Relationship-Based Access Control (ReBAC or RAC)

Prof. Ravi Sandhu
Executive Director and Endowed Chair

March 4, 2016

ravi.sandhu@utsa.edu
www.profsandhu.com
RAC Models

U2U, U2R, R2R with attributes

URRACA

Cheng et al 2014

U2U, U2R, R2R

URRAC

Cheng et al 2012-09

U2U

UURAC

Cheng et al 2012-07
Cheng et al 2015
RAC Models

U2U, U2R, R2R
with attributes

URRACA

Cheng et al 2014

U2U, U2R, R2R

URRAC

Cheng et al 2012-09

U2U

URRAC

Cheng et al 2012-07
Cheng et al 2015

© Ravi Sandhu

World-Leading Research with Real-World Impact!
UURAC Motivation

• Users in Online Social Networks (OSNs) are connected by social relationships (user-to-user relationships U2U)

• Owner of a resource can control its release based on U2U relationships between the access requester and the owner
UURAC Motivation

• OSNs keep massive resources and support varied activities for users
• Users want to regulate access to their resources and activities related to them (as a requester or target)
• Some related users also expect control on how the resource or user can be exposed
UURAC Motivation

• What current friend-of-friend approach cannot do?
  – User who is tagged in a photo wants to keep her image private (Related User’s Control)
  – Mom doesn’t want her kid to become friend with her colleagues (Parental Control)
  – Employee promotes his resume to headhunters without letting his current employer know (Allowing farther users but preventing closer users)
Characteristics of Access Control in OSNs

• **Policy Individualization**
  – Users define their own privacy and activity preferences
  – Related users can configure policies too
  – Collectively used by the system for control decision

• **User and Resource as a Target**
  – e.g., poke, messaging, friendship invitation, etc.

• **User Policies for Outgoing and Incoming Actions**
  – User can be either requester or target of activity
  – Allows control on 1) activities w/o knowing a particular resource and 2) activities against the user w/o knowing a particular access requestor
  – e.g., block notification of friend’s activities; restrict from viewing violent contents

• **Relationship-based Access Control**
UURAC Approach

• Using regular expression-based path pattern for arbitrary combination of relationship types
• Given relationship path pattern and hopcount limit, graph traversal algorithm checks the social graph to determine access
Related Works

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Relationship Category</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multiple Relationship Types</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Directional Relationship</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U2U Relationship</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>U2R Relationship</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td><strong>Model Characteristics</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Policy Individualization</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User &amp; Resource as a Target</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Outgoing/Incoming Action Policy</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Relationship Composition</strong></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Relationship Depth</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Relationship Composition</td>
<td>0 to 2 f, f of f</td>
<td>0 to n exact type sequence</td>
<td>1 to n path of same type</td>
<td>1 to n exact type sequence</td>
<td>0 to n path pattern of different types</td>
</tr>
</tbody>
</table>

- The advantages of this approach:
  - Passive form of action allows outgoing and incoming action policy
  - Path pattern of different relationship types make policy specification more expressive

World-Leading Research with Real-World Impact!
Related Works

• Not to forget PGP web of trust (mid 1990s)
Social Networks

- Social graph is modeled as a directed labeled simple graph $G=<U, E, \Sigma>$
  - Nodes $U$ as users
  - Edges $E$ as relationships
  - $\Sigma=\{\sigma_1, \sigma_2, \ldots, \sigma_n, \sigma_1^{-1}, \sigma_2^{-1}, \ldots, \sigma_n^{-1}\}$ as relationship types supported
Policy Taxonomy

Access Control Policy

User-specified Policy
- Policy for Resource
  - Incoming Action Policy (Target Resource Policy)
- Policy for User
  - Outgoing Action Policy (Accessing User Policy)

System-specified Policy
- Policy for Resource
- Policy for User
  - Incoming Action Policy (Target User Policy)
UURAC Model Components

