Towards a Framework for Group-Centric Secure Collaboration

Ram Krishnan (George Mason University)
Ravi Sandhu, Jianwei Niu, William Winsborough
(University of Texas at San Antonio)

CollaborateCom 2009, Nov 11th – 14th 2009, Crystal City, Washington DC
Group-Centric Collaboration

- Share/Collaborate for a specific purpose or mission
  - E.g. Collaboration in joint product design, merger and acquisition, etc.

- Emerging needs in Government and Commercial Organizations
  - E.g. Mission critical operations post 9/11, Inter-organizational collaboration, etc.

- Brings users & objects together in a group
  - Secure Meeting Room
  - Subscription Model
Group-Centric Collaboration (contd)

**Operational aspects**

- **Group Characteristics**
  - Core properties
  - Membership semantics
  - Membership renewal semantics
  - g-SIS specification

- **Object Model**
  - Read-only
  - Read-write (versioning?)

- **User-Subject Model**
  - User: Representation of human in the system
  - Subject: Programs/processes (untrusted)

**Administrative aspects**

- **Group Lifecycle**
- **Group Membership**

**Inter-group relations**

- Subordination
- Conditional Membership
- Mutual Exclusion
## Object Model

<table>
<thead>
<tr>
<th>No Versioning</th>
<th>Versioning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Multiple users may update, latest write is committed (destructive write).</td>
<td>1. Multiple users may update, each update creates a new version.</td>
</tr>
</tbody>
</table>
| 3. Tricky issues if read allowed after leave.  
  3.1 Fix: No read after write | 3. No such issues. Past users may continue to read versions authorized at leave time. No access to new versions after leave. |
| 4. Write after Leave? | 4. Write after Leave? |
Objective

- Systematically study authorization aspects in a simple inter-organizational collaboration scenario

Objective diagram:

- Administrative Model
- Operational Model

Collaboration Group:
- Establish/Disband
- Join User
- Leave User
- Add Version
- Remove Version
- Merge Version
- Substitute User
- Create RO/RW Subject
- Kill Subject
- Create Object
- Read/Update Version
- Suspend/Resume Version

ORG A:
- Import Version

ORG B:
- Import Version
Merge Vs Export of Object Versions

- ORG A
  - Merge
  - Add
  - Export
- Collaboration Group
  - Copy?
- ORG B
  - Merge
  - Add
  - Export
- Newly created group object
- ORG C
  - Add
Read-only Vs Read-Write Subjects

• Read Only subjects can read from multiple groups/entities
• Read-Write subjects restricted to one group
### Attribute Definitions

**Universal sets of names:**
- UNIV_ORG: The universe of organizations
- UNIV_CG: The universe of collaboration groups
- UNIV_U: The universe of users
- UNIV_S: The universe of subjects
- UNIV_O: The universe of objects
- UNIV_V: The universe of versions

**Existing sets of names:**
- ORG: Set of all existing collaborating organizations
- CG: Set of all existing groups established between orgs in ORG
- U: Set of all existing users
- S: Set of all existing subjects
- O: Set of all existing objects

| User Attributes: $\text{Att}(U) = \{\text{uorg, ucg, orgadmin, cgadmin}\}$ |
|-----------------|-----------------|
| $\text{uorg : } U \rightarrow \text{ORG}$ |
| $\text{ucg : } U \rightarrow 2^{\text{CG}}$ |
| $\text{orgadmin : } U \rightarrow \{\text{True, False}\}$ |
| $\text{cgadmin : } U \rightarrow 2^{\text{CG}}$ |

| Objects Attributes: $\text{Att}(O) = \{\text{member, currV}\}$ |
|-----------------|-----------------|
| $\text{member : } O \rightarrow \text{ORG} \cup \text{CG}$ |
| $\text{currV : } O \rightarrow 2^{\text{UNIV_V}}$ |

