

# Poster Abstract: LEAP – Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks

Sencun Zhu  
Center for Secure Information  
Systems  
George Mason University  
Fairfax, VA 22030  
szhu1@gmu.edu

Sanjeev Setia  
Department of Computer  
Science  
George Mason University  
Fairfax, VA 22030  
setia@gmu.edu

Sushil Jajodia  
Center for Secure Information  
Systems  
George Mason University  
Fairfax, VA 22030  
jajodia@gmu.edu

## ABSTRACT

In this paper, we describe LEAP (Localized Encryption and Authentication Protocol), a key management protocol for sensor networks that is designed to support *in-network processing* techniques such as passive participation. LEAP includes support for multiple symmetric keying mechanisms including individual keys, pairwise shared keys, cluster keys, and a group key. This design is based on the observation that different types of messages exchanged between sensor nodes have different security requirements, and a single keying mechanism is not suitable for meeting these different security requirements.

## Categories and Subject Descriptors

C.2.0 [Computer-Communication Networks]: General—*Security and protection*

## General Terms

Design, Security

## Keywords

Security Mechanism, Sensor Networks, Key Management, In-network Processing

## 1. INTRODUCTION

Many sensor systems are deployed in unattended and often adversarial environments. Hence, security mechanisms that provide confidentiality and authentication are critical for the operation of many sensor applications. Providing security is particularly challenging in sensor networks due to the resource limitations of sensor nodes. Thus, key management protocols for sensor networks are based upon symmetric key algorithms.

A fundamental issue that must be addressed for using key management protocols based on symmetric shared keys is the mechanism used for establishing the shared keys in the first place. The constrained energy budgets and the limited computational and communication capacities of sensor nodes make protocols such as TLS and Kerberos developed

for wired networks impractical for use in large-scale sensor networks. At present, the most practical approach for bootstrapping secret keys in sensor networks is to use pre-deployed keying in which keys are loaded into sensor nodes before they are deployed.

A unique issue that arises in sensor networks that needs to be considered while selecting a key sharing approach is its impact on the effectiveness of in-network processing and passive participation. In many applications, sensors in the network are organized into a data fusion or aggregation hierarchy for efficiency. Readings or messages from several sensors are processed at a data fusion node and aggregated into a more compact report before being relayed to the parent node in the data fusion hierarchy. Passive participation is another form of in-network processing in which a sensor node can take certain actions based on overheard messages, e.g., a sensor can decide to not report an event if it overhears a neighboring node reporting the same event.

Particular keying mechanisms may preclude or reduce the effectiveness of in-network processing. To support passive participation, it is essential for intermediate nodes be able to decrypt and authenticate an encrypted message exchanged between two sensor nodes. Thus, passive participation of encrypted or authenticated messages is only possible if multiple nodes share the keys used for encryption and authentication. On the other hand, if a pairwise shared key is used for encrypting or authenticating a message, it effectively precludes passive participation in the sensor network.

In this work, we describe the design of LEAP (Localized Encryption and Authentication Protocol), a key management protocol for sensor networks that is designed to support *in-network processing*. In addition, LEAP includes support for multiple keying mechanisms. The design of the protocol is motivated by the observation that different types of messages exchanged between sensor nodes have different security requirements, and that a single keying mechanism is not suitable for meeting these different security requirements. More specifically, LEAP supports the establishment of the following four types of keys for each sensor node.

**Individual Key** Every node has a unique key that it shares pairwise with the base station. This key is used for secure communication between them. For example, a node may send an alert to the base station if it observes any abnormal or unexpected behavior by a neighboring node, or the base station can use this key to encrypt keying material and then send it to an individual node.

**Group Key** This is a globally shared key that is used by the base station for encrypting messages that are broadcast to the whole group. For example, the base station issues missions, sends queries and interests. Note that from the confidentiality point of view there is no advantage to separately encrypting a message transmitted to all the nodes using the individual key of each node. However, since the group key is shared among all the nodes in the network, an efficient rekeying mechanism is necessary for updating this key after a compromised node is revoked.

