Resilience of sink filtering scheme in wireless sensor networks

Original: Computer Communications, 06
Author: Miao Ma
Reporter: Yu-Hang Liu
Date: 2007/05/23
Outline

- Motivation
- Problem Definition
- State of the Art
- Assumption and network model
- Sink filtering scheme
- Resilience analysis
- Overhead analysis
- Conclusion
Motivation

- **Wireless Sensor Network (WSN)**
  - Major application
    - Monitor environment events
    - Transmit events to Sink
  - Node could be captured and compromised
    - False data injection attack
      - False alarm
      - Wastes sensor’s energy
Motivation

- Small low-powered sensor are constrained in their capabilities

- Some papers proposed
  - IHA and SEF, break down if attacker has compromised T nodes.
  - LBRS, it needs other location scheme to obtain sensor’s geographic location.
Problem Definition

- Goal
  - Defend against false data injection attack
  - Scheme is resilient to an increasing number of compromised nodes, without breakdown problem
Assumption and network model

- Heterogeneous sensor network
  - Basic sensor
    - Inexpensive
    - power-limited
  - CH (Cluster-Head)
    - More capabilities
    - Richer power supply
    - More compromised-resilient
Assumption and network model

- Uniform distribution
- Sink at center
- Define Deployment density as $n$
- $N = \lceil nA^2/\pi r_s^2 \rceil$
- $s = N/C$

<table>
<thead>
<tr>
<th>$N$</th>
<th>The total number of basic sensor</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_c$</td>
<td>The communication range of a basic sensor</td>
</tr>
<tr>
<td>$r_s$</td>
<td>The sensing range of a basic sensor</td>
</tr>
<tr>
<td>$n$</td>
<td>The average number of basic sensors within $r_s$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$C$</th>
<th>The number of grids</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s$</td>
<td>The number of sensor in cluster</td>
</tr>
</tbody>
</table>
Assumption and network model

- Key management
  - Before deployment
    - Each sensor $i$ shares a secret key with sink, $K_{(ID_i,S)}$.
    - $CH_j$ knows 8 neighborhood $CH_k$ and their pairwise keys $K_{(CH_j,CH_k)}$.
  - After deployment
    - Each sensor $i$ establishes a pairwise key with its one-hop neighboring basic sensors $j$, $K_{(ID_i,ID_j)}$.
    - Each $CH_j$ establishes a pairwise key with every basic sensor $i$ within its cluster, $K_{(CH_j,ID_j)}$. 
Assumption and network model

- Sensing coverage
  - A real event occurs, \( n \) basic sensors within the sensing range can sense it.
  - In reality, if \( n \) basic sensors in different cluster, CHs do inner working.

- Routing
  - Ring-by-ring through the CHs
Sink filtering scheme

Node$_i (1, \ldots, n)$

CH$_j$

1. Verifies each MAC
2. If all legitimate, generates

- $ID_{CH_j} | R_{CH_j}$
- $ID_1 | R_1 | MAC(K_{(ID_1,CH_j)}, ID_{CH_j} | R_{CH_j} | ID_1 | R_1)$
- $\vdots$
- $ID_n | R_n | MAC(K_{(ID_n,CH_j)}, ID_{CH_j} | R_{CH_j} | ID_n | R_n)$

$n$
- Number of basic sensors within a sensing range

$K_{(i,j)}$
- The pairwise key between $i$ and $j$

MAC($K,M$)
- Message authentication code of message $M$ with a symmetric key $K$

$R_i$
- Sensing result

$R_{CH_j}$
- Aggregation report
Sink filtering scheme

1. Verifies corresponding MAC
2. Generates

\[
\text{ID}_i \mid \text{MAC(} K_{(i,S)}, \text{ID}_{\text{CH}_j} | R_{\text{CH}_j} | \text{ID}_i) \times n' \rightarrow 1. \text{ Collects all } n' \text{ proofs } \\
2. \text{ Generates }
\]

\[
\text{ID}_{\text{CH}_j}, E( K_{(S,\text{CH}_j)}, \text{ID}_{\text{CH}_j} | R_{\text{CH}_j} | \text{ID}_1 | \text{MAC(} K_{(\text{ID}_1,S)}, \text{ID}_{\text{CH}_j} | R_{\text{CH}_j} | \text{ID}_1) \\
\vdots \\
\text{ID}_{n'}, \text{MAC(} K_{(\text{ID}_{n'},S)}, \text{ID}_{\text{CH}_j} | R_{\text{CH}_j} | \text{ID}_{n'}) \rightarrow \text{Sink}
\]

| \(n'\) | \(m < n' \leq n\), where \(m\) is the threshold of endorsements for a legitimate report |
| E(\(K,M\)) | Encryption of message \(M\) using a symmetric key \(K\) |
| \(S\) | Sink |
Resilience behavior within a cluster

- Intruding basic sensor
  - Fail to provide a valid MAC
- Intruding CH
  - Cannot forge an aggregation report

- Copy some compromised basic sensors from $k$th cluster into $j$th cluster.
  - $CH_j$ is able to recognize that.
  - Cannot know keys
- Copy some compromised CH from $k$th cluster into $j$th cluster
  - Cannot know key
Resilience behavior within a cluster

- Compromised CH
  - False data injection attack
    \[ P_f = \sum_{i=m}^{s} \binom{s}{i} p^i (1 - p)^{s-i}, \]
    where \( p = \frac{1}{2^e} \) and \( e \) is the size for each MAC in bits.
  - Selective forwarding attack
    - Watchdog mechanism are applicable to deal with it.
Resilience behavior within a cluster

