# A Comparison of Logical-Formula and Enumerated Authorization Policy ABAC Models

Prosunjit Biswas, Ravi Sandhu and Ram Krishnan

The University of Texas at San Antonio

DBSec 2016

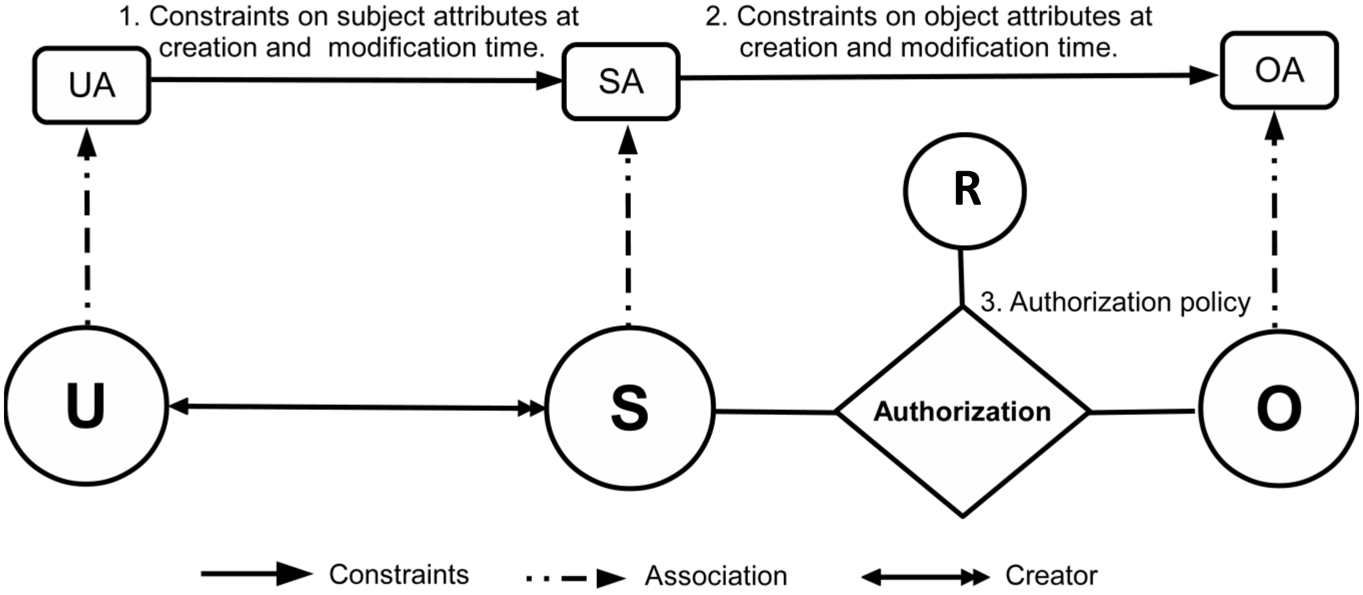
July 18, 2016

UTSA Engineering

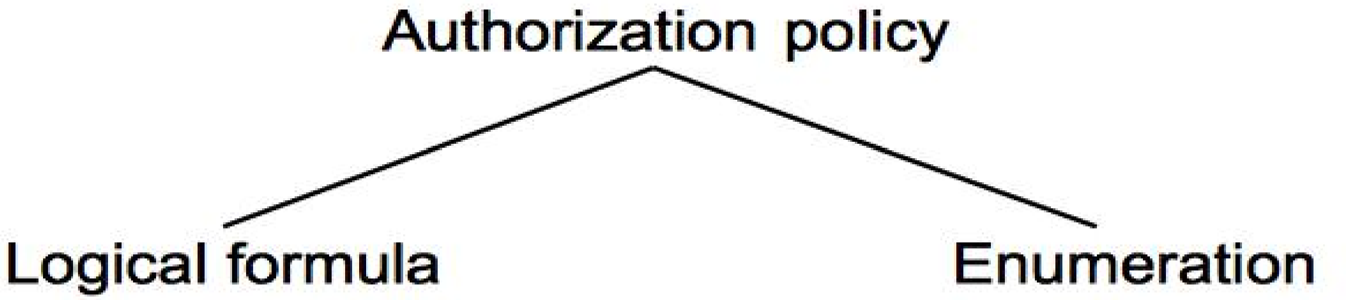
I·C·S
The Institute for Cyber Security

# ABAC

- ABAC model components
  - Users (U), subjects (S), objects (O), their attributes (UA, SA, OA) and access rights (R)
  - Authorization policies…