**Cluster Key** A cluster key is a key shared by a node and all its neighbors, and it is mainly used for securing locally broadcast messages, e.g., routing control information, or securing sensor messages which can benefit from passive participation. For example, a node that overhears a neighboring sensor node transmitting the same reading as its own current reading can elect to not transmit the same. In responding to aggregation operations such as MAX, a node can also suppress its own reading if its reading is not larger than an overheard one. For passive participation to be feasible, neighboring nodes should be able to decrypt and authenticate some classes of messages, e.g., sensor readings, transmitted by their neighbors. This means that such messages should be encrypted or authenticated by a *locally shared* key. Therefore, in LEAP each node possesses a unique cluster key that it uses for securing its messages, while its immediate neighbors use the same key for decryption or authentication of its messages.

**Pairwise Shared Key** Every node shares a pairwise key with each of its immediate neighbors. In LEAP, pairwise keys are used for securing communications that require privacy or source authentication. For example, a node can use its pairwise keys to secure the distribution of its cluster key to its neighbors, or to secure the transmissions of its sensor readings to an aggregation node. Note that the use of pairwise keys precludes passive participation.

**Inter-node Traffic Authentication** LEAP also includes an efficient protocol for inter-node traffic authentication based on the use of one-way key chains. Traffic authentication is essential for defending against denial-of-service attacks launched by outsider adversaries. A salient feature of the authentication protocol is that it supports source authentication without preventing in-network processing and passive participation.

## 2. PROTOCOL DESIGN

In this section, we discuss our techniques for key establishment and for inter-node traffic authentication.

**Establishing Individual Keys** An individual key is generated and pre-loaded into each node prior to its deployment.

**Establishing Pairwise Keys** Our approach for establishing pairwise keys between neighboring sensor nodes exploits the special property of sensor networks consisting of stationary nodes that the set of neighbors of a node is relatively static, and that a sensor node that is being added to the network will discover most of its neighbors at the time of its initial deployment. Second, it is our belief that a sensor node deployed in a security critical environment must be designed to sustain possible break-in attacks at least for a short interval (say several seconds) when captured by the adversary. Therefore, we assume there exists a lower bound on the time interval  $T_{min}$  that is necessary for an adversary

to compromise a sensor node. Given these assumptions, in our scheme a node can establish pairwise keys with its neighbors very efficiently.

**Establishing Cluster Keys** The cluster key establishment phase follows the pairwise key establishment phase. A node first generates its cluster key, and then transmits it to each of its neighbors, encrypted with the pairwise key shared with that neighbor respectively.

**Establishing Group Key** The base station generates a group key, then securely transmits it to all its neighboring nodes, encrypted with its cluster key. Each node that receives the group key forwards the key to its own neighbors encrypting the group key with its own cluster key. Thus, the group key is propagated in the network in a hop-by-hop fashion. To prevent a compromised node from broadcasting a group key while impersonating the base station, we use  $\mu$ TESLA [3] for authenticating the group key.

**Inter-node Traffic Authentication** Our scheme for traffic authentication combines the use of cluster keys with one-way key chains, i.e., each cluster key used for authentication is part of a one-way key chain. In this scheme, a node generates a one-way key chain [1] and uses each key in the key chain for authenticating only one packet. Our scheme can provide source authentication similar to that provided by pairwise keys, but without precluding passive participation.

## 3. CONTRIBUTIONS

In this work, we proposed multiple keying mechanisms for supporting the differing security requirements of various sensor applications. The protocols used for establishing these keys for each node are communication- and energy-efficient, and minimize the involvement of the base station. Our scheme supports important in-network processing mechanisms such as passive participation and localizes the impact of node compromise. It can prevent or mitigate many of the attacks on sensor networks discussed by Karlof and Wagner [2]. We refer the reader to [4] for a more detailed analysis of the performance and security of our scheme.

## 4. REFERENCES

- [1] L. Lamport. Password authentication with insecure communication. *Communications of the ACM*, 24(11):770-772, Nov., 1981.
- [2] C. Karlof and D. Wagner. Secure Routing in Sensor Networks: Attacks and Countermeasures. In *Proc. of First IEEE Workshop on Sensor Network Protocols and Applications*, May 2003.
- [3] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. Tygar. SPINS: Security Protocols for Sensor Networks. In *Proc. of Mobicom 2001*, July 2001.
- [4] S. Zhu, S. Setia and S. Jajodia. LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks. In *Proc. of the 10th ACM Conference on Computer and Communications Security (CCS '03)*, Washington D.C., October, 2003.