

# .Sense, A Secure Framework for Sensor Network Data Acquisition, Monitoring and Command

M. Salajegheh, H. Soroush, A. Thomos, T. Dimitriou, I. Krontiris  
Algorithms and Security Lab  
Athens Information Technology  
{msal, hsor, ntom, tdim, ikro}@ait.edu.gr

## ABSTRACT

We present .Sense, an end-to-end security framework for sensor network data acquisition, monitoring and command. In order to provide security service inside the sensor network two security protocols are implemented. The first is a key establishment algorithm in which sensor nodes agree on common keys to use for securing communications among them. The second is a scheme in which the base station can issue commands in authenticated manner to the network. We are also using typical security schemes such as SSL to connect the end-users to the system.

## Categories and Subject Descriptors

C.2.0 [Computer-Communication Networks]: General – Security and protection C.2.1 [Computer-Communication Networks]: Network Architecture and Design – Distributed networks.

## General Terms

Algorithms, Design, Security.

## Keywords

Sensor Networks, Security Mechanism, Fault-tolerance.

## 1. INTRODUCTION

As sensor networks are usually deployed in hostile environments, many of their applications require that data must be exchanged in a secure and authenticated manner. However, the threat to a sensor network is different from the threat to a mobile ad-hoc network (or other types of networks). Existing network security mechanisms, including those developed for mobile ad-hoc networks, are a poor fit for this domain. Establishing secure communications between sensor nodes becomes a challenging task, given their limited processing power, storage, bandwidth and energy resources. Research into authentication and confidentiality mechanisms designed specifically for sensor data and network control protocols is needed [1].

Our main contribution is .Sense, a solution that aims to provide an end-to-end security service in a sensor network data acquisition

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.  
REALWSN'06, June 19, 2006, Uppsala, Sweden.  
Copyright 2006 ACM 1-59593-431-6/06/0006...\$5.00.

and command application. To the best of our knowledge no other work has attempted to provide end-to-end security in a sensor network application. We believe that the results of this on-going work will be certainly useful for the designers of similar sensor network applications who keep an eye on end-to-end security. A secondary contribution of this work is a succinct overview of the techniques that aim to secure sensor networks and provide for overall sensor network operations and efficiency. Given the fact that little prior work exists in this respect, there is a need both to identify the problems and challenges and propose solution techniques.

## 2. Design Goals and System Architecture

.Sense is a distributed system which acts as a tool for sensor network data acquisition and command providing a transparent end-to-end security service. The goal is also to be simple and user-friendly enough to be used by non-advanced users. Overall, .Sense was designed with the following characteristics in mind:

*Security*: The system should provide data integrity, confidentiality and authentication. *Fault-tolerance*: The system should handle potential faults of its different components neatly. *Distributed access to sensed information*: The system should allow concurrent access to sensed data according to user privileges. *User friendliness*: The system should be easy to use for non-advanced users. *Scalability*: The design should scale to a huge number of nodes and bear network changes (additions or deletions of nodes).

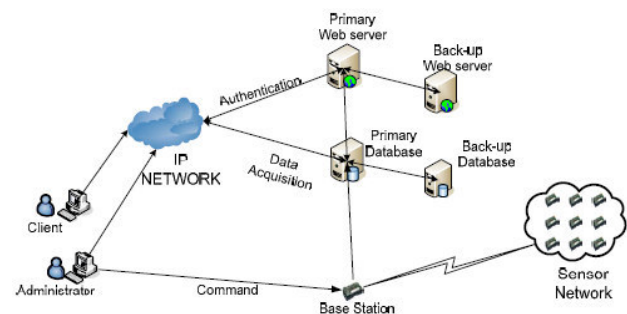


Figure 1: General architecture of .Sense

A simple high level scenario of what happens in the system environment is as follows (Figure 1). Mica2 motes collect data at a given sample rate. This data can be light intensity, sound volume, temperature, etc. In addition to environmental data, WSN meta-data such as mote's power level, signal strength and number of neighbors can be sent to the base station. The base station will store sensed data and meta-data in a replicated external database. Authorized users can access some part of the data according to

their privileges. These users will interact with the system through a user-friendly Java Applet. The administrator of the system, on the other hand, can also send different mote control commands such as radio power or sampling rate adjustments as well as issuing start or stop sensing commands in a separate stand alone java application.

Web servers are mainly used through the process of end-user authentication. The authenticated user will run a Java Applet on its browser which is downloaded from the web server to its local machine. The applet then, creates a secure connection to the database in order to represent data appropriately.

