Security in Sensor Networks

11.1.2006

Presentation by Ziroli Plutschow
Outline

• Introduction into Sensor Networks
  Security Issues
• Overview: Key Establishment Schemes
• Secure Information Aggregation (SIA)
  Problem definition
  Attacker model
  Excursion: cryptography
  General Approach
  Example: Median
  Hierarchical Aggregation
Sensor Networks
Sensor Network Concept

Home Server  Base Station  Sensor Nodes
Applications of Sensor Networks

- Traffic Monitoring
- Wildlife Tracking
- Weather Monitoring
- Military Applications
- Building Security
- Building Automation
Special Security Set-Up

• No Public Key Cryptography
  Use symmetric cryptography

• Attacker has physical access to Sensor Node
  Use independent shared keys for any potential communication channel. (-scalability)
  → Key Establishment Schemes
  Tamper resistant packaging for key (-expensive)
Research Topics

- Key Establishment Schemes
- Secure Routing
- Secure Information Aggregation
- Efficient Cryptographic Primitives
  hash- / one-way - functions, PRG
  Public-Key (elliptic curve)
• Every node shares a key with each other node
  \( \rightarrow O(n^2) \) different keys, memory \( O(n) \) per node

• Location Information
  - node shares keys with neighbors
    (maybe base station, home server, aggregator)
  \( \rightarrow \) memory \( O(\text{const}) \)

• Probabilistic
  - node holds a subset of the generated keys
  - node has \( d \) neighbors \( \rightarrow \) memory \( O(n/d) \)
Key Establishment Schemes 2

• Peer intermediary
  node i has \((x_i, y_i)\)-position
  \(\rightarrow\) memory \(O(n^{1/2})\), but trust every node

• Polynomial based
  random 2-dim polynomial \(p(x, y)\)
  gets \(p(x_i, y)\) and \(p(x, y_i)\)
  degree \(t\): \(\rightarrow\) memory \(O(t)\)
  allows \(t\) compromised sensor nodes
  TinyKeyMan for TinyOS
Key Establishment Benchmark

From Paper: *Establishing Pairwise Keys in Distributed Sensor Networks* by D.Liu and P.Ning, NCSU
Secure Information Aggregation

• Problem Setting:

→ **Goal:** Home server accepts only true value
SIA: Attacker Model

- Corrupted / compromised aggregator
  Attacker has full control (stealthy attack)
- Corrupted / compromised sensor nodes
  Attacker has full control (stealthy attack)
- No DoS
  Radio based communication → physical
- Routing
  Uncorrupted nodes are connected
SIA: Key Setup

- Each Sensor Node
  - Unique Id
  - Shares a key with home server and aggregator

→ Home server and aggregator are able to authenticate the messages from sensor nodes.
SIA: Example, compute average

- 12 sensors, range 1...9, honest

Home Server

average: $v=5$

Base Station / Aggregator

Sensor Nodes
SIA: Example, compute average

- $n$ sensors, range $a...b$, $n'$ corrupted sensors
- max error $\varepsilon$ can be bounded exactly

$$\varepsilon = \frac{n'}{n}(b-a)$$

Home Server

Base Station / Aggregator

Sensor Nodes
SIA: Example, compute average

- $n$ sensors, range $a...b$, corrupted aggregator
- max error: $\epsilon = b-a$

$\nu \pm \epsilon = a...b$

$\rightarrow$ SIA can help
Minimize $\varepsilon$ (corrupted aggregator)

- Aggregator sends all signed sensor values to home server.
  - very inefficient
- SIA: Agg. proves that he aggregated correct
  Cryptographic techniques
  - commitment scheme
  - interactive proof
Cryptographic Hash Function

