INTRUSION TOLERANCE BASED ON ARCHITECTURAL HYBRIDIZATION

Miguel Nuno Dias Alves Pupo Correia
Faculdade de Ciências da Universidade de Lisboa
December 2003

Summary

• Intrusion tolerance
• Architectural hybridization
• TTCB
• Reliable Multicast
• Consensus
• Membership Service
• View-Synchronous Atomic Multicast
• Conclusions and future work
Intrusion Tolerance

- Security typically tries to prevent intrusions from happening
- Intrusion tolerance applies the fault tolerance paradigm to security
  - Assume the system has vulnerabilities
  - Attacks on components can happen and cause intrusions
  - Attacks, intrusions and vulnerabilities are faults
  - Tolerate these faults to ensure that the system remains secure and operational

Asynchronous Byzantine model

- Processes
  - Can fail arbitrarily: stop, disobey the protocol, send contradictory messages, collude with other malicious processes
- Network
  - Messages can be corrupted due to accidental faults
  - An attacker can modify, delete, replay or insert messages
- Time
  - No bounds on processing delays
  - No bounds on communication delays
  - No bounds on clock drift rates
Architectural Hybridization

- Most of the system is asynchronous Byzantine
- There is a subsystem built in such a way that it has stronger properties

TTCB Model

- The thesis studies the notion of architectural hybridization with the Trusted Timely Computing Base
  - Distributed component with its own network/channel
  - Secure (can only fail by crashing)
  - Real-time (there are known delays for its operations)
**TTCB and Intrusion Tolerance**

- The TTCB serves to support the execution of intrusion-tolerant protocols/applications
  - They run mostly in the payload system that can be attacked
  - They use the TTCB to execute some critical steps securely

**TTCB Services**

- The TTCB provides a limited set of services
- Security-related services
  - Trusted Block Agreement (TBA)
  - Local Authentication
  - Trusted Random Number Generation
- Time-related services
  - Trusted Absolute Timestamping
  - Trusted Duration Measurement
  - Trusted Timely Execution
  - Trusted Timing Failure Detection
COTS-based TTCB

- First implementation of the TTCB
  - PCs
  - Real-time kernel (Real-Time Linux or RTAI)
  - Fast-Ethernet
- Basic idea
  - Local TTCB is a set of tasks inside the kernel
  - The kernel is protected
  - The control network is a second LAN (secure inside premises)
- Reasonable assumption coverage
- Easy to deploy and test

Validation of the Approach

- Design of intrusion-tolerant middleware components using the TTCB as a runtime support component
- Solve two classical distributed systems problems
  - Reliable multicast
  - Consensus
- A group communication system
  - Membership service
  - View-synchronous atomic multicast
**Reliable Multicast**

- **Objective**
  - To deliver the same messages to all correct processes

- **Protocol**
  - Uses the TBA to deliver a trustworthy hash of the message to all processes

**Benefits**

- **Resilience**
  - Reliable Multicast with Byzantine faults requires...
    - Asynchronous systems: $n > 3f$
    - Synchronous systems: $n \geq f$
    - Architectural hybridization with the TTCB: also $n \geq f$

- **No public-key cryptography**
- **Low message complexity**
- **Low latency**
  - ~9-10 ms
  - Previous works: ~50 ms
Consensus

- **Objective**
  - To agree on a value despite a number of process failures
- **Protocol**
  - Run TBAs until enough processes give a value, then choose the value of the majority
- **Benefits**
  - Low latency degree: 1 or 2 against at least 4 in previous works
  - Low message complexity
  - No public-key cryptography

Membership Service

- **Objective**
  - To supply the list of the current group members
  - To allow members to join/leave; remove failed members
- **Protocol**
  - Multicasts information about the operations
  - Uses the TBA to make agreement on a new membership
- **Benefits**
  - Agreement is made cooperatively by all correct processes
  - Arguably good performance (no public-key cryptography)
    - ~10-20 ms
    - ~200-950 ms in previous works
**Atomic Multicast**

- **Objective**
  - To deliver the same messages in the same order in the same view to all correct groups members

- **Protocol**
  - Variation of the reliable multicast delivers the same messages to the members
  - Variation of the membership makes agreement on the messages to deliver

- **Benefits**
  - Essentially the same as for the membership service

---

**Conclusions**

- Principles of modeling architectural-hybrid distributed systems
- Concept of architectural hybridization and its illustration using the TTCB
- New protocol design methods for architectural-hybrid systems using two classical problems: reliable multicast and consensus
- The design and evaluation of an intrusion-tolerant group communication system
- A proof-of-concept implementation of the TTCB using COTS components
Future Work

- Implementation of the TTCB using *hardware appliance boards*; large-scale TTCB on WANs
- Exploring *new benefits* of architectural hybridization
  - State machine approach using \( n > 2f \) replicas
- Study TTCB-like components simpler and with weaker properties
- Building intrusion-tolerant systems using TTCB-based protocols

Publications