### 3. FEATURES OF .Sense

**Security** In order to provide an end-to-end security service, .Sense has to first provide security for communications happening in components of the system which are outside of the sensor network. This is done using standard SSL connections. These secure connections are present in the user's GUI to the database communication, as well as the communication between the base station and the database and even the master/slave databases.

More challenging is to provide security *inside* the sensor network. For this, two security protocols have been implemented. The first one is a key establishment protocol in which sensor nodes agree on common keys to use for securing communications among them. This protocol allows for support of data with various sensitivity levels. For example, our protocol supports the establishment of three types of keys for each sensor: a key that can be used for communication with the base station, pairwise keys that can be used to communicate with immediate neighbors and group keys that allow for secure *in-network* processing. Such separation allows for efficient resource management that is essential for wireless sensor networks. The second protocol is a simple scheme in which the base station can issue commands in authenticated manner to the network.

While there are many types of key establishment protocols in the literature, (see for example [2,3]), we make use of a global key  $K$  that is used once to dynamically derive various types of keys, after which the global key is *deleted* [4, 5] (details omitted).

The various primitives (node discovery, encrypt/decrypt operation, etc.) have been implemented in a system *library* which can be used to secure *any* sensor network application in a straightforward manner. We believe this is another important contribution of our work.

**Failure handling** This feature happens at two major components of the .Sense system:

*Web servers:* In order to have a fault tolerant web server, we provide primary/secondary web servers. Both primary and secondary web servers receive the login command of users. The primary one sends the user information to the database in order to authenticate him/her. After authentication takes place, the primary sends its log file of the specific client to the secondary web server. If the primary fails, the client program retries the other web server.

*Databases:* .Sense uses MySQL 5.0 as its databases. We use MySQL master/slave configuration in order to have data replication so that whenever the master database is down for any reason, data gathered by the sensor network remains accessible via the slave database.

**Scalability** The design should scale to hundreds or thousands of sensor nodes. Furthermore, one must pay sufficient attention to the impact of topology changes inside the sensor network when designing protocols or implementing applications for sensor networks. Even in the case that the motes are not mobile, they might be added or removed from the sensor network. This should be taken into account when designing practical key establishment protocols like the ones we have implemented in .Sense.

**Current State of the System** The administrator application is implemented to support various commands (start/stop sensing, turn on/off, number of neighbors, query for data, etc.)

All commands are sent from the base station to the desirable motes in a *secure* and *authenticated* manner. The routing protocol to send command messages is flooding using an edited version of TinyOS *Bcast* library (tos/lib/Bcast). To route messages we use TinyOS *MintRoute* (tos/lib/MintRoute). *MintRoute* bases its routing decisions mainly on link-quality estimates rather than minimum hop count. The link quality is used to select a parent that minimizes the expected number of transmissions to reach the base station. In addition to environmental data, battery level, number of neighbors and the parent of each mote is stored in the database which can be used for monitoring network health. At this point database configuration and replication is completed. User authentication and interface is also completed, but some enhancements are needed to visualize the WSN data and meta-data.

### 4. Conclusions and Future Work

.Sense is a protocol suite for sensor networks with an emphasis on fault-tolerance and end-to-end security. To the best of our knowledge no other work in the past has attempted to combine all these different technologies to build secure sensor network applications. When .Sense is complete, we believe the lessons learned from our experience can be valuable to other researchers as well.

We intend to improve .Sense by enhancing some of the key features previously described. For example, the basic scheme used for authenticating base stations commands has to be extended to build a *broadcast* tree in a secure manner.

Another improvement is to make the base station fault tolerant by having a primary-backup approach. Yet, several issues need to be addressed here, as for example how to integrate this feature with the security characteristics mentioned before.

### 5. REFERENCES

- [1] H. Chan, A. Perrig, Security and privacy in sensor networks, *IEEE Computer*, Volume 36, Issue 10, Oct. 2003.
- [2] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in Proceedings of the 9th ACM conference on Computer and communications security, pp. 41– 47, 2002.
- [3] H.Chan, A.Perrig, and D.Song, "Random key predistribution schemes for sensor networks," in IEEE Symposium on Security and Privacy, pp. 197–213, May 2003.
- [4] Tassos Dimitriou, I. Krontiris, "A Localized, Distributed Protocol for Secure Information Exchange in Sensor Networks", 5th IEEE WMAN, 2005.
- [5] Tassos Dimitriou, D. Foteinakis. "Secure In-Network Processing in Sensor Networks", IEEE BASENETS, 2004.