Context-Sensitive Access Control for Open Multi-Agent Systems

Christine Julien
Jamie Payton
Gruia-Catalin Roman

SELMAS
May 24, 2004

Washington University in St. Louis
School of Engineering & Applied Science
The Nature of Open MAS

- Highly dynamic nature
- Need for scalability
- Unpredictability
- Transient interactions
- Limited guarantees

- *Epitomized by ad hoc mobile networks*
A Computational Model

Host 1

Agent 1

Profile
Location
Access Control Function
Local Tuple Space

Agent 2

Profile
Location
Access Control Function
Local Tuple Space

Host 2

Profile
Location
Control Function

Agent 1
Agent 2
Host 1
Host 2
Coordination in MAS

- Linda model popularized use of tuple spaces for agent coordination
- Extensions for use in open MAS
  - MARS: logically mobile agents coordinate through tuple spaces at stationary hosts
  - LIME, EgoSpaces: logically mobile agents move fluidly over physically mobile hosts
- Coordination through tuple spaces associated with networked components
The Basics of Tuple Spaces

- Tuples contain fields that store data values
- Patterns select tuples by constraining fields
- Agents issue operations on tuple spaces
  - Tuple generation creates tuples either locally or remotely
  - Tuple access through variations of `rd` and `in` uses patterns to select tuples either locally or remotely
Security in Open MAS

- Protecting hosts from malicious agents
  - Public-key cryptography for authenticating incoming agents (D’Agents)
- Protecting agents from tampering hosts
  - Undetachable threshold signatures
- Securing Data
  - Ensuring data integrity
    - Encrypted communication within data spaces (Yalta)
  - Controlling data access
Common Access Control Strategies

- Administrator-based solutions (e.g., access control matrices)
  - Rows and columns correspond to users and objects
  - Cell indicates user’s rights on object
- Do not scale well
  - Large numbers of objects or users lead to unmanageable matrices
- Do not handle unpredictability
  - A priori knowledge of participants and data is required
Access Control Function

• Basic idea:
  - Requesting agent provides information about itself with access request
  - Owner of data uses this information and other context information to determine access
• Allows individualized control
• Builds on tuple/pattern foundation
• Access decision can be context-sensitive
Access Control Function

\[
\text{access} = \text{ACF}(\text{credentials}_r, \text{operation}, \text{tuple}, \text{pattern}, \text{profile}_o)
\]

- **Credentials**
  - Requesting agent provides information about itself
- **Operation**
  - Type of access is important (e.g., read vs. delete)
- **Tuple**
  - Considering tuple provides a fine granularity
- **Pattern**
  - Knowledge of data structure can be important
- **Profile**
  - Current state of owner can affect access decision
Personalization

• Owner agents control access to their data
  ✷ Decentralized mechanism for open environments

• Requesting agents control how much information is provided
  ✷ Credentials are a chosen subset of the profile
Dynamic Adaptation

- Access control policy can depend on aspects of the environment
  - Owner’s profile
- Owner can change the policy over time
- Agents need not know in advance who they will encounter
  - Access decisions not necessarily based on identity
AccessControlPolicy policy =
    new AccessControlPolicy();
policy.addPropertyConstraint("Passphrase",
    new EquivalencyConstraint(encryptedPhrase));
policy.addPermittedOperation("RDP");
acf.addPolicy(policy);

Credentials c =
    new Credentials(getAgentID());
c.addProperty("Passphrase", encryptedPhrase);
Conclusions

• Open multi-agent systems require dynamic access control mechanisms
  ❖ Must allow coordinating agents to adapt to changing conditions
• Traditional approaches to access control do not move directly to open MAS
  ❖ Require a priori knowledge
  ❖ Do not address scalability concerns in open systems
• Our mechanism places access control decisions in an individual agent’s control
Questions?

• Christine Julien
  - julien@wustl.edu
  - http://www.cse.wustl.edu/~julien

• Mobile Computing Laboratory
  - http://www.cse.wustl.edu/mobilab