Ενσωμάτωση ασφαλούς επικοινωνίας σε ασύρματους κόμβους με χρήση του MQTT

Στο διαδίκτυο των πραγμάτων IoT (Internet of Things) υπάρχει μεγάλη ανάγκη για την για την ασφάλεια μεταφοράς δεδομένων. Το πρότυπο IEEE 802.15.4 [1] για το στρώμα ζεύξης δεδομένων (Link Layer) που χρησιμοποιούν πολλοί ασύρματοι αισθητήρες WSN (Wireless Sensor Nodes), προβλέπει τη δυνατότητα διαύλων προστατευμένων με κρυπτογράφηση χρησιμοποιώντας το AES (Advanced Encryption Standard) σε διάφορες παραλλαγές του [2], όπως το CBC (Cipher Block Chaining) [3] ή CCM* (Counter with CBC-MAC) [4]. Εν τούτοις δεδομένου ότι η επικοινωνία μεταξύ των κόμβων δεν είναι συνεχής, η προστασία των μηνυμάτων (message security) μπορεί να είναι περισσότερο κατάλληλη από την προστασία των καναλιών (channel security).

Στο περιβάλλον IoT, η οικονομία πόρων στη μεταφορά μηνυμάτων (μετρήσεων) από τους αισθητήρες είναι κύριας σημασίας. Για αυτό το λόγο χρησιμοποιούνται «Ουρές Μηνυμάτων» (Message Queues), όπου οι αισθητήρες αντί να στέλνουν τιμές στους ενδιαφερόμενους αποδέκτες τις στέλνουν σε *ενδιάμεσους*, οι οποίοι φροντίζουν να τις αποστείλουν στους τελικούς αποδέκτες. Μια διαδεδομένη υλοποίηση είναι το MQTT (Message Queue Telemetry Protocol) [5]. Το MQTT είναι ένα *ελαφρύ* πρωτόκολλο μετάδοσης μηνυμάτων πάνω από το TCP/IP, με τη χρήση ενός σχήματος δημοσίευσης/εγγραφής (publish/subscribe), όπου ο *ενδιάμεσος* (Broker) μπορεί να μοιράζει τις μετρήσεις σε περισσότερους από ένα αποδέκτες. Η παραλλαγή του, το MQTT-SN [6] είναι μια τροποποίηση για sensor nodes που χρησιμοποιεί πρωτόκολλο μεταφοράς UDP αντί το TCP και χρησιμοποιείται ευρέως σε IoT που υλοποιείται με WSN. Το Eclipse Mosquitto [7] είναι μια υλοποίηση ανοιχτού κώδικα για MQTT και MQTT-SN εξυπηρετητές και gateways.

Το Contiki OS [8] είναι ένα ανοικτού κώδικα λειτουργικό σύστημα για περιβάλλοντα IoT, με μικρές απαιτήσεις μνήμης, power management και soft real-time. Μαζί με το Contiki προσφέρεται ο Cooja Network Simulator [9], που επιτρέπει στους χρήστες να κάνουν εξομοιώσεις σε μεγάλη κλίμακα και πλήρως εξομοιούμενες συσκευές.

Σκοπός της διπλωματικής είναι να χρησιμοποιηθεί ο Cooja simulator για το λειτουργικό σύστημα contiki ώστε να γίνουν δοκιμές υλοποίησης AES με κρυπτογράφηση δεδομένων AES-CBC, AES-OCB ενεργοποιώντας Link layer encryption με AES-CCM και εναλλακτικά με κρυπτογράφηση μηνύματος στην ουρά μηνυμάτων (Message-Queue) είτε με χρήση MQTT [10] είτε με χρήση MQTT-SN [11].

Σχετικοί σύνδεσμοι:
[1] https://en.wikipedia.org/wiki/IEEE_802.15.4
[2] http://www.libelium.com/security-802-15-4-zigbee/
[3] https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation
[4] https://en.wikipedia.org/wiki/CCM_mode
[5] http://mqtt.org/
[6] http://mqtt.org/2013/12/mqtt-for-sensor-networks-mqtt-sn
[7] https://projects.eclipse.org/projects/technology.mosquitto
[8] http://www.contiki-os.org/
[9] https://github.com/contiki-os/contiki/wiki/An-Introduction-to-Cooja
[10] https://github.com/esar/contiki-mqtt
[11] https://github.com/adamrenner/mqtt-sn-tools-contiki

Επικοινωνία: Ε. Δ. Συκάς (sykas@cn.ntua.gr), Δ. Καλογεράς (dkalo@noc.ntua.gr)