LEAP: EFFICIENT SECURITY MECHANISMS FOR LARGE-SCALE DISTRIBUTED SENSOR NETWORKS

· In *ACM Transaction on Sensor Networks (TOSN)*, Nov. 2006

Presented by Huang Lu
Advisor: Dr. Jie Li
OSDP Lab, Dept. of Computer Science,
Graduate School of SIE, U. of Tsukuba
October 25th, 2010
OUTLINE

- Introduction
- Security mechanism and approaches for WSNs
- LEAP protocol
- Key management
**Introduction**

- WSN devices have severe resource constraints in terms of energy, computation and memory.

- Many Security-critical applications that depend on key management processes, demand a high level of fault tolerance when a node is compromised.
WSN Applications

- Battle field and homeland security
  - Enemy movement (tanks, soldiers, terrorists etc)

- Environmental monitoring
  - Habitat monitoring
  - Early bush fire detection

- Hospital tracking systems
  - Tracking patients, doctors, drug administrators

- Traffic congestions monitoring
  - Traffic flow and jams
SECURITY MECHANISM FOR WSNs

Reference:

Huang LU, OSDP Lab, U. of Tsukuba – Oct. 2010
Security and Operational Requirements for Key Management

- **Confidentiality** - Nodes should not reveal data to any unintended recipients.
- **Integrity** - Data should not be changed between transmissions due to environment or malicious activity.
- **Authentication** - Data used in decision making process should originate from correct source.
- **Availability** - Network should not fail frequently.
- **Robustness** - When some nodes are compromised the entire network should not be compromised.
- **Self-organization** - Nodes should be flexible enough to be self-organizing (autonomous) and self-healing (failure tolerant)

Huang LU, OSDP Lab, U. of Tsukuba – Oct. 2010
CONTINUED.

- **Data Freshness** - Old data should not be used as new.
- **Time Synchronization** - The protocols in nodes should be manipulated simultaneously to produce incorrect data.
- **Secure Localization** - Nodes should be able to accurately and securely acquire location information.
- **Accessibility** - Intermediate nodes should be able to perform data aggregation by combining data from different nodes.
- **Flexibility** - Nodes should be replaceable when compromised.
- **Scalability** - WSN should concurrently support thousands of nodes even with key management in place.
CURRENT APPROACHES

- [Key management], by Eschenauer et al. in ACM CCS’02.
- [SPINS], by Perrig et al. in Wireless Networks Journal (WINE), 2002.
- [Random Key Assignment], by pietro et al. in ACM SASN’03.
- [Establishing Pair-wise Keys], by Liu et al. in ACM CCS’03.
- [Pair-wise Key Pre-distribution], by Du et al. in ACM CCS’03.
- [Random Key Predistribution], by Chan et al. in IEEE S&P’03
- [Deployment knowledge], by Du et al. in IEEE INFOCOM’4.
- [TinySec], by Karlof et al, in IEEE SenSys’04
- [LEAP], by Zhu et al. in ACM Transaction on Sensor Networks (TOSN). 2006.
- [Message Authentication], by Zhang et al. in IEEE INFOCOM’08

......
......

Huang LU, OSDP Lab, U. of Tsukuba – Oct. 2010
LEAP

LEAP: Localized Encryption and Authentication Protocol

Motivation:
- Observation that different types of messages exchanged between sensor nodes have different security requirements
- A single keying mechanism is not suitable for meeting these different requirements.
- Asymmetric keys cost too much computational time

A KEY management protocol for sensor networks

Four types of keys for each sensor node
LEAP FEATURES

- The establishing and updating part of the protocol is communication and energy-efficient and minimizes the involvement of the BS (base station).

- The authentication part of the protocol supports source authentication without precluding in-network processing.

- The inter-node traffic authentication is based on the use of one-way key chains.

