Research on the authentication strategy in wireless network environment: sensor network and roaming system

CSIE, National Chiayi University

Chih-Hung Wang   Ph.D.
wangch@mail.nctu.edu.tw
Agenda

- Introduction of Wireless Sensor Networks
  - Applications
  - Security considerations
- Key Distribution Strategies in WSNs
- New Approach
  - [CW07][Che07] by Y. J. Chenoup
  - Proposed scheme
  - Rekeying
- Heterogeneous Sensor Networks
- Roaming System and Mobile Payment
  - Delegation-based Signature
  - Group Signature
- Conclusions
Introduction

Internet 前 Internet 後
PART 1: SENSOR NETWORKS
Introduction

Features of WSNs [WLSC06]

- Very limited resources
  - Limited memory and storage space (8-bit, 4MHz CPU, 8KB memory)
  - Power limitation

- Unreliable communication
  - Unreliable transfer
  - Conflicts
  - Latency

- Unattended operation
  - Exposure to physical attacks
  - Managed Remotely
  - No central management point
Introduction

Challenges

- Vulnerability of nodes to physical capture
- Lack of a-priori knowledge of post-deployment configuration
- Limited bandwidth and transmission power
Introduction: Applications

Applications of WSNs

- Be able to monitor a wide variety of ambient conditions
  - Temperature
  - Humidity
  - Vehicular movement
  - Lightning condition
  - Pressure
  - Soil makeup
  - Noise levels
  - …
WSN Applications

- Military applications
  - Monitoring friendly forces, equipment and ammunition, battlefield surveillance

- Environmental applications
  - Forest fire detection, biocomplexity mapping of the environment, flood detection

- Health applications
  - Telemonitoring of human physiological data, tracking and monitoring doctors and patients, drug administration in hospitals

- Home applications
  - Home automation, smart environment

- Other commercial applications
  - Environmental control in office buildings, Detecting and monitoring car thefts, …
Security Considerations (1)

General security requirements [CY05]
- Availability
- Authentication
- Integrity
- Confidentiality
- Non-repudiation

Specific requirements
- Survivability
- Degradation of security services
Security Considerations (2)

Security requirements for key distribution

- Scalability
- Efficiency
- Key connectivity
  - Probability that two (or more) sensor nodes store the same key or keying material
- Resilience
  - Resistance against node capture. Compromise of security credentials should not reveal information about security of any other links in WSN.
Key Distribution Strategies in WSNs (1)

Models [CY05]

- Hierarchical WSN

Data flow:
- Pair-wise
- Group-wise
- Network-wise
Key Distribution Strategies in WSNs (2)

- Distributed WSN
  - No fixed infrastructure
  - Network topology is not known prior to deployment
  - Sensors are usually randomly scattered all over the target area
  - Once deployed, each sensor scans its radio coverage area to figure out its neighbors.

![Distributed WSN Diagram]
Previous Works

**Single network-wide key** [CPS04]

Key preloading

After deployment
Single Network-wide Key

Properties

- Minimal memory storage required
- No additional protocol steps are necessary
- Resistant against DoS, packet injection
  - MAC protection

Drawbacks

- Compromise of a single node causes the compromise of the entire network
Fully Pairwise-shared keys (1)

- **Number of keys**
  - In a network with \( n \) nodes
    - Every node stores \( n-1 \) nodes
    - Total of \( \binom{n}{2} \) keys
Fully Pairwise-shared keys (2)

Properties

- Perfect resilience to node capture
- Compromised keys can be revoked
- Only uses symmetric cryptography
  - 8-bit, 4MHZ CPU with only 8K (total) of memory and disk space.
Random Key Pre-distribution (1)

- Eschenauer & Gligor 2002 [EG02]

Connectivity

- $p$: the probability that a shared key exists between two sensor nodes
- $n$: number of network nodes

Key Pool

|S| keys

Random $m$ keys

Information Security Lab., CSIE, NCYU, Taiwan, R.O.C.
Random Key Pre-distribution (2)

Connectivity

- **d**: the expected degree of a node $= p \times (n - 1)$
- **$G(n, p)$**: a graph of $n$ nodes for which the probability that a link exists between two nodes is $p$.
- Erdos and Renyi [ER60] shows
  - A desired probability $P_c$ for graph connectivity, ex. 0.99999
  - $P_c = \lim_{n \to \infty} \Pr[G(n, p) \text{ is connected}] = e^{e^{-c}}$
  - $p = \frac{\ln(n)}{n} + \frac{c}{n}$
  - $2 \times 10^{-3}$, $n = 10000$, $d = 2 \times 10^{-3} \times 9999 = 20$

