ένα αλφαριθμητικό C το οποίο αποτελεί το κρυπτογραφημένο μήνυμα, ή μήνυμα λάθους.

- Παράγει ένα προσωρινό ζευγάρι κλειδιών (k_U, R_U), όπου $k_U \in F_q$ και $R_U = (x_R, y_R)$ με βάση τις παραμέτρους της ελλειπτικής καμπύλης που έχει προσυμφωνηθεί. Το k_U αποτελεί το μυστικό κλειδί ενώ το R_U το δημόσιο κλειδί της οντότητας U.
- Πολλαπλασιάζοντας το μυστικό της κλειδί με το δημόσιο κλειδί Q_V της οντότητας V δημιουργεί το διαμοιραζόμενο μυστικό κλειδί $z \in F_q$.
- Μετατρέπει το μυστικό κλειδί z σε ένα αλφαριθμητικό Z, αποτελούμενο από οκτάδες (bytes).
- Χρησιμοποιεί τη συνάρτηση παραγωγής κλειδιού KDF για να παράγει ένα κλειδί K με μήκος $enckeylen + mackeylen$. Τα αριστερότερα enckeylen bytes αποτελούν το κλειδί κρυπτογράφησης EK και τα υπόλοιπα mackeylen bytes αποτελούν το MAC (Message Authentication Code) κλειδί.
- Χρησιμοποιεί τη πράξη XOR μεταξύ του μηνύματος M και του κλειδίου EK για την μετατροπή του μηνύματος M στο μήνυμα EM.
- Χρησιμοποιεί το MAC κλειδί με το προσυμφωνημένο MAC σχήμα για να παράξει μια ετικέτα (tag) D για το κρυπτογραφημένο μήνυμα EM.
- Παράγει ώς έξοδο την τριπλέτα $C = (R||EM||D)$.

Διαδικασία Από-Κρυπτογράφησης

Η οντότητα V χρησιμοποιεί το μυστικό της κλειδί d_V για να παράγει το κοινό διαμοιραζόμενο κλειδί. Εν συνεχεία, με χρήση του προσυμφωνημένου σχήματος συμμετρικής κρυπτογράφησης αποκωδικοποιεί το κρυπτογραφημένο μήνυμα C στο αρχικό μήνυμα M.

Είσοδος: Η τριπλέτα $C = (R||EM||D)$ η οποία παράχθηκε κατά τη διαδικασία κρυπτογράφησης και το μυστικό κλειδί d_V της οντότητας V.

Έξοδος: Ένα αλφαριθμητικό M το οποίο αποτελεί το αρχικό μήνυμα, ή μήνυμα λάθους.

- Αναλύει την τριπλέτα $C = (R||EM||D)$ και εξάγει το δημόσιο κλειδί R_U της οντότητας U.
- Πολλαπλασιάζοντας το μυστικό της κλειδί με το δημόσιο κλειδί της οντότητας U δημιουργεί το διαμοιραζόμενο μυστικό κλειδί $z \in F_q$.
- Μετατρέπει το μυστικό κλειδί z σε ένα αλφαριθμητικό Z, αποτελούμενο από οκτάδες (bytes).
- Χρησιμοποιεί τη συνάρτηση παραγωγής κλειδιού KDF (Key Derivation Function) για να παράγει ένα κλειδί K με μήκος $enckeylen + mackeylen$. Τα αριστερότερα enckeylen bytes αποτελούν το κλειδί από-κρυπτογράφησης EK και τα υπόλοιπα mackeylen bytes αποτελούν το MAC (Message Authentication Code) κλειδί.
- Χρησιμοποιώντας το MAC κλειδί και το προσυμφωνημένο MAC (Sha1-Mac) σχήμα, ελέγχει εαν η ετικέτα D είναι έγκυρη για το κρυπτογραφημένο μήνυμα EM. Εάν δεν είναι έγκυρη ο αλγόριθμος βγάζει μήνυμα λάθους.
- Χρησιμοποιεί τη πράξη XOR μεταξύ του μηνύματος M και του κλειδίου EK για την μετατροπή του μηνύματος EM στο αρχικό μήνυμα M.
- Παράγει ώς έξοδο το μήνυμα M.

6.3.4 Ο Αλγόριθμος Ψηφιακής Υπογραφής ECDSA

Ο αλγόριθμος ECDSA (Elliptic Curve Digital Signature Algorithm) αποτελεί ένα σχήμα ψηφιακής υπογραφής με βάση τις ελλειπτικές καμπύλες. Ο αλγόριθμος αυτός λαμβάνει χώρα μεταξύ δύο οντοτήτων, U και V. Οι οντότητες αυτές, έχουν προσυμφωνήσει στις παραμέτρους της ελλειπτικής καμπύλης που θα χρησιμοποιθεί κατά τη διάρκεια του αλγορίθμου και σε μιά συνάρτηση κατακερματισμού HASH() η οποία παράγει αλφαριθμητικά μήκους (σε bytes hashlen).

Προετοιμασία Κλειδιών

Η οντότητα U παράγει ένα ζευγάρι κλειδιών (d_U, Q_U) με βάση τις παραμέτρους της ελλειπτικής καμπύλης που έχει προσυμφωνηθεί. Το Q_U, το οποίο αποτελέι το δημόσιο κλειδί της γίνεται γνωστό στην οντότητα V με κάποιο τρόπο αυθεντικοποίησης.

Διαδικασία Υπογραφής

Η οντότητα U παράγει μια υπογραφή S στο μήνυμα M χρησιμοποιώντας το μυστικό της κλειδί d_U και ένα προσωρινό ζευγάρι κλειδιών (k, R) που παράγει.

Έισοδος : Ένα αλφαριθμητικό M το οποίο αποτελεί το μήνυμα που θα υπογραφεί και το μυστικό κλειδί d_U.

Έξοδος : Δύο ακέραιοι r, s οι οποίοι αποτελούν την υπογραφή $S = (r, s)$ στο μήνυμα M, ή μήνυμα λάθους.

- Παράγει ένα προσωρινό ζευγάρι κλειδιών (k, R), όπου $k \in F_q$ και $R = (x_R, y_R)$ βάσει των παραμέτρων της ελλειπτικής καμπύλης που έχει προσυμφωνηθεί.

- Υπολογίζει την ποσότητα $r = x_R \, (mod \, n)$. Εάν $r = 0$, ο αλγόριθμος πηγαίνει πάλι στο προηγούμενο βήμα.
- Χρησιμοποιώντας τη συνάρτηση κατακερματισμού, υπολογίζει το αλφαριθμητικό $H = HASH(M)$.
- Μετατρέπει το αλφαριθμητικό H σε έναν ακέραιο e.
- Υπολογίζει την ποσότητα $s = k^{-1} \cdot (e + r \cdot d_U) \, (mod \, n)$. Εαν $s = 0$, ο αλγόριθμος επιστρέφει στο πρώτο βημα.
- Παράγει ως έξοδο την υπογραφή $S = (r, s)$.

Διαδικασία Επαλήθευσης Υπογραφής

Η οντότητα V επαληθεύει την υπογραφή S πάνω στο μήνυμα M χρησιμοποιώντας το δημόσιο κλειδί Q_U της οντότητας U.

Είσοδος : Το αλφαριθμητικό M το οποίο αποτελεί το υπογεγραμένο μήνυμα και η υπογραφή $S = (r, s)$ πάνω στο M.

Έξοδος : Μια ένδειξη για το αν η υπογραφή είναι έγκυρη ή όχι.

- Ελέγχει εάν οι ποσότητες r, s είναι και οι δύο ακέραιοι αριθμοί στο διάστημα $[1, n-1]$. Εάν όχι, παράγει στην έξοδο ένδειξη άκυρης υπογραφής.
- Χρησιμοποιώντας τη συνάρτηση κατακερματισμού, υπολογίζει το αλφαριθμητικό $H = HASH(M)$.
- Μετατρέπει το αλφαριθμητικό H σε έναν ακέραιο e.
- Υπολογίζει τις ποσότητες $u_1 = e \cdot s^{-1} \, (mod \, n)$ και $u_2 = r \cdot s^{-1} \, (mod \, n)$.
- Υπολογίζει το σημείο $R = (x_R, y_R) = u_1 \cdot G + u_2 \cdot Q_U$. Εάν $R = O$, παράγει στην έξοδο ένδειξη άκυρης υπογραφής.
- Υπολογίζει την ποσότητα $u = x_R \, (mod \, n)$.
- Συγκρίνει τις ποσότητες u και r. Εάν $u = r$ παράγει στην έξοδο ένδειξη έγκυρης υπογραφής αλλιώς ένδειξη άκυρης υπογραφής.

Κεφάλαιο 7

Πειραματικά Αποτελέσματα

Για να εξετάσουμε την απόδοση της υλοποίησής μας καθώς και για να αποδείξουμε τη γενικότητα της, εκτελέσαμε τους υλοποιημένους κρυπτογραφικούς αλγορίθμους σε πραγματικές συσκευές των σειρών iSense και TelosB που εκτελούν διαφορετικά λειτουργικά συστήματα (iSense OS και Contiki Sky αντίστοιχα) και μετρήσαμε τους χρόνους εκτέλεσης των βασικών τους πράξεων σαν το μέσο όρο 100 επαναλήψεων.

Στη συνέχεια, ακολουθώντας μια θεωρητική προσέγγιση υπολογίσαμε την ενέργεια που απαιτείται για την εκτέλεσή τους. Η προσέγγιση αυτή, βασίζεται στις τιμές κατανάλωσης ρεύματος που μας παρέχει το data sheet του κάθε μικροεπεξεργαστή. Παραδείγματος χάριν, στο data sheet του μικροεπεξεργαστή JN5139 παρατηρούμε ότι η κατανάλωση ρεύματος της συσκευής όταν ο μικροεπεξεργαστής λειτουργεί στο μέγιστο φόρτο είναι 12.7 mA, δεδομένης τάσης εισόδου 3V και θερμοκρασίας $25°C$. Η τάση εισόδου 3V αποτελεί μια λογική επιλογή αν λάβουμε υπόψιν ότι δυο μπαταρίες AA παρέχουν τόση τάση. Αντίστοιχα, ο μικροελεγκτής MSP430 καταναλώνει περίπου 1.8 mA όταν εκτελεί λειτουργίες. Έτσι, θεωρώντας την Εξίσωση

$$E = V \cdot I \cdot t \tag{7.1}$$

όπου V είναι η τάση, I είναι το ρεύμα και t ο χρόνος εκτέλεσης κάθε πράξης, μπορούμε να υπολογίσουμε την κατανάλωση ενέργειας των βασικών πράξεων κάθε αλγορίθμου.

7.1 Ο Αλγόριθμος Κρυπτογράφησης AES

Ο Πίνακας 7.1, παρουσιάζει τους χρόνους εκτέλεσης και την κατανάλωση ενέργειας των διαδικασιών κρυπτογράφησης και αποκρυπτογράφησης του αλγορίθμου AES όταν εκτελείται στις συσκευές iSense και TelosB, σε συνάρτηση με το μέγεθος κλειδιού που χρησιμοποιείται. Όπως αναφέρθηκε στην περιγραφή του αλγορίθμου, το μπλοκ που μπορεί να κρυπτογραφήσει ο αλγόριθμος AES είναι 128 bit ενώ το μέγεθος του κλειδιού κρυπτογράφησης μπορεί να είναι 128, 192 ή 256 bit. Εν συνεχεία, στον Πίνακα 7.2 φαίνεται το μέγεθος του μεταφρασμένου κώδικα για τις διαδικασίες κρυπτογράφησης και αποκρυπτογράφησης του αλγορίθμου.

Μέγεθος Κλειδιού	Πράξη	Χρόνος Εκτέλεσης		Κατανάλωση Ενέργειας	
		iSense	TelosB	iSense	TelosB
128 bit	Κρυπτογράφηση	29.92 msec	99.85 msec	1.139 mJ	0.539 mJ
	Αποκρυπτογράφηση	34.31 msec	118.92 msec	1.307 mJ	0.642 mJ
192 bit	Κρυπτογράφηση	36.22 msec	119.39 msec	1.379 mJ	0.644 mJ
	Αποκρυπτογράφηση	40.63 msec	142.85 msec	1.548 mJ	0.771 mJ
256 bit	Κρυπτογράφηση	41.92 msec	139.29 msec	1.597 mJ	0.752 mJ
	Αποκρυπτογράφηση	49.3 msec	167.74 msec	1.878 mJ	0.905 mJ

Πίνακας 7.1: Χρόνος Εκτέλεσης και Κατανάλωση Ενέργειας του Αλγορίθμου AES Συναρτήσει του Μεγέθους Κλειδιού στις Συσκευές iSense και TelosB.

Μέγεθος Κλειδιού	Πράξη	Μέγεθος Κώδικα	
		iSense	TelosB
128 bit	Κρυπτογράφηση	2644 bytes	4466 bytes
	Αποκρυπτογράφηση	3280 bytes	4600 bytes
192 bit	Κρυπτογράφηση	2648 bytes	4474 bytes
	Αποκρυπτογράφηση	3284 bytes	4608 bytes
256 bit	Κρυπτογράφηση	2656 bytes	4482 bytes
	Αποκρυπτογράφηση	3292 bytes	4616 bytes

Πίνακας 7.2: Μέγεθος Μεταφρασμένου Κώδικα των Διαδικασιών Κρυπτογράφησης και Αποκρυπτογράφησης του Αλγορίθμου AES Συναρτήσει του Μεγέθους Κλειδιού στις Συσκευές iSense και TelosB.

