

A Framework for Defending Against Node Compromises in Distributed Sensor Networks

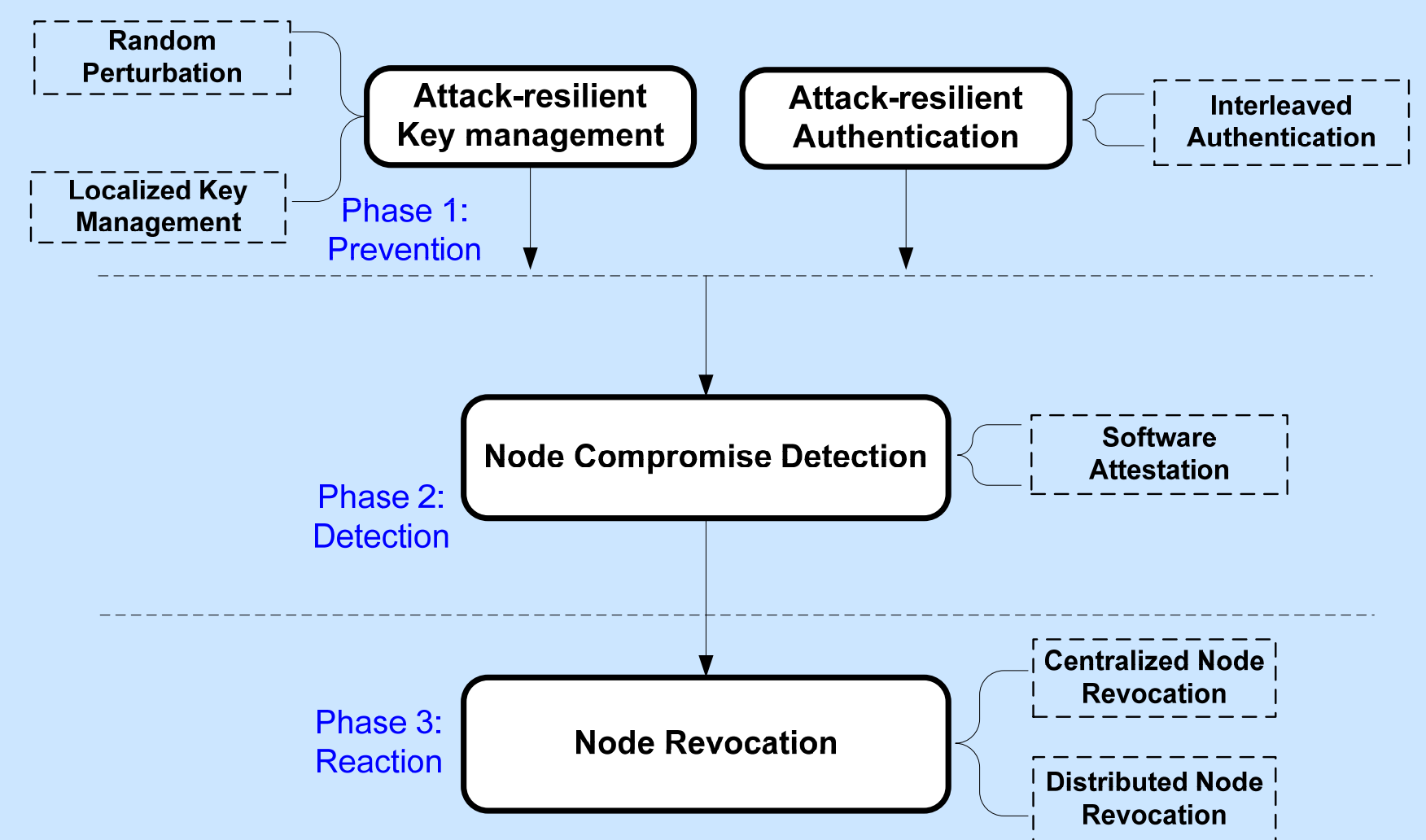


Sencun Zhu and Guohong Cao, The Pennsylvania State University, <http://mcn.cse.psu.edu> (CNS-0524156)

Project Description

The goal of this project is to develop a framework for defending against node compromises in wireless sensor networks. The framework consists of security mechanisms spanning three phases of defense: prevention, detection, and reaction. The PIs will investigate the following four issues:

- Attack resilient key management. By adding random perturbation, the proposed solution ensures any two nodes directly establish pairwise keys in a flexible and efficient way and resilient to collusion attacks.
- Investigates an interleaved hop-by-hop data authentication protocol which allows relaying nodes to detect and discard false data packets.
- Examines compromise detection techniques, where an efficient software attestation scheme is proposed to verify the suspects.
- Distributed group rekeying techniques which do not involve any on-site key server.