$|S| = 100000$, $m = 15$
Random Key Pre-distribution (3)

- Neighborhood connectivity constraints
  - $n'$: the number of nodes in a neighborhood
  - $p'$: the probability of sharing a key between any two nodes in a neighborhood
  - $p' = \frac{d}{n'-1} \gg p$
Random Key Pre-distribution (4)

- The connectivity model is probability
- A slight change that the graph may not be fully connected
- The deployment area has unpredictable physical obstacles to communication.
Random Pairwise Key Scheme

- Assign a unique key to each pair of nodes
- Provide node-to-node authentication

Chan, Perring and Song [CPS03]

- Each node needs only $np$ pairwise keys
- If a node can store $m$ keys, then the maximum supportable network size is

$$n = \frac{m}{p}$$
Pairwise Key Establishing (1)

- Liu and Ning [LN03a]
- Polynomial-based Key Predistribution Scheme [Blundo et al.] [BSHKVY93]
- The (key) setup server randomly generates a (bivariate $t$ degree) polynomial

$$f(x, y) = \sum_{i=0}^{t} \sum_{j=0}^{t} a_{i,j} x^i y^j$$

over a finite field $F_q$, where $q$ is a prime number, such that it has the property of $f(x, y) = f(y, x)$.
- For each sensor $i$, the setup server computes a polynomial share of $f(i, y)$.
- Node $i$ evaluate $f(i, y)$ at point $j \rightarrow f(i, j)$

\| node $j$ evaluate $f(j, y)$ at point $i \rightarrow f(j, i)$
Each sensor node $i$ needs to store a $t$-degree polynomial $f(i, x)$, which occupies $(t + 1) \log q$ bits storage space.

To establish a pairwise key, both sensor nodes need to evaluate the polynomial at the ID of the other sensor node.

It can only tolerate no more than $t$ compromised nodes, where the value of $t$ is limited by the memory available in sensor nodes.
Pairwise Key Establishing (3)

Random subset assignment

- Set $F$ of bivariate $t$-degree polynomials over finite field $F_q$ is generated. $\rightarrow$ polynomial pool (size = $s$)
- Each polynomial is assigned a unique id.
- For each sensor node a subset of $s'$ polynomial is randomly chosen from $F$.
- For each polynomial in the chosen subset a polynomial share is loaded into nodes memory.
Pairwise Key Establishing (4)

Polynomial Pool $F$

Sensor nodes
Du, Deng, Han and Varshney [DDHV06]

- $N$ sensor nodes are divided into $t \times n$ groups $G_{i,j}$, for $i = 1, \ldots, t$ and $j = 1, \ldots, n$

  - is deployed from the deployment point with index $(i, j)$
  - $(x_i, x_j)$: the deployment point for group
Using Deployment Knowledge (2)

Follow two-dimensional Gaussian distribution
(Normal distribution)
Using Deployment Knowledge (3)

**Blom’s scheme** [Blo85]
- Public matrix $G$
- Private matrix $D$ (symmetric).

Let $A = (D \ G)^T$

$$A \ G = (D \ G)^T \ G = G^T \ D^T \ G = G^T \ D \ G = (A \ G)^T$$

$N : \text{Network size}$
Using Deployment Knowledge (3)

\[ A = (D \ G)^T \]

\[ X = A = (D \ G)^T \ G (D \ G)^T \ G \]

Node i carries:

Node j carries:

Information Security Lab., CSIE, NCYU, Taiwan, R.O.C.
Using Deployment Knowledge (4)

Two nonneighboring key-space pools share no key space.

Horizontal

Vertical

Diagonal

\[ 0 \leq a \leq 0.25 \]
\[ 0 \leq b \leq 0.25 \]
\[ 4a + 4b = 1 \]
Using Deployment Knowledge (5)

- Select keys for each key pool $S_{i,j}$
- Global key pool $S$
- Overlapping factor $a$ and $b$

| $|S_c|\text{ keys}$ | $|S_c|\text{ keys}$ | $|S_c|\text{ keys}$ | $|S_c|\text{ keys}$ |
|------------------|------------------|------------------|------------------|
| $a|S_c|\text{ keys}$ | $|S_c|\text{ keys}$ | $b|S_c|\text{ keys}$ | $|S_c|\text{ keys}$ |
| $1-(a+b)|S_c|\text{ keys}$ | $1-2(a+b)|S_c|\text{ keys}$ | $1-a|S_c|\text{ keys}$ | $1-2(a+b)|S_c|\text{ keys}$ |
The Exclusion Basis System (EBS)