- Hash $y = h(x): \{0,1\}^* \rightarrow \{0,1\}^n$
  - one-way: given $y$, you can not calculate $x$
  - $2^{nd}$ pre-image resistance: given $x$ and $y$, you can not calculate a $x'$ with $h(x') = y$
  - collision resistance: you can not find $x \neq x'$ where $h(x) = h(x')$
Commitment Scheme

\[ v_{0,0} = H(v_{1,0} \parallel v_{1,1}) \]

\[ v_{1,0} \]
\[ v_{1,1} \]
\[ v_{2,0} \]
\[ v_{2,1} \]
\[ v_{2,2} \]
\[ v_{2,3} \]
\[ v_{3,0} \]
\[ v_{3,1} \]
\[ v_{3,2} \]
\[ v_{3,3} \]
\[ v_{3,4} \]
\[ v_{3,5} \]
\[ v_{3,6} \]
\[ v_{3,7} \]

\[ H(m_0) \]
\[ H(m_1) \]
\[ H(m_2) \]
\[ H(m_3) \]
\[ H(m_4) \]
\[ H(m_5) \]
\[ H(m_6) \]
\[ H(m_7) \]
SIA: General Approach

Aggregate

Commit

Proof

result

commitment

interactive

proof

sensor value

signatures
SIA: Two proofs

• Correct values as input for hash tree
  \((a_1, a_2, \ldots, a_n) = (m_1, m_2, \ldots, m_n)\)
  \(\rightarrow\) check signature of randomly chosen values

• Correct calculation of aggregation function
  result = \(f(m_1, m_2, \ldots, m_n)\)
  \(\rightarrow\) approximate with the randomly chosen values
SIA: Merkle hash tree 2

Verify $m_2$

$v_0,0 = H(v_1,0 || v_1,1)$

$v_1,0$

$v_2,0$

$v_3,0$

$m_0$

$v_2,1$

$v_3,1$

$m_1$

$v_2,2$

$v_3,2$

$m_2$

$v_2,3$

$v_3,3$

$v_3,4$

$m_3$

$v_3,5$

$m_4$

$v_3,6$

$m_5$

$v_3,7$

$m_6$

$m_7$
SIA: General Solution

- Allows to verify if the aggregator is honest
  If he cheats the result is rejected.
- Works for any aggregation function $f(a_1, a_2, ..., a_n)$, that can be approximated by a random subset of the input.
  - for concrete $f$, we can find better approx
  - example: median...
Median (General Approach)

• n sensors with distinct values
  if not distinct, use pair (value, sensor-Id)
  sorted sequence \((a_1, a_2, \ldots, a_n)\), median\(=a_{n/2}\)

• \(n'\) corrupted sensors
  can cause a result \(n'\) positions away form true median
  \(\rightarrow\) focus on corrupted aggregator

• General Approach: test \(m\) values
  Accept, if median of chosen set is close to the reported median.
Median (General Approach) 2

• Analyze the General Approach
  
  $n$ values, sorted sequence $(a_1, a_2, ..., a_n) = A$

  uniform sample $S$ of $m$ values from $A$

  allowed approximation fault $\varepsilon$:
    
    median($S$) is in $A$ between positions $n/2 \pm \varepsilon n$

  $\delta = \Pr[\text{detect violating approx. fault}]$

  $\rightarrow \delta \geq 1 - \left(\frac{2}{e^{2m\varepsilon^2}}\right)$

• For $\varepsilon$-approximation with constant probability $\delta$

  Choose size of sample $S$: $m = O(1/\varepsilon^2)$
• Trick: aggregator commits sorted sequence $\mathcal{A}$
• Check $m$ elements (if seq. is sorted + signature)
• Analysis
  Cheat-result is out of range $n/2 \pm \varepsilon n \rightarrow$ at least $\varepsilon n$
  elements are in wrong half of sequence.
  $\rightarrow \delta = \Pr[\text{detect cheating}] \geq 1-(1-\varepsilon)^m$
• For constant $\delta > 0.5$, we choose $m=O(1/\varepsilon)$
• Median method can be used for any position $k$ of a sequence, not only median at pos. $n/2$.

• The paper proposes specialized methods for
  - median
  - average
  - min/max
  - counting distinct elements
    (counting network size)
Secure Hierarchical Aggregation

- i) 1 verifies 2, ii) HS verifies 1

- (Not-) hierarchical aggregatable functions
  min/max, average, count vs. median
  → compute median of medians
Forward Secure Authentication

• Querying past data
  became interesting later / no connection
  sensor stored (data, sig(k, data))
  sensor could be compromised since that time

• Update k with one-way function $k_{\text{new}} = \text{OW}(k_{\text{old}})$
  Define time interval
  → Attacker must answer correct, or keep silent.
Thank you for your attention!

Questions?