Παρατηρώντας τον Πίνακα 7.1, συμπεραίνουμε οτι η συσκευή iSense εκτελεί γρηγορότερα τον αλγόριθμο AES σε σχέση με τη συσκευή TelosB. Αυτό είναι λογικό καθώς ο μικροεπεξεργαστής που χρησιμοποιεί (JN5139) λειτουργεί σε μεγαλύτερη συχνότητα σε σχέση με τον MSP430 της συσκευής TelosB. Παρά το γεγονός αυτό, η συσκευή TelosB είναι πιο οικονομική στο θέμα κατανάλωσης ενέργειας. Επιπλέον, στον Πίνακα 7.2 παρατηρούμε ότι το μέγεθος κώδικα είναι διαφορετικό για κάθε συσκευή. Αυτό είναι λογικό αν σκεφτούμε ότι η κάθε συσκευή τρέχει το δικό της λειτουργικό σύστημα (iSense OS, Contiki Sky) και χρησιμοποιεί διαφορετικό μεταγλωττιστή (ba-elf-g++, mspgcc).

7.2 Ο Αλγόριθμος Κατακερματισμού SHA-1

Στον Πίνακα 7.3 φαίνονται ο χρόνος εκτέλεσης και η κατανάλωση ενέργειας που απαιτούνται για τον κατακερματισμό ενός μπλοκ μεγέθους 160 bit με χρήση του αλγορίθμου SHA-1 στις συσκευές iSense και TelosB. Ακολούθως, στον Πίνακα 7.4 φαίνεται το μέγεθος του μεταφρασμένου κώδικα για τη διαδικασία κατακερματισμού ενός μπλοκ δεδομένων 160 bit.

Πράξη	Χρόνος Εκτέλεσης		Κατανάλωση Ενέργειας	
	iSense	TelosB	iSense	TelosB
Κατακερματισμός (20 bytes)	6 msec	12.1 msec	0.22 mJ	0.06 mJ

Πίνακας 7.3: Χρόνος Εκτέλεσης και Κατανάλωση Ενέργειας του Αλγορίθμου SHA-1 στις Συσκευές iSense και TelosB.

Πράξη	Μέγεθος Κώδικα	
	iSense	TelosB
Κατακερματισμός (20 bytes)	2144 bytes	2766 bytes

Πίνακας 7.4: Μέγεθος Μεταφρασμένου Κώδικα της Διαδικασίας Κατακερματισμού του Αλγορίθμου SHA-1 στις Συσκευές iSense και TelosB.

Στην περίπτωση του αλγορίθμου SHA-1, παρατηρούμε ότι η συσκευή iSense είναι και πιο γρήγορη στην εκτέλεσή του αλλά και πάλι η συσκευή TelosB είναι πιο οικονομική από πλευράς κατανάλωσης ενέργειας λόγω του αργότερου μικροεπεξεργαστή.

7.3 Κρυπτογραφικοί Αλγόριθμοι με Βάση τις Ελλειπτικές Καμπύλες

Όπως αναφέρθηκε προηγουμένως, για την υλοποίηση των βασικών πράξεων ελλειπτικών καμπυλών μεταφέραμε τον κώδικα της εργασίας TinyECC [16] στο προγραμματιστικό περιβάλλον της βιβλιοθήκης Wiselib. Με βάση τον κώδικα αυτό ο χρήστης μπορεί να αρχικοποιήσει μια από τρεις διαφορετικές ελλειπτικές καμπύλες ορισμένες πάνω από πεδία F_p, μεγέθους 128, 160 και 192 bit αντίστοιχα. Οι παράμετροι των ελλειπτικών καμπυλών έχουν παρουσιαστεί από την ερευνητική ομάδα προτύπων Standards for Efficient Cryptography Group της Certicom Research [19].

Ακολουθώντας τον πρότυπο συμβολισμό, οι παράμετροι μιας ελλειπτικής καμπύλης ορισμένης πάνω από πεδίο F_p παρουσιάζονται ως μια εξάδα $T = (p, a, b, G, n, h)$ όπου p είναι ο πρώτος αριθμός που ορίζει το πεδίο F_p, a, b είναι οι αριθμοί που καθορίζουν την ελλειπτική καμπύλη σύμφωνα με την εξίσωση

$$E : y^2 = x^3 + a \cdot x + b \, (mod \, p) \tag{7.2}$$

, $G = (x_G, y_G)$ ένα σημείο γεννήτορας (βάσης) στο $E(F_p)$, n ένας πρώτος αριθμός που υποδεικνύει την τάξη του G και h ο συμπαράγοντας έτσι ώστε $h = \sharp E(F_p)/n$.

Οι παράμετροι της καμπύλης μεγέθους 128 bit δίνονται από την εξάδα $T = (p, a, b, G, n, h)$ όπου

$$p = FFFFFFFDFFFFFFFFFFFFFFFFFFFFFFFF = 2^{128} - 2^{97} - 1,$$

$$a = 0xFFFFFFFDFFFFFFFFFFFFFFFFFFFFFFFC$$

και

$$b = 0xE87579C11079F43DD824993C2CEE5ED3.$$

Το σημείο βάσης G δίνεται από τις συντεταγμένες

$$x_G = 0x161FF7528B899B2D0C28607CA52C5B86 \text{ και}$$

$$y_G = 0xCF5AC8395BAFEB13C02DA292DDED7A83.$$

Τέλος,

$$n = 0xFFFFFFFE0000000075A30D1B9038A115 \text{ και } h = 0x01.$$

Οι παράμετροι της καμπύλης μεγέθους 160 bit δίνονται από την εξάδα $T = (p, a, b, G, n, h)$ όπου

$$p = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7FFFFFFF = 2^{160} - 2^{31} - 1,$$

$$a = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7FFFFFFC$$
και

$$b = 0x1C97BEFC54BD7A8B65ACF89F81D4D4ADC565FA45.$$

Το σημείο βάσης G δίνεται από τις συντεταγμένες

$$x_G = 0x4A96B5688EF573284664698968C38BB913CBFC82 \text{ και}$$

$$y_G = 0x23A628553168947D59DCC912042351377AC5FB32.$$

Τέλος,

$$n = 0x0000000100000000000000000001F4C8F927AED3CA752257 \text{ και } h = 0x01.$$

Για την καμπύλη μεγέθους 192 bit, οι παράμετροι δίνονται από την εξάδα $T = (p, a, b, G, n, h)$ όπου

$$p = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEFFFFEE37 = 2^{192} - 2^{32} - 2^{12} - 2^8 - 2^7 - 2^6 - 2^3 - 1,$$

$$a = 0x000000000000000000000000000000000000000000000000 \text{ και}$$

$$b = 0x000000000000000000000000000000000000000000000003.$$

Το σημείο βάσης G δίνεται από τις συντεταγμένες

$$x_G = 0xDB4FF10EC057E9AE26B07D0280B7F4341DA5D1B1EAE06C7D$$
και

$$y_G = 0x9B2F2F6D9C5628A7844163D015BE86344082AA88D95E2F9D.$$

Τέλος,

$$n = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFE26F2FC170F69466A74DEFD8D$$
και $h = 0x01$.

Καθώς ο βασικός στόχος της βιβλιοθήκης Wiselib, είναι να παραμένει όσο το δυνατόν γενικότερη και να είναι μεταφέρσιμη αποφασίσαμε κατά τη μεταφορά του κώδικα της εργασίας TinyECC να μην χρησιμοποιήσουμε τις διάφορες βελτιστοποιήσεις λογισμικού (προβολικές συντεταγμένες, μέθοδος ολισθαίνοντος παραθύρου, υβριδικό πολλαπλασιασμό κτλ) που παρέχονται. Αυτή η απόφαση λήφθηκε με στόχο την επίτευξη μικρότερου μεγέθους μεταφρασμένου κώδικα έτσι

ώστε οι διάφορες εφαρμογές να μην επιβαρύνονται αρκετά από τις κρυπτογραφικές πράξεις με ελλειπτικές καμπύλες. Η μόνη *παράμετρος* που θεωρήσαμε ότι έχει νόημα να υφίσταται στη βιβλιοθήκη Wiselib, είναι η επιλογή από το χρήστη της ψηφιολέξης που υποστηρίζει ο μικροεπεξεργαστής για τον οποίο μεταφράζεται ο κώδικας. Έτσι, για έναν 32 bit μικροελεγκτή όπως ο JN5139 της συσκευής iSense οι αριθμητικές πράξεις μπορούν να γίνονται ανά ψηφιολέξεις των 32 bit (ή μικρότερες) ενώ σε ένα μικροελεγκτή 16 bit όπως ο MSP430 της συσκευής TelosB οι πράξεις μπορούν να γίνονται ανά ψηφιολέξεις των 16 bit (ή μικρότερες). Τα πειραματικά αποτελέσματα για τους κρυπτογραφικούς αλγόριθμους που βασίζονται στις ελλειπτικές καμπύλες παρουσιάζονται στη συνέχεια.

7.3.1 Παραγωγή Ζευγαριού Κλειδιών με Ελλειπτικές Καμπύλες

Στον Πίνακα 7.5 παρουσιάζονται ο χρόνος εκτέλεσης και η κατανάλωση ενέργειας κατά την παραγωγή ζευγαριού κλειδιών ελλειπτικών καμπυλών ανάλογα με το μέγεθος της καμπύλης και το μέγεθος της ψηφιολέξης που ορίζουμε για τους μικροεπεξεργαστές των συσκευών iSense και TelosB. Ο μικροελεγκτής της συσκευής iSense μπορεί να εκτελεί πράξεις με ψηφιολέξεις μεγέθους 8, 16 ή 32 bit ενώ αυτός της συσκευής TelosB μπορεί να εκτελεί πράξεις με ψηφιολέξεις μεγέθους 8 ή 16 bit. Όπως φαίνεται και από τον Πίνακα 7.5, η διαδικασία παραγωγής δημοσίου κλειδιού είναι η πιο απαιτητική ενέργεια αφού στην ουσία αποτελεί το βαθμωτό πολλαπλασιασμό του ιδιωτικού κλειδιού με το σημείο γεννήτορα της καμπύλης. Η ασφάλεια της κρυπτογραφίας με ελλειπτικές καμπύλες βασίζεται σε αυτήν την πράξη καθώς το αποτέλεσμα της είναι *δύσκολο* να αντιστραφεί.

Μέγεθος Καμπύλης	Μέγεθος Λέξης	Πράξη	Χρόνος Εκτέλεσης		Κατανάλωση Ενέργειας	
			iSense	TelosB	iSense	TelosB
128 bit	8 bit	Αρχικοποίηση Καμπύλης	5 msec	0.101 sec	0.19 mJ	0.54 mJ
		Παραγωγή Μυστικού Κλειδιού	4 msec	0.1 sec	0.15 mJ	0.54 mJ
		Παραγωγή Δημοσίου Κλειδιού	4.394 sec	35.96 sec	0.16 J	0.19 J
	16 bit	Αρχικοποίηση Καμπύλης	5 msec	0.1 sec	0.19 mJ	0.54 mJ
		Παραγωγή Μυστικού Κλειδιού	4 msec	0.099 msec	0.15 mJ	0.53 mJ
		Παραγωγή Δημοσίου Κλειδιού	2.42 sec	21.57 sec	0.09 J	0.11 J
	32 bit	Αρχικοποίηση Καμπύλης	4 msec	-	0.15 mJ	-
		Παραγωγή Μυστικού Κλειδιού	7 msec	-	0.26 mJ	-
		Παραγωγή Δημοσίου Κλειδιού	1.781 sec	-	0.06 J	-
160 bit	8 bit	Αρχικοποίηση Καμπύλης	4 msec	0.1 sec	0.15 mJ	0.54 mJ
		Παραγωγή Μυστικού Κλειδιού	4 msec	0.1 msec	0.15 mJ	0.54 mJ
		Παραγωγή Δημοσίου Κλειδιού	8.716 sec	67.27 sec	0.33 J	0.36 J
	16 bit	Αρχικοποίηση Καμπύλης	3 msec	0.09 sec	0.11 mJ	0.53 mJ
		Παραγωγή Μυστικού Κλειδιού	3 msec	0.1 sec	0.11 mJ	0.54 mJ
		Παραγωγή Δημοσίου Κλειδιού	4.58 sec	45.21 sec	0.17 J	0.24 J
	32 bit	Αρχικοποίηση Καμπύλης	4 msec	-	0.15 mJ	-
		Παραγωγή Μυστικού Κλειδιού	6 msec	-	0.22 mJ	-
		Παραγωγή Δημοσίου Κλειδιού	2.483 sec	-	0.09 J	-
192 bit	8 bit	Αρχικοποίηση Καμπύλης	3 msec	99 msec	0.11 mJ	0.53 mJ
		Παραγωγή Μυστικού Κλειδιού	5 msec	99 msec	0.19 mJ	0.53 mJ
		Παραγωγή Δημοσίου Κλειδιού	14.024 sec	114.37 sec	0.53 J	0.61 J
	16 bit	Αρχικοποίηση Καμπύλης	4 msec	0.1 sec	0.15 mJ	0.54 mJ
		Παραγωγή Μυστικού Κλειδιού	5 msec	0.1 sec	0.19 mJ	0.54 mJ
		Παραγωγή Δημοσίου Κλειδιού	7.53 sec	76.82 sec	0.28 J	0.41 J
	32 bit	Αρχικοποίηση Καμπύλης	6 msec	-	0.22 mJ	-
		Παραγωγή Μυστικού Κλειδιού	4 msec	-	0.15 mJ	-
		Παραγωγή Δημοσίου Κλειδιού	4.05 sec	-	0.15 J	-

Πίνακας 7.5: Χρόνος Εκτέλεσης και Κατανάλωση Ενέργειας για την Παραγωγή Ζευγαριού Κλειδιών με Ελλειπτικές Καμπύλες σε Σχέση με το Μέγεθός τους στις Συσκευές iSense και TelosB.