- Ex. $M_0$ has been compromised
- Rekeying messages
  
  \[ Message_1 : E_{K_4}(S', E_{K_1}(K_1'), E_{K_2}(K_2'), E_{K_3}(K_3')) \]
  
  \[ Message_2 : E_{K_5}(S', E_{K_1}(K_1'), E_{K_2}(K_2'), E_{K_3}(K_3')) \]

  $S'$: new session key

[EHMS04]
# New Approach

## Concept

<table>
<thead>
<tr>
<th>Key Management scheme</th>
<th>Pairwise Key Distribution</th>
<th>Group Key Distribution</th>
<th>New Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[CPS03], [DDHV03], [DDHV06], [HMMH04], [LN03a], [LN03b], [LND05].</td>
<td>EBS [EHMS04], LiSP [PS04], SPINS [PSWCT02], SHELL [YGE06]</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rekeying</th>
<th>Less</th>
<th>More</th>
<th>More</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication key</td>
<td>Pairwise key</td>
<td>Session key</td>
<td>Pairwise key</td>
</tr>
</tbody>
</table>

Information Security Lab., CSIE, NCYU, Taiwan, R.O.C.
Architecture

Sensor Nodes

Controller Nodes

Base Station

Information Security Lab., CSIE, NCYU, Taiwan, R.O.C.
Description (1)

<table>
<thead>
<tr>
<th>Link type</th>
<th>Polynomial type</th>
<th>Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>In-group</td>
<td>$f_r(x, y), r = 1, \ldots, n$ (n: # of groups)</td>
<td>$t$</td>
</tr>
<tr>
<td>Inter-group</td>
<td>$g(x, y)$</td>
<td>$t'$</td>
</tr>
</tbody>
</table>

The scheme consists four phases:

- Key predistribution
- EBS formation and administrative key distribution
- Pairwise key establishment
- Rekeying
## Description (2)

<table>
<thead>
<tr>
<th>Key type</th>
<th>Notation</th>
<th>Construction stage</th>
<th>Major function</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Session key</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>In-group</td>
<td>$(SK_{\text{in-group}})_r$</td>
<td>Key predistribution</td>
<td>Generate in-group pairwise key</td>
</tr>
<tr>
<td>Inter-group</td>
<td>$(SK_{\text{inter-group}})_{i,j}$</td>
<td></td>
<td>Generate inter-group pairwise key</td>
</tr>
<tr>
<td><strong>Administrative key</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$K_{\ell}$</td>
<td>EBS formation</td>
<td>Rekeying</td>
</tr>
<tr>
<td><strong>Pairwise key</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>In-group</td>
<td></td>
<td>Pairwise key establishment</td>
<td>Communication key between nodes in same group</td>
</tr>
<tr>
<td>Inter-group</td>
<td></td>
<td></td>
<td>Communication key between nodes from neighboring group</td>
</tr>
</tbody>
</table>
Collusion Prevention (1)

EBS is highly vulnerable to collusion attacks

Ex:

<table>
<thead>
<tr>
<th>Colluding</th>
<th>Revealed keys</th>
<th>Hamming distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_a ) and ( p_b )</td>
<td>( K_1, K_5 )</td>
<td>2</td>
</tr>
<tr>
<td>( p_a ) and ( p_c )</td>
<td>( K_2, K_3, K_4, K_6 )</td>
<td>4</td>
</tr>
<tr>
<td>( p_a ) and ( p_d )</td>
<td>( K_1, K_2, K_3, K_4, K_5, K_6 )</td>
<td>6</td>
</tr>
</tbody>
</table>
Collusion Prevention (2)

Example

$k = 2, m = 3$

<table>
<thead>
<tr>
<th>KC</th>
<th>$K_1$</th>
<th>$K_2$</th>
<th>$K_3$</th>
<th>$K_4$</th>
<th>$K_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>KC_1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>KC_2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>KC_3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>KC_4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>KC_5</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>KC_6</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>KC_7</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>KC_8</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>KC_9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>KC_{10}</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Mutual Hamming distance among KCs

<table>
<thead>
<tr>
<th>KC</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>
Collusion Prevention (3)

**SHELL** [YGE06]
Collusion Prevention (4)

The diagram illustrates a network of nodes (A, B, C, D, E, F, G, H) with keys (KC1, KC2, KC3, KC4, KC5, KC6, KC7, KC8, KC9, KC10) indicating the relationships and connections between the nodes. Each connection is labeled with a key value, such as (KC2, 1) or (KC3, 2), and the numbers in parentheses represent the cardinality or significance of the connection.
Collusion Prevention (5)

Our scheme

The percentage of number of Hamming distance is 2 (the best) in our scheme almost is better than SHELL (with about 3.56% advantage).