1. Constraints on subject attributes at creation and modification time.
2. Constraints on object attributes at creation and modification time.
UA
SA
OA
R
3. Authorization policy
U
S
Authorization
O
Constraints
Association
Creator

# ABAC Auth Policies



Authorization policy

Logical formula

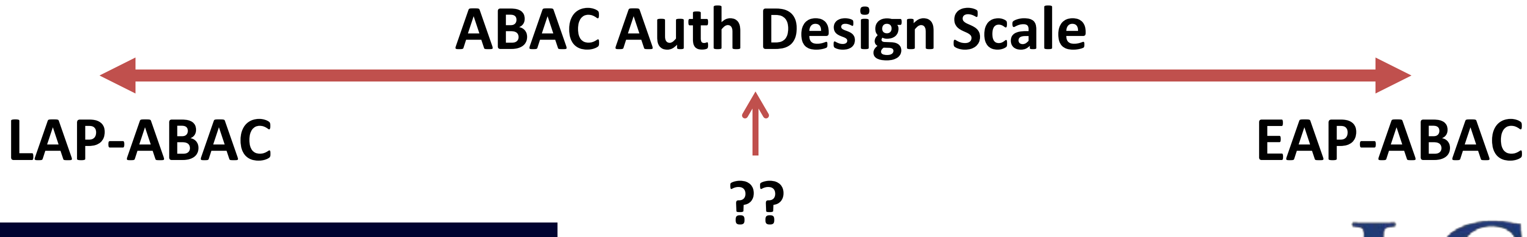
Enumeration

- Boolean expression
- E.g.: $auth(u,o,read) \leftrightarrow age(u)>18 \wedge age(u) <25$
- $\text{ABAC}_{\alpha}$ (Jin et al, DBSec 2012), HGABAC (Servos et al, FPS 2014)

- Set of authorizing tuples
- E.g.: {(age(u),19), (age(u),20), …(age(u),24)}
- Policy Machine (JSA 2011), 2-sorted-RBAC (Kuijper et al, SACMAT 2014)

# Objective

- Gain insights into different forms of ABAC auth policy representations
  - Logical Authorization Policy ABAC (LAP-ABAC)
  - Enumerated Authorization Policy ABAC (EAP-ABAC)
- Quantitative and qualitative comparison
  - Expressive power, ease of administration, etc.



