Reachability Analysis For Role Based Administration of User Attributes

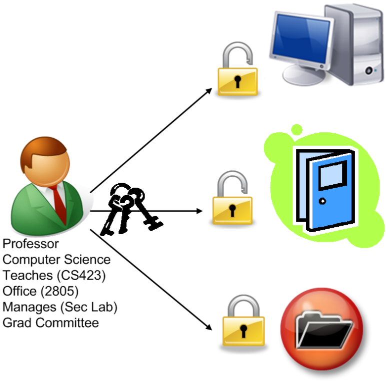
Xin Jin, Ram Krishnan, Ravi Sandhu

Institute of Cyber Security (ICS)
University of Texas at San Antonio (UTSA)
San Antonio, Texas, USA

April 22, 2014

Content

What is User Attributes



Related research with User Attributes

- Attribute based access control (**ABAC**): Jin *et al*(DBSEC 12), Wang *et al*(FSME 04), Hu *et al* (NIST draft model 2013), Chadwick *et al* (WETICE 06), XACML 3.0 (06), Pirretti *et al* (CCS 06), Li *et al* (Oakland 02)
- Attribute based encryption (**ABE**): Goyal *et al* (CCS 06), Bethencourt *et al* (Oakland 07), Ostrovsky *et al* (CCS 07), Rouselakis *et al* (CCS 13), Liu *et al* (CCS 13)
- **Identity management**: Chadwick *et al* (Computer 09)
- **Usage control**: $UCON_{ABC}$ by Park *et al* (TISSEC 04)

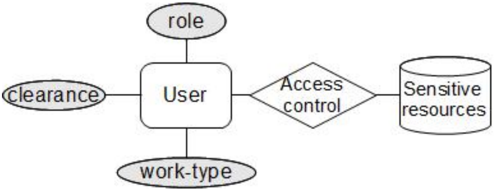
Attribute Administration

- In each organization, certain administrators have to **assign** user attributes values when the user is provisioned and **modify** user attributes values thereafter.
- Attributes of the same user **constrain** each other. Administration rules are specified to regulate attribute modifications.

Example Rule

clearance attribute of users can be assigned to “topsecret” **IF**: “officer” $\in$ role(u) $\wedge$ clearance(u) == “secret” $\wedge$ work-type(u) == “full-time”.

Motivation for Reachability Problem



Example Authorization Policy

read(*sub*, *obj*) $\rightarrow \neg$(clearance(u) == "topsecret" $\wedge$ work-type(u) == "part-time")

Questions

Given a predefined administrative rules, will Alice ever be able to access *obj* in the future? It is equivalent to ask whether Alice's attribute can reach conditions which satisfies the authorization policy.

Background and Motivation

Attributes Assignment Constraints

- *Rule 1*: assign clearance(u) to “topsecret” **IF**:
  “officer” $\in$ role(u) $\wedge$ clearance(u) == “secret” $\wedge$ work-type(u) == “full-time”.
- *Rule 2*: assign work-type(u) to “part-time” **IF** “officer” $\in$ role(u).

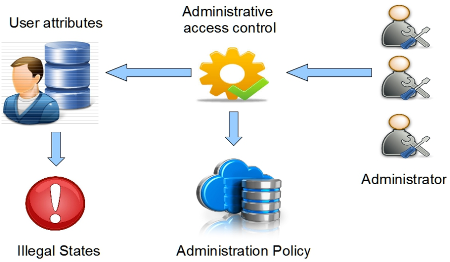
Transition by Rule 1

From rule 1, it seems that the user will never get access to *obj*.

Transition by Rule 2

“officer” $\in$ role(Alice), clearance(Alice) == “topsecret”,
work-type(Alice) == “full-time”
$\rightarrow$ “officer” $\in$ role(Alice), clearance(Alice) == “topsecret” ,
work-type(Alice) == “part-time”.

Background and Motivation



- Given a **large set** of administration rules, it is hard to tell whether user attributes can reach certain values as expected.
- Constraints (Crampton *et al*(SACMAT 03), Ahn *et al* (TISSEC), Bijon *et al* (PASSAT 13)) can be deployed on user attributes assignment. It prevent values to be assigned. Reachability is still important. Help understand what each assignment enables indirectly and also help design constraints.
- **Reachability analysis** help solves this problem by determining whether user attributes can reach certain value based on given policies.