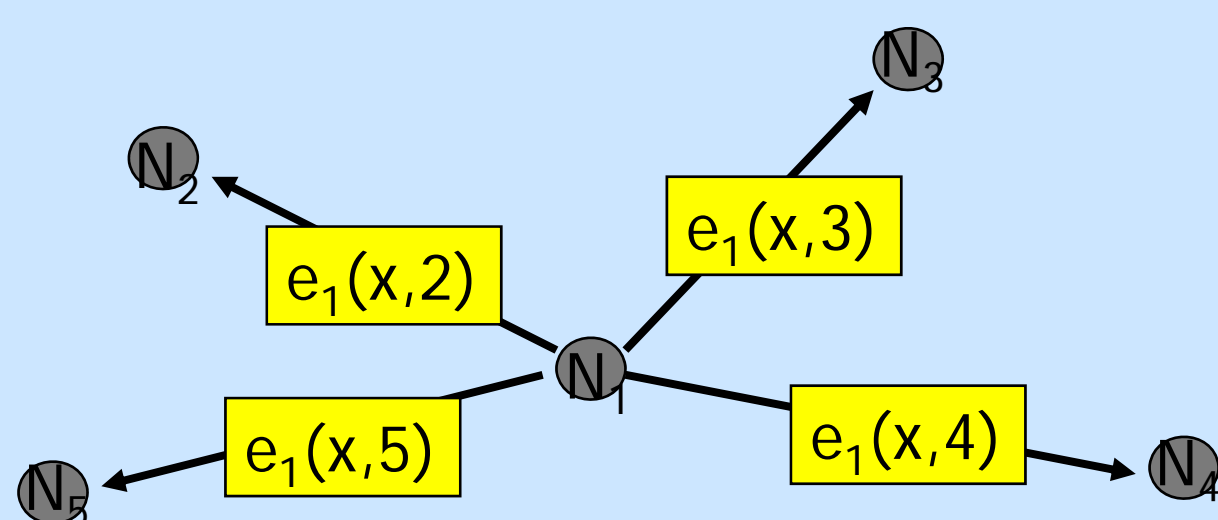
Approach and Impact

This project will provide fundamental security services covering key management, authentication, compromise detection, and revocation. These services are essential for the successful deployment of sensor networks. The proposed solutions either provide unique security properties or provide strong security not supported by the existing work.

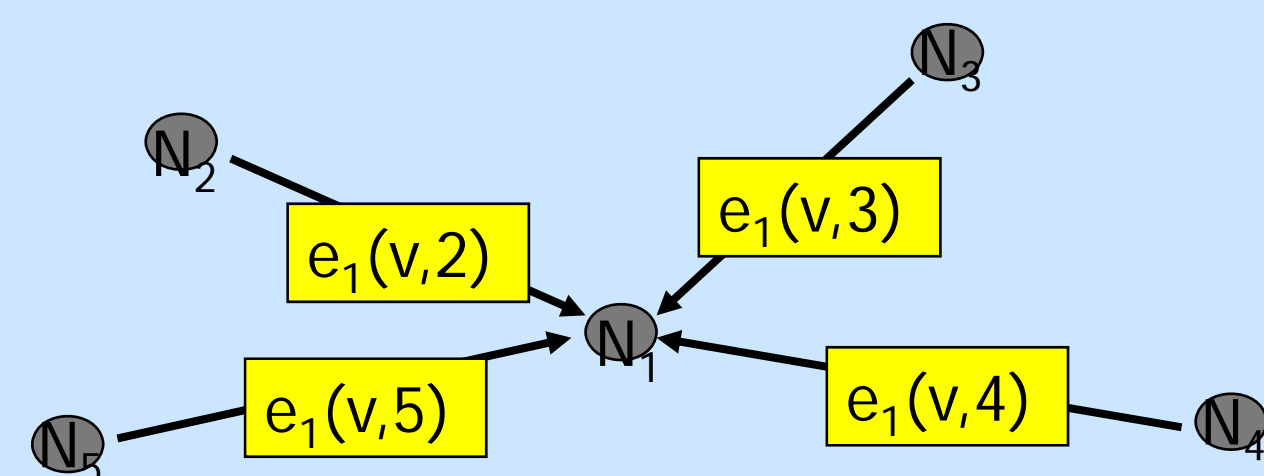
Technical Details of Group Rekeying

Before nodes are deployed, the current and all future group keys are preloaded. All the keys are represented by a polynomial $g(x)$, where $g(0)$ is the current key and $g(i)$ is the key of version i .

