Verification in Sensor Networks

Frank Werner

KeY Symposium 2009
Speyer, 19. May 2009

Motivation

Frank Werner
Logik und Formale Methoden
Prof. Schmitt

model based simulation
- not exhaustive
- difficult to estimate coverage
- easy to conduct

Formal Methods

testing
- adaptable, depending on internals (white-box, etc)
- analysis of fully implemented final product

theorem proving
- lead of proofs time consuming
- non-discrete values, not limited to natural numbers

model checking
- Concurrent system modeling
- exhaustive
- huge state space → counter measures (PO, sym. Red.,…)

Sensor Networks

cheap in price
- today, $50 to $100 per node,
- re-engineering, Moore's Law, volume production:
  < $5 over the next five years. [Aberer, Karl 2006]

Secure Query Dissemination with AQF

probabilistic authenticator
- faked query: accepted with probability $p_f$
- Still secure in the presence of corrupted nodes
- gradual security: $p_f$ can be adjusted by user
- longer authenticator
  - $p_f$ smaller
  - more authentic results

Overview

• AQF (authenticated query flooding)
• sAQF (simple authenticated query flooding)
  Correctness, safety, liveness, energy

Concast Protocols
- Automatic Model Generation
- Software Behavior Model Verification
ZeuS Properties of Interest

- assure using formal methods:
  - efficiency:
    - is the application of AQF efficient?
    - compare with traditional strategies
  - safety:
    - if a SN accepts q as legitimate, then with probability p, it is legitimate
  - liveness:
    - every legitimate query will be received by every node with probability p

ZeuS Markov Model: Efficiency

use specific energy in each state
- p_c: probability of forwarding a fake query
- p_l: prob. packet is lost

ZeuS Security/Energy Tradeoff

ZeuS AQF Liveness

Liveness Property
- base station and nodes must share keys
- Not every legitimate query is reached by every node
- share no key
- share >= 3 keys

ZeuS New Approach: sAQF

- Design of a new algorithm simpleAQF
  - reduce key pool
  - use all keys of key pool for authenticator construction
  - better suitable for small networks
  - Safety and Liveness properties trivially hold
  - Show Correctness for forwarding probability

\[
P_{\text{AQF}} = \left(1 - \frac{1}{2}(1 - \frac{1}{k})\right)^n\]

- k keys per node
- l size keypool
- n compromised nodes
- p probability of forwarding forged query

ZeuS sAQF: Correctness (1)
Problem
- the derived formula from the original work is wrong
  - anticipates independency of two parameters (k's correlate)
  - wrong results
- exact formalism is derived using a hyper geometric distribution
  - proof its correctness

Correctness (2)

Formal Methods in WSN - Overview
- AQF (authenticated query flooding)
- sAQF (simple authenticated query flooding)
  - Correctness, safety, liveness, energy

Concast Protocols
- Automatic Model Generation
- Software Behavior Model Verification

Embedded Software Verification
- Automated model generation from TinyOS
  - abstraction to ANSI C (nesC Compiler, abstraction platform)
    - Behavior model of a sensor node (5k loc)
    - Property verification with CBMC

Verification Tasks
- STATUS packets are correctly processed
  - STATUS_SET(num_nodes,k,p)
  - STATUS_SETAGG(node_id,parent_id)
  - STATUS_GO(interval)
- ESAWN packets are correctly processed
  - ESAWN(from=3|value=7|from=4|value=4)
  - forall i <= w: (agg=aggni?true:ALARM(8))
  - if(alarm at node i): alarm broadcasted to n_{i-1} ... n_{i-w-1} nodes

CBMC: Results

<table>
<thead>
<tr>
<th>packet type</th>
<th>check</th>
<th>successful</th>
<th>injected error</th>
<th># claims</th>
</tr>
</thead>
<tbody>
<tr>
<td>status.set</td>
<td>unwinding</td>
<td>yes</td>
<td>found</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>bounds</td>
<td>yes</td>
<td>-</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>pointer</td>
<td>no</td>
<td>-</td>
<td>181</td>
</tr>
<tr>
<td>status.setag</td>
<td>unwinding</td>
<td>yes</td>
<td>-</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>assertions</td>
<td>no</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>bounds</td>
<td>yes</td>
<td>-</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>pointer</td>
<td>no</td>
<td>-</td>
<td>177</td>
</tr>
<tr>
<td>status.go</td>
<td>unwinding</td>
<td>yes</td>
<td>-</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>assertions</td>
<td>yes</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>bounds</td>
<td>yes</td>
<td>-</td>
<td>59</td>
</tr>
<tr>
<td></td>
<td>pointer</td>
<td>no</td>
<td>-</td>
<td>175</td>
</tr>
</tbody>
</table>

and none of the ESAWN properties could be proved
- missing treatment of formalisms in CBMC
- receiving/sending is complex even in the SN world

Questions..?