I·C·S
The Institute for Cyber Security

Related Work

UTSA

- **The Harrison Ruzzo Ullman (HRU) model:** Safety problem regarding leakage of a specific right. Others are TAM, ATAM by Sandhu *et al*(Oakland 92).
- **Role Based Trust Management (RT):** safety analysis on trust relationships: Li *et al* (Oakland 02, 03)
- **ARBAC97 Related:** Safety analysis on role administration rules: Stoller *et al* (CCS 07, ESORICS 10, CSFW 06, SACMAT 09), Alberti *et al* (ASIACCS 2011), Armando *et al* (DBSEC 2012), Li *et al* (SACMAT 04)
- **Others:** policy mis-configuration detection, model checking, policy analysis, etc.

Limitations

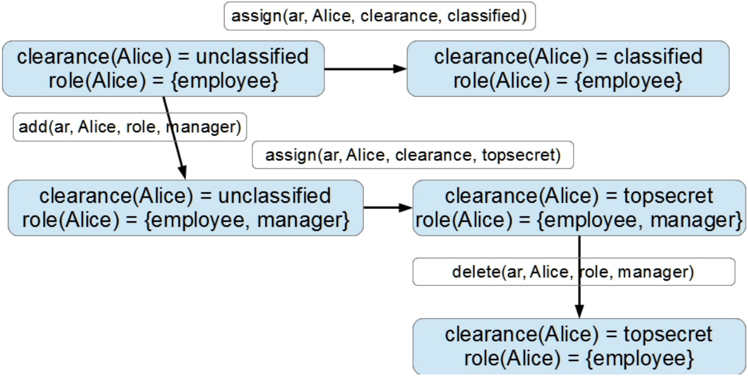
- Analysis on only rules for one user attribute——role, and is for RBAC authorization policy, i.e., role represents permissions.
- There is connection between those work and reachability analysis for attributes. But it is not intuitive and has not been studied.
- Attribute reachability is beyond the safety analysis of role as defined in related work.

Contributions of Our Paper

Our Contributions

- Formally define user attribute administration as state transition system.
- Define two kinds of reachability problems in the context of attribute administration Model.
- Provide formal proof for problem complexity. Most problems are in PSPACE-complete.
- Discover practical restrictions on policies and design polynomial time solvable algorithms.

Attributes, State, State Transition and Rules



State Transition Rules

User attributes changes as guided by some models. We take a restricted version of the Generalized User-Role Assignment Model (GURA) (Jin *et al* WSRAS12) here. It is simple while the reachability problem is not obvious.

$$can\_add_{sua} \subseteq \mathrm{AR} \times \mathrm{C} \times \mathrm{SCOPE}_{sua}$$
$$can\_delete_{sua} \subseteq \mathrm{AR} \times \mathrm{C} \times \mathrm{SCOPE}_{sua}$$
$$can\_assign_{aua} \subseteq \mathrm{AR} \times \mathrm{C} \times \mathrm{SCOPE}_{aua}$$

sua: a set-valued attribute, *aua*: an atomic-valued attribute, AR: administrative role, C: preconditions on attributes of users.

- **if** $\langle$ hr, clearance(u) = secret $\wedge$ employee $\in$ role(u), manager$\rangle \in$ $\mathrm{can\_add}_{role}$
  then *add*(hr, Alice, role, manager) is allowed if clearance(Alice) == secret $\wedge$ employee $\in$ role(Alice).

The rGURA_0 Schemes

For preconditions in each can_assign$_{aua}$ relation:

$$\varphi ::= \neg\, \varphi \mid \varphi \wedge \varphi \mid aua(u) = avalue$$
$$avalue ::= aval_1 \mid aval_2 \;\; \ldots \mid aval_n$$

where $\mathrm{SCOPE}_{aua} = \{aval_1,\ aval_2,\ \ldots,\ aval_n\}$.
For preconditions in each can_add$_{sua}$ and can_delete$_{sua}$ relations:

$$\varphi ::= \neg\, \varphi \mid \varphi \wedge \varphi \mid svalue \in sua(u)$$
$$svalue ::= sval_1 \mid sval_2 \mid \ldots \mid sval_m$$

where $\mathrm{SCOPE}_{sua} = \{sval_1,\ sval_2,\ \ldots,\ sval_m\}$.