- Every node N_i encrypts its $g(x)$ using a randomly picked encryption polynomial $e_i(x,y)$.
- After encryption, every node only keeps the encrypted group key polynomial $g'_i(x)$, where $g'_i(x) = g(x) + e_i(x,i)$, and distributes the shares of $e_i(x,y)$ to its neighbors
- At the time for updating group key from version $v-1$ to version v , node N_i receives one share of $e_i(x,y)$ (i.e., $e_i(v,j)$) from each trusted neighbor N_j .
- Based on received shares, N_i can reconstruct $e_i(v,y)$, and compute the new group key $g_i(v) = g'_i(v) - e_i(v,i)$.



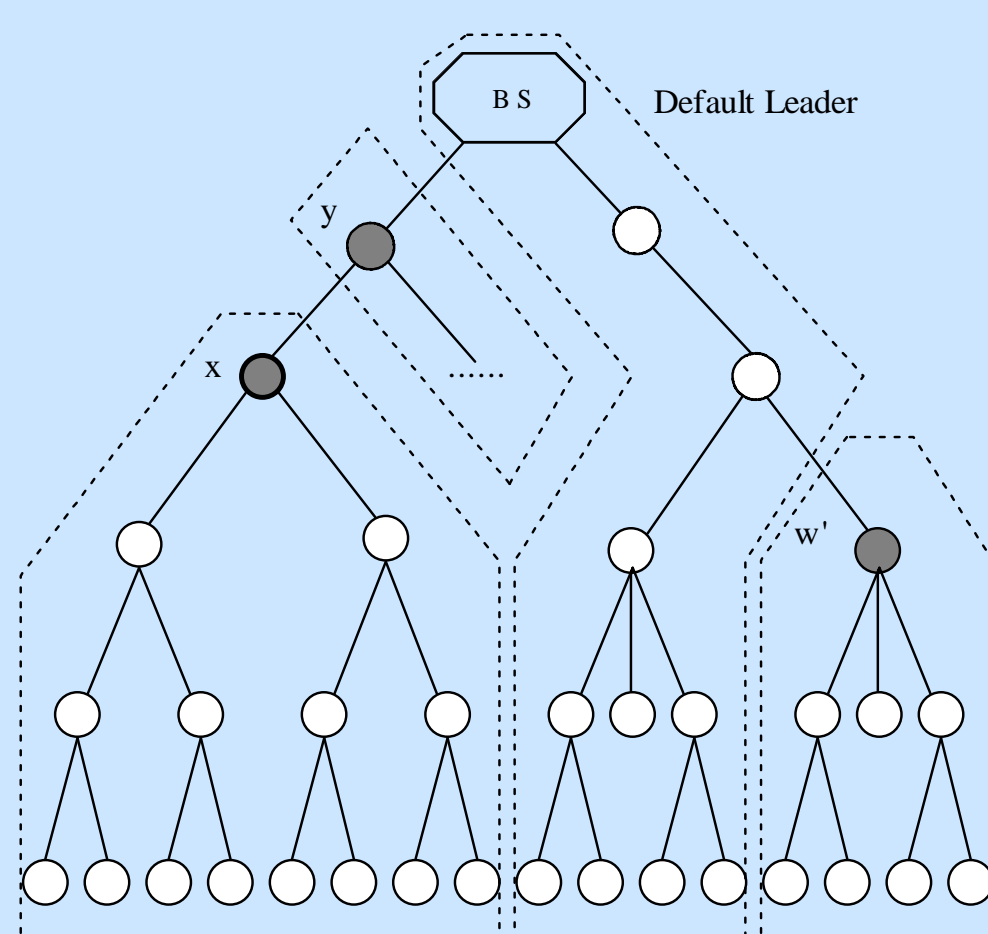
* N_1 keeps $g'_1(x)$, but removes $g_1(x)$ and $e_1(x,y)$.



N_1 computes the new group key $g_1(v) = g'_1(v) - e_1(v,1)$.

Secure Data Aggregation

- Aggregates computed by a higher-level node are from more low-level nodes
- If a compromised node is closer to BS, false value from it has more impact on the final result computed by BS



Our Solution: divide and conquer; commit and attest

- Tree construction and query dissemination
- Probabilistic grouping
 - Partition nodes in the tree into multiple logical groups (subtrees) of similar size
- Hop-by-hop aggregation
 - Each group generates a commitment which cannot be denied later
- Attestation between BS and suspicious groups
 - BS identifies abnormal groups from the set of received group commitments
 - Groups under suspicion prove the correctness of submitted commitments to BS
 - BS discards commitments from groups failing to support previous values when computing final aggregates