SECURE AND FAULT TOLERANT VOTING IN DISTRIBUTED INFORMATION SYSTEMS
14 September 2001

Shambhu Upadhyaya
SPIDER Lab
CSE Department
University at Buffalo
Overview

- Problem Addressed
  - Replication and voting for fault tolerance
  - Secure voting is essential (AFRL/IF & University at Buffalo)

- Techniques Used
  - Distributed monitoring and voter isolation
What About Voting?

Execute an “Al Gore” - ithm?

Just get an answer and “stop beating around the Bush”?
Requirements of Secure Voting

- Eliminating single point of failure in the voting complex
- Correctness and completion of the new voting algorithm -- a formal methodology
- Implementation and application
2-phase CommitProtocols

• Voters exchange their votes and determine a majority result; an arbitrary voter then commits that result to the user

• Fault-Tolerance: what if the committing voter experiences a failure during committal?

• Security: what if the committing voter is compromised?
2-Phase Commit Protocol

- Phase 1: processors distribute their results and vote on them such that each processor determines the majority.

User waits for majority result

User is sent majority result - LAUNCH

- Phase 2: processor in the majority commits result to the user.
Safe Operation

Time duration of committal
Danger of 2-Phase Commit Protocol

- Phase 1: processors distribute their results and vote on them such that each processor determines the majority.

User waits for majority result

- Phase 2: processor in the majority commits result to the user.

User is sent malicious result - SELF-DESTRUCT
Timed-Buffer Distributed Voting
(Ref: Hardekopf, Kwiat, Upadhyaya, IEEE Aero 2001)

- Addresses “last mile” of distributed voting
- Buffer until “silence is consent”
- Reverses 2-phase commit protocol
  - Instead of voting then committing - commits first (to buffer) then votes (period of dissension)
  - Prevents disastrous commit phase - unlikely for classical fault tolerance but not information attack

Integrity restored and buffer released

Suspect results buffered
• Used TLA+ to write a formal specification of the algorithm

• Used the specification and TLA to prove both partial correctness and termination
Implementation and Application

- Being transferred to Assured Communications Research Center
  - Instantiation of TB-DVA in Configurable Protocol Stack for the Software Radio Development System (SoRDS)
ACRC Application of TB-DVA

SECURE SERVER

GATEWAY

WIRELESS CLIENT

SECURE WIRED LINK

SECURE WIRELESS LINK

SECURE DATA IS EXPOSED

(when translated from IP standards to wireless and vice-a-versa)

• Apply fault tolerance techniques to protect, detect, and react to attacks and enable service restoration
Multiple Wireless Hubs

Timed-buffer distributed voting algorithm assures integrity, availability, and helps curtail confidentiality leaks.
Conclusion

• Problems Addressed
  – Security enhancement in distributed voting

• Techniques Used
  – Guaranteeing owner’s intended result by distributed monitoring and voter isolation