Distributed Authorization Framework for Mobile Agents

G. Navarro[1] J. A. Ortega-Ruiz[2] J. Ametller[1] S. Robles[1]

[1]SeNDA *Security for Network and Distributed Applications* Group
Dept. of Information and Communications Engineering
Universitat Autònoma de Barcelona, Spain.

[2]Institute for Space Studies of Catalonia (IEEC),
80034 Barcelona, Spain.

MATA 2005, Montreal.

Outline

Introduction

DAFMA

Distributed **A**uthorization **F**ramework for **M**obile **A**gents

- Provides an authorization framework for multiagent systems supporting agent mobility.
- Based on **trust management** systems.
- Provides Role-Based Access Control (RBAC) services.

Introduction (II)

- Agent mobility introduces limitations. Mobile agents:
  - May be able to directly perform cryptographic operations?
  - May have cryptographic keys?

Introduction (II)

- Agent mobility introduces limitations. Mobile agents:
  - May be able to directly perform cryptographic operations?
  - May have cryptographic keys?
- MAs in DAFMA do not need to perform cryptographic operations.

Naming schema

- Strongly influenced by SPKI/SDSI's **local names**.

Naming schema

- Strongly influenced by SPKI/SDSI's **local names**.
- Each principal, is identified by:
  - Public key: non-MA principals. They have a cryptographic key pair.
  - Hash of an object: MA principals.

Local names

- Each entity as a **name container** for defining local names.

```
(<principal>, <local-name>)
```

Local names

- Each entity as a **name container** for defining local names.

```
(<principal>, <local-name>)
```

- Example: PK_0 defines a local name *partner* for PK_1:

$$PK_0\text{: } (PK_1\text{, partner})$$

Local names

- Each entity as a **name container** for defining local names.

```
(<principal>, <local-name>)
```

- Example: PK_0 defines a local name *partner* for PK_1:

$$PK_0\text{: } (PK_1\text{, partner})$$

- Third parties can use a *fully qualified name* to refer to PK_0's *partner*:

$$PK_0 \text{ partner}$$

Local names

- Each entity as a **name container** for defining local names.

```
(<principal>, <local-name>)
```

- Example: PK_0 defines a local name *partner* for PK_1:

$$PK_0\text{: } (PK_1, \text{partner})$$

- Third parties can use a *fully qualified name* to refer to PK_0's *partner*:

$$PK_0 \text{ partner}$$

- Name assertion: local names can be made public through a signed statement → **name certificate**:

$$\{(PK_1, \text{partner})\}_{PK_0^{-1}}$$

Group/role Management

- Groups/roles created as local names:

$$\{(PK_1,\ friends)\}_{PK_{adm}^{-1}}$$
$$\{(PK_2,\ friends)\}_{PK_{adm}^{-1}}$$
$$\{(PK_0\ partner,\ friends)\}_{PK_{adm}^{-1}}$$

Group/role Management

- Groups/roles created as local names:

$$\{(PK_1,\ friends)\}_{PK_{adm}^{-1}}$$
$$\{(PK_2,\ friends)\}_{PK_{adm}^{-1}}$$
$$\{(PK_0\ partner,\ friends)\}_{PK_{adm}^{-1}}$$

- Support for group/role **hierarchies** by means of inclusion.
  $\longrightarrow$ support for RBAC-like systems.

$$\{(PK_{adm}\ family,\ friends)\}_{PK_{adm}^{-1}}$$

Authorization certificates

- Authorization expressed through **authorization certificates** (SPKI/SDSI).

Authorization certificates

- Authorization expressed through **authorization certificates** (SPKI/SDSI).
- Includes a boolean delegation statement: determines if the authorization may be further delegated or not.

Authorization certificates

- Authorization expressed through **authorization certificates** (SPKI/SDSI).
- Includes a boolean delegation statement: determines if the authorization may be further delegated or not.
- Delegation adds a lot of flexibility to the system:
  - But introduces complexity: we need reduction and resolution algorithms.
  - Currently used a simplified version of SPKI/SDSI's certificate chain discovery algorithms.

DAFMA Architecture

- Divided in 4 components:

  AM Authorization Manager.
  RM Role Manager.
  RC Resource Controller.
  CRM Certificate Repository Manager.