Information Security Lab., CSIE, NCYU, Taiwan, R.O.C.
Key Pre-distribution (1)

The key materials generated by Base Station

- $f_r(x, y)$, $g(x, y)$, $(SK_{\text{in-group}})_r$, and $(SK_{\text{inter-group}})_{i,j}$, where $r = 1, ..., n$ and $i, j$ is the index of neighboring groups

- $f_r(u, y)$: the polynomial share of $f_r(x, y)$ to the node $u \in G_r$.

- $g(G_r, y)$: the polynomial share of $g(x, y)$ to the group $G_r$.
Key Pre-distribution (2)

For controller node $CN_4$, load:

- $f_4(CN_4,y)$ [In-group]
- $g(G_4,y)$ [Inter-group]
- $g(G_1,CN_4)$, $g(G_5,CN_4)$
- $g(G_2,CN_4)$, $g(G_6,CN_4)$
- $g(G_3,CN_4)$, $g(G_7,CN_4)$

Current session key $(SK_{\text{in-group}})_4$, $(SK_{\text{inter-group}})_{1,4}$, $(SK_{\text{inter-group}})_{2,4}$, $(SK_{\text{inter-group}})_{3,4}$, $(SK_{\text{inter-group}})_{4,5}$, $(SK_{\text{inter-group}})_{4,6}$, $(SK_{\text{inter-group}})_{4,7}$
Proposed scheme -- key predistribution

For node $p_a$, load:

- $f_4(p_a, y)$ [In-group]
- $g(G_4, y)$ [Inter-group]
- $g(G_1, p_a)$ $g(G_5, p_a)$
- $g(G_2, p_a)$ $g(G_6, p_a)$
- $g(G_3, p_a)$ $g(G_7, p_a)$

Current session key ($SK_{\text{in-group}})_4$, 
($SK_{\text{inter-group}})_{1,4}$, ($SK_{\text{inter-group}})_{2,4}$,
($SK_{\text{inter-group}})_{3,4}$, ($SK_{\text{inter-group}})_{4,5}$,
($SK_{\text{inter-group}})_{4,6}$, ($SK_{\text{inter-group}})_{4,7}$
EBS Key
Administrative key distribution (ex. group $G_4$)

1. Tabulate the sensor ID & location
2. construct EBS table

$E_{(SK_{in-group})_4}$ (ID, Location)

$E_{K_{CN4,p_a}}$ (administrative keys for $p_a$)

$K_{CN4,p_a}$ : the pairwise key of $CN_4$ and $p_a$
If two nodes are in the same group $G_i$, derives a pairwise key by polynomial $f_i(x, y)$.

<table>
<thead>
<tr>
<th>Group ID</th>
<th>$G_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node ID</td>
<td>$P_a$</td>
</tr>
<tr>
<td>Polynomial share of $f_4(x, y)$ preloaded</td>
<td>$f_4(P_a, y)$</td>
</tr>
</tbody>
</table>

In-group pairwise key:

$$H(f_4(P_a, P_b) || (SK_{in-group})_4)$$

$$compute f_4(P_a, P_b) = compute f_4(P_b, P_a)$$
If two nodes are in the different group, derives a pairwise key by polynomial $g(x, y)$.

- Group ID: $G_4$
- Node ID: $G_5$
- Polynomial share of $g(x, y)$ preloaded: $g(G_4, y)$
- Secret information preloaded: $g(G_5, p_a)$

Inter-group pairwise key:

$$H(g(G_4, q) \| g(G_5, p_a) \| (SK_{inter-group})_{4,5})$$
Rekeying (1)

Ex. \( p_c \in G_4 \) has been compromised

<table>
<thead>
<tr>
<th>( K_1 )</th>
<th>( \ldots )</th>
<th>( p_a )</th>
<th>( \ldots )</th>
<th>( p_b )</th>
<th>( \ldots )</th>
<th>( p_c )</th>
<th>( \ldots )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( K_2 )</td>
<td>( \ldots )</td>
<td>1</td>
<td>( \ldots )</td>
<td>0</td>
<td>( \ldots )</td>
<td>1</td>
<td>( \ldots )</td>
</tr>
<tr>
<td>( K_3 )</td>
<td>:</td>
<td>0</td>
<td>:</td>
<td>0</td>
<td>:</td>
<td>1</td>
<td>:</td>
</tr>
<tr>
<td>( K_4 )</td>
<td>:</td>
<td>1</td>
<td>:</td>
<td>1</td>
<td>:</td>
<td>0</td>
<td>:</td>
</tr>
<tr>
<td>( K_5 )</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( K_6 )</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Rekeying of in-group links
  - In-group session key \( [(SK_{in-group})_4] \)
  - Administrative keys of compromised nodes \( (p_c) \)
Rekeying (2)