- UA: Accessing User
- UT: Target User
- UC: Controlling User
- RT: Target Resource
- AUP: Accessing User Policy
- TUP: Target User Policy
- TRP: Target Resource Policy
- SP: System Policy

World-Leading Research with Real-World Impact!
Access Request and Evaluation

• Access Request <u_a, action, target>
  – u_a tries to perform action on target
  – Target can be either user u_t or resource r_t

• Policies and Relationships used for Access Evaluation
  – When u_a requests to access a user u_t
    • u_a’s AUP, u_t’s TUP, SP
    • U2U relationships between u_a and u_t
  – When u_a requests to access a resource r_t
    • u_a’s AUP, r_t’s TRP (associated with u_c), SP
    • U2U relationships between u_a and u_c
Policy Representations

<table>
<thead>
<tr>
<th>Policy Representation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessing User Policy</td>
<td>(&lt;\text{action}, \text{start, path rule}&gt;)</td>
</tr>
<tr>
<td>Target User Policy</td>
<td>(&lt;\text{action}^{-1}, \text{start, path rule}&gt;)</td>
</tr>
<tr>
<td>Target Resource Policy</td>
<td>(&lt;\text{action}^{-1}, r_t, \text{start, path rule}&gt;)</td>
</tr>
<tr>
<td>System Policy for User</td>
<td>(&lt;\text{action}, \text{start, path rule}&gt;)</td>
</tr>
<tr>
<td>System Policy for Resource</td>
<td>(&lt;\text{action}, r.\text{type, start, path rule}&gt;)</td>
</tr>
</tbody>
</table>

- \(\text{action}^{-1}\) in TUP and TRP is the passive form since it applies to the recipient of action
- TRP has an extra parameter \(r_t\) to distinguish the actual target resource it applies to
  - \(\text{owner}(r_t)\) → a list of \(u_c\) → U2U relationships between \(u_a\) and \(u_c\)
- SP does not differentiate the active and passive forms
- SP for resource needs \(r.\text{type}\) to refine the scope of the resource
Graph Rule Grammar

\[
\text{GraphRule} ::= \text{"(" } \text{< StartingNode } \text{\text{",\text{" } < PathRule } \text{\text{"\text{)}}}
\]

\[
\text{PathRule} ::= \text{< PathSpecExp } \text{\text{| < PathSpecExp } \text{\text{< Connective } < PathRule }}\text{\text{)}}
\]

\[
\text{Connective} ::= \lor | \land
\]

\[
\text{PathSpecExp} ::= \text{< PathSpec } \text{\text{| ~ < PathSpec }\text{\text{)}}}
\]

\[
\text{PathSpec} ::= \text{"(" } \text{< Path } \text{\text{",\text{" } < HopCount } \text{\text{"\text{)}}}\text{\text{|{"\text{"}} < EmptySet } \text{\text{",\text{" } < Hopcount } \text{\text{)}}\text{\text{)}}}
\]

\[
\text{HopCount} ::= \text{< Number }
\]

\[
\text{Path} ::= \text{< TypeExp } \text{\text{| < TypeExp } \text{\text{< Path >}}}
\]

\[
\text{EmptySet} ::= \emptyset
\]

\[
\text{TypeExp} ::= \text{< TypeSpecifier } \text{\text{| < TypeSpecifier } \text{\text{< Wildcard >}}}
\]

\[
\text{StartingNode} ::= u_a | u_t | u_c
\]

\[
\text{TypeSpecifier} ::= \sigma_1 | \sigma_2 | \ldots | \sigma_n | \sigma_1^{-1} | \sigma_2^{-1} | \ldots | \sigma_n^{-1} | \Sigma \text{ where } \Sigma = \{ \sigma_1, \sigma_2, \ldots, \sigma_n, \sigma_1^{-1}, \sigma_2^{-1}, \ldots, \sigma_n^{-1} \}
\]

\[
\text{Wildcard} ::= \text{"*"|"?"|"+"}
\]

