



RKMP: A New Resilient Key Management Protocol for WSN

C. Bekara and M. Laurent-Maknavicius
Institut National des Télécommunications d'Evry
France
chakib.bekara@int-edu.eu

Outlines

- Introduction
- Security requirements for WSN
- Key management protocols for WSN
- A Resilient Key Management Protocol for WSN
 - Main idea, assumptions
 - Pair-wise key establishment and path-key establishment protocols
 - Resilience analysis of RKMP
- Comparison with existing works
- Conclusion and future works

Introduction

- Wireless sensor networks (WSN)
 - Infrastructure-less networks
 - Hundreds/Thousands of tiny devices (sensors)
 - Deployed any-where, and work unattended
 - Used in military, industrial and health areas.
- Sensors devices
 - Have limited energy, memory and computation resources
 - Non tamper-resistant devices (physical compromising)

Security requirements for WSN

- Authentication, confidentiality, access control and reply attacks protection
- Resistance to nodes compromising: communications between non-compromised nodes remain secure.
- **Resilience to nodes compromising:** Preventing an attacker from populating the network with **clones** of compromised nodes, or **new** injected nodes with **non-existing** IDs., using the key materials it retrieves from compromised nodes.

Key management protocols for WSN

- Keys creation, establishment, renewal and revocation.
- Key management protocols (KMP) must provide
 - All the needed security services for WSN
 - The less computation, transmission and memory overheads
- Few KMP for WSN are *resilient* to nodes compromising
 - Assume sensors are tamper-resistant devices
 - Assume a prior knowledge of nodes deployment
 - Introduce a heavy communication and computation overhead

Overview of some Key Management Protocols for WSN

Protocol	Description	Drawbacks
Bhuse et al., 2003	<p>Nodes self-destroy when they are under attack</p> <p>All nodes share a master key P</p> <p>Nodes self-organize on clusters</p> <p>Use of PKC to establish cluster keys</p>	<p>Self-destroying sensors are very expensive</p> <p>The use of PKC is highly energy consuming</p>
Liu et al., 2006	<p>Nodes know their location coordinates, and are static</p> <p>Some nodes act as key servers, and are not compromised</p> <p>Use of symmetric bivariate polynomials and PKC</p>	<p>GPS-enabled sensors are highly energy consuming</p> <p>Tamper-resistant sensors are expensive</p>

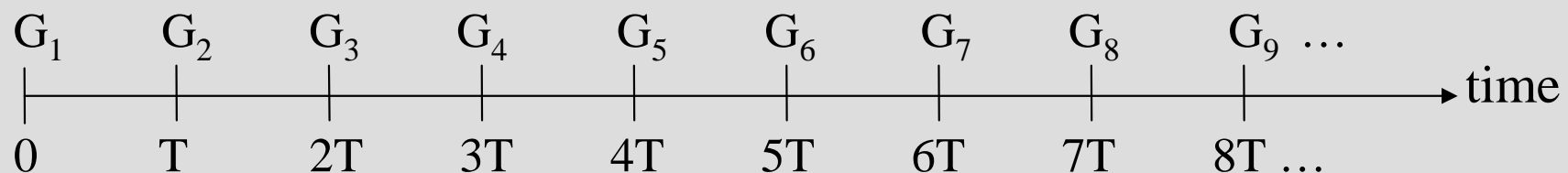
RKMP: A New Resilient Key Management Protocol for WSN

Main idea

- Group-based (generation-based) deployment of sensors
- Only nodes of the newly deployed generation are able to ask for key establishment.
- Restrict the time for key establishment to protect from the compromising of newly deployed node.
- Use bivariate symmetric polynomials for pair-wise key establishment, and to protect against the injection of nodes with fake IDs in the network

Assumptions

- At most n deployed generations: G_1, G_2, \dots, G_n . Static nodes.
- T_{est} : maximum time needed for pair-wise keys establishment
- $T_{comp} > T_{est}$: the minimum time for node compromising
- Time synchronization between nodes & BS, through a periodically broadcasted authenticated beacon
- Periodic scheduling of generations deployment, of periodicity $T > T_{est}$.



Initialization phase (before nodes deployment)

- The BS, creates a symmetric secret bivariate t -degree polynomial:

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

- The BS affects nodes of the network to the different generations
- The BS loads u in G_i with its unique secret polynomial share:

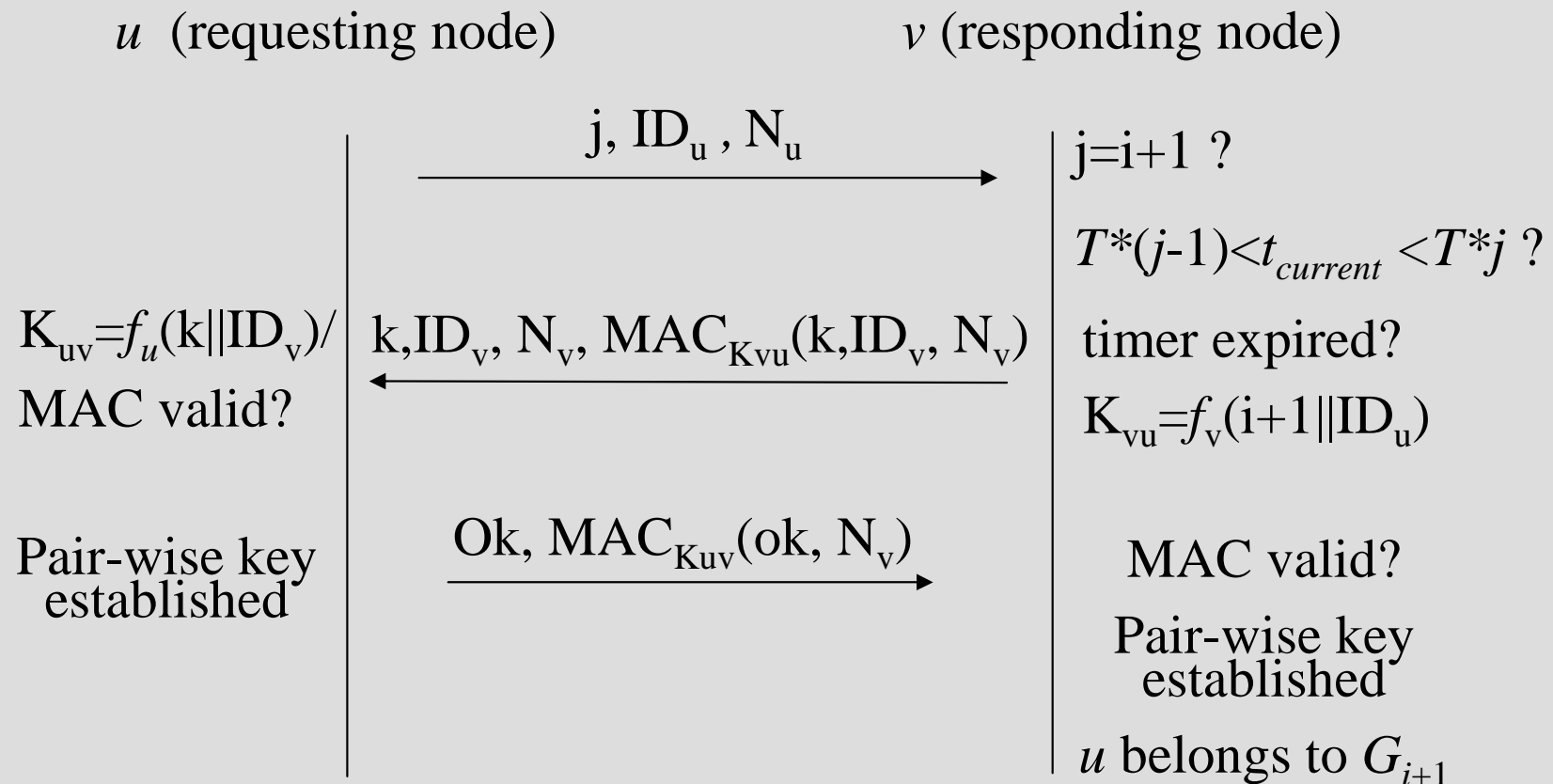
$$f_u(y) = f(i \| ID_u, y)$$

- If u in G_i and v in G_j : $K_{uv} = f_u(j \| ID_v) = f_v(i \| ID_u) = K_{vu}$

Pair-wise key establishment (1/2)

- G_1, \dots, G_i already deployed, G_{i+1} just deployed
- Each u in G_{i+1} sets a timer to T_{est} .
- Each v in $G_j, j \leq i$ sets a timer to t_{max} , with $t_{est} \leq t_{max} < t_{comp}$.
- Let u in G_j and v in G_k , where $k \leq i+1$, two neighboring nodes

Pair-wise key establishment (2/2)



Path-key establishment

- A pair-wise key established between non-neighboring nodes u and v
- Find a secure path of already established pair-wise keys

$$u \rightarrow N_1 \rightarrow N_2 \rightarrow N_3 \rightarrow \dots \rightarrow N_{w-1} \rightarrow N_w \rightarrow v$$

- Follow the pair-wise key establishment process, where all exchanged messages between u and v are authenticated through the path

Resilience analysis of RKMP (1/4)

- Resilience to nodes injecting with new IDs
 - We use a t -degree polynomial based key establishment protocol
 - ⇒ An attacker needs to compromise at least t nodes in the network, in-order to inject nodes with new IDs in the network
 - ⇒ The greater is the t value, the more resilient is our protocol.

Resilience analysis of RKMP (2/4)

- Resilience to nodes cloning
 - Three scenarios for key establishment according to generation of requesting/responding node
 - Either requesting or responding is a cloned node

<i>Requesting node</i>	<i>Respaning node</i>
New	New
New	Old
Old	Old/New



Resilience analysis of RKMP (3/4)

- New (cloned) vs New
 - u, v in G_{i+1}
 - An attacker compromises u in at least T_{comp}
 - v needs at most $T_{est} < T_{comp}$ for pair-wise keys establishment
- Old (cloned) vs Old/New
 - u in G_i and v in G_j , $j \leq i$ or $j > i$
 - An attacker compromises u
 - All nodes know the number of the highest deployed generation

Resilience analysis of RKMP (4/4)

- New vs Old (cloned)

- u in G_{i+1} , and v in G_j a cloned node, $j \leq i$




- $N1, N2$ two well-behaving nodes




- $u, v, N1, N2$ neighboring nodes

- v launches a silent attack and responds to u 's key establishment request

- v didn't ask for key establishment with $N1$ and $N2$

- $N1$ and $N2$ can detect it and inform u

Comparison with existing works

Protocol	Resilience	Tamper-resistant devices	Location information	Computation cost
Liu et al.	yes	yes (partially)	yes	high (use of PKC)
Bhuse et al.	yes	yes (all nodes)	no	medium (use of PKC)
RKMP	yes	no	no	low (symmetric cryptography)

Conclusion and futur works

- Our protocol is resilient to nodes cloning and injecting attacks
 - Without assuming tamper-resistant/self-destroying sensors
 - Without relying on any location information
 - Without introducing a heavy computation or communication costs
- Our protocol can provide a framework for trust management
- Adapt our protocol to handle dynamic WSN (nodes movements)
- Extend our protocol with a distributed revocation mechanism

Questions ?