**Object Version Attributes: $\text{Att}(O, \text{UNIV_V}) = \{\text{vMember, vSuspended, importable}\}$**

| $\text{vMember : } O \times \text{UNIV_V} \rightarrow 2^{\text{ORG} \cup \text{CG}}$ |
| $\text{vSuspended : } O \times \text{UNIV_V} \rightarrow \{\text{True, False}\}$ |
| $\text{importable : } O \times \text{UNIV_V} \rightarrow \{\text{True, False}\}$ |

As shown, vMember, vSuspended and importable are partial functions that are undefined for object versions that do not currently exist.

| Group Attributes: $\text{Att}(\text{CG}) = \{\text{assoc}\}$ |
|-----------------|-----------------|
| $\text{assoc : } \text{CG} \rightarrow 2^{\text{ORG}}$ |

| Subject Attributes: $\text{Att}(S) = \{\text{sOwner, type, belongsTo}\}$ |
|-----------------|-----------------|
| $\text{sOwner : } S \rightarrow U$ |
| $\text{type : } S \rightarrow \{\text{ro, rw}\}$ |
| $\text{belongsTo : } S \rightarrow \text{ORG} \cup \text{CG}$ |

**Specified a complete authorization model: Administrative and Operational**
What can be guaranteed?

- A set of core safety properties can be guaranteed for group-centric collaboration models
- That is, we have shown that the specified authorization model satisfies the core safety properties
Core Properties

- **Authorization Persistence**
  - *Authorization cannot change if no group event occurs*

  \[
  \varphi_0 = \Box (\text{Authz} \rightarrow (\text{Authz} \land \text{Join} \lor \text{Leave} \lor \text{Add} \lor \text{Remove}))
  \]

  \[
  \varphi_1 = \Box (\neg \text{Authz} \rightarrow (\neg \text{Authz} \land \text{Join} \lor \text{Leave} \lor \text{Add} \lor \text{Remove}))
  \]

- **Authorization Provenance**
  - *Authorization can begin to hold only after a simultaneous period of user and object membership*

  \[
  \varphi_2 = (\neg \text{Authz} \land (\text{Authz} \land (\neg \text{Leave} \land \text{Join}) \land (\neg \text{Remove} \land \text{Add})))
  \]
Core Properties

- **Bounded Authorization**
  - *Authorization cannot grow during non-membership periods*

\[
\varphi_3 = \Box((\text{Leave} \land \neg \text{Authz}) \rightarrow (\neg \text{Authz} \lor \text{Join}))
\]

\[
\varphi_4 = \Box((\text{Remove} \land \neg \text{Authz}) \rightarrow (\neg \text{Authz} \lor \text{Add}))
\]

- **Availability**
  - *On add, authorization should hold for all existing users at add time*

\[
\varphi_5 = \Box(\text{Join} \rightarrow (\text{Add} \rightarrow ((\text{Authz} \lor \text{Leave}) \lor \text{Leave})))
\]
Richer Group-Centric Models

• “Begin Collaboration” Phase
  • Collaboration Group (CG) administration
  • Collaboration Structure
    • Flat group (no differentiation)
    • Flat group with differentiation (e.g. clearance/classification)
    • Structured groups with constraints (subordination, mutual exclusion etc.)
  • Participation Policy (users from non-collaborating orgs?)

• “Collaboration” Phase
  • Authentication to CG (Local Vs Federated)
  • CG membership (Local Vs Federated)
  • CG permissions (read-only, read-write, create, etc.)

• “End Collaboration” Phase (Publish Vs No Publish)
  • Tear down
  • Suspend
Conclusion and Future Work

- Group-Centric models are a natural fit for many collaboration scenarios
  - Practical applications might require additional access control aspects
    - E.g. DAC, LBAC, RBAC, ABAC, etc.
- Future Work
  - Inter-group Relations: Subordination, Conditional Membership, Mutual Exclusion
  - Handling authorizations in case of change in relations
  - Study information flow