Εν συνεχεία, στον Πίνακα 7.6 παρουσιάζεται το μέγεθος του μεταφρασμένου κώδικα των διαδικασιών αρχικοποίησης καμπύλης, παραγωγής μυστικού και δημοσίου κλειδιού για τις συσκευές iSense και TelosB.

Μέγεθος Καμπύλης	Μέγεθος Λέξης	Πράξη	Μέγεθος Κώδικα	
			iSense	TelosB
128 bit	8 bit	Αρχικοποίηση Καμπύλης	964 bytes	716 bytes
		Παραγωγή Μυστικού Κλειδιού	872 bytes	550 bytes
		Παραγωγή Δημοσίου Κλειδιού	5476 bytes	4524 bytes
	16 bit	Αρχικοποίηση Καμπύλης	680 bytes	474 bytes
		Παραγωγή Μυστικού Κλειδιού	1032 bytes	550 bytes
		Παραγωγή Δημοσίου Κλειδιού	5580 bytes	4778 bytes
	32 bit	Αρχικοποίηση Καμπύλης	596 bytes	-
		Παραγωγή Μυστικού Κλειδιού	1004 bytes	-
		Παραγωγή Δημοσίου Κλειδιού	6028 bytes	-
160 bit	8 bit	Αρχικοποίηση Καμπύλης	1116 bytes	870 bytes
		Παραγωγή Μυστικού Κλειδιού	1000 bytes	550 bytes
		Παραγωγή Δημοσίου Κλειδιού	5472 bytes	4524 bytes
	16 bit	Αρχικοποίηση Καμπύλης	700 bytes	530 bytes
		Παραγωγή Μυστικού Κλειδιού	1032 bytes	554 bytes
		Παραγωγή Δημοσίου Κλειδιού	5576 bytes	4784 bytes
	32 bit	Αρχικοποίηση Καμπύλης	684 bytes	-
		Παραγωγή Μυστικού Κλειδιού	1004 bytes	-
		Παραγωγή Δημοσίου Κλειδιού	6024 bytes	-
192 bit	8 bit	Αρχικοποίηση Καμπύλης	1072 bytes	810 bytes
		Παραγωγή Μυστικού Κλειδιού	1000 bytes	550 bytes
		Παραγωγή Δημοσίου Κλειδιού	5476 bytes	4524 bytes
	16 bit	Αρχικοποίηση Καμπύλης	848 bytes	632 bytes
		Παραγωγή Μυστικού Κλειδιού	1032 bytes	554 bytes
		Παραγωγή Δημοσίου Κλειδιού	5580 bytes	4784 bytes
	32 bit	Αρχικοποίηση Καμπύλης	764 bytes	-
		Παραγωγή Μυστικού Κλειδιού	1004 bytes	-
		Παραγωγή Δημοσίου Κλειδιού	6028 bytes	-

Πίνακας 7.6: Μέγεθος Μεταφρασμένου Κώδικα για την Παραγωγή Ζευγαριού Κλειδιών με Ελλειπτικές Καμπύλες σε Σχέση με το Μέγεθός τους στις Συσκευές iSense και TelosB.

Από τον Πίνακα 7.5 παρατηρούμε πως βελτιστοποιείται ο χρόνος εκτέλεσης των αριθμητικών πράξεων όταν ορίζουμε μεγαλύτερη ψηφιολέξη την οποία μπορεί να υποστηρίζει το υλικό. Σαν παράδειγμα αναφέρουμε ότι στη συσκευή iSense η απαιτητική πράξη της δημιουργίας δημοσίου κλειδιού στην ελλειπτική καμπύλη μεγέθους 160-bit απαιτεί 8 sec όταν ορίζουμε 8-bit ψηφιολέξη, 4 sec όταν ορίζουμε 16-bit ψηφιολέξη και 2 sec όταν ορίζουμε 32-bit ψηφιολέξη και εκμεταλλευόμαστε πλήρως τον μικροεπεξεργαστή JN5139.

7.3.2 Το Σχήμα Συμφωνίας Κλειδιών ECDH

Για την πειραματική αξιολόγηση του σχήματος συμφωνίας κλειδιών ECDH στις συσκευές iSense και TelosB χρησιμοποιήσαμε την ελλειπτική καμπύλη μεγέθους 160 bit. Στον Πίνακα 7.7, παρουσιάζονται ο χρόνος εκτέλεσης και η κατανάλωση ενέργειας κατά την παραγωγή ενός διαμοιραζόμενου κλειδιού ελλειπτικών καμπυλών με χρήση του σχήματος συμφωνίας κλειδιών ECDH ανάλογα με το μέγεθος της ψηφιολέξης που ορίζουμε για τους μικροεπεξεργαστές των συσκευών iSense και TelosB. Με βάση την περιγραφή του σχήματος ECDH, παρατηρούμε ότι η διαδικασία παραγωγής διαμοιραζόμενου κλειδιού αποτελείται από ένα πολλαπλασιασμό στην ελλειπτική καμπύλη και μια πράξη κατακερματισμού για τη δημιουργία του κλειδιού. Λαμβάνοντας υπόψιν αυτή την παρατήρηση αυτή και τους χρόνους που παρουσιάστηκαν στον Πίνακα 7.5 οι χρόνοι που παρουσιάζονται στον Πίνακα 7.7 είναι λογικοί.

Μέγεθος Καμπύλης	Μέγεθος Λέξης	Πράξη	Χρόνος Εκτέλεσης		Κατανάλωση Ενέργειας	
			iSense	TelosB	iSense	TelosB
160 bit	8 bit	Παραγωγή Διαμοιραζόμενου Κλειδιού	8.864 sec	69.567 sec	0.337 J	0.375 J
	16 bit	Παραγωγή Διαμοιραζόμενου Κλειδιού	4.701 sec	47.462 sec	0.179 J	0.256 J
	32 bit	Παραγωγή Διαμοιραζόμενου Κλειδιού	2.995 sec	-	0.114 J	-

Πίνακας 7.7: Χρόνος Εκτέλεσης και Κατανάλωση Ενέργειας για την Παραγωγή Διαμοιραζόμενου Κλειδιού με το Πρωτόκολλο ECDH στις Συσκευές iSense και TelosB.

Στη συνέχεια, μετρήσαμε το μέγεθος του μεταφρασμένου κώδικα της διαδιασίας παραγωγής μυστικού κοινού κλειδιού στις συσκευές iSense και TelosB. Τα αποτελέσματα φαίνονται στον Πίνακα 7.8.

Μέγεθος Καμπύλης	Μέγεθος Λέξης	Πράξη	Μέγεθος Κώδικα	
			iSense	TelosB
160 bit	8 bit	Παραγωγή Διαμοιραζόμενου Κλειδιού	2604 bytes	3522 bytes
	16 bit	Παραγωγή Διαμοιραζόμενου Κλειδιού	2592 bytes	3544bytes
	32 bit	Παραγωγή Διαμοιραζόμενου Κλειδιού	2604 bytes	-

Πίνακας 7.8: Μέγεθος Μεταφρασμένου Κώδικα για την Παραγωγή Διαμοιραζόμενου Κλειδιού με το Πρωτόκολλο ECDH στις Συσκευές iSense και TelosB.

7.3.3 Ο Αλγόριθμος Κρυπτογράφησης ECIES

Για την πειραματική αξιολόγηση του σχήματος συμφωνίας κλειδιών ECDH στις συσκευές iSense και TelosB χρησιμοποιήσαμε την ελλειπτική καμπύλη μεγέθους 160 bit. Στον Πίνακα 7.9, παρουσιάζονται ο χρόνος εκτέλεσης και η κατανάλωση ενέργειας για τις διαδικασίες κρυπτογράφησης και αποκρυπτογράφησης ενός μπλοκ δεδομένων με χρήση του αλγορίθμου κρυπτογράφησης ECIES ανάλογα με το μέγεθος της ψηφιολέξης που ορίζουμε για τους μικροεπεξεργαστές των συσκευών iSense και TelosB. Με βάση την περιγραφή του αλγορίθμου ECIES, παρατηρούμε ότι η διαδικασία κρυπτογράφησης ενός μπλοκ δεδομένων αποτελείται από μια παραγωγή ενός ιδιωτικού κλειδιού, μια παραγωγή ενός δημοσίου κλειδιού, ένα πολλαπλασιασμό πάνω στην καμπύλη, μια πράξη XOR και 2 πράξεις κατακερματισμού. Αντίστοιχα, η διαδικασία αποκρυπτογράφησης ενός κρυπτογραφημένου μπλοκ δεδομένων αποτελείται από ένα πολλαπλασιασμό στην ελλειπτική καμπύλη, μια πράξη XOR και 2 πράξεις κατακερματισμού. Με βάση τις παρατηρήσεις αυτές και τα αποτελέσματα του Πίνακα 7.5 επιβεβαιώνουμε τους χρόνους που παρουσιάζονται στον Πίνακα 7.9.

Μέγεθος Καμπύλης	Μέγεθος Λέξης	Πράξη	Χρόνος Εκτέλεσης		Κατανάλωση Ενέργειας	
			iSense	TelosB	iSense	TelosB
160 bit	8 bit	Κρυπτογράφηση	17.978 sec	137.303 sec	0.684 J	0.741 J
		Αποκρυπτογράφηση	8.935 sec	69.793 sec	0.34 J	0.376 J
	16 bit	Κρυπτογράφηση	9.642 sec	96.572 sec	0.367 J	0.521 J
		Αποκρυπτογράφηση	4.613 sec	46.416 sec	0.175 J	0.25 J
	32 bit	Κρυπτογράφηση	5.503 sec	-	0.209 J	-
		Αποκρυπτογράφηση	2.69 sec	-	0.102 J	-

Πίνακας 7.9: Χρόνος Εκτέλεσης και Κατανάλωση Ενέργειας των Διαδικασιών Κρυπτογράφησης και Αποκρυπτογράφησης του Αλγορίθμου ECIES στις Συσκευές iSense και TelosB.

Ακολούθως, στον Πίνακα 7.10 παρουσιάζεται το μέγεθος μεταφρασμένου κώδικα για τις διαδικασίες κρυπτογράφησης και αποκρυπτογράφησης του αλγορίθμου ECIES στις συσκευές iSense και TelosB.

7.3.4 Ο Αλγόριθμος Ψηφιακής Υπογραφής ECDSA

Στον Πίνακα 7.11, παρουσιάζονται ο χρόνος εκτέλεσης και η κατανάλωση ενέργειας για τις διαδικασίες παραγωγής και επαλήθευσης υπογραφής για ένα μπλοκ δεδομένων που αποτελεί το μήνυμα με χρήση του αλγορίθμου ψηφιακής υπογραφής

Μέγεθος Καμπύλης	Μέγεθος Λέξης	Πράξη	Μέγεθος Κώδικα	
			iSense	TelosB
160 bit	8 bit	Κρυπτογράφηση	3440 bytes	4868 bytes
		Αποκρυπτογράφηση	4192 bytes	6828 bytes
	16 bit	Κρυπτογράφηση	3428 bytes	4754 bytes
		Αποκρυπτογράφηση	4204 bytes	8014 bytes
	32 bit	Κρυπτογράφηση	3440 bytes	-
		Αποκρυπτογράφηση	4188 bytes	-

Πίνακας 7.10: Μέγεθος Μεταφρασμένου Κώδικα των Διαδικασιών Κρυπτογράφησης και Αποκρυπτογράφησης του Αλγορίθμου ECIES στις Συσκευές iSense και TelosB.

ECDSA ανάλογα με μέγεθος της ψηφιολέξης που ορίζουμε για τους μικροεπεξεργαστές των συσκευών iSense και TelosB. Με βάση την περιγραφή του αλγορίθμου ECDSA, παρατηρούμε ότι η διαδικασία παραγωγής υπογραφής για ένα μπλοκ δεδομένων αποτελείται από μια παραγωγή ενός ιδιωτικού κλειδιού, μια παραγωγή ενός δημοσίου κλειδιού, και μια πράξη κατακερματισμού. Αντίστοιχα, η διαδικασία επαλήθευσης μιας υπογραφής για ένα μπλοκ δεδομένων αποτελείται από δύο πολλαπλασιασμούς στην ελλειπτική καμπύλη, μια πρόσθεση σημείων στην ελλειπτική καμπύλη και μια πράξη κατακερματισμού. Με βάση τις παρατηρήσεις αυτές και τα αποτελέσματα του Πίνακα 7.5 επιβεβαιώνουμε τους χρόνους που παρουσιάζονται στον Πίνακα 7.11.