Rekeying of in-group pairwise key

Message1: $E_{K_2}( (SK_{\text{in-group}})'_4, E_{K_1}(K_1'), E_{K_3}(K_3'), E_{K_6}(K_6'))$

Message2: $E_{K_4}( (SK_{\text{in-group}})'_4, E_{K_1}(K_1'), E_{K_3}(K_3'), E_{K_6}(K_6'))$

Message3: $E_{K_5}( (SK_{\text{in-group}})'_4, E_{K_1}(K_1'), E_{K_3}(K_3'), E_{K_6}(K_6'))$

$(SK_{\text{in-group}})'_4: \text{new in-group session key}$

New in-group pairwise key ($p_a$ and $p_b$)

$H(f_4(p_a, p_b) \parallel (SK_{\text{in-group}})'_4)$
Rekeying (3)

Rekeying of inter-group links
- The polynomial share of $g(x, y)$ [$g(G_4, y)$]
- Session keys sharing with the neighboring groups

\[
E_{H(c_1||c_2||\ldots||c_{t+1})}((SK_{\text{inter-group}})_{1,4}', (SK_{\text{inter-group}})_{2,4}', (SK_{\text{inter-group}})_{3,4}', (SK_{\text{inter-group}})_{4,5}', (SK_{\text{inter-group}})_{4,6}', (SK_{\text{inter-group}})_{4,7}', g(G_4^*, y))
\]

$H$: a one-way hash function
$c_1, c_2, \ldots, c_{t+1}$: the corresponding coefficients of $f_4(CN_4, y)$

Base Station $\rightarrow$ $CN_4$

$CN_i \leftarrow E_{K_{CN_4,CN_i}}((SK_{\text{inter-group}})_{4,i}')$

$i$: the index of the neighboring groups of $G_4$
Rekeying (4)

New inter-group pairwise key ($p_a \in G_4$ and $q \in G_5$)

Group ID

Node ID

Polynomial share of $g(x, y)$ preloaded

Polynomial share of $g(x, y)$ after rekeying

Secret information preloaded

compute $g(G_4^*, q)$

Inter-group pairwise key after rekeying

$H(g(G_4^*, q) || g(G_5, p_a) || SK_{inter-group}^{'}_{4,5})$
Multi-level: Summary of Network Model(1)

- There are totally $\sum_{k=0}^{L} N_k$ nodes arranged into $|D|$ deployment groups, where $k$ is the level of sensor nodes, and $N_k$ is the number of the nodes of level $k$. The lower index represents a higher ability node, and vice versa. The level 0 represents the sink of this network, and the level $L$ is the lowest sensor nodes.
- The ability: $A_1 >> A_2 > \cdots > A_k > \cdots > A_L$
- The number of nodes: $N_1 < N_2 < \cdots < N_k < \cdots < N_L$
- The deployment groups can be arranged as a grid or hexagonal distribution. The distance between the deployment point and cell’s border is $2\sigma$. 
Multi-level: Summary of Network Model (2)

- The resident point means the final location of the deployed node. The node \( n \) in the deployment group \( D_i \) follows a pdf \( pdf(x, y \mid n \in D_i) \) of two-dimensional Gaussian distribution:

\[
pdf(x, y) = \frac{1}{2\pi\sigma_k^2} e^{-\left[(x-x_i)^2 + (y-y_i)^2 / 2\sigma_k^2\right]}
\]

where \((x, y)\) is a coordinates and the \( \sigma_k^2 \) is the variance of distribution for the nodes in class \( k \).

- In the network group, different level has different number of sensor nodes, and there is a security degree \( t \) used in the entire system.
Polynomial Hash Tree Key Distribution Strategy (1)

Sketch

- Two network groups with two CHs
Polynomial Hash Tree Key Distribution Strategy (2)

- The proposed system employs the m-ary polynomial hash tree (mPHT).
- An m-ary tree is obviously less deep than the binary tree and convenient to represent and implement the complex network topology.
Polynomial Hash Tree Key Distribution Strategy (3)

Example
The steps of building PHT

- The system generates a long integer \( P \in \{0,1\}^k \) as \( P(1,1) \), where \( k \) is the number of bits. Let \(|H|\) denote the number of bits of the hash function and \(|D|\) denote the number of deployment groups. \( k \) must satisfy the equation of \( k \geq (|D| + 1) \times |H| \)

- For the nodes of \( P(2, *) \), the system divides \( P \) into \((\text{Child}_1, \text{Child}_2, \ldots, \text{Child}_{|D|}, \text{Child}_{|D| + 1})\), and all parts of \( P \) except for the last part are distributed to the \( P(2, *) \) nodes.