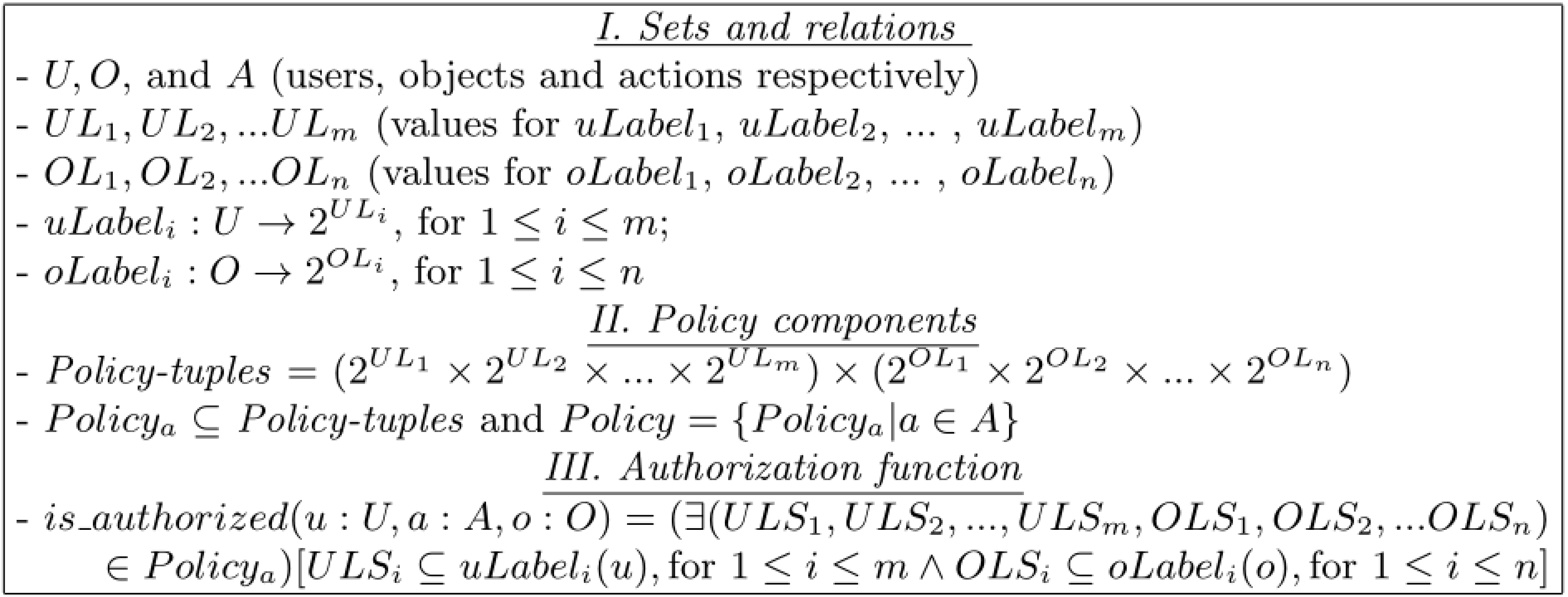
# Attribute Domain

- Assume attributes as functions
  - UA = {age,clr,friends}
  - Range(age) = {1…100}, Range(clr) = {H,L}, and Range(friends) = U
- Example finite domain attributes
  - Age of user, roles of user, object classification, etc.
- Example unbounded domain attributes
  - Friends of user, editors of objects, etc.
- We assume attribute domains to be finite

# Contributions and Results Summary

- Candidate LAP-ABAC and EAP-ABAC models for the purpose of this investigation
- LAP-ABAC and EAP-ABAC are equally expressive (recall finite domain)
  - Single (e.g. UA = {age}) and multi-attribute (e.g. UA = {age,group,clr}) ABACs are equally expressive
- However, LAP-ABACs and EAP-ABACs have their pros and cons on qualitative aspects

# EAP-ABAC$_{m,n}$



*I. Sets and relations*

- $U, O$, and $A$ (users, objects and actions respectively)
- $UL_1, UL_2, ...UL_m$ (values for $uLabel_1$, $uLabel_2$, ... , $uLabel_m$)
- $OL_1, OL_2, ...OL_n$ (values for $oLabel_1$, $oLabel_2$, ... , $oLabel_n$)
- $uLabel_i : U \rightarrow 2^{UL_i}$, for $1 \leq i \leq m$;
- $oLabel_i : O \rightarrow 2^{OL_i}$, for $1 \leq i \leq n$