Μέγεθος Καμπύλης	Μέγεθος Λέξης	Πράξη	Χρόνος Εκτέλεσης		Κατανάλωση Ενέργειας	
			iSense	TelosB	iSense	TelosB
160 bit	8 bit	Δημιουργία Υπογραφής	8.981 sec	70.32 sec	0.342 J	0.379 J
		Επαλήθευση Υπογραφής	18.65 sec	142.029 sec	0.71 J	0.766 J
	16 bit	Δημιουργία Υπογραφής	4.746 sec	48.595 sec	0.18J	0.262 J
		Επαλήθευση Υπογραφής	10.143 sec	98.930 sec	0.386 J	0.534 J
	32 bit	Δημιουργία Υπογραφής	2.846 sec	-	0.108 J	-
		Επαλήθευση Υπογραφής	5.988 sec	-	0.228 J	-

Πίνακας 7.11: Χρόνος Εκτέλεσης και Κατανάλωση Ενέργειας των Διαδικασιών Δημιουργίας και Επαλήθευσης Υπογραφής του Αλγορίθμου ECDSA στις Συσκευές iSense και TelosB.

Τέλος, στον Πίνακα 7.12 φαίνεται το μέγεθος μεταφρασμένου κώδικα για τις διαδικασίες δημιουργίας και επαλήθευσης υπογραφής του αλγορίθμου ECDSA ανάλογα με το μέγεθος ψηφιολέξης που ορίζουμε για τις συσκευές iSense και TelosB.

Μέγεθος Καμπύλης	Μέγεθος Λέξης	Πράξη	Μέγεθος Κώδικα	
			iSense	TelosB
160 bit	8 bit	Δημιουργία Υπογραφής	3376 bytes	4864 bytes
		Επαλήθευση Υπογραφής	916 bytes	1894bytes
	16 bit	Δημιουργία Υπογραφής	3364 bytes	5022bytes
		Επαλήθευση Υπογραφής	920 bytes	2066 bytes
	32 bit	Δημιουργία Υπογραφής	3372 bytes	-
		Επαλήθευση Υπογραφής	912 bytes	-

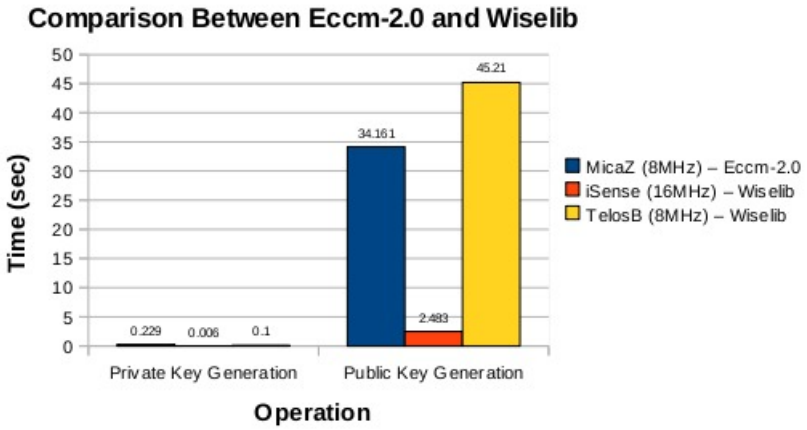
Πίνακας 7.12: Μέγεθος Μεταφρασμένου Κώδικα των Διαδικασιών Δημιουργίας και Επαλήθευσης Υπογραφής του Αλγορίθμου ECDSA στις Συσκευές iSense και TelosB.

7.4 Σύγκριση με Προηγούμενες Εργασίες

Για να ελέγξουμε την απόδοση της υλοποίησης μας, συγκρίνουμε τα πειραματικά μας αποτελέσματα με αυτά που παρουσιάζονται στις εργασίες EccM-2.0 [9] και TinyECC[16].

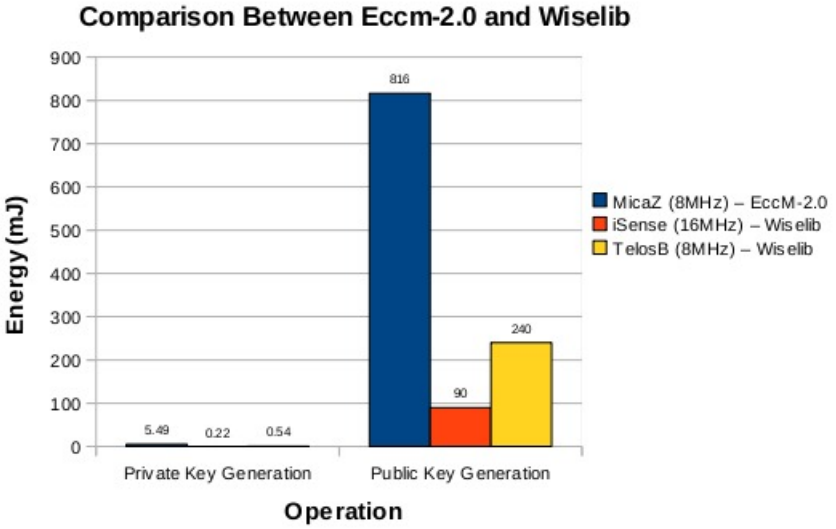
7.4.1 Σύγκριση της Wiselib με την Εργασία EccM-2.0

Αρχικά, συγκρίνουμε το απαιτούμενο χρόνο για τη δημιουργία ενός ζευγαριού ελλειπτικών καμπυλών μεταξύ της εργασίας EccM-2.0 και της υλοποίησης μας στη βιβλιοθήκη Wiselib. Στο σημείο αυτό αναφέρουμε, ότι ο κώδικας της εργασίας EccM-2.0 ορίζει μια ελλειπτική καμπύλη πάνω από πεδίο F_{2^m} μεγέθους 163-bit και η απόδοση της υλοποίησης μετράται στη συσκευή MICA-2 (8MHz). Η δικιά μας υλοποίηση ορίζει μια ελλειπτική καμπύλη πάνω απο πεδίο F_p μεγέθους 160-bit και η απόδοση της ελέγχεται στις συσκευές iSense (16MHz) και TelosB (8MHz). Τα αποτελέσματα φαίνονται στον Σχήμα 7.1.



Σχήμα 7.1: Σύγκριση Χρόνου Εκτέλεσης των Διαδικασιών Δημιουργίας Προσωπικού και Δημοσίου Κλειδιού των EccM-2.0 και Wiselib.

Στη συνέχεια, συγκρίνουμε την κατανάλωση ενέργειας για τη δημιουργία ενός ζευγαριού ελλειπτικών καμπυλών μεταξύ της εργασίας EccM-2.0 και της υλοποίησης μας στη βιβλιοθήκη Wiselib. Τα αποτελέσματα φαίνονται στο Σχήμα 7.2.



Σχήμα 7.2: Σύγκριση Κατανάλωση Ενέργειας (mJ) των Διαδικασιών Δημιουργίας Προσωπικού και Δημοσίου Κλειδιού των EccM-2.0 και Wiselib.

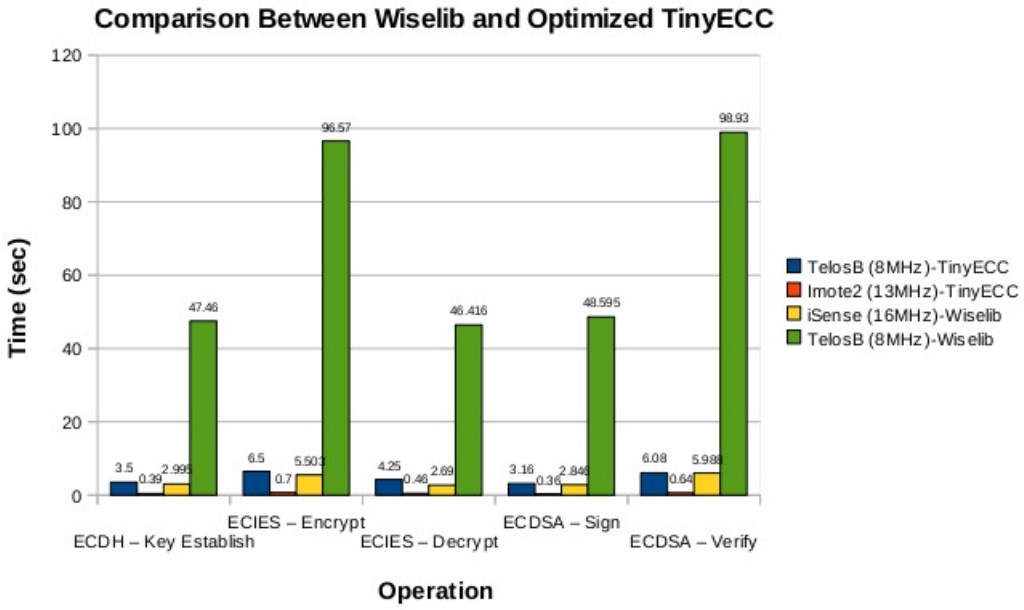
Από τα Σχήματα 7.1 και 7.2, παρατηρούμε ότι η συσκευή iSense (16 MHz) είναι η πιο γρήγορη στον υπολογισμό ενός ζευγαριού κλειδιών ελλειπτικής καμπύλης και η πιο οικονομική από άποψη κατανάλωσης ενέργειας. Επίσης, παρατηρούμε ότι η συσκευή TelosB είναι πιο γρήγορη στον υπολογισμό του ιδιωτικού κλειδιού σε σχέση με τη συσκευή MICA-2. Αυτό ισχύει επειδή ο μικροεπεξεργαστής της συσκευής TelosB λειτουργεί στα 16-bit, ενώ αυτό της MICA-2 στα 8-bit. Στον υπολογισμό του δημοσίου κλειδιού η συσκευή MICA-2 είναι πιο γρήγορη καθώς ο βαθμωτός πολλαπλασιασμός σε ελλειπτικές καμπύλες ορισμένες πάνω από πεδία F_{2^m} εκτελείται πιο γρήγορα σε σχέση με την ίδια πράξη πάνω από πεδία F_p. Παρά το γεγονός αυτό η συσκευή TelosB είναι πιο οικονομική ως προς την κατανάλωση ενέργειας σε σχέση με τη συσκευή MICA-2.

7.4.2 Σύγκριση της Wiselib με την Εργασία TinyECC

Εν συνεχεία, συγκρίναμε τον απαιτούμενο χρόνο εκτέλεσης, την απαιτούμενη ενέργεια και το μέγεθος μεταφρασμένου κώδικα των βασικών διαδικασιών των αλγορίθμων ECDH, ECIES και ECDSA μεταξύ των υλοποιήσεων TinyECC και Wiselib. Πρώτα, έγινε η σύγκριση στην περίπτωση που το TinyECC έχει ενεργοποιημένες όλες τις βελτιστοποιήσεις και στη συνέχεια όταν τις έχει απενεργοποιημένες. Αναφέρουμε ότι το μέγεθος και είδος καμπύλης που χρησιμοποιείται και στις δυο υλοποιήσεις είναι το ίδιο (160-bit ελλειπτική καμπύλη πάνω από πεδίο F_p) και ότι για τη σύγκριση χρησιμοποιήσαμε τα αποτελέσματα που παρουσιάζονται για τις συσκευές TelosB (8 MHz) και Imote2 (13 MHz) οι οποίες είναι περίπου ισοδύναμες με τις αντίστοιχες TelosB (8 MHz) και iSense που χρησιμοποιήσαμε στα δικά μας πειράματα. Τα αποτελέσματα φαίνονται στα ακόλουθα σχήματα.

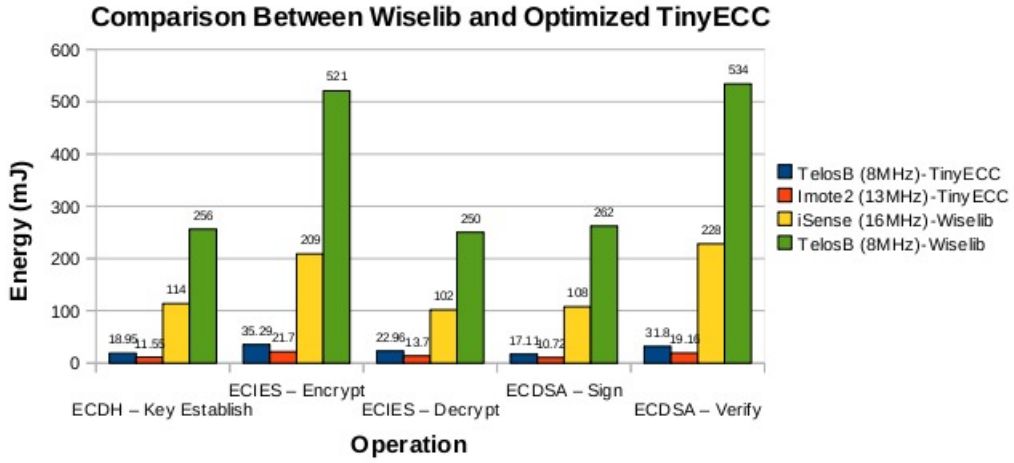
Wiselib vs Optimized TinyECC

Αρχικά, συγκρίνουμε την απόδοση των υλοποιήσεων της Wiselib με αυτή του TinyECC όταν είναι ενεργοποιημένες όλες οι βελτιστοποιήσεις λογισμικού.



