Analysis and Evaluation of Key management Protocols for Wireless Sensor Networks

by

Sridhar Sarsan & Gopal Addada
PROTOCOLS ANALYZED


- Donggang Liu and Peng Ning (North Carolina State University, USA), “Location-Based Pairwise Key Establishments for Relatively Static Sensor Networks”. In Proceedings of ACM SASN WorkShop’03 at GMU
Protocols For Presentation

- *Basic Scheme* by Gligor et al.
- *q-composite scheme* by Perrig et al.
- *Multi-path reinforcement scheme* by Perrig et al.
- *Random Pair-wise scheme* by Perrig et al.
- *Grid-based pre-distribution scheme* by Peng Ning et al.
What’s different about Sensor nets?

- Low powered battery
- Low speed processor (Eg: 4 MHz 8-bit)
- Small size memory (Eg: 8 kB)
- Low communication range (Eg: 20 feet)
- Tamper-resistant
- Lack of pre-defined network topology
- Asymmetric systems (public-key) are not suitable
  Eg: On Motorola “DragonBall” processor
  - 1024-bit RSA consumes 42 mJ
  - 1024-bit AES consumes 0.104 mJ
- KDC-type third party key distribution schemes unsuitable for sensor nets
What’s different about Key-Management in Sensor nets?

- Requires *key pre-distribution*
- Naive approaches
  - single network-wide mission key
  - pair wise private keys between every set of nodes

**Trade offs**

- *Single mission key*: Compromise of single node compromises entire network
- Pair-wise keys between every pair of nodes requires storage of $(n-1)$ keys per node ( $n \rightarrow$ size of network)
Basic Scheme by Gligor

Key Management in Sensor Networks

$\text{shared keys}$

$\text{key ring 1}$

$\text{key ring 2}$

$\text{key pool}$

$(P \text{ keys})$

$k = 6$
Random key Pre-distribution Scheme by Perrig

- Consists of 3 independent schemes:
  - $q$-composite key pre-distribution scheme
  - multi-path key reinforcement scheme
  - random-pairwise keys scheme
**q-composite scheme**

Link key $K = \text{hash}( K_1 || K_2 || \ldots || K_{q'})$

$q'$ is the number of keys shared by 2 nodes.
Multi-path key Reinforcement scheme

\[ k' = k \oplus v_1 \oplus v_2 \oplus \ldots \oplus v_j \]
Modification to basic pairwise key scheme

Instead of “n-1” only “m” pairwise key are pre-deployed in each node

where \( m = n \times p \)

Each node only stores a random set of “np” pairwise keys (i.e. key ring size is np)

Each node ‘A’ s identity is matched up with \( m \) other randomly selected node ids

Each node ‘A’ is stored with \((B, K_{A,B}), (c, K_{A,c}), (D, K_{A,D})\)……
Polynomial Grid-based pre-distribution scheme

Background

- Polynomial based pre-distribution uses bivariate t-degree polynomials to establish shared pair-wise key
- \( f(x,y) = \sum a_{ij}x^iy^j \) where \( i = 0,1,2, ..., t \) and \( j = 0,1,2, ..., t \)
- Polynomial \( f(x,y) \) is chosen over finite field \( F_q \)
- Important property: \( f(x,y) = f(y,x) \)
- How shared key established using polynomials?
Polynomial Grid-based pre-distribution scheme

How to establish shared key?

Let ‘A’ and ‘B’ wants to establish shared key

- F(x,y) is a bivariate polynomial
- ‘A’ is loaded with its share on polynomial - F(A,y)
- ‘B’ is loaded with its share on polynomial - F(B,y)
- ‘A’ calculates shared key by evaluating polynomial share F(A,y) at point ‘B’ i.e. $K_{A,B} = F(A,B)$
- ‘B’ calculates shared key by evaluating polynomial share F(B,y) at point ‘A’ i.e. $K_{A,B} = F(B,A)$
- Shared key $K_{A,B} = F(A,B) = F(B,A)$
**Polynomial Grid-based pre-distribution scheme**

