UML Modeling and Formal Verification of Secure Group Communication Protocols

P. de Saqui-Sannes, T. Villemur, B. Fontan, S. Mota,
M.S. Bouassida, N. Chridi, I. Chrisment, L. Vigneron

pdss@laas.fr

UML-FM’09
Rio de Janeiro, RJ, Brasil
8/12/2009
Working together, a PMR terminal in hand
Challenges and Bottlenecks

• Secured Group Communication system (SGC)
  – Key-based security
  – Group management
    • Dynamic group
    • Hierarchy

• Architecture design and validation
  – Protocol mechanisms proposed by TelecomParis, UTC, LORIA
  – Model-based validation
    • Performance evaluation by LORIA (NS) and UTC (Matlab)
    • Security flaws detection (AVISPA @ LORIA)
    • Deadline violation detection (TURTLE @ LAAS-CNRS)

• A verification-centric UML method for SGC design
  – Reuse of AVISPA and TURTLE
UML method for SGC design

Before using UML

Using UML

Design (behaviors)

Design (architecture)

Requirement capture

Formal verification
- security: AVISPA
- timeliness: TURTLE

Analysis (use-case)

Analysis (scenarios)
Pattern – Key Management
Pattern – Group Management
AVISPA

High Level Protocol Specification Language (HLPSL)

HLPSL2IF Translator

Intermediate Format (IF)

On-the-Fly Model-Checker OFMC
CL-based Attack Searcher AtSe
SAT-based Model-Checker SATMC
Tree Automated based Protocol Analyser TA4SP

Answer
<table>
<thead>
<tr>
<th>Requirement capture</th>
<th>SysML requirement diagrams, chronograms</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Automatic synthesis of observers</td>
</tr>
<tr>
<td>Use-case driven analysis, scenarios</td>
<td>Rendezvous and FIFO, Time intervals</td>
</tr>
<tr>
<td></td>
<td>Formal verification (RTL, CADP, UPPAL)</td>
</tr>
<tr>
<td></td>
<td>Automatic synthesis of design diagrams</td>
</tr>
<tr>
<td>Object-oriented design Architecture, Behaviors</td>
<td>Object composition (process algebra)</td>
</tr>
<tr>
<td></td>
<td>Synchronization actions, Time intervals</td>
</tr>
<tr>
<td></td>
<td>Formal verification (RTL, CADP, UPPAL)</td>
</tr>
<tr>
<td>Rapid prototyping Components, deployment nodes</td>
<td>Java annotations</td>
</tr>
<tr>
<td></td>
<td>Java and System C code generators</td>
</tr>
</tbody>
</table>
The Upgrade service

Security flaw
- Man in the Middle attack
- fixed

Does the system meet the deadline?
- low rate PMR: no
  Forget it!
- medium rate PMR: yes
  1 requirement relaxed
  implement it!
# Upgrade: Formal Verification

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Limit duration (ms)</th>
<th>Upgrade protocol on average-rate network (Execution time 331 ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detecting an integrity violation</td>
<td>10 000</td>
<td>Widely validated</td>
</tr>
<tr>
<td>Detecting a replay</td>
<td>10 000</td>
<td>Widely validated</td>
</tr>
<tr>
<td>Accessing to a multimedia group</td>
<td>350</td>
<td>Shortly validated</td>
</tr>
<tr>
<td>Accessing to textual message groups</td>
<td>60 000</td>
<td>Very widely validated</td>
</tr>
</tbody>
</table>
Conclusions

• A method for Secure Group Communication system design
  – Requirement, analysis and design patterns
  – A verification-centric method
  – An annotated UML model with security and temporal requirements

• SAFECAST: joint use of UML and formal verification tools
  – UML has made communication among partners easier than expected
  – Acknowledged benefits of formal verification
    • Security flaws were detected and fixed (HLPSL, AVISPA)
    • Secured configurations were eliminated because of unmet deadlines (TURTLE, TTool and RTL)
    • EADS has saved development time

• Future work
  – The method is not restricted to SAFECAST system
  – Audio-video multicast streaming application within ad hoc networks
    • Quality of Service
  – TURTLE & network coding
Acknowledgements

SAFECAST partners

AVISPA developers

TTool developer