Σχήμα 7.3: Σύγκριση Χρόνου Εκτέλεσης των Αλγορίθμων ECDH, ECIES, ECDSA των TinyECC (Optimized) και Wiselib.

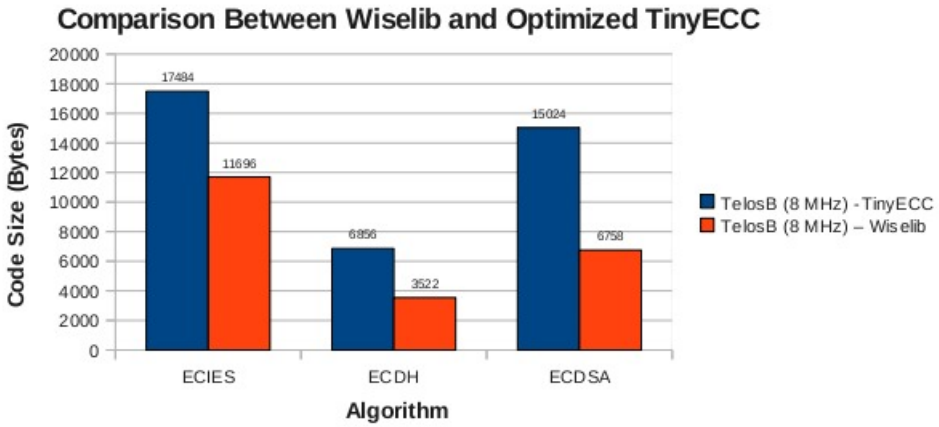
Στο Σχήμα 7.3, παρατηρούμε ότι όταν οι βελτιστοποιήσεις του TinyECC είναι ενεργοποιημένες η συσκευή Imote2 είναι η πιο γρήγορη στην εκτέλεση των βασικών διαδικασιών των αλγορίθμων ECDH, ECIES και ECDSA. Η συσκευή iSense παρά τη γενική υλοποίηση της Wiselib επιτυγχάνει λίγο καλύτερους χρόνους από τη συσκευή TelosB καθώς έχει πιο γρήγορο επεξεργαστή (16 MHz) ο οποίος υποστηρίζει ψηφιολέξη των 32-bit. Τέλος, η συσκευή TelosB όταν εκτελεί τη γενική υλοποίηση της Wiselib είναι η πιο αργή στην εκτέλεση αυτών των διαδικασιών.



Σχήμα 7.4: Σύγκριση Κατανάλωσης Ενέργειας των Αλγορίθμων ECDH, ECIES, ECDSA των TinyECC (Optimized) και Wiselib.

Παρατηρώντας το Σχήμα 7.4, συμπεραίνουμε ότι η γενική υλοποίση της Wiselib στις συσκευές iSense και TelosB απαιτεί πολύ περισσότερη ενέργεια για την ολοκλήρωση των βασικών διαδικασιών των αλγορίθμων ECDH, ECIES και ECDSA. Η συσκευή iSense παρά το γεγονός ότι επιτυγχάνει καλύτερους χρόνους από την TelosB - TinyECC απαιτεί περισσότερη ενέργεια λόγω του γρηγορότερου μικροεπεξεργαστή. Η συσκευή TelosB - Wiselib είναι η πιο απαιτητική σε ενέργεια καθώς εκτελεί τις διαδικασίες των αλγορίθμων πολύ πιο αργά από τις υπόλοιπες συσκευές.

Για τη σύγκριση του μεγέθους μεταφρασμένου κώδικα, χρησιμοποιήσαμε τα αποτελέσματα των Wiselib και TinyECC για τη συσκευή TelosB. Το Σχήμα 7.5 παρουσιάζει τα αποτελέσματα.

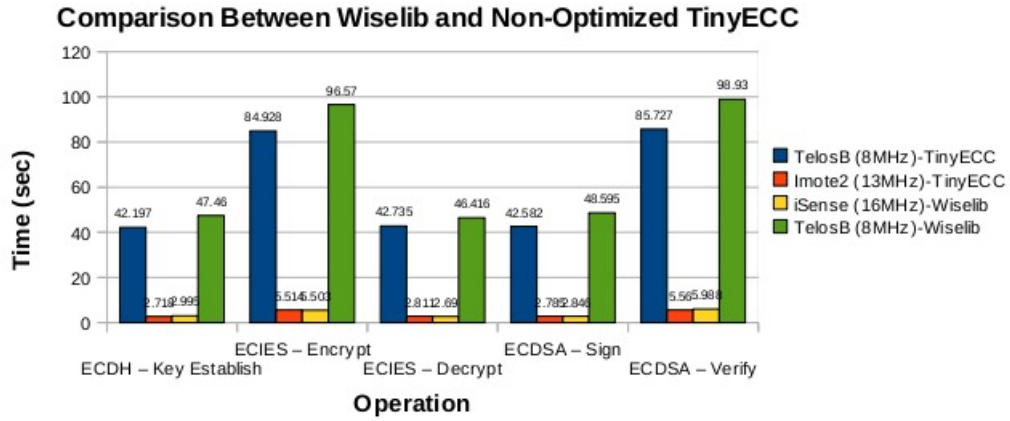


Σχήμα 7.5: Σύγκριση Μεγέθους Μεταφρασμένου Κώδικα των Αλγορίθμων ECDH, ECIES, ECDSA των TinyECC (Optimized) και Wiselib.

Παρατηρώντας το Σχήμα 7.5, συμπεραίνουμε ότι η γενική υλοποίηση της Wiselib επιτυγχάνει πολύ μικρότερο μέγεθος κώδικα για τους αλγορίθμους ECIES, ECDH και ECDSA σε σχέση με το Optimized TinyECC. Αυτό είναι λογικό αν αναλογιστεί κανείς ότι το Optimized TinyECC χρησιμοποιεί διάφορες βελτιστοποιήσεις λογισμικού. Αυτός μάλιστα είναι και ο βασικός λόγος για τον οποίο αποφασίσαμε να μην μεταφέρουμε αυτές τις βελτιστοποιήσεις στη γενικού σκοπού βιβλιοθήκη Wiselib.

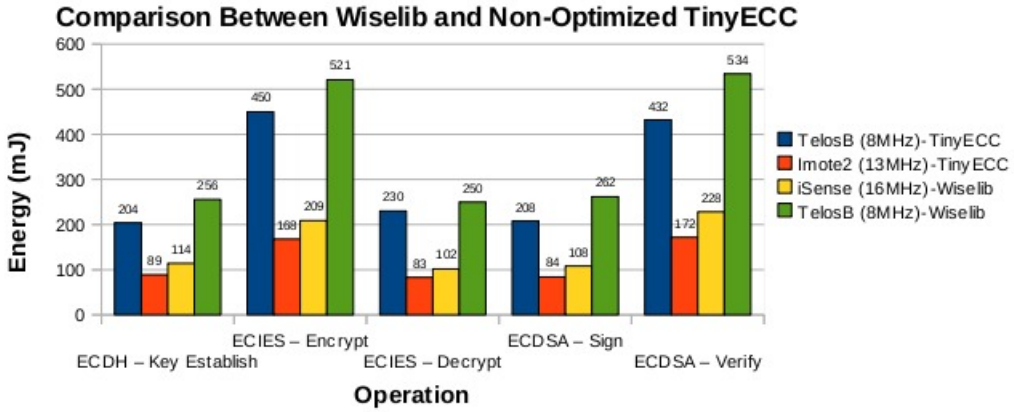
Wiselib vs Non-Optimized TinyECC

Ακολούθως, συγκρίνουμε την απόδοση των υλοποιήσεων της Wiselib με αυτή του TinyECC όταν είναι απενεργοποιημένες όλες οι βελτιστοποιήσεις λογισμικού.



Σχήμα 7.6: Σύγκριση Χρόνου Εκτέλεσης των Αλγορίθμων ECDH, ECIES, ECDSA των TinyECC (Non-Optimized) και Wiselib.

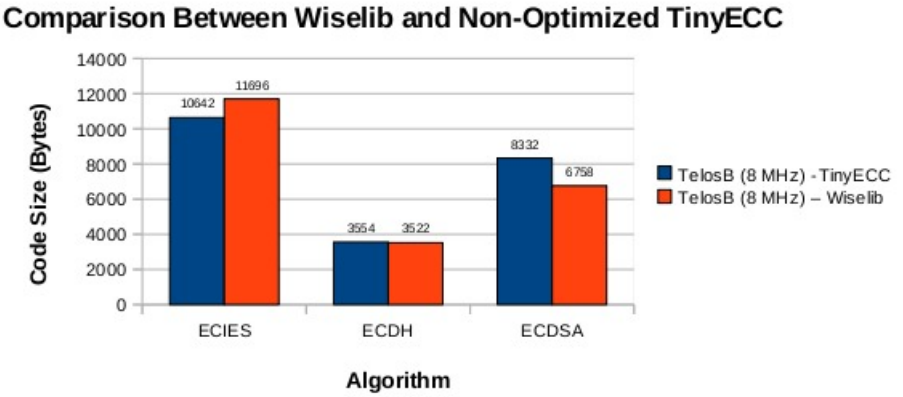
Στο Σχήμα 7.6, παρατηρούμε ότι όταν οι βελτιστοποιήσεις του TinyECC είναι απενεργοποιημένες οι δύο υλοποιήσεις επιτυγχάνουν παρόμοιους χρόνους στις αντίστοιχες συσκευές. Για παράδειγμα, η συσκευή iSense για την ολοκλήρωση μιας πράξης κρυπτογράφησης με τον αλγόριθμο ECIES απαίτει 5.5 sec περίπου όσα και η αντίστοιχης επεξεργαστικής δύναμης συσκευή Imote2. Αντίστοιχα, η συσκευή TelosB - TinyECC απαιτεί 43 sec για τη δημιουργία διαμοιραζόμενου κλειδιού με τον αλγόριθμο ECDH όσα περίπου απαιτεί και η TelosB - Wiselib.



Σχήμα 7.7: Σύγκριση Κατανάλωσης Ενέργειας των Αλγορίθμων ECDH, ECIES, ECDSA των TinyECC (Non-Optimized) και Wiselib.

Συγκρίνοντας την κατανάλωση ενέργειας μεταξύ των υλοποιήσεων Wiselib και TinyECC - Non Optimized παρατηρούμε από το Σχήμα 7.7 ότι οι δύο υλοποιήσεις

επιτυγχάνουν παρεμφερή αποτελέσματα. Η συσκευή Imote2 απαίτει για την κρυπτογράφηση με τον αλγόριθμο ECIES 160 mJ ενώ η συσκευή iSense απαιτεί 200 mJ. Αυτό είναι λογικό αν σκεφτούμε ότι ο επεξεργαστής της συσκευής Imote2 δουλεύει σε λίγο μικρότερη συχνότητα από αυτόν της iSense, οπότε απαιτεί λιγότερη ενέργεια. Αντίστοιχα, η συσκευή TelosB - TinyECC για την παραγωγή ενός διαμοιραζόμενου κλειδιού με τον αλγόριθμο ECDH απαίτει 204 mJ ενώ η TelosB - Wiselib 250 mJ για την ίδια πράξη.



Σχήμα 7.8: Σύγκριση του Μεγέθους Μεταφρασμένου Κώδικα των Αλγορίθμων ECDH, ECIES, ECDSA των TinyECC (Non-Optimized) και Wiselib.

Τέλος, παρατηρούμε ότι το μέγεθος του μεταφρασμένου κώδικα των αλγορίθμων ECDH, ECIES και ECDSA για τις υλοποιήσεις Wiselib και Non Optimized TinyECC είναι περίπου το ίδιο.

Περίληψη Σύγκρισης

Από τις παραπάνω συγκρίσεις, συμπεραίνουμε ότι η υλοποίηση TinyECC επιτυγχάνει πολύ καλύτερη απόδοση (χρόνος, ενέργεια) σε σχέση με την υλοποίση της Wiselib όταν όλες οι τεχνικές βελτιστοποίησης είναι ενεργοποιημένες. Αυτό το αποτέλεσμα ήταν αναμενόμενο καθώς κατά την υλοποίηση Wiselib αποφασίσαμε να μην χρησιμοποιήσουμε τις βελτιστοποιήσεις αυτές. Αυτή η απόφαση έγινε με σκοπό την αποφυγή μεγάλων μεγεθών μεταφρασμένου κώδικα που θα επιβάρυναν τις εφαρμογές ασυρμάτων δικτύων αισθητήρων. Αυτό μάλιστα αποδεικνύεται όταν συγκρίνουμε το μέγεθος μεταφρασμένου κώδικα των αλγορίθμων για τη συσκευή TelosB.