Example rGURA_0 Instance

UA = {clearance, dept, role, project}
U = {Alice}
AR = $\{ar_1, ar_2\}$

- can_assign_{dept}: $\{\langle\, ar_1, dept(u) = finance, IT \rangle\}$
- can_add_{role}: $\{\langle\, ar_2, employee \in role(u) \wedge \neg(manager \in role(u)), leader \rangle\}$
- $can\_delete_{project}$: $\{\langle\, ar_1, prj_1 \in project(u) \wedge \neg\,(prj_2 \in project(u)), prj_3 \rangle, \langle\, ar, \neg(prj_1 \in project(u)) \wedge \neg\,(prj_2 \in project(u)), prj_4 \rangle\}$

The rGURA_1 Schemes

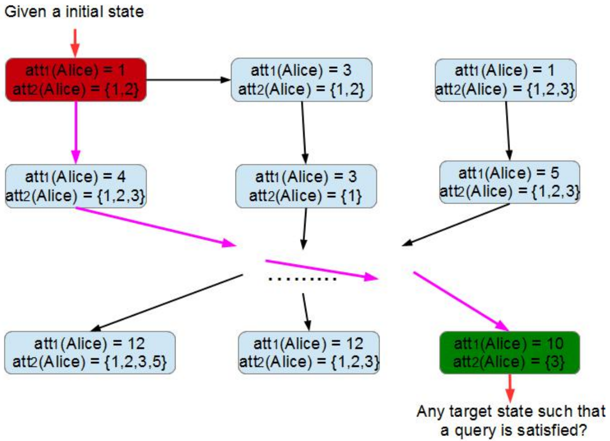
For preconditions in all relations:

$$\varphi ::= \neg\varphi \mid \varphi \wedge \varphi \mid aua(u) = avalue \mid svalue \in sua(u)$$

Example rGURA_1 **instance**:
UA = {clearance, dept, role, project}
U = {Alice}
AR = $\{ar_1, ar_2\}$

- $\mathrm{can\_assign}_{dept}$: $\langle$ ar, dept(u) = finance $\wedge$ $\neg$(prj_1 $\in$ project(u)) $\wedge$ $\neg$ (prj_2 $\in$ project(u))$\wedge$ employee $\in$ role(u) $\wedge$ $\neg$(manager$\in$role(u)) , IT$\rangle$

Reachability Problem Example



Two types of Reachability Problems (RP)

A **query** is a state or subset of a state. Given a initial state and a query of the following types :

- $\mathrm{RP}_{=}$: All attributes should be the same.
- $\mathrm{RP}_{\supseteq}$: For set-valued attribute, the target state may contain additional values.

Example:
Initial state: $att_1(Alice) = 1$, $att_2(Alice) = \{1,2\}$
Query: $att_1(Alice) = 1$, $att_2(Alice) = \{1,3\}$
Target States that satisfy the query:

- $\mathrm{RP}_{=}$: $att_1(Alice) = 1$, $att_2(Alice) = \{1,3\}$
- $\mathrm{RP}_{\supseteq}$: $att_1(Alice) = 1$, $att_2(Alice) = \{1, 3, 4\}$ **OR**
  $att_1(Alice) = 1$, $att_2(Alice) = \{1, 3, 5\}$ **OR**
  $att_1(Alice) = 1$, $att_2(Alice) = \{1, 3, 6\}$

Content of Analysis

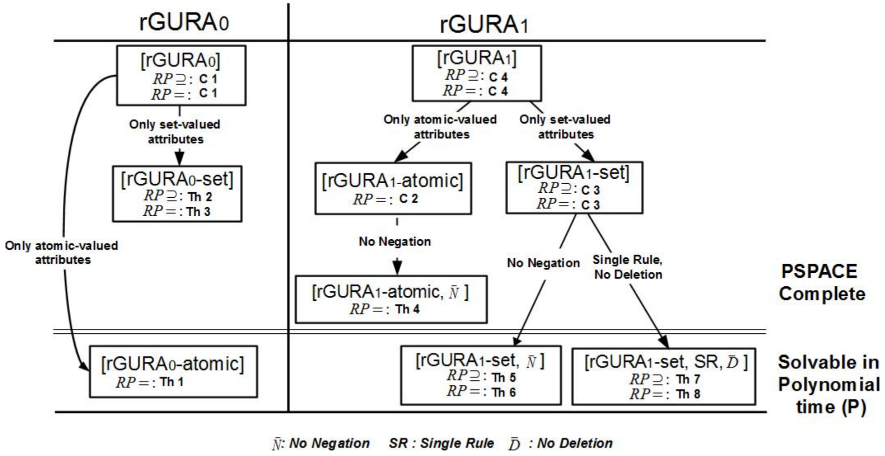
We use [rGURA_x-[atomic, set], `Restriction`] denote a specialized rGURA scheme.