\[
\text{Number} ::= [0 - 9]+\]
Example

- Alice’s policy $P_{Alice}: \langle \text{poke}, (u_a, (f*, 3)) \rangle \langle \text{poke}^{-1}, (u_t, (f, 1)) \rangle \langle \text{read}, (u_a, (\Sigma*, 5)) \rangle \langle \text{read}^{-1}, \text{file1}, (u_c, (cf*, 4)) \rangle$
- Harry’s policy $P_{Harry}: \langle \text{poke}, (u_a, (cf*, 5) \lor (f*, 5)) \rangle \langle \text{poke}^{-1}, (u_t, (f*, 2)) \rangle \langle \text{read}^{-1}, \text{file2}, (u_c, -(p+, 2)) \rangle$
- System’s policy $P_{Sys}: \langle \text{poke}, (u_a, (\Sigma*, 5)) \rangle \langle \text{read}, \text{photo}, (u_a, (\Sigma*, 5)) \rangle$

- “Only Me”
  - $\langle \text{poke}, (\text{ua}, (\emptyset, 0)) \rangle$ says that ua can only poke herself
  - $\langle \text{poke}^{-1}, (\text{ut}, (\emptyset, 0)) \rangle$ specifies that ut can only be poked by herself
- The Use of Negation Notation
  - $(\text{fff}c \land \neg\text{fc})$ allows the coworkers of the user’s distant friends to see, while keeping away the coworkers of the user’s direct friends
Policy Collecting

• To authorize \((u_a, \text{action}, \text{target})\) if \(\text{target} = u_t\)
  – E.g., \((\text{Alice}, \text{poke}, \text{Harry})\)

World-Leading Research with Real-World Impact!
Policy Collecting

- To authorize \((u_a, \text{action}, \text{target})\) if \(\text{target} = u_t\)
  - E.g., \((\text{Alice}, \text{poke}, \text{Harry})\)

- **AUP**
  - \(<\text{poke}, (u_a, (f^*,3))>\>

- **TUP**
  - \(<\text{poke}^{-1}, (u_t, (f^*,2))>\>

- **SP**
  - \(<\text{poke}, (u_a, (\Sigma^*,5))>\>

- **PAlice**
  - \(<\text{poke}^{-1}, (u_a, (f^*,3))>\>

- **PHarry**
  - \(<\text{poke}, (u_a, (cf^*,5) \lor (f^*,5))>\>

- **P Sys**
  - \(<\text{poke}, (u_a, (f^*,5))>\>

World-Leading Research with Real-World Impact!
Policy Collecting

• To authorize \((u_a, \text{action}, \text{target})\) if \(\text{target} = r_t\)
  – Determine the controlling user for \(r_t\):
    • \(u_c \leftarrow \text{owner}(r_t)\)
  – E.g., \((\text{Alice}, \text{read}, \text{file2})\)
Policy Collecting

- To authorize \((u_a, \text{action}, \text{target})\) if target = \(r_t\)
  - Determine the controlling user for \(r_t\):
    - \(u_c \leftarrow \text{owner}(r_t)\)
    - E.g., (Alice, read, file2)
Policy Extraction

• Policy: \(<\text{action}, \text{r.type}, \text{graph rule}\>\)

• Graph Rule: \(\text{start}, \text{path rule}\)

• Path Rule: \(\text{path spec} \land | \lor \text{path spec}\)

• Path Spec: \(\text{path}, \text{hopcount}\)
Policy Evaluation

• Evaluate a combined result based on conjunctive or disjunctive connectives between path specs

• Make a collective result for multiple policies in each policy set.
  – Policy conflicts may arise. We assume system level conflict resolution strategy is available (e.g., disjunctive, conjunctive, prioritized).