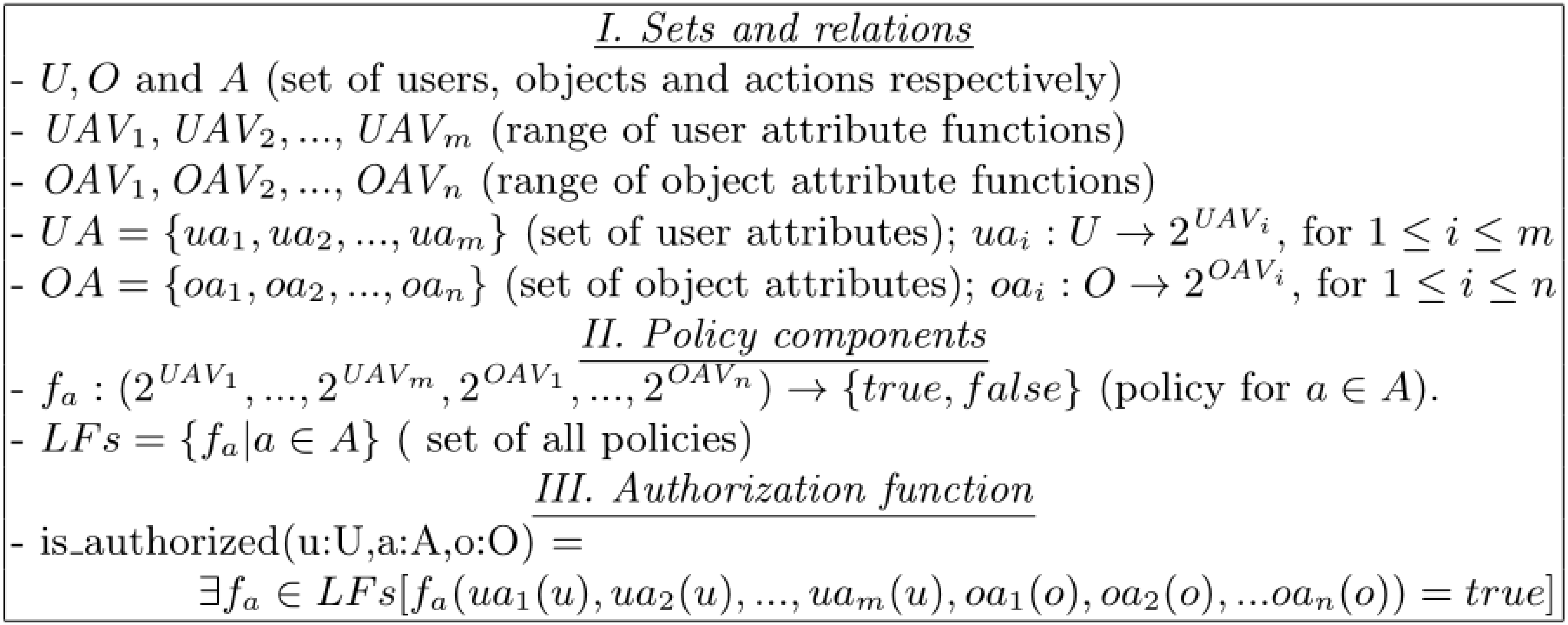
*II. Policy components*

- $Policy\text{-}tuples = (2^{UL_1} \times 2^{UL_2} \times ... \times 2^{UL_m}) \times (2^{OL_1} \times 2^{OL_2} \times ... \times 2^{OL_n})$
- $Policy_a \subseteq Policy\text{-}tuples$ and $Policy = \{Policy_a | a \in A\}$

*III. Authorization function*

- $is\_authorized(u : U, a : A, o : O) = (\exists(ULS_1, ULS_2, ..., ULS_m, OLS_1, OLS_2, ...OLS_n)$
  $\in Policy_a)[ULS_i \subseteq uLabel_i(u), \text{for } 1 \leq i \leq m \wedge OLS_i \subseteq oLabel_i(o), \text{for } 1 \leq i \leq n]$

# LAP-ABAC$_{m,n}$



*I. Sets and relations*

- $U, O$ and $A$ (set of users, objects and actions respectively)
- $UAV_1, UAV_2, ..., UAV_m$ (range of user attribute functions)
- $OAV_1, OAV_2, ..., OAV_n$ (range of object attribute functions)
- $UA = \{ua_1, ua_2, ..., ua_m\}$ (set of user attributes); $ua_i : U \rightarrow 2^{UAV_i}$, for $1 \leq i \leq m$
- $OA = \{oa_1, oa_2, ..., oa_n\}$ (set of object attributes); $oa_i : O \rightarrow 2^{OAV_i}$, for $1 \leq i \leq n$