- Compromised basic sensors \((t < m)\)
  - Forge raw data attack (with valid MAC)
    - Robust aggregation algorithm can tolerate the number of incorrect raw sensing results
  - Wrong MAC attack
    - Any intermediate CH node is not allowed to drop packets
Resilience behavior within a cluster

- Compromised CH and basic sensors simultaneously ($t < m$)
  - False data injection attack
    
    \[
    P_f = \sum_{i=m-t}^{s-t} \binom{s-t}{i} p^i (1-p)^{s-t-i}
    \]

    where $p = \frac{1}{2^e}$ and $e$ is the size for each MAC in bits

- Even if a cluster is fully compromised ($t \geq m$), the forging ability is only limited in this cluster.
Resilience analysis

- **Proposition 5.1**
  \[ E[D_t] = \frac{C(s \choose t) \cdot (N-s \choose k-t)}{N \choose k} \approx C \cdot e^{-\frac{k}{c}} \cdot \left(\frac{k}{C}\right)^t \]

- **Corollary 5.2**
  \[ E[Y] = \sum_{t=m}^{s} D_t = \sum_{t=m}^{s} \frac{C(s \choose t) \cdot (N-s \choose k-t)}{N \choose k} \approx \sum_{t=m}^{s} C \cdot e^{-\frac{k}{c}} \cdot \left(\frac{k}{C}\right)^t \]
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A^2$</td>
<td>Deployment area</td>
<td>$1000m \times 1000m$</td>
</tr>
<tr>
<td>$a^2$</td>
<td>Cluster area</td>
<td>$50m \times 50m$</td>
</tr>
<tr>
<td>$r_s$</td>
<td>Sensing range of a basic sensor</td>
<td>$20m$</td>
</tr>
<tr>
<td>$r_c$</td>
<td>Communication range of a basic sensor</td>
<td>$40m$</td>
</tr>
<tr>
<td>$n$</td>
<td>Number of basic sensors within a sensing range</td>
<td>$12$</td>
</tr>
<tr>
<td>$N$</td>
<td>Number of basic sensors in entire area</td>
<td>$9600$</td>
</tr>
<tr>
<td>$C$</td>
<td>Number of CHs (i.e., clusters)</td>
<td>$400$</td>
</tr>
<tr>
<td>$L$</td>
<td>Number of rings</td>
<td>$10$</td>
</tr>
<tr>
<td>$h$</td>
<td>Average number of hops from a CH to sink</td>
<td>$7.15$</td>
</tr>
<tr>
<td>$s$</td>
<td>Number of basic sensors per cluster</td>
<td>$24$</td>
</tr>
<tr>
<td>$d$</td>
<td>Number of basic sensors within a communication range</td>
<td>$48$</td>
</tr>
<tr>
<td>$e$</td>
<td>Length of each MAC in bits</td>
<td>$32$</td>
</tr>
</tbody>
</table>
Quantitative analysis
Quantitative analysis

![Graph showing the relationship between the number of compromised basic sensors in deployment area (k out of 9600) and the expected value (E[Y] out of 400). The graph includes lines for different values of m (5, 6, 7, 8) and distinguishes between simulation and analytical results. The parameters are n = 12, C = 400.](image-url)
Poisson approximation
Poisson approximation

\[ E[Y] \text{ (out of 400)} \]

\[ \text{Number of compromised basic sensors in deployment area, } k \text{ (out of 9600)} \]

- \( m = 5 \), simulation
- \( m = 6 \), simulation
- \( m = 7 \), simulation
- \( m = 8 \), simulation
- \( m = 5 \), Poisson Approximation
- \( m = 6 \), Poisson Approximation
- \( m = 7 \), Poisson Approximation
- \( m = 8 \), Poisson Approximation

\( n = 12, C = 400 \)
Parameters selection

- Choice of C
Parameters selection

- Choice of $n$
## Overhead analysis

<table>
<thead>
<tr>
<th>Overhead</th>
<th>CH</th>
<th>Basic sensor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication</td>
<td>709.2 bytes [8+(2+4) \times n]+[8+8+(2+4) \times n] \times h</td>
<td>12 bytes [(2+4)+(2+4)]</td>
</tr>
<tr>
<td>Computation</td>
<td>1 aggregation 1 encryption and 24 MACs</td>
<td>3 MACs</td>
</tr>
<tr>
<td>Storage</td>
<td>264 bytes [1+8+s] \times 8</td>
<td>392 bytes [1+1+(d-1)] \times 8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Size of basic sensor ‘s ID</th>
<th>2 bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size of CH ‘s ID</td>
<td>8 bytes</td>
</tr>
<tr>
<td>Key size</td>
<td>8 bytes</td>
</tr>
</tbody>
</table>
Comparison

- Compare with SEF
  - Communication
    - The number of hops routing to the sink in heterogeneous sensor network is smaller than in a homogeneous one.
    - Forwarding report only happen on the powerful CH nodes.
  - Computation
    - Relay CH don’t need do verification when forwarding data
  - Storage
    - SEF need a large number of key to increase the probability of key sharing between nodes.
Conclusion

- Present a sink filtering scheme in clusters of heterogeneous sensor networks
  - Solve the threshold breakdown problem.
  - Graceful performance degradation
  - Scalable and efficient in communication, computation and storage

- Using Poisson Approximation to investigate the performance tradeoff between resilience and overall cost
  - Dividing into more clusters is helpful to improve the resilience, but with a significant sacrifice on the overall cost
  - Much denser deployment of basic sensors is a feasible way toward resilient security
State of the Art