- For the node under the \( P(2,*) \):
\[
P(i, j) = H(\text{its parent node} \oplus \text{ID of the node})
\]
Polynomial Hash Tree Key Distribution Strategy (5)

For example, after deployment, CH1 will be assigned the node $P(3,1)$, which is generated by

$$H(\text{Child}_1 \oplus \text{ID of node CH}_1)$$

The node of this PHT represents a polynomial seed, which can be used to build a polynomial by calculating $H(\text{LeftShift}(P(i,j),1))$ until calculating enough coefficients.

If any pair of CHs wants to calculate their co-polynomial, they have to exchange their polynomial seeds for each other.

- For instance, if CH2 tries to build the co-polynomial shared with CH3, it can ask CH3 transmitting $P(3,3)$ and use $H(P(3,2) \oplus P(3,3))$ to calculate the co-seed.
Use of Deployment Knowledge

The deployment group D5 will get the seeds of its all neighbors.
Analysis and Discussion

The Network Connectivity

- The connectivity is much related to the **coverage rate**.
- There are two kinds of connectivity: local and global connectivities.
- **Due to the use of higher ability of the CHs, the coverage in our approach is almost 100%.**
- The rate of connectivity is approximated to 1 (about 0.999).
- We use **java platform** to build our simulation environment.
  - The most popular tool for network simulation, NS2, is not suitable for our requirement due to the incomplete support of key distribution and applications.
- We only simulate **the two-level environment** for simplifying the complexity.
The Experiment Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area Size</td>
<td>3600m * 3600m (Area Size = Group Size * # of Groups)</td>
</tr>
<tr>
<td># of Groups</td>
<td>36</td>
</tr>
<tr>
<td># of Sensor Nodes</td>
<td>3600</td>
</tr>
<tr>
<td>Group Size</td>
<td>100m * 100m</td>
</tr>
<tr>
<td># of Sensor Nodes/Group</td>
<td>100</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>Group Distance/4 (100/4)</td>
</tr>
<tr>
<td>Sensing Range of L-Node</td>
<td>40m</td>
</tr>
<tr>
<td>Transmission Range of H-Node</td>
<td>112m</td>
</tr>
<tr>
<td>PBT height (two-level and binary tree)</td>
<td>7(total 64 leaves)</td>
</tr>
<tr>
<td>Ratio of CHs/Group</td>
<td>0.05</td>
</tr>
</tbody>
</table>
The Coverage

The coverage of our sensor network environment.

- There are 3600 sensor nodes deployed into 36 deployment groups. Following two-dimensional Gaussian distribution with default standard deviation, we can get this topology.
- The coverage rate is near to $1 - \frac{16}{(3600-180)} \approx 0.996$, and the connectivity is about 99%.
Resilience Ability (1)

In our mechanism, the number of keys of a certain node is equal to the number of its neighbors, thus the resilience ability can be regarded as the average number of the neighbors.

The compromised link fraction is defined as the following equation:

\[
\frac{\text{# of effected edge}}{\text{# of linked edge}}
\]

The resilience ability in our scheme is linear because the pairwise key is only used in one link.
The following figure shows the comparisons between our scheme and the random key pool of EG scheme.

In the random key pool system (EG), the number of compromised links is quickly increasing because compromising one key may destroy multiple links.
Efficiency (1)

Storing overheads

- In our approach, all the nodes must maintain their key rings.
- The intermediate nodes have to maintain and extend the polynomial seeds for building a polynomial, co-polynomial and generate the polynomial seeds for their children.
- Moreover, all the sensor nodes have to store one master key before deployment.
- The storage requirement is considered only for storing overheads of the keys and polynomial seeds.
Communication costs

- The intermediate sensor node needs to take longer time in hello phase because it has to wait for the message from higher or lower sensor nodes.
- Moreover, the sensor node must transmit its collection information to its KDC.
- The computation overhead is required for the intermediate nodes and CHs because they have to deploy and distribute the key rings and polynomial seeds to their lower nodes after deployment.
PART2: ROAMING PAYMENT
Introduction

Since the mobile user may stay in an untrusted domain when roaming, the security requirements such as authentication, anonymity, privacy and confidentiality should be paid more attention to and be developed.

In 2005, Lee and Yeh[1] proposed a new delegation-based authentication protocol for PCSs in which the home location register (HLR) delegates the signing power to the mobile user so that the authentication and secure communication between the mobile user and visitor location register (VLR) can be established.

Introduction

Lee et al. [2] pointed out that the protocol of Lee and Yeh has a weakness for a malicious VLR to forge the authentication messages in off-line authentication process.

