Cross-Tenant Trust Models in Cloud Computing

Bo Tang and Ravi Sandhu

IRI
Aug 14-16, 2013
San Francisco, CA
OUTLINE

- Introduction
- Background and Motivation
- Cross-Tenant Trust Model (CTTM)
  - Tenant Trust Relations
  - Formalized Model
  - Role-Based CTTM (RB-CTTM)
- Related Work
- Conclusion and Discussion
OUTLINE

- Introduction
  - Background and Motivation
  - Cross-Tenant Trust Model (CTTM)
    - Tenant Trust Relations
    - Formalized Model
    - Role-Based CTTM (RB-CTTM)
- Related Work
- Conclusion and Discussion
Cloud Computing

- **Shared infrastructure**
  - [$$] -----> [$|$|$]$

- **Multi-Tenancy**
  - Virtually dedicated resources

- **Data Locked-in**
  - Collaborations can only be achieved through desktop.
  - E.g.: create/edit Word documents in Dropbox.

- A suitable fine-grained cross-tenant access control model is essential

Industry Solutions

- Microsoft and IBM: Fine-grained data sharing in SaaS using DB schema
  - Only feasible in DB

- NASA: RBAC + OpenStack (Nebula)
  - Lacks ability to support multi-org collaborations

- Salesforce (Force.com): Single Sign-On + SAML
  - Focus on authentication and simple authorization
  - Heavy management of certificates

http://nebula.nasa.gov/blog/2010/06/03/nebulas-implementation-role-based-access-control-rbac/
OUTLINE

- Introduction
- Background and Motivation
  - Cross-Tenant Trust Model (CTTM)
    - Tenant Trust Relations
    - Formalized Model
    - Role-Based CTTM (RB-CTTM)
- Related Work
- Conclusion and Discussion

© ICS at UTSA World-Leading Research with Real-World Impact! 6
Figure 1. A car renting example of cross-tenant accesses
On-Demand Self-Service

- Centralized facility
  - Resource pool
- Multi-tenancy
  - Unilateral and automatic provisioning as needed
  - Dynamically assigned virtual resources
- Temporary users and tenants
Scope and Assumptions

- Standardized APIs
  - Cross-tenant accesses are functionally available
- Authenticated Users
- Removable assumptions:
  - One Cloud Service
    - But extensible to multi-cloud
  - Two Tenant Trust (rather than federation)
  - Unidirectional Trust Relations (like follow in Twitter)
  - Unilateral Trust Relations (trustor or trustee)
Introduction

Background and Motivation

Cross-Tenant Trust Model (CTTM)
- Tenant Trust Relations
- Formalized Model
- Role-Based CTTM (RB-CTTM)

Related Work

Conclusion and Discussion
Tenant Trust Relations

- Tenant Trust (TT) relation is not partial order
- It is
  - Reflexive: $A \sqsubseteq A$
  - But not transitive: $A \sqsubseteq B \land B \sqsubseteq C \nRightarrow A \sqsubseteq C$
  - Neither symmetric: $A \sqsubseteq B \nRightarrow B \sqsubseteq A$
  - Nor anti-symmetric: $A \sqsubseteq B \land B \sqsubseteq A \nRightarrow A \equiv B$
Four potential trust types:

- **Type-α**: trustor can give access to trustee.
- **Type-β**: trustee can give access to trustor.
- **Type-γ**: trustee can take access from trustor.
- **Type-δ**: trustor can take access from trustee.

No meaningful use case, since the trustor holds all the control of the cross-tenant assignments of the trustee’s permissions.
Figure 2. Cross-Tenant Trust Management model
AA \subseteq U \times P, a many-to-many user-to-permission assignment relation, also written as \( \leftarrow \), requiring that \( u \leftarrow p \) only if

\[
\begin{align*}
\text{permOwner}(p) & \equiv \text{userOwner}(u) \lor \\
\text{permOwner}(p) & \sqsubseteq_{\alpha} \text{userOwner}(u) \lor \\
\text{userOwner}(u) & \sqsubseteq_{\beta} \text{permOwner}(p) \lor \\
\text{permOwner}(p) & \sqsubseteq_{\gamma} \text{userOwner}(u),
\end{align*}
\]

where only one of the \( \sqsubseteq \) requirements can apply depending on the nature of \( TT \).

Example: \texttt{Bob@UTSA} \leftarrow \texttt{discount%AVIS}
Figure 3. Role-Based Cross-Tenant Trust Management model
- $UA \subseteq U \times R$, is a many-to-many user-to-role assignment relation;

- $PA \subseteq P \times R$, is a many-to-many permission-to-role assignment relation requiring that $(p, r) \in PA$ only if

  \[permOwner(p) \sqcup roleOwner(r) \lor permOwner(p) \sqcup_{\alpha} roleOwner(r) \lor roleOwner(r) \sqcup_{\beta} permOwner(p) \lor permOwner(p) \sqcup_{\gamma} roleOwner(r),\]

  where only one of the $\sqcup$ requirements can apply depending on the nature of $TT$;
RH ⊆ R × R, is a partial order on R called role hierarchy or role dominance relation, also written as “≥”, requiring that r₂ ≥ r₁ only if

roleOwner(r₁) ≡ roleOwner(r₂) ∨ roleOwner(r₁) ⊴ α roleOwner(r₂) ∨ roleOwner(r₁) ⊴ β roleOwner(r₂) ∨ roleOwner(r₁) ⊴ γ roleOwner(r₂),

where only one of the ≤ requirements can apply depending on the nature of TT;
Feasibility in the Cloud

Figure 4. Multi-Tenant Authorization as a Service (MTAaaS) Architecture
OUTLINE

- Introduction
- Background and Motivation
- Cross-Tenant Trust Model (CTTM)
  - Tenant Trust Relations
  - Formalized Model
  - Role-Based CTTM (RB-CTTM)
- Related Work
- Conclusion and Discussion
Related Work

➢ RBAC
  ❖ CBAC, GB-RBAC, ROBAC (e.g.: player transfer in NBA)
  ❖ Require central authority managing collaborations

➢ Delegation Models
  ❖ dRBAC and PBDM (e.g.: allowing subleasing)
  ❖ Lacks agility (which the cloud requires)

➢ Grids
  ❖ CAS, VOMS, PERMIS
  ❖ Absence of centralized facility and homogeneous architecture (which the cloud has)
Related Work (Contd.)

 ➢ Role-based Trust
  ❖ RT (Type-α trust relation)
  ❖ MTAS (Type-β trust relation)
  ❖ MT-RBAC (Type-γ trust relation)
  ❖ Suits the cloud (out-sourcing trust)
Introduction

Background and Motivation

Cross-Tenant Trust Model (CTTM)
  - Tenant Trust Relations
  - Formalized Model
  - Role-Based CTTM (RB-CTTM)

Related Work

Conclusion and Future Work
Conclusion

- Needs of cross-tenant access control
- On-demand self-service model
- Tenant trust relation and types
- CTTM and RB-CTTM models
  - Formalization
  - Feasibility in the cloud
- Mapping to related work
  - RT, MTAS and MT-RBAC
Future Work

- Other models compatible with MTAaaS platform
- Implementation MTAaaS in OpenStack
Q & A
Thank You!