Securing Distributed Adaptation

Jun Li, Mark Yarvis, and Peter Reiher
University of California,
Los Angeles
{ljun, yarvis, reiher}@cs.ucla.edu

Presentation by: Mark Yarvis
Adapting with Conductor
Distributed Adaptation

Requires stream access!

Prioritize
Shortest job first? Text before images?
Characteristics of Conductor

• Wide variety of possible adaptations
  – Compress, encrypt, prefetch
  – Distill a video stream to black-and-white
  – Remove and store e-mail attachments
  – Power down wireless interface during predicted query response latency

• Distributed planning architecture
  – Efficiently address end-to-end network conditions
  – Prevent adaptation conflicts
  – Security is needed to ensure adaptation is exactly as desired
What should be protected?

• Protect the secrecy and integrity of the user data
  – But, still allow adaptation
• Protect the nodes from misbehaving adaptors
  – Leverage existing research
• Protect the user from misbehaving nodes
  – Allow only desired adaptations
What nodes can we trust?

- Various levels of trust possible
  - See or modify plain text
  - See or modify encrypted text
  - None
- Implicitly trust endpoints (typically)
- Trusting other nodes
  - Requires some type of authentication
  - Static list, dynamic trust model
Complications of Distributed Adaptation

- Users require different levels of security
- Adaptation may span administrative domains
  - No ubiquitous authentication infrastructure
  - Many choices; how do we agree securely?
- Must allow *limited* stream access within the network
  - Only desired adaptations
  - Typically restricted to trusted nodes
Adapting with Conductor

Node A  Node B  Node C  Node D

Formulate Plan

Time

Node Information  The Plan  User Data
SecurityinConductor

• Determine which nodes to trust
  – Select an authentication mechanism
  – Authenticate each node to the planner
  – Authenticate the planner to each node
• Protect planning from untrusted nodes
• Adapt plaintext only at trusted nodes
• Encrypt user data between trusted nodes
Security Architecture

Planning Messages → Planning Module → Planning Messages

Security Module

Security Module A
Security Module B
Security Module C
Security Module D
Authentication

• Goals:
  – Verifiable node identity
  – Digital signature capability

• Plug-in modules provide various authentication schemes
  – Null
  – Public-key based: tree, chain of trust
  – Kerberos based
Tree-based Authentication
Chain-of-trust Authentication

A (CA-1) -> B (CA-2) -> C (CA-3) -> D (CA-4)

CA-1 -> CA-2
CA-2 -> CA-3
CA-3 -> CA-4

A (CA-1) -> B (CA-2) -> C (CA-3) -> D (CA-4)
Selecting an Authentication Scheme

• The client node selects the desired scheme

• Conductor must ensure that all nodes use the desired scheme
  – No external mechanism available
  – Nodes must not be fooled into using null security
  – Not sufficient for the client to sign its request
SecurePlanning

Node A       Node B       Node C       Node D

Authenticate

Verify & Select

Formulate Plan

Authentication Scheme
Signed Authentication Scheme
Authentication Information
Signed Plan
Signed Node Information
SecurePlanning

• Protocol features
  – Ensures trusted nodes (and their planning information) can be identified
  – Ensures the specified authentication scheme was used by the planner
  – Ensures an authentic plan is distributed
  – Self selecting and self enforcing
• A random session id is used to prevent replay attacks
• Still required: protection for the user data …
VirtualLinkEncryption

• Allow plaintext adaptation only at trusted nodes
• Encrypt between points of adaptation
  – Use encryption adaptors
• Requires:
  – Selection of trusted nodes
  – Encryption adaptor selection and deployment
  – Secure key distribution
SecureKeyDistribution

- Each deployed encryption adaptor requires a particular type of key
- Several keys may be required per session
  - Typically one of each type
- The planner uses adaptor code to generate a set of keys
- Each key is encrypted and signed for each recipient node
  - Use public/private key or shared secret from authentication
SecureKeyDistribution
Implementation

- Fully integrated into Conductor
- Security modules
  - Null
  - RSA/SHA-1: static, tree, chain-of-trust
- Encryption/decryption adaptors
  - DES
- Environment
  - Cryptix, Java, Linux
The Cost of Secure Planning

- Increased setup latency
- Increased bandwidth use

- Compare Conductor in four cases:
  - No security
  - Null authentication
  - Tree-based authentication
    - Tree height = 3
  - Chain-of-trust authentication
    - With maximum chain length
Plan Setup Latency

- None
- Null
- Tree
- Chain w/ cache
Bandwidth Used

![Graph showing bandwidth used vs. number of conductors (number of links + 1).](image)

- **None**
- **Null**
- **Tree**
- **Chain** (with optimal caching)
Conclusion

- Adaptation is a powerful capability that introduces new avenues of attack
- Open architectures require comprehensive security
  - Protect the user data
  - Protect the node from malicious users
  - Protect the user from malicious nodes
- Conductor provides a flexible security mechanism for distributed adaptation