Secure Embedded Wireless Networks

Prof. Saurabh Bagchi
School of Electrical & Computer Engineering, Purdue University
September 22, 2010

Product/Service

- Communication and reprogramming protocol that can fit within the constraints of embedded wireless devices
- Secure ⇒ No eavesdropping, no masquerading, no compromised nodes
Advantages

- Based on Advanced Encryption Standard (AES) protocol already widely used and National Security Agency (NSA) approved
- Fastest AES encryption method available
- Provides secure communication; even when a node has been compromised (local monitoring)
- Science novelty:
  - Software optimizations, including compiler
  - Distributed software requiring no central controller
- Cost effective and easily modified since it is software-based

Schematic of Solution

- Wireless Communication
- Encrypt
- Decrypt
- Authenticate
- Overhear to detect suspicious pattern from a compromised node
Ad Hoc Wireless Networks (AHWN)

- Consist of a number of nodes that communicate with each other over a wireless channel
  - Each node operates not only as a host but also as a router
- Are easily deployable, decentralized, and self-configured
- Suitable for a variety of applications that avoid infrastructure because
  - Establishing infrastructure is impossible
    - E.g., battlefield, natural-disaster areas, natural habitat, etc.
  - Establishing infrastructure is not cost-effective
    - E.g., rural areas, temporary events such as a sport match or a conference

Security Vulnerability of AHWN

- Adversary can physically capture and tamper with ad hoc nodes
  - Ad hoc nodes are often deployed in insecure locations
    - Mesh routers are deployed on rooftops or attached to streetlights
    - Nodes may be deployed in a hostile environment, e.g., in a battlefield
  - Ad hoc nodes are typically low-cost devices with a lack of strong hardware protection (e.g., anti-tamper hardware)
- Compromised nodes can be exploited to launch a variety of attacks
  - Deny the network protocols such as the back-off rule at MAC layer or packet-relaying duty, etc.
  - Inject malicious traffic, e.g. worms, into networks
Motivation

• Behavior-based Detection
  – Nodes overhear communications in their neighborhood
  – Then, determine if the behaviors of the neighbors are legitimate

• Use of Multiple Channels and Multiple Radios
  – Nodes are equipped with multiple radios tuned on different non-overlapping channels
  – Can significantly increase the capacity of AHWNs

• Question that arises:
  In order to execute the behavior-based detection, on which channel does a node overhear?

Problem Statement

• Framework for behavior-based detection in multi-channel AHWNs
  – We use a set of trusted nodes (called monitoring nodes) to execute the behavior detection for monitoring the network
  – Monitoring nodes can be dedicated nodes for security purposes or reliable system nodes

• Optimal placement of monitoring nodes and selection of channels
  Where to place a given number of monitoring nodes among several possible locations of the network and which channels to tune their radios to, in order to maximize the detection coverage?

Equivalently, given a set of monitoring nodes deployed in the network,

How to choose a subset of monitoring nodes to be activated and the channels for the chosen monitoring nodes, in order to maximize the detection coverage?
Contributions

<table>
<thead>
<tr>
<th></th>
<th>GR-MCSC (existing work)</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AR: 1-1/e</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>MCSC</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>MCMC</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Centralized</td>
<td>GR-MCMC</td>
<td>PRA</td>
<td>DRA</td>
</tr>
<tr>
<td>AR: 0.5</td>
<td>AR: 1-1/e, Probabilistically</td>
<td>AR: 1-1/e, Deterministically</td>
<td></td>
</tr>
<tr>
<td><strong>MCMC</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Distributed</td>
<td>DGR-MCMC</td>
<td>DPRA</td>
<td>DDRA</td>
</tr>
<tr>
<td>AR: 0.5</td>
<td>AR: $\alpha \cdot (1-1/e)$, Probabilistically</td>
<td>AR: $\alpha \cdot (1-1/e)$, Deterministically</td>
<td></td>
</tr>
</tbody>
</table>

- Best approximation ratio for MCSC: 1-1/e
- Best approximation ratio for MCMC: 1-1/e
- $\alpha$ is an accuracy parameter that can be set to $\in (0, 1)$

Sensor Network Reprogramming

- Uploading new software while the nodes are *in situ*, embedded in their sensing environment
  - Fix software bugs
  - Adapt to changing user needs and environmental conditions in which the network is deployed
  - Shorten software development phase
  - Make software robust
  - Fine-tune algorithms
  - Complete application replacement
Requirements of Network Reprogramming

- For correctness, all nodes in the network should receive the code completely
  - Reliable dissemination using unreliable wireless channels is challenging
- For performance, code upload should minimize
  - *reprogramming time* so that sensor nodes can quickly resume their normal function
  - *reprogramming energy* spent in disseminating code through the network since sensor nodes have limited energy
- For security, malicious nodes should be unable to insert code
- Solution should fit within computation, memory, and bandwidth constraints of sensor nodes

Experiment

- Softbaugh DZ1611 Zigbee Demo Board
- ROM (Code) Size
  - `msp430-objdump`
- RAM (Memory) Usage
  - `msp430-gdb printing stack pointer`
- Execution Time
  - Set I/O line on start, clear on end
  - Measure on oscilloscope
AES Comparison

- Improved both speed and code size (RAM unknown)
- Note that our measurements seem to have varied significantly from published numbers in some cases

<table>
<thead>
<tr>
<th>Implementation</th>
<th>Reference paper</th>
<th>Measured ROM Usage</th>
<th>Published ROM Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[6]</td>
<td>5968 bytes</td>
<td>n/a</td>
</tr>
<tr>
<td>2</td>
<td>[12]</td>
<td>6780 bytes</td>
<td>12616 bytes</td>
</tr>
<tr>
<td>3</td>
<td>[14]</td>
<td>6848 bytes</td>
<td>3322 bytes</td>
</tr>
<tr>
<td>4</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>5</td>
<td>Our implementation</td>
<td>5160 bytes</td>
<td>n/a</td>
</tr>
</tbody>
</table>

Our Work

Internal Malicious Node Detection

- Malicious node $M$ – has all the cryptographic keys – thwarting communication to base node $B$
Applications

- Military/Homeland Security
  - Secure ad-hoc networking
  - Secure wide area networking
  - Emergency/disaster communications
- Corporate entities where secure wireless communication is a concern
- Hospitals/pharmacies
- E-commerce

Team

- Faculty principal investigator: Prof. Saurabh Bagchi
- Technical staff: Aaron Ault
- Students: Shammi Didla (undergraduate), DongHoon Shin, Matthew Tan Creti
- Alumni students: Rajesh Panta
Next Steps

- Patent issued
- Development timelines:
  - Prototype developed & tested, including comparative evaluation
  - Commercial viability = ~1 man year
- Future plans for development
  - Software for the two parts (secure communication, detection of compromised nodes) needs to be ported to a common platform

Summary

- Secure communication between nodes using AES that is faster than the wireless network speed
- Detection and isolation of internal compromised node through decentralized protocol
Opportunity

- Licensing or start-up opportunity
  - Contact Hilton Turner, Technology Manager
  - 765-496-7539
  - haturner@prf.org