Όταν οι τεχνικές βελτιστοποίησης είναι απενεργοποιημένες οι δυο υλοποιήσεις επιτυγχάνουν παρεμφερή αποτελέσματα (χρόνος, ενέργεια, μέγεθος μεταφρασμένου κώδικα). Παρά το γεγονός αυτό, το πλεονέκτημα της υλοποίησης Wiselib είναι η ετερογένεια και η γενικότητα καθώς ο ίδιος κώδικας μπορεί να μεταφραστεί για διάφορα λειτουργικά συστήματα (TinyOS, iSense OS, Contiki Sky) και να εκτελεστεί σε διάφορες συσκευές όπως TelosB, iSense, ScatterWeb, MicaZ, Tmote Sky. Όπως αναφέραμε και προηγουμένως η υλοποίηση TinyECC μπορεί να εκτελεστεί μόνο σε συκευές που μπορούν να τρέξουν το λειτουργικό σύστημα

TinyOS. Η γενικότητα του κώδικα της Wiselib αποδείχθηκε με την πειραματική αξιολόγηση των υλοποιημένων κρυπτογραφικών αλγορίθμων σε δύο συσκευές (iSense, TelosB) που τρέχουν διαφορετικά λειτουργικά συστήματα (iSense OS και Contiki Sky αντίστοιχα).

Κεφάλαιο 8

Εφαρμογές των Κρυπτογραφικών Αλγορίθμων

Εν συνεχεία, παρουσιάζουμε τρεις εφαρμογές των υλοιποιημένων κρυπτογραφικών αλγορίθμων στα πλαίσια της βιβλιοθήκης Wiselib. Οι εφαρμογές αυτές αποδεικνύουν την ευκολία χρήσης των αλγορίθμων ασφάλειας που υλοποιήσαμε σε απλές λειτουργίες ασυρμάτων δικτύων αισθητήρων έτσι ώστε να δημιουργούνται ασφαλείς εφαρμογές.

8.1 Ασφαλής Δρομολόγηση

Στην εφαρμογή αυτή, παρουσιάζουμε τα πλεονεκτήματα της C++ και του σχεδιασμού με βάση τα πρότυπα (templates) που εφαρμόζεται στη βιβλιοθήκη Wiselib, εξετάζοντας δυο είδη αλγορίθμων: αλγόριθμους δρομολόγησης και κρυπτογραφίας. Αρχικά, περιγράφουμε και τα δύο είδη αλγορίθμων σαν απλά concepts και στη συνέχεια δείχνουμε πως μπορούν να συνδυαστούν ώστε να δημιουργηθούν αλγόριθμοι ασφαλής δρομολόγησης.

Αλγόριθμοι Δρομολόγησης: Το concept για τον ορισμό ενός αλγόριθμου δρομολόγησης αποτελείται από τις παρακάτω μεθόδους.

- Μια μέθοδο η οποία δημιουργεί το δείκτη για το μοντέλο λειτουργικού συστήματος (set_os()) που απαιτείται όταν καλούνται στατικές συναρτήσεις της εξωτερικής διεπαφής.
- Δύο μεθόδους για την ενεργοποίηση (enable()) και απενεργοποίηση (disable()) του αλγόριθμου δρομολόγησης, οι οποίες είναι χρήσιμες όταν η δρομολόγηση πρέπει να εκτελεστεί σε συγκεκριμένα χρονικά διαστήματα (π.χ. για θέματα εξοικονόμησης ενέργειας).
- Δύο μεθόδους για την εγγραφή (register()) και απεγγραφή (unregister()) μιας επανάκλησης για την παραλαβή μηνυμάτων.
- Μια μέθοδο για την αποστολή μηνυμάτων (send()) σε άλλους κόμβους του δικτύου.

Επίσης, το concept του αλγόριθμου δρομολόγησης εξειδικεύει το Radio Concept και είναι ως εξής:

```
1 concept Routing {
2   void set_os(OsModel* os);
3   void enable(void);
4   void disable(void);
5   void send(node_id_t receiver, size_t len, data_t* data);
6   template <class Callee, void (Callee::*Method)
7                         (node_id_t, size_t, data_t*)>
8   int reg_recv_callback(T *obj_pnt);
9   void unreg_recv_callback(int);
10 };
```

Αλγόριθμοι Κρυπτογράφησης: Το γενικό concept για ένα αλγόριθμο κρυπτογράφησης αποτελείται από τις εξής μεθόδους.

- Μια μέθοδο η οποία δημιουργεί το δείκτη για το μοντέλο λειτουργικού συστήματος (set_os()) που απαιτείται όταν καλούνται στατικές συναρτήσεις της εξωτερικής διεπαφής.
- Δύο μεθόδους για την ενεργοποίηση (enable()) και απενεργοποίηση (disable()) του αλγόριθμου κρυπτογράφησης, οι οποίες είναι χρήσιμες όταν η κρυπτογράφηση πρέπει να εκτελεστεί σε συγκεκριμένα χρονικά διαστήματα (π.χ. για θέματα εξοικονόμησης ενέργειας).
- Μια μέθοδο για την αρχικοποίηση και δημιουργία (key_setup()) των κρυπτογραφικών κλειδιών.
- Δύο μεθόδους για την κρυπτογράφηση (encrypt()) και την αποκρυπτογράφηση (decrypt()) μπλοκ δεδομένων.

Το concept του αλγόριθμου κρυπτογράφησης είναι ως εξής:

```
1 concept Crypto {
2    void set_os(OsModel* os);
3    void enable(void);
4    void disable(void);
5    void key_setup(node_id_t, data_t* key);
6    void encrypt(data_t* in, data_t* out, size_t length);
7    void decrypt(data_t* in, data_t* out, size_t length);
8 };
```

Ασφαλής Δρομολόγηση: Ακολούθως, περιγράφουμε πως τα concepts των αλγορίθμων δρομολόγησης και κρυπτογράφησης μπορούν να συνδυαστούν με αποτέλεσμα τη δημιουργία αλγορίθμων ασφαλής δρομολόγησης. Στο σημείο αυτό επισημαίνουμε ότι οποιοσδήποτε αλγόριθμος δρομολόγησης μπορεί να συνδυαστεί με οποιοδήποτε αλγόριθμο κρυπτογράφησης χωρίς αλλαγές στον κωδικά τους.

Το concept της ασφαλής δρομολόγησης δέχεται ως πρότυπες παραμέτρους έναν αλγόριθμο δρομολόγησης και ένα αλγόριθμο κρυπτογράφησης και εσωτερικά του χρησιμοποιούμε αυτούς τους τύπους. Για παράδειγμα, όταν ενεργοποιείται η ασφαλής δρομολόγηση, ενεργοποιούνται με τη σειρά οι αλγόριθμοι δρομολόγησης και κρυπτογράφησης. Όταν ένα μήνυμα πρόκεται να σταλεί, τα bytes δεδομένων πρώτα κρυπτογραφούνται με τον αλγόριθμο κρυπτογράφησης και στη συνέχεια αποστέλλονται στον προορισμό τους με τον αλγόριθμο δρομολόγησης. Αντίστοιχα, όταν ένα μήνυμα παραλαμβάνεται σε ένα προορισμό, πρώτα αποκρυπτογραφείται και στη συνέχεια παραδίδεται στους προοριζόμενους παραλήπτες. Το concept της ασφαλής δρομολόγησης είναι ως εξής:

```
1  template<typename Routing,
2           typename Crypto>
3  class SecureRouting {
4     void set_os(OsModel* os);
5     void enable(void);
6     void disable(void);
7     void send(node_id_t receiver, size_t len, data_t* data);
8     template <class Callee, void (Callee::*Method)
9                         (node_id_t, size_t, data_t*)>
10    int reg_recv_callback(T *obj_pnt);
11    void unreg_recv_callback(int);
12    Routing routing_;
13    Crypto crypto_;
14 };
```

Καθώς η κλάση ασφαλούς δρομολόγησης υλοποιεί το concept της δρομολόγησης, μπορεί να χρησιμοποιηθεί από οποιαδήποτε εφαρμογή που έχει να κάνει με δρομολόγηση. Παρόλα αυτά, οι διαδικασίες κρυπτογράφησης και αποκρυπτογράφησης είναι τελείως διαφανείς.

8.1.1 Πειραματικά Αποτελέσματα

Για την εφαρμογή αυτή κάναμε πειραματικές μετρήσεις στις συσκευές iSense για να διαπιστώσουμε το overhead που δημιουργεί η Wiselib κατά τον συνδυασμό αλγορίθμων. Μετρήσαμε την καθυστέρηση που υπάρχει για την αποστολή και παραλαβή 200 μηνυμάτων και το μέγεθος του μεταφρασμένου κώδικα όταν: α) χρησιμοποιείται ένας dummy αλγόριθμος δρομολόγησης και όταν χρησιμοποιείται ένας dummy αλγόριθμος δρομολόγησης σε συνδυασμό με ένα dummy αλγόριθμο κρυπτογράφησης, β) όταν χρησιμοποιείται ο αλγόριθμος δρομολόγησης DSDV και όταν χρησιμοποιείται ο αλγόριθμος DSDV σε συνδυασμό με ένα dummy αλγόριθμος κρυπτογράφησης και γ) όταν συνδυάζεται ο αλγόριθμος δρομολόγησης DSDV με τον αλγόριθμο κρυπτογράφησης AES. Τα αποτελέσματα φαίνονται στον Πίνακα 8.1.

	Dummy Routing	Dummy Routing, Dummy Crypto	DSDV Routing	DSDV Routing, Dummy Crypto	DSDV Routing, AES
Καθυστέρηση	6.08 msec	6.09 msec	6.72 msec	6.75 msec	70.95 msec
Μέγεθος Κώδικα	1892 bytes	2428 bytes	4912 bytes	6544 bytes	7974 bytes

Πίνακας 8.1: Overhead που Παράγει η Wiselib Κατά το Συνδυασμό Αλγορίθμων Δρομολόγησης και Κρυπτογράφησης στις Συσκευές iSense.

Από τα αποτελέσματα του Πίνακα 8.1, παρατηρούμε ότι η χρονοκαθυστέρηση που εισάγει η Wiselib κατά το συνδυασμό dummy αλγορίθμων είναι ελάχιστη. Άρα, συμπεραίνουμε ότι η δημιουργία σωρού αλγορίθμων είναι εφικτή στη Wiselib. Επιπλέον, βλέπουμε ότι το μέγεθος μεταφρασμένου κώδικα αυξάνει ανάλογα με το μέγεθος των συνδυαζόμενων αλγορίθμων κάτι που είναι λογικό.

8.2 Εγκαθίδρυση Κλειδιού Ομάδας με Ελλειπτικές Καμπύλες

Σαν δεύτερη εφαρμογή της κρυπτογραφικής μας βιβλιοθήκης, υλοποιήσαμε το πρωτόκολλο εγκαθίδρυσης κλειδιού ομάδας με χρήση ελλειπτικών καμπυλών το οποίο παρουσιάζεται στην εργασία [6]. Το πρωτόκολλο αυτό αποτελεί μια παραλλαγή του πρωτοκόλλου Group Diffie Hellman - 3.0 [20] και βασίζεται στην κρυπτογραφία

με ελλειπτικές καμπύλες για την αποδοτική του εφαρμογή του σε ασύρματα δίκτυα αισθητήρων.

Σύμφωνα με το πρωτόκολλο, οι συσκευές του δικτύου έχουν διακριτές ταυτότητες και γνωρίζουν μόνο τοπικές πληροφορίες όπως π.χ. τις ταυτότητες των γειτόνων τους. Καθολική πληροφορία για το μέγεθος ή τη δομή του δικτύου δεν είναι διαθέσιμη σε κανένα κόμβο. Το πρωτόκολλο ακολουθεί μια κατανεμημένη προσέγγιση η οποία δεν απαιτεί επικοινωνία *many-to-many* και δεν βασίζεται σε κάποια διάταξη των κόμβων του δικτύου. Το πρωτόκολλο αποτελείται από δυο βασικά στάδια τα οποία περιγράφονται στη συνέχεια:

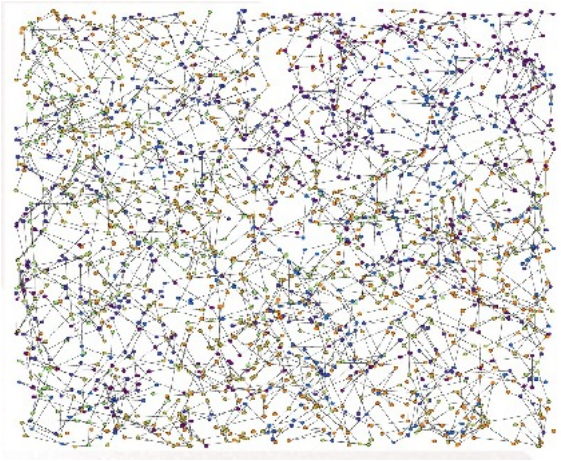
- Στο πρώτο στάδιο, κάθε κόμβος της ομάδας M_i δημιουργεί ένα τυχαίο μυστικό k_i. Ο κόμβος M_1 υπολογίζει το σημείο $Q_1 = k_1 \cdot G$ στην προσυμφωνημένη ελλειπτική καμπύλη και το στέλνει στο κόμβο M_2. Ο κόμβος M_2 πολλαπλασιάζει το σημείο που έλαβε με το δικό του μυστικό δηλ. $Q_2 = k_1 \cdot k_2 \cdot G$ και το προωθεί στον κόμβο M_3. Η διαδικασία αυτή συνεχίζεται μέχρις ότου να φτάσει στον κόμβο M_n της ομάδας. Το σημείο Q_n που θα υπολογιστεί από τον κόμβο M_n αποτελεί το κοινό κλειδί της ομάδας.