• Compose the final result from the result of each policy set (AUP, TUP/TRP, SP)
Path Checking Algorithm Overview

- Parameters: $G$, path, hopcount, $s$, $t$
- Traversal Order: Depth-First Search
  - Why not BFS?
    - Activities in OSN typically occur among people with close distance
    - DFS needs only one pair of variables to keep the current status and history of exploration
    - Hopcount limit prevents DFS from lengthy useless search
Path Checking Algorithm Complexity

- Time complexity is bounded between $[O(d_{\text{min}}^{\text{Hopcount}}), O(d_{\text{max}}^{\text{Hopcount}})]$, where $d_{\text{max}}$ and $d_{\text{min}}$ are maximum and minimum out-degree of node
  
  - Users in OSNs usually connect with a small group of users directly, the social graph is very sparse
  
  - Given the constraints on the relationship types and hopcount limit, the size of the graph to be explored can be dramatically reduced
RAC Models

U2U, U2R, R2R

with attributes

U2U, U2R, R2R

U2U

URRAC$_A$

URRAC

UURAC

Cheng et al 2014

Cheng et al 2012-09

Cheng et al 2012-07

Cheng et al 2015
Limitation of U2U Relationships

• We rely on **the controlling user and ownership** to regulate access to resources in UURAC (U2U Relationship-based-based AC)

• Needs more flexible control
  – Parental control, related user’s control (e.g., tagged user)
  – User relationships to resources (e.g., U-U-R)
  – User relationships via resources (e.g., U-R-U)
Beyond U2U Relationships

• There are various types of relationships between users and resources in addition to U2U relationships and ownership
  – e.g., share, like, comment, tag, etc

• U2U, U2R and R2R

• U2R further enables relationship and policy administration
Access Scenarios

Access in OSNs

User as Target

Resource as Target

Entity on the path

Direct

U in between

R in between

U and R in between

(b) Taxonomy based on Entity on the path

Direct

U in between

R in between

U and R in between

UU

UU+U

UR+U

U(U|R)+U

UU

UU+R

UR+R

U(U|R)+R

World-Leading Research with Real-World Impact!
Related Works

• Access Control Models for OSNs

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiple Relationship Types</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Directional Relationship</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>U2U Relationship</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>U2R Relationship</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

| Model Characteristics         | ✓         | ✓               | ✓              | ✓                 | ✓     | ✓     |
| Policy Individualization      | ✓         | ✓               | ✓              | ✓                 | ✓     | ✓     |
| User & Resource as a Target   | ✓         | ✓               | ✓              | ✓                 | ✓     | ✓     |
| Outgoing/Incoming Action Policy| ✓         | ✓               | ✓              | ✓                 | ✓     | ✓     |

<table>
<thead>
<tr>
<th>Relationship Composition</th>
<th>0 to 2 f, f of f</th>
<th>0 to n exact type sequence</th>
<th>1 to n path of same type</th>
<th>1 to n exact type sequence</th>
<th>0 to n path pattern of different types</th>
<th>0 to n path pattern of different types, hopcount skipping</th>
</tr>
</thead>
</table>

• The advantages of URRAC:
  – Path pattern of different relationship types and hopcount skipping make policy specification more expressive
  – System-level conflict resolution policy

World-Leading Research with Real-World Impact!
UURAC Model Components

Uₐ: Accessing User
Uₜ: Target User
Uₖ: Controlling User
Rₜ: Target Resource
AUP: Accessing User Policy
TUP: Target User Policy
TRP: Target Resource Policy
SP: System Policy

World-Leading Research with Real-World Impact!
URRAC Model Components

AU: Accessing User
AS: Accessing Session
TU: Target User
TS: Target Session
O: Object
P: Policy
PAU: Accessing User Policy
PAS: Accessing Session Policy
PTU: Target User Policy
PTS: Target Session Policy
PO: Object Policy
PP: Policy for Policy
PSys: System Policy

Attached to
Used as input
1-to-n mapping
Constrained by (e.g., subset)

Social Graph (SG)
Decision Module (DM)