DAFMA Architecture

- Divided in 4 components:

  AM Authorization Manager.
  RM Role Manager.
  RC Resource Controller.
  CRM Certificate Repository Manager.

- Implemented as static agents.

DAFMA Architecture

- Divided in 4 components:

  AM Authorization Manager.
  RM Role Manager.
  RC Resource Controller.
  CRM Certificate Repository Manager.

- Implemented as static agents.
- Interaction between components: FIPA Agent Communication Language and Ontologies.

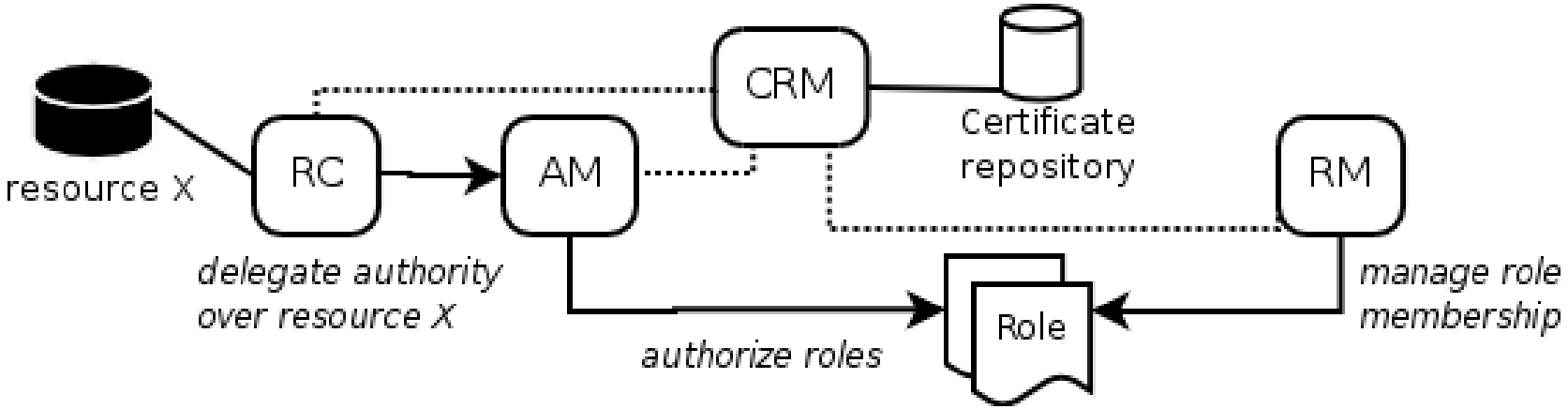
AM and RM

- Authorization Manager:
  - **Authorization** $\longrightarrow$ **roles**.
  - Follows local authorization policy.
- Role Manager:
  - **Role membership**.
  - Follows local role policy.

RC and CRM

- Resource Controller:
  - Acts like a Policy Enforcement Point.
  - Simple authorization policy: determines AMs allowed to manage authorizations for the resource.
- Certificate Repository Manager:
  - Certificate repository and related services.

Simple DAFMA Architecture



Establishing MA's role membership

- Two supported role membership establishment:
  - **User-managed** role:
    - RM gives full control to a user to manage its role membership.
  - **RM-managed** role:
    - The users needs to query the RM each role membership.

Conclusions

- DAFMA provides an authorization system for distributed environments taking into account agent mobility.

Conclusions

- DAFMA provides an authorization system for distributed environments taking into account agent mobility.
- Mobile agents do not need to perform cryptographic operations nor need to carry sensitive information (private keys).

Conclusions

- DAFMA provides an authorization system for distributed environments taking into account agent mobility.
- Mobile agents do not need to perform cryptographic operations nor need to carry sensitive information (private keys).
- Based on trust management and RBAC.

Conclusions

- DAFMA provides an authorization system for distributed environments taking into account agent mobility.
- Mobile agents do not need to perform cryptographic operations nor need to carry sensitive information (private keys).
- Based on trust management and RBAC.
- Current prototype implemented on top of JADE, supporting FIPA specifications.

Thanks

Thanks. Questions?