- Assumes the existence of a group (global) broadcast authentication protocol. e.g., μTESLA [Perrig et al. 2001].
Different Security Requirements

- The packets can be classified into several categories based on different criteria
  - Control packets vs. Data packets
  - Queries or commands vs. Sensor Readings

- Authentication is required for all types of packets whereas confidentiality may only be required for some type of packets.
  - Routing control info (does not require confidentiality)
  - Readings transmitted by a sensor node and queries sent by the BS need to be confidential
**FOUR TYPES OF KEYS**

- **Individual key:** shared with BS, used for secure communications.
  - e.g.: 1. *abnormal or unexpected event* report; 2. encryption of *sensitive data* that are broadcast to the whole group (by BS).
FOUR TYPES OF KEYS

- **Group Key:** (or Global Key), used by BS for encryption of broadcast.
  
  e.g.: 1. the BS issues the missions, send queries and interests; 2. useful to update new key after one of them is compromised.
**Four Types of Keys**

- **Cluster Key**: shared by a node and all its neighbors, used for securing locally broadcast messages.
  - e.g.: 1. routing control info 2. benefit from the passive participation
**FOUR TYPES OF KEYS**

- **Pair-wise Shared Key**: shared with its immediate neighbors
e.g.: transfer/distribute its cluster keys to its neighbors.

![Diagram of network with CH1, CH2, CH3, CH4, CH5, CH6 nodes and BS (Base Station)]
## Notations

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N )</td>
<td>The number of nodes in the network</td>
</tr>
<tr>
<td>( u, v )</td>
<td>Principals such as communicating nodes</td>
</tr>
<tr>
<td>( {f_k} )</td>
<td>A family of pseudo-random function</td>
</tr>
<tr>
<td>( {s}_k )</td>
<td>Encrypting message with a key ( k )</td>
</tr>
<tr>
<td>( MAC(k, s) )</td>
<td>The message authentication code of a message using a symmetric key ( k )</td>
</tr>
</tbody>
</table>
**Establishing Individual Node Keys**

- $K_u$ is the individual key for a node $u$
- $f$ is a pseudo-random function
- $K_s$ is a master key only known to the controller

Then we have

$$K_u = f_{K_s}(u)$$

If the BS want to communicate with $u$, it computes $K_u$ on the fly.
4 Steps to establish pair-wise key

- Step 1: Key pre-distribution: the controller generates an initial key $K_I$ and loads each node with this key. Each sensor node $u$ drives a master key $K_u = f_{K_I}(u)$
Establishing Pair-wise Shared Keys

- Step 2: when a node $u$ is deployed, it first initialize a timer after time $T_{\text{min}}$ (using for example, $\mu$TESLA).

It tries to discover its neighbors. It broadcast $HELLO$ messages and wait for neighbor nodes’ responses.

$u \rightarrow * : u, \text{Nonce } u$

$v \rightarrow u : v, \text{MAC (}K_v, \text{Nonce } U | v\text{)}$

$u$ then stores all $K_v = f_{K_l}(v)$

- $T_{\text{min}}$: is the time necessary for an adversary to compromise a sensor node.
- LEAP assumes that $T_{\text{min}}$ is larger than the maximum time to finish the key distribution
Establishing Pair-wise Shared Keys

- **Step 3**: since $K_v$ was got, compute $K_{uv} = f_{Kv}(u)$

- **Step 4**: when its timer expires, node $u$ erases $K_I$ and all the keys $K_v$ it computed in the neighbor discovery phase.

Huang LU, OSDP Lab, U. of Tsukuba – Oct. 2010
Establishing Cluster Keys

Suppose node u want to create a cluster key

Algorithm:

- Node $u$ generates a random key $K_u^c$
- For all immediate neighbor

$$u \rightarrow v_i : (K_u^c)_{K_{uv_i}}$$

* If a neighborhood node is revoked, node $u$ generates a new cluster key and transmits it to all the remaining neighbors in the same way.
Establishing Group Keys

- First approach:
  distribute a message $M$ securely to all the nodes using hop-by-hop transmission. The BS encrypts $M$ with its cluster key and then broadcast the message.

Drawback:
  each node need to consume a quite amount of energy on computation and time.
Establishing Group Keys

- Second approach:
  Pre-load every node with the group key.

Drawback:
the group key must be changed and distributed to remaining nodes in a secure, reliable, and timely fashion.
Security Analysis - Survivability

- When a sensor node $u$ is compromised, the group rekeying scheme can revoke node $u$ from the group.

- Compromise Detection
  - Pair-wise and cluster keys win the trust of only neighbor nodes.
  - Group keys allow to decrypt broadcast messages (but the information level is low).
  - Adversary could not flood the network (Hello attack) because it employs $\mu$TESTLA scheme.
Thank You
For listening!