- Στο δεύτερο στάδιο, ο κόμβος M_n κρυπτογραφεί το σημείο Q_n με το δημόσιο κλειδί του κόμβου M_{n-1} και του το στέλνει. Ο κόμβος M_{n-1} αποκρυπτογραφεί το κοινό κλειδί της ομάδας Q_n με το μυστικό του κλειδί. Στη συνέχεια κρυπτογραφεί το κοινό κλειδί με το δημόσιο κλειδί του κόμβου M_{n-2} και του το προωθεί. Η διαδικασία συνεχίζεται μέχρι το κοινό κλειδί της ομάδας να φτάσει στον κόμβο M_1.

Πιο συγκεκριμένα, χρησιμοποιείται ένας αλγόριθμος που υλοποιεί μια ακολουθιακή τραβέρσα του δικτύου η οποία επισκέπτεται κάθε κόμβο της ομάδας έτσι ώστε όλοι να συνεισφέρουν στη δημιουργία του κοινού κλειδιού. Όταν η τραβέρσα ολοκληρωθεί και έχουν επισκεφτεί όλοι οι κόμβοι του δικτύου, το πρωτόκολλο φτάνει στο κόμβο M_n ο οποίος υπολογίζει το κοινό κλειδί Q_n. Τέλος, για να προωθηθεί το κοινό κλειδί στους υπόλοιπους κόμβους του δικτύου ακολουθείται η τραβέρσα με ανάστροφη σειρά. Κάθε κόμβος που μαθαίνει το κοινό κλειδί, το κρυπτογραφεί με το δημόσιο κλειδί του προηγούμενου και του το προωθεί. Η διαδικασία ολοκληρώνεται όταν το κοινό κλειδί της ομάδας φτάσει στον κόμβο από τον οποίο ξεκίνησε η τραβέρσα.

Για να υλοποιήσουμε το παραπάνω πρωτόκολλο ακολουθήσαμε την βασισμένη σε κομμάτια προσέγγιση που χρησιμοποιεί η Wiselib για την εύκολη και γρήγορη κατασκευή αλγορίθμων ομαδοποίησης. Η προσέγγιση αυτή περιγράφεται αναλυτικά στην τεχνική αναφορά [1]. Από την προσέγγιση αυτή χρησιμοποιήσαμε το module που μας παρέχει για την πιθανοτική εκλογή αρχηγών και το module που μας παρέχει για την εκτέλεση κατά βάθους τραβέρσας του δικτύου (Depth First Search), του οποίου η λειτουργία φαίνεται στο Σχήμα 8.1. Επιπλεόν, υλοποιήσαμε ένα module το οποίο ονομάζεται GKE και παρέχει τις κρυπτογραφικές πράξεις που απαιτεί το πρωτόκολλο. Συνδυάζοντας, τα 3 αυτά modules με τη χρήση προτύπων που υποστηρίζει η Wiselib υλοποιείται το πρωτόκολλο εγκαθίδρυσης κλειδιού ομάδας που περιγράφηκε προηγουμένως, μέσα στην κάθε ομάδα που δημιουργείται από το module DFS. Έτσι, στο τέλος της εκτέλεσης του πρωτοκόλλου, ένα δίκτυο είναι διαχωρισμένο σε ομάδες και οι κόμβοι που ανήκουν στην ίδια ομάδα διαμοιράζονται ένα κοινό κλειδί το οποίο μπορεί να χρησιμοποιηθεί για ασφαλή επικοινωνία. Από τη στιγμή που δημιουργηθεί το κοινό κλειδί με ελλειπτικές καμπύλες, οι

κόμβοι μπορούν να επικοινωνούν κρυπτογραφώντας τα μηνύματα με αυτό το κλειδί και ένα πιο αποδοτικό αλγόριθμο όπως π.χ. τον AES.



Σχήμα 8.1: Η Οργάνωση Ενός Δικτύου Μετά την Εκτέλεση του Module Depth First Search (DFS).

8.2.1 Πειραματικά Αποτελέσματα

Με βάση τα αποτελέσματα που παρουσιάσαμε στο προηγούμενο κεφάλαιο, δείχνουμε στον Πίνακα 8.2 το χρόνο εκτέλεσης και την κατανάλωση ενέργειας μιας συσκευής iSense για τις βασικές κρυπτογραφικές πράξεις που υλοποιεί το module GKE. Για τα αποτελέσματα αυτά, χρησιμοποιήσαμε την ελλειπτική καμπύλη μεγέθους 160-bit.

Πράξη	Δημιουργία Ιδιωτικού Κλειδιού	Βαθμωτός Πολλαπλασιασμός	Κρυπτογράφηση	Αποκρυπτογράφηση
Χρόνος Εκτέλεσης	6 msec	2.4 sec	5.5 sec	2.6 sec
Κατανάλωση Ενέργειας	0.22 mJ	0.09 J	0.2 J	0.1 J

Πίνακας 8.2: Χρόνος Εκτέλεσης και Κατανάλωση Ενέργειας των Κρυπτογραφικών Πράξεων που Παρέχει το Module GKE στις Συσκευές iSense.

Το μεταφρασμένο μέγεθος κώδικα του module που καλεί τις διαδικασίες της κρυπτογραφικής μας βιβλιοθήκης μετρήθηκε στα 14.24 KB. Τέλος, εκτελέσαμε ένα πείραμα σε μια πραγματική τοπολογία 10 κόμβων iSense στις εγκαταστάσεις του ΙΤΥ και μετρήσαμε το συνολικό χρόνο για τη δημιουργία ενός κοινού κλειδιού ομάδας περίπου στα 100 sec.

8.3 Αυθεντική Μετάδοση με Αλυσίδες Κλειδιών Κατακερματισμού

Σαν μια τρίτη εφαρμογή της κρυπτογραφικής μας βιβλιοθήκης, δείχνουμε πως μπορεί να χρησιμοποιηθεί ο αλγόριθμος κρυπτογραφικού κατακερματισμού SHA-1 για την επίτευξη αυθεντικής μετάδοσης με αλυσίδες κλειδιών κατακερματισμού. Η ιδέα

αυτή παρουσιάστηκε για πρώτη φορά στην εργασία των Adrian Perrig et. al [18] και έχει χρησιμοποιηθεί από τότε σε διάφορες εργασίες όπως [10] και [2].

Η εργασία αυτή υποστηρίζει ότι η αυθεντική μετάδοση με βάση ασυμμετρική κρυπτογραφία (π.χ. ψηφιακές υπογραφές) δεν αποτελεί πρακτική λύση λόγω του υψηλού overhead σε επεξεργαστική ισχύ, ενέργεια και μνήμη. Δεδομένης αυτής της υπόθεσης και της μικρής ισχύος ασύρματης επικοινωνίας που χρησιμοποιείται στα ασύρματα δίκτυα αισθητήρων προτείνει τη μέθοδο μTESLA για την αυθεντική μετάδοση.

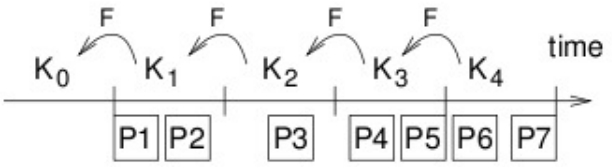
Το μTESLA εισάγει ασυμμετρία χρησιμοποιώντας συμμετρικά κλειδιά τα οποία τα αποκαλύπτει με κάποια χρονική καθυστέρηση. Βασική προυπόθεση της μεθόδου είναι ότι ο σταθμός βάσης και οι κόμβοι του δικτύου είναι συγχρονισμένοι και ότι κάθε κόμβος γνωρίζει ένα άνω όριο για το μέγιστο σφάλμα συγχρονισμού. Για την αποστολή ενός αυθεντικού πακέτου, ο σταθμός βάσης υπολογίζει ένα MAC (message authentication code) στο πακέτο με χρήση ενός κλειδιού το οποίο είναι μυστικό στο δεδομένο χρονικό διάστημα. Όταν ένας κόμβος λάβει το πακέτο, μπορεί να επαληθεύσει ότι το κλειδί αυτό δεν έχει αποκαλυφθεί ακόμα (με βάση το συγχρονισμό και το πρόγραμμα με το οποίο γίνεται αποκάλυψη κλειδιών). Έτσι, ο κόμβος είναι βέβαιος ότι το κλειδί αυτό το κατέχει μόνο ο σταθμός βάσης και κανένας επιτιθέμενος δεν έχει μεταβάλει το πακέτο. Εν συνεχεία, ο κόμβος αποθηκεύει το πακέτο σε κάποιο buffer. Τη στιγμή αποκάλυψης του κλειδιού ο σταθμός βάσης αποκαλύπτει το κλειδί επαλήθευσης σε όλους τους κόμβους οι οποίοι το χρησιμοποιούν για να επιβεβαιώσουν την εγκυρότητα των πακέτων που είχαν λάβει σε προηγούμενα χρονικά διαστήματα.

Κάθε MAC κλειδί είναι ένα κομμάτι μιας αλυσίδας κλειδιών η οποία δημιουργείται με κάποια συνάρτηση μιας κατεύθυνσης (π.χ. μια συνάρτηση κατακερματισμού). Για τη δημιουργία αλυσίδας κλειδιών μιας κατεύθυνσης, ο σταθμός βάσης δημιουργεί τυχαία το τελευταίο κλειδί της αλυσίδας K_n και εφαρμόζει επανειλημμένα τη συνάρτηση μιας κατεύθυνσης για τον υπολογισμό των υπολοίπων κλειδιών: $K_i = F(K_{i+1})$.

Ένα παράδειγμα της μεθόδου μTESLA φαίνεται στο Σχήμα 8.2. Κάθε κλειδί της αλυσίδας αντιστοιχεί σε ένα χρονικό διάστημα και όλα τα πακέτα που στέλνονται μέσα σε αυτό αυθεντικοποιούνται με το κλειδί του διαστήματος. Το χρονικό διάστημα αποκάλυψης κλειδιών στο συγκεκριμένο παράδειγμα είναι 2 διαστήματα. Ο κόμβος παραλήπτης είναι συγχρονισμένος με το σταθμό βάσης και γνωρίζει το κλειδί K_0, μια δέσμευση στην αλυσίδα κλειδιών κατακερματισμού. Τα πακέτα P_1, P_2 στέλνονται στο χρονικό διάστημα 1 και περιέχουν ένα MAC με βάση το κλειδί K_1. Το πακέτο P_3 περιέχει ένα MAC με βάση το κλειδί K_2. Μέχρι στιγμής, ένας παραλήπτης δε μπορεί να αυθεντικοποιήσει κανένα πακέτο. Ας υποθέσουμε ότι τα πακέτα P_4, P_5, P_6 χάνονται όπως και το πακέτο που αποκαλύπτει το κλειδί K_1. Έτσι, ο παραλήπτης δε μπορεί να αυθεντικοποιήσει τα πακέτα P_1, P_2, P_3. Στο χρονικό διάστημα 4, ο σταθμός βάσης αποκαλύπτει το κλειδί K_2 το οποίο ο παραλήπτης αυθεντικοποιεί ως $K_0 = F(F(K_2))$, οπότε υπολογίζει επίσης και το $K_1 = F(K_2)$. Τώρα ο παραλήπτης μπορεί να αυθεντικοποιήσει τα πακέτα P_1, P_2 με το κλειδί K_1 και το πακέτο P_3 με το κλειδί K_2.

Έτσι, αντί να εισάγεται ένα κλειδί σε κάθε πακέτο δεδομένων, η αποκάλυψη κλειδιών γίνεται ανά δεδομένα χρονικά διαστήματα και είναι ανεξάρτητη από την αποστολή πακέτων. Στα πλαίσια του μTESLA ο σταθμός βάσης μεταδίδει περιοδικά το τρέχον κλειδί με ένα ειδικό πακέτο.

Για την εφαρμογή αλυσίδων κλειδιών κατακερματισμού υλοποίησαμε στα πλαίσια της Wiselib ένα module που παρέχει τις κρυπτογραφικές λειτουργίες τους το



Σχήμα 8.2: Παράδειγμα Μιας Σχετικής με το Χρόνο Αλυσδίδας Κλειδιών για Αυθεντικοποίηση Αποστολέα.