- The subscript x takes a value of 0 or 1.
- `Restriction` represents possible combinations of SR, $\overline{D}$ and $\overline{N}$.

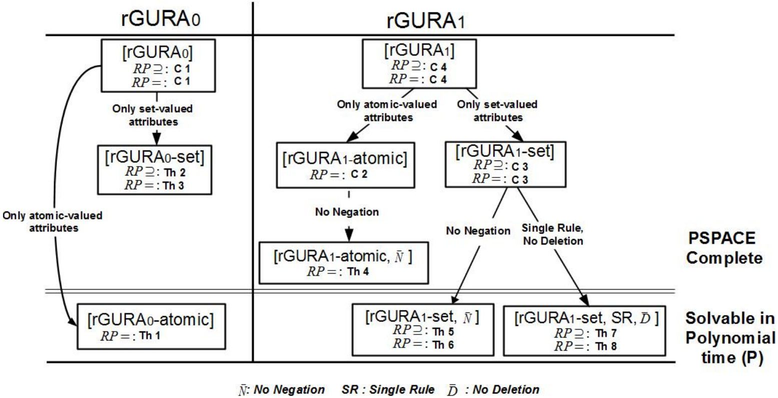
Example

[rGURA_1-atomic, $\overline{N}$] denotes an rGURA_1 scheme where only atomic-valued attributes are defined and the administrative rules satisfy $\overline{N}$.

Analysis Results



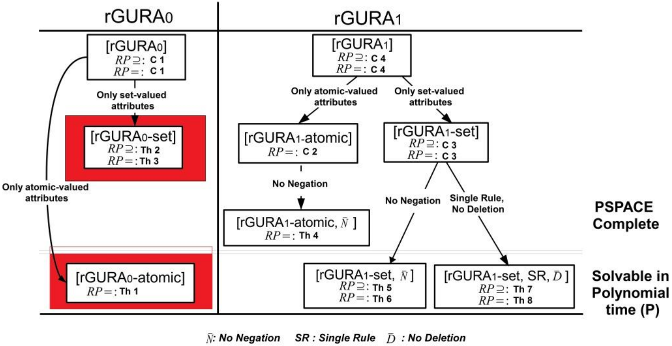
Result 1



Lemma 1: All problems are within PSPACE.

Non-deterministic Turing Machine can simulate the algorithm. Polynomial space is needed. Thus, it is NPSPACE (NPSPACE = PSPACE).

Result 2

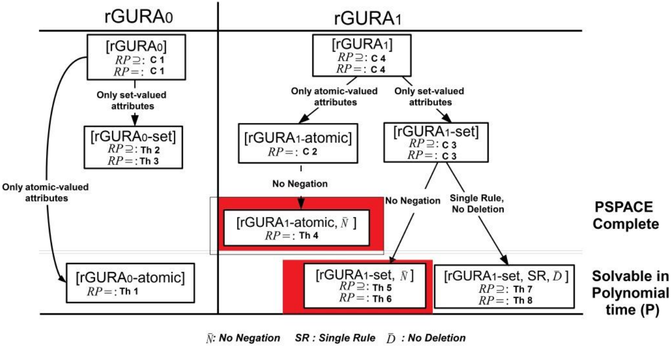


$\mathrm{RP}_=$ *in [*rGURA_0*-set] is a reduction from ARBAC97 analysis problem as proved in CSFW06 by S. Stoller.*

$\mathrm{RP}_=$ *in [*rGURA_0*-atomic] is equivalent to path search problem in directed graph.*

Contributions of Our Paper

Result 3



$\mathrm{RP}_{=}$ *in [*rGURA_1*-set,* $\overline{N}$*] can be solved by policy traversal.*
$\mathrm{RP}_{=}$ *in [*rGURA_1*-atomic,* $\overline{N}$*] is a reduction from SAS planning problem in AI.*

Conclusion and Future Work

Our contribution

- Motivate user attributes reachability analysis.
- Define reachability problems based on a restricted version of GURA model.
- Formal proof and polynomial time solvable algorithm design.

Interesting future work

- Heuristic algorithm to solve the general case $\mathrm{RP}_{=}$ and $\mathrm{RP}_{\supseteq}$ in $[\mathrm{rGURA}_1]$.
- Bring Authorization Policy into consideration.
- Bring ABAC into consideration such as subject attributes and its constraints.

Thanks
Any Questions?