**Grid Based scheme (contd...)**

- Construct a *grid* of size $m \times m$
- $m = N^{1/2}$
- Place each sensor at a vertex of the *grid*
- Each Row $i$ assigned with a bivariate polynomial $f_r^i(x,y)$
- Each column $j$ assigned with a bivariate polynomial $f_c^j(x,y)$
- Each node at vertex $(i,j)$ is assigned with its shares on polynomials $f_r^i(x,y)$ and $f_c^j(x,y)$
ANALYSIS
<table>
<thead>
<tr>
<th>Scheme</th>
<th>Resiliency to Node capture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Scheme</td>
<td><img src="Chart" alt="Basic Scheme" /> □Less resilient as each compromised node compromises ‘k’ shared keys between other pairs of nodes</td>
</tr>
<tr>
<td>Q-composite scheme</td>
<td><img src="Chart" alt="Q-Composite Scheme" /> □To compromise a shared key more number of nodes are to be compromised</td>
</tr>
<tr>
<td>Multipath Key reinforcement</td>
<td><img src="Chart" alt="Multipath Key reinforcement" /> □Need to compromise all the paths between two nodes to compromise shared key</td>
</tr>
<tr>
<td>Random Pairwise key</td>
<td><img src="Chart" alt="Random Pairwise key" /> □Since pair-wise key is unique. Hence, perfectly resilient to node capture</td>
</tr>
<tr>
<td>Bi-variate Polynomial</td>
<td><img src="Chart" alt="Bi-variate Polynomial" /> □Unique key shared between any two nodes □Difficult to attacker</td>
</tr>
</tbody>
</table>
### Secure Node Revocation

<table>
<thead>
<tr>
<th>Basic Scheme</th>
<th>Q-Composite Scheme</th>
<th>Multipath Key reinforcement</th>
<th>Random Pairwise key</th>
<th>Bi-variate Polynomial</th>
</tr>
</thead>
</table>
| - Base station initiated node revocation  
  - Broadcasts key ring of compromised nodes  | - Base station initiated revocation  
  - Very slow due to high latency between nodes and base station  | - Always used in conjunction with either basic scheme or q-composite  
  - No special technique required  | - Voting scheme is used  
  - Fast  | - Simple as only ID of compromised ID is broadcasted  
  - Nodes remember revoked IDs |

**Key Management in Sensor Networks**
Secure Addition of Nodes

<table>
<thead>
<tr>
<th>Basic Scheme</th>
<th>Q-Composite Scheme</th>
<th>Multipath Key Reinforcement</th>
<th>Random Pairwise Key</th>
<th>Bi-variate Polynomial</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple as new node pre deployed with randomly chosen $k$ keys</td>
<td>Same as basic scheme</td>
<td>Always used in conjunction with either basic scheme or q-composite</td>
<td>simple matching with ‘m’ existing nodes and key generation</td>
<td>Limited number of nodes can be added using left over IDs after first deployment</td>
</tr>
</tbody>
</table>

Key Management in Sensor Networks
<table>
<thead>
<tr>
<th>Scheme</th>
<th>Basic Scheme</th>
<th>Q-Composite Scheme</th>
<th>Multipath Key reinforcement</th>
<th>Random Pairwise key</th>
<th>Bi-variate Polynomial</th>
</tr>
</thead>
</table>
|                               | Depends connectivity of random graphs                                      | Higher than basic scheme with same key pool and key ring sizes                     | Significant boost in network size when used in conjunction with basic scheme than with q-composite | Network size limited by initially generated node Ids                              | Depends on size of grid matrix
|                               | High if large key pool and key ring are provided                            |                                                    |                                                                                               |                                                                                   | Less than basic scheme                                                                      |

Depends connectivity of random graphs

High if large key pool and key ring are provided

Key Management in Sensor Networks
<table>
<thead>
<tr>
<th>Basic Scheme</th>
<th>Q-Composite Scheme</th>
<th>Multipath Key reinforcement</th>
<th>Random Pairwise key</th>
<th>Bi-variate Polynomial</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requires $k$ encryptions and $k^2$ decryptions for shared key discovery</td>
<td>Additional overhead than basic scheme due to link key calculation (hash operation)</td>
<td>Additional computation required for X-OR operation</td>
<td>keys are predeployed no additional computation required for key discovery</td>
<td>LARGE computational overhead for bi-variate polynomial evaluation</td>
</tr>
</tbody>
</table>

*Key Management in Sensor Networks*
Bivariate polynomial

Random Pairwise key

Multipath Keyreinforcement

q-composite scheme

Basic Scheme

Memory utilization

<table>
<thead>
<tr>
<th>Basic Scheme</th>
<th>Q-Composite Scheme</th>
<th>Multipath Key reinforcement</th>
<th>Random Pairwise key</th>
<th>Bi-variate Polynomial</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low compared to all other schemes analyzed</td>
<td>Same memory requirement as basic scheme</td>
<td>Little extra storage required for random number generated for each path</td>
<td>Additional storage required for storing vote messages and member hash values</td>
<td>Large memory utilized for storing polynomial shares of row and column</td>
</tr>
</tbody>
</table>

Key Management in Sensor Networks 20