World-Leading Research with Real-World Impact!
Action and Access Request

- $ACT = \{act_1, act_2, \ldots, act_n\}$ is the set of OSN supported actions
- Access Request $<s, act, T>$
  - $s$ tries to perform $act$ on $T$
  - Target $T \subseteq (2^{TU \cup R} - \emptyset)$ is a non-empty set of users and resources
    - $T$ may contain multiple targets
Authorization Policy

<table>
<thead>
<tr>
<th>Accessing User Policy</th>
<th>&lt;act, graphrule &gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessing Session Policy</td>
<td>&lt;act, graphrule &gt;</td>
</tr>
<tr>
<td>Target User Policy</td>
<td>&lt;act(^{-1}), graphrule &gt;</td>
</tr>
<tr>
<td>Target Session Policy</td>
<td>&lt;act(^{-1}), graphrule &gt;</td>
</tr>
<tr>
<td>Object Policy</td>
<td>&lt;act(^{-1}), graphrule &gt;</td>
</tr>
<tr>
<td>Policy for Policy</td>
<td>&lt;act(^{-1}), graphrule &gt;</td>
</tr>
<tr>
<td>System Policy for User</td>
<td>&lt;act, graphrule &gt;</td>
</tr>
<tr>
<td>System Policy for Resource</td>
<td>&lt;act, o.type, graphrule &gt;</td>
</tr>
</tbody>
</table>

where o.type is optional

- \( action^{-1} \) in TUP, TSP, OP and PP is the passive form since it applies to the recipient of action
- SP does not differentiate the active and passive forms
- SP for resource needs \( o.type \) to refine the scope of the resource
Six degrees of separation
- Any pair of persons are distanced by about 6 people on average. (4.74 shown by recent study)
- Hopcount for U2U relationships is practically small

U2R and R2R relationships may form a long sequence
- Omit the distance created by resources
- Local hopcount stated inside “[[[]]]” will not be counted in global hopcount.
- E.g., “([f*,3][[c*, 2]],3)”, the local hopcount 2 for c* does not apply to the global hopcount 3, thus allowing f* to have up to 3 hops.
Policy Conflict Resolution

• System-defined conflict resolution for potential conflicts among user-specified policies

• Disjunctive, conjunctive and prioritized order between relationship types
  – $\land$, $\lor$, $>$ represent disjunction, conjunction and precedence
  – $@$ is a special relationship “null” that denotes “self”
Policy Conflict Resolution (cont.)

\(< read^{-1}, (own \land tag) >\)

The more rigid one between the owner’s and the tagged users’ “read\(^{-1}\)” policies over the photo is honored.

\(< friend\_request, (parent > @) >\)

When child attempts friendship request to someone, parents’ policies get precedence over child’s own will.

\(< share^{-1}, (own \lor tag \lor share) >\)

A weblink is sharable if either the original owner, or any of the tagged users or shared users allows.
RAC Models

U2U, U2R, R2R with attributes → URRAC\textsubscript{A} → URRAC → UURAC

- Cheng et al 2014
- Cheng et al 2012-09
- Cheng et al 2012-07
- Cheng et al 2015
• ReBAC usually relies on type, depth, or strength of relationships, but cannot express more complicated topological information

• ReBAC lacks support for attributes of users, resources, and relationships

• Useful examples include common friends, duration of friendship, minimum age, etc.
Attributes in OSNs

• Node attributes
  • Define user’s identity and characteristics: e.g., name, age, gender, etc.

• Edge attributes
  • Describe the characteristics of the relationship: e.g., weight, type, duration, etc.

• Count attributes
  • Depict the occurrence requirements for the attribute-based path specification, specifying the lower bound of the occurrence of such path
RAC Models

- U2U, U2R, R2R
  - with attributes
    - URRAC
      - UURAC
        - URRACA
          - Cheng et al 2014
  - U2U, U2R, R2R
    - URRAC
      - Cheng et al 2012-09
  - U2U
    - UURAC
      - Cheng et al 2012-07
      - Cheng et al 2015