οποίο αποκαλούμε HKC. Το module αυτό το συνδυάσαμε με το module ομαδοποίησης το οποίο καλείται HDL (Hop Distance Layer) το οποίο οργανώνει το δίκτυο σε κυκλικούς τομείς χωρίς αρχηγούς με βάση την απόστασή τους (μετρημένη σε hops) από το σταθμό βάσης (Σχήμα 8.3). Ο σταθμός βάσης, χρησιμοποιώντας τη συνάρτηση κατακερματισμού SHA-1 δημιουργεί μια αλυσίδα κλειδιών μεγέθους όσο ο αριθμός των τομέων που θέλει να οργανώσει το δίκτυο. Ο σταθμός βάσης, ζητάει αρχικά από τους κόμβους που απέχουν 1 hop από αυτόν να συμμετέχουν στον πρώτο τομέα. Το μήνυμα αυτό έχει ένα MAC με βάση το κλειδί k_1. Οι κόμβοι που το λαμβάνουν εφαρμόζουν τη συνάρτηση κατακερματισμού μια φορά έτσι ώστε να αυθεντικοποιήσουν το σταθμό βάσης και κατόπιν δέχονται να συμμετέχουν στον πρώτο τομέα. Εν συνεχεία, ο σταθμός βάσης δημιουργεί ένα μήνυμα για να ζητήσει από τους κόμβους που απέχουν 2 hops από αυτόν να συμμετάσχουν στο δεύτερο τομέα. Το μήνυμα αυτό περιέχει ένα MAC με βάση το κλειδί k_2. Οι κόμβοι του πρώτου τομέα απλά προωθούν το μήνυμα και οι κόμβοι του δεύτερου τομέα εφαρμόζουν δυο φορές τη συνάρτηση κατακερματισμού ώστε να αυθεντικοποιήσουν το σταθμό βάσης. Στη συνέχεια, δέχονται να συμμετέχουν στον δεύτερο τομέα. Η διαδικασία συνεχίζεται μέχρι το δίκτυο να οργανωθεί στους τομείς που επιθυμεί ο σταθμός βάσης. Στο σημείο αυτό αναφέρουμε ότι για την ασφαλή αρχικοποίηση του πρωτοκόλλου το πρώτο κλειδί της αλυσίδας μπορεί να σταλεί από το σταθμό βάσης με χρήση του αλγορίθμου ψηφιακής υπογραφής ECDSA, δεδομένου ότι οι κόμβοι γνωρίζουν εξάρχής το δημόσιο κλειδί του σταθμού βάσης.

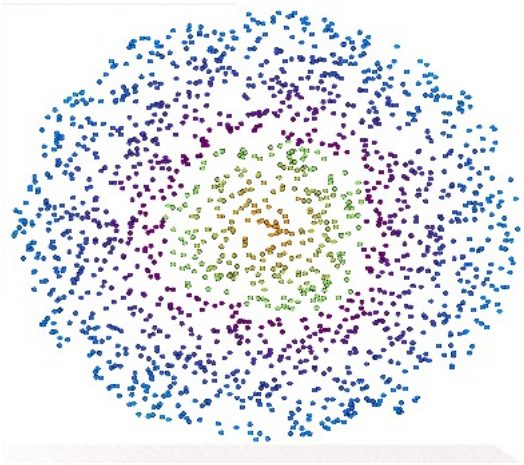
8.3.1 Πειραματικά Αποτελέσματα

Κάνοντας πειραματικές μετρήσεις σε πραγματικές συσκευές της σειράς iSense υπολογίσαμε τους χρόνους εκτέλεσης και την κατανάλωση ενέργειας που απαιτούνται για τις βασικές πράξεις που παρέχει το module HKC δηλαδή: τη δημιουργία του αρχικού κλειδιού, τη δημιουργία μιας αλυσίδας κλειδιών κατακερματισμού καθώς και τον έλεγχο αυθεντικότητας ενός κλειδιού. Τα αποτελέσματα φαίνονται στον Πίνακα 8.3.

Πράξη	Χρόνος	Ενέργεια
Δημιουργία Αρχικού Κλειδιού (160-bit)	4 msec	0.15 mJ
Δημιουργία Αλυσίδας Κλειδιών (Μεγέθους 100)	0.6 sec	22.8 mJ
Έλεγχος Αυθεντικότητας Κλειδιού	7 msec	0.26 mJ

Πίνακας 8.3: Χρόνος Εκτέλεσης και Κατανάλωση Ενέργειας των Πράξεων που Παρέχει το Module HKC στις Συσκευές iSense.

Τέλος, το μέγεθος του μεταφρασμένου κώδικα του module που υλοποιεί τις



Σχήμα 8.3: Η Οργάνωση Ενός Δικτύου Μετά την Εκτέλεση του Module Hop Distance Layer (HDL).

αλυσίδες κλειδιών κατακερματισμού με βάση τον αλγόριθμο SHA-1 μετρήθηκε στα 2856 bytes για τις συσκευές iSense.

Κεφάλαιο 9

Συμπεράσματα και Μελλοντική Δουλειά

Στη διπλωματική εργασία αυτή, αναπτύξαμε κρυπτογραφικούς αλγορίθμους για ετερογενή δίκτυα αισθητήρων στο προγραμματιστικό περιβάλλον που παρέχει η γενική βιβλιοθήκη αλγορίθμων για ασύρματα δίκτυα αισθητήρων Wiselib. Η Wiselib [3] είναι υλοποιημένη σε C++ και με τη χρήση τεχνικών όπως τα πρότυπα και οι inline συναρτήσεις, επιτρέπει τη συγγραφή γενικού κώδικα ο οποίος αναλύεται και δεσμεύεται κατά τη διαδικασία μεταγλώτισσης χωρίς να δημιουργεί πλεονασμό μνήμης ή υπολογισμού. Η κρυπτογραφική βιβλιοθήκη που αναπτύξαμε παρέχει αλγορίθμους συμμετρικής και ασυμμετρικής κρυπτογραφίας λόγω της συμπληρωματικότητας των πλεονεκτημάτων τους. Επειδή οι συσκευές που απαρτίζουν τα δίκτυα αισθητήρων συνήθως παρέχουν περιορισμένους πόρους (υπολογιστικής ισχύς, μνήμης, ενέργειας), χρησιμοποιήσαμε το σύστημα ελλειπτικών καμπυλών για την υλοποίηση κρυπτογραφίας δημοσίου κλειδιού. Οι ελλειπτικές καμπύλες αποτελούν ιδανικό σύστημα για υλοποίηση κρυπτογραφίας δημοσίου κλειδιού σε ενσωματωμένα περιβάλλοντα, διότι προσφέρουν το ίδιο επίπεδο ασφάλειας με άλλα συστήματα (π.χ. RSA) χρησιμοποιώντας μικρότερα μεγέθη κλειδιών. Έτσι, καταλαβαίνει κανείς ότι λόγω του γεγονότος αυτού οι υλοποιήσεις αλγορίθμων με ελλειπτικές καμπύλες εξοικονομούν υπολογιστική ισχύ, μνήμη και κατόπιν ενέργεια που είναι περιορισμένες στις συσκευές που αποτελούν τα δίκτυα αισθητήρων.

Οι αλγόριθμοι που παρέχει η κρυπτογραφική μας βιβλιοθήκη είναι οι εξής: ο αλγόριθμος συμμετρικής κρυπτογράφησης AES, ο αλγόριθμος κατακερματισμού SHA-1, το σχήμα συμφωνίας κλειδιών Diffie Hellman (ECDH), ο αλγόριθμος ασυμμετρικής κρυπτογράφησης ECIES και το σχήμα ψηφιακής υπογραφής ECDSA. Για να επαληθεύσουμε την ορθότητα και τη γενικότητα της κρυπτογραφικής μας βιβλιοθήκης διεξάγουμε μια εκτενή πειραματική αξιολόγηση (χρόνος, ενέργεια, μέγεθος μεταφρασμένου κώδικα) των παραπάνω αλγορίθμων σε δυο συσκευές (iSense, TelosB) διαφορετικών δυνατοτήτων (ισχύς, μνήμης) που εκτελούν διαφορετικά λειτουργικά συστήματα (iSense OS, Contiki Sky). Στη συνέχεια, για να αποδείξουμε την ευκολία χρήσης της βιβλιοθήκης μας παρουσιάζουμε τρεις εφαρμογές οι οποίες συνδυάζουν τους αλγόριθμους κρυπτογράφησης με αλγόριθμους δρομολόγησης και ομαδοποίησης που μας παρέχει η βιβλιοθήκη Wiselib.

Όσον αφορά στη μελλοντική εργασία, σκοπεύουμε να χρησιμοποιήσουμε την κρυπτογραφική μας βιβλιοθήκη και ειδικότερα τις υλοποιήσεις ελλειπτικών καμπυ-

λών για την ανάπτυξη νέων κρυπτογραφικών πρωτοκόλλων όπως π.χ. αποδείξεις μηδενικής γνώσης [7]. Οι αποδείξεις μηδενικής γνώσης, είναι πρωτόκολλα που χρησιμοποιούνται για την προστασία της ιδιωτικότητας. Εκτελούνται μεταξύ δύο οντοτήτων: ενός αποδεικνύων και ενός επαληθευτή και κατά την εκτέλεσή τους, ο αποδεικνύων μπορεί να πείσει τον επαληθευτή για την κατοχή ευαίσθητης πληροφορίας χωρίς όμως να του την αποκαλύπτει. Τέλος, επεκτείνοντας τέτοιου είδους πρωτόκολλα ώστε να αποδεικνύουν ακόμα πιο σύνθετες ιδιότητες (όπως π.χ. ότι η τιμή που προστατεύω είναι μεγαλύτερη από κάποια άλλη τιμή) στοχεύουμε να υλοποιήσουμε πιστοποιητικά με βάση τα γνωρίσματα [5], [4], σε συσκευές περιορισμένων πόρων.

Βιβλιογραφία

[1] D. Amaxilatis, I. Chatzigiannakis, C. Koninis, and A. Pyrgelis. Component based clustering in wireless sensor networks. Computing Research Repository (CORR) abs/1105.3864, ACM, October 2010.

[2] K. Bairaktaris, I. Chatzigiannakis, V. Liagkou, and P. Spirakis. Adaptive probabilistic secure routing in mobile wireless sensor networks. In *16th International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2008)*, pages 208–212. IEEE, IEEE, September 2008.

[3] T. Baumgartner, I. Chatzigiannakis, S. P. Fekete, C. Koninis, A. Kröller, and A. Pyrgelis. Wiselib: A generic algorithm library for heterogeneous sensor networks. In *EWSN*, pages 162–177, 2010.

[4] S. A. Brands. *Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy.* MIT Press, Cambridge, MA, USA, 2000.

[5] J. Camenisch and E. Van Herreweghen. Design and implementation of the idemix anonymous credential system. In *Proceedings of the 9th ACM conference on Computer and communications security*, CCS '02, pages 21–30, New York, NY, USA, 2002. ACM.

[6] I. Chatzigiannakis, E. Konstantinou, V. Liagkou, and P. G. Spirakis. Design, analysis and performance evaluation of group key establishment in wireless sensor networks. *Electr. Notes Theor. Comput. Sci.*, 171(1):17–31, 2007.

[7] I. Chatzigiannakis, A. Pyrgelis, P. G. Spirakis, and Y. C. Stamatiou. Elliptic curve based zero knowledge proofs and their applicability on resource constrained devices. In *MASS*, pages 715–720. IEEE, 2011.

[8] Crypto++ Library 5.6.1. http://www.cryptopp.com/.

[9] M. W. David J. Malan and M. D. Smith. Implementing public-key infrastructure for sensor networks. *TOSN*, 4(4), 2008.

[10] T. Dimitriou. Securing communication trees in sensor networks. In *ALGOSENSORS*, pages 47–58, 2006.

[11] D. Eastlake and P. Jones. US Secure Hash Algorithm 1 (SHA1). Technical Report 3174, Sept. 2001.

[12] GNUMP: GNU Multiple Precision Arithmetic Library. http://gmplib.org/.

[13] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz. Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs. pages 119–132. 2004.

[14] E. Konstantinou, Y. Stamatiou, and C. Zaroliagis. A software library for elliptic curve cryptography. In R. Möhring and R. Raman, editors, *Algorithms — ESA 2002*, volume 2461 of *Lecture Notes in Computer Science*, pages 101–130. Springer Berlin / Heidelberg, 2002. 10.1007/3-540-45749-6-55.

[15] A. Kröller, D. Pfisterer, C. Buschmann, S. P. Fekete, and S. Fischer. Shawn: A new approach to simulating wireless sensor networks. *CoRR*, abs/cs/0502003, 2005.

[16] A. Liu and P. Ning. Tinyecc: A configurable library for elliptic curve cryptography in wireless sensor networks. In *IPSN*, pages 245–256, 2008.

[17] P. Papaioannou, P. E. Nastou, Y. C. Stamatiou, and C. D. Zaroliagis. Secure elliptic curve generation and key establishment on a 802.11 wlan embedded device. In *ISADS*, pages 41–48, 2009.

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

[19] Standards for Efficient Cryptography Group. http://www.secg.org/.

[20] M. Steiner, G. Tsudik, and M. Waidner. Diffie-hellman key distribution extended to group communication. In *ACM Conference on Computer and Communications Security*, pages 31–37, 1996.

[21] A. S. Wander, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz. Energy analysis of public-key cryptography for wireless sensor networks. In *Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications*, pages 324–328, Washington, DC, USA, 2005. IEEE Computer Society.