Lee et al. presented an enhanced one to avoid this kind of security flaw.
Youn and Lim [3] in 2010 added a nice property of unlinkability to fully protect the mobile user’s location privacy.

In this paper, we further point out Youn and Lim's protocol suffers from the denial of service (DoS) attack, since they change the keys stored in HLR each round but do not consider the possible interruption in data transmissions.
Introduction

- Three contributions of the paper
  - It has the property of user privacy with unlinkability and can avoid the denial of service attacks.
  - It supports multiple-VLR payments by using UOBT (Unbalanced One-way Binary Tree).
  - It is efficient since combines the anonymous authentication and payment verification.
Related Work

Unbalanced One-way Binary Tree (Yen et al. 1999)

The UOBT is specially useful in reducing storage of a resource constraint device, such as the PDA or smart phone, because only the root value of the tree has to be stored.

\[
P_1 \quad P_2 \quad \ldots \quad P_{a-1} \quad P_a
\]

\[
h_1 \quad h_1 \quad h_1 \quad h_1
\]

\[
P_{1,b} \leftarrow P_{2,b} \leftarrow \ldots \leftarrow P_{a-1,b} \leftarrow P_{a,b}
\]

\[
P_{1,b-1} \quad P_{2,b-1} \quad P_{a-1,b-1} \quad P_{a,b-1}
\]

\[
\vdots \quad \vdots \quad \vdots \quad \vdots
\]

\[
P_{1,1} \quad P_{2,1} \quad P_{a-1,1} \quad P_{a,1}
\]

\[
P_{1,0} \quad P_{2,0} \quad P_{a-1,0} \quad P_{a,0}
\]
Related Work

The DoS Attacks on Youn and Lim’s Protocol[3]

- Youn and Lim proposed an improved protocol with unlinkability by changing the key values each round.
- However, the **asynchronization problem** of the keys may occur if the last transmission step in the authentication phase is interrupted by a malicious attacker.
- HLR has updated his secret value $K$ with $K'$ and $\sigma$ with $\sigma'$, but the mobile user has no information about that and keep his secret value as the old $(K, \sigma)$.

The Proposed Protocol

- Let $p$ and $q$ are two distinct large primes and satisfy $q \mid p - 1$.
- $g$ is a generator of the subgroup of $Z_p^*$ with the order $q$.
- The identities of VLR and HLR are $ID_V$ and $ID_H$.
- HLR selects a random $x$ as his private key.
- The corresponding public key is $\nu = g^x \pmod p$. 
The Proposed Protocol

Registration

- The mobile user MU subscribes to his HLR and generates a unbalanced one-way binary tree.
- MU sends the anchor values $P_{1,0}, P_{2,0}, \ldots P_{a,0}$ and the maximum spending unit $b$ to HLR.
- HLR stores these anchor values to his database and links this record to MU’s account.
The Proposed Protocol

Registration

- the value of $P_{i,0}$, which means the anchor of the hash chain $P_i$, is used for the payment to $VLR_i$.
- HLR then selects a random number $k$ and computes $K = g^k \pmod{p}$ and $\sigma = x + kK \pmod{q}$.
- $\sigma$ is the secret key shared between HLR and MU, $K$ can be regarded as the pseudonym of MU.
- HLR stores $(K, \sigma)$ into the SIM card of MU and sends the SIM card to MU.
The Proposed Protocol

**Authentication**

- When MU roams to the foreign area, he must connect to VLR for authentication.
- MU prepares a random $r_1$ and computes a hash chain $(h^{(1)}(r_1), h^{(2)}(r_1), \ldots h^{(n+1)}(r_1))$, where $h^{(i+1)} = h(h^{(i)}(r_1))$ and $h^{(n+1)} = R$.
The Proposed Protocol

Authentication steps

\[ R = h^{(n+1)}(r_i) \]

1. \( K \)
2. \( ID_{V_i}, r_2 \)

3. compute \( u = g^i \) (mod \( p \))

\[ s = \sigma \times h(R \parallel r_2 \parallel ID_{V_i} \parallel P_{i,0} \parallel b \parallel P_{i,\text{start}}) + t \times u \) (mod \( q \))

\((u, s), K, R, ID_{H}, P_{i,0}, b, P_{i,\text{start}}\)

4. verify \( g^s = (vK^K)^{h(R||r_2||ID_{V_i}||P_{i,0}||b||P_{i,\text{start}})} \) \( u^u \) (mod \( p \))

compute \( CC_1 = [R \parallel r_2 \parallel K \parallel P_{i,\text{start}} \parallel b]_{K_{HV_i}} \)

\( CC_1, ID_H, ID_{V_i} \)
The Proposed Protocol

Authentication steps

5. decrypt $CC_1$

5.1. if $K$ can be found in $DBT_{new}$

- retrieve $\sigma_{new}$ from $DBT_{new}$ according to $K_{new}$
- $K_{old} \leftarrow K_{new}$, $\sigma_{old} \leftarrow \sigma_{new}$
- store $K_{old}, \sigma_{old}$ to $DBT_{old}$
- perform Step 5.2

else if $K$ can be found in $DBT_{old}$

- retrieve $\sigma_{old}$ from $DBT_{old}$ according to $K_{old}$
- perform Step 5.2

else reject the request and quit

5.2. choose $k'_{new}$

- compute $K'_{new} = g^{k'_{new}}, \sigma'_{new} = x + k'_{new}K'_{new}$
- $K_{new} \leftarrow K'_{new}$, $\sigma_{new} \leftarrow \sigma'_{new}$
- store $K_{new}, \sigma_{new}$ in $DBT_{new}$
- choose a random $r_3$

- compute $C_1 = h(R || r_2 || r_3 || \sigma_{old})$ as the session key
- $CC_2 = [R || r_3 || ID_{K_{i}} || \sigma_{new} || K_{new} || P_{i,start}]_{\sigma_{old}}$
- $CC_3 = [CC_2 || r_2 || R || C_1 || P_{i,start}]_{K_{i,HVR}}$
The Proposed Protocol

**Authentication steps**

6. decrypt $CC_3$
   - verify $r_2, R$

7. decrypt $CC_2$
   - compute $C_1 = h(R || r_2 || r_3 || \sigma_{old})$ as the session key
   - update $(K, \sigma)$ with $(K_{new}, \sigma_{new})$
The Proposed Protocol

Roaming Payment steps:

1. \([h^{(n-\lambda+1)}(r_1) \parallel P_{i,start+L} \parallel L]_{C_{\lambda}}\)

2. check if \(h(h^{(n-\lambda+1)}(r_1)) = h^{(n-\lambda+2)}(r_1)\)
   update the stored chain with \(h^{(n-\lambda+1)}(r_1)\)
   check if \(\text{start} + L \leq b\)
   check if \(h_2^{(L)}(P_{i,start+L}) = P_{i,start}\)
   update the stored payment chain with \(P_{i,start+L}\)
   set \(\text{start} = \text{start} + L\)
   update session key \(C_{\lambda+1} = h(h^{(n-\lambda+1)}(r_1), C_{\lambda})\)
   set \(\lambda = \lambda + 1\) and check if \(\lambda \leq n\)
Security Analysis And Discussion

- We explain that the proposed protocol is secure against the DoS attack and can be protected from forging payments.

- In Step 6 of the authentication phase, if the transmission messages from VLR to MU are interrupted by the attacker, MU cannot receive the updating information for \((K, \sigma)\).

- HLR stores the old keys of \((K_{\text{old}}, \sigma_{\text{old}})\) in the database, he can use these keys to identify MU.
Security Analysis And Discussion

- By using the hash chain and UOBT, the attacker cannot forge a unused payment since he has no idea of reversing the hash function.

- The anchor value is signed by MU (using the delegation key from HLR) upon applying roaming service, MU can only send the committed hash chain values to VLR in roaming payment that can effectively reduce the computational cost.
## Security Analysis And Discussion

### THE COMPARISON

<table>
<thead>
<tr>
<th></th>
<th>Unlinkability</th>
<th>Against DoS</th>
<th>Roaming</th>
<th>Anonymous Authentication</th>
<th>Payment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lee and Yeh[1]</td>
<td>no</td>
<td>-</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Lee et al.[2]</td>
<td>no</td>
<td>-</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Youn and Lim[3]</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Ours</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>


Roaming Payment Using Group Signatures

- Applying the group signature scheme and revocation list to the roaming authentication to achieve both anonymity and untraceability. It has low communication cost.

- Propose a new model of roaming payment protocol that felicitously integrates the roaming authentication by group signatures and PayWord-based micropayments. The group signature can be used to sign a commitment of a hash chains and the untraceability can be preserved in our protocol.

- Furthermore, we use Unbalanced One-way Binary Tree (UOBT) to provide an efficient mechanism for multiple vendors architecture and a convenient paying procedure in the fast authentication phase.

Information Security Lab., CSIE, NCYU, Taiwan, R.O.C.
Conclusions (1)

Important security considerations of WSNs

- Pairwise key
- Deployment knowledge
- Rekeying
- Resilience
- Heterogeneous Models
Conclusions (2)

Mobile payment with roaming by using delegation-based signature and group signature schemes.
References (1)

References (2)


References (3)


Information Security Lab., CSIE, NCYU, Taiwan, R.O.C.
References (4)