*II. Policy components*

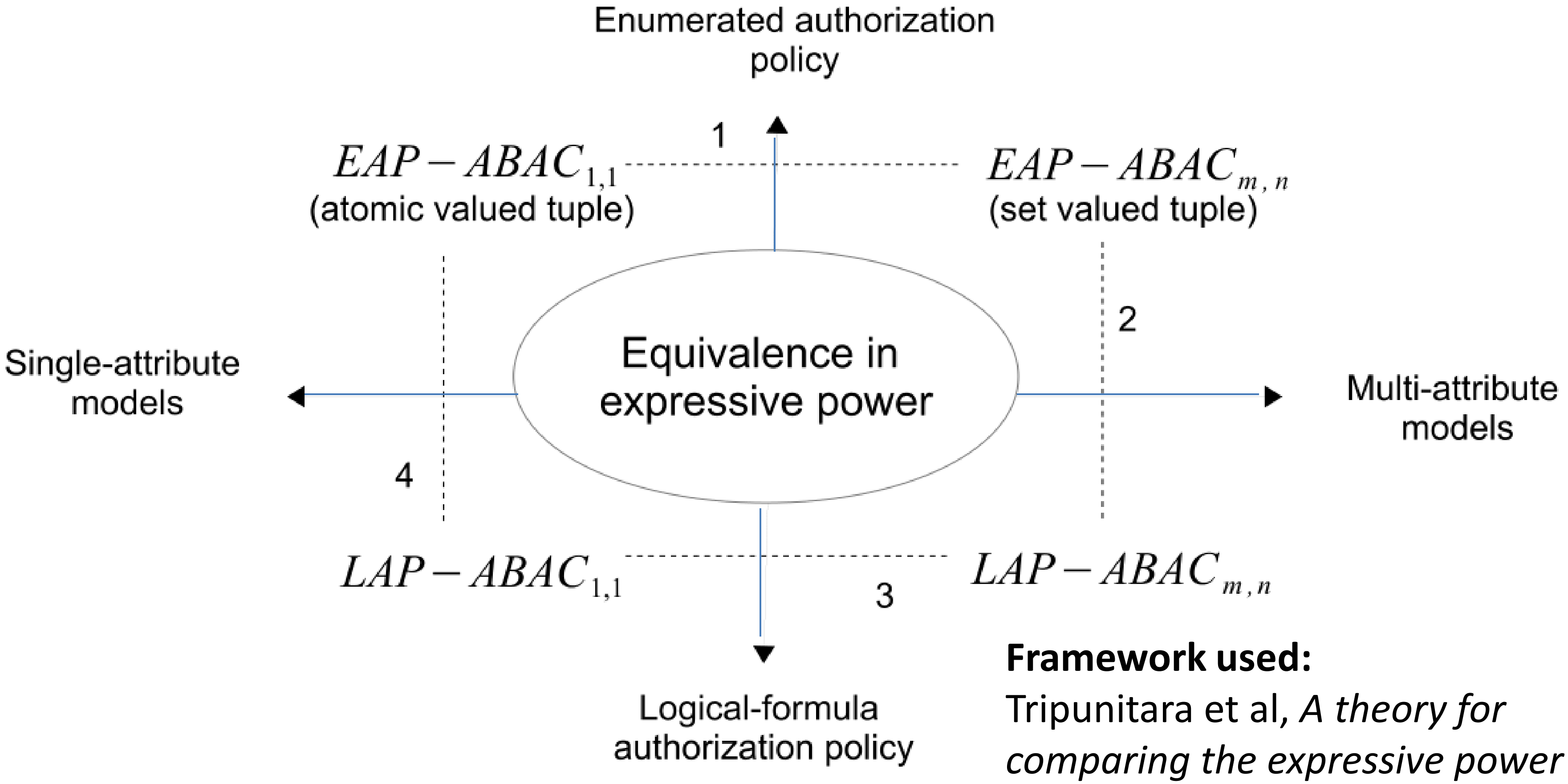
- $f_a : (2^{UAV_1}, ..., 2^{UAV_m}, 2^{OAV_1}, ..., 2^{OAV_n}) \rightarrow \{true, false\}$ (policy for $a \in A$).
- $LFs = \{f_a | a \in A\}$ ( set of all policies)

*III. Authorization function*

- is_authorized(u:U,a:A,o:O) =

$$\exists f_a \in LFs[f_a(ua_1(u), ua_2(u), ..., ua_m(u), oa_1(o), oa_2(o), ...oa_n(o)) = true]$$

# Expressive Power Equivalence



**Framework used:**
Tripunitara et al, *A theory for comparing the expressive power of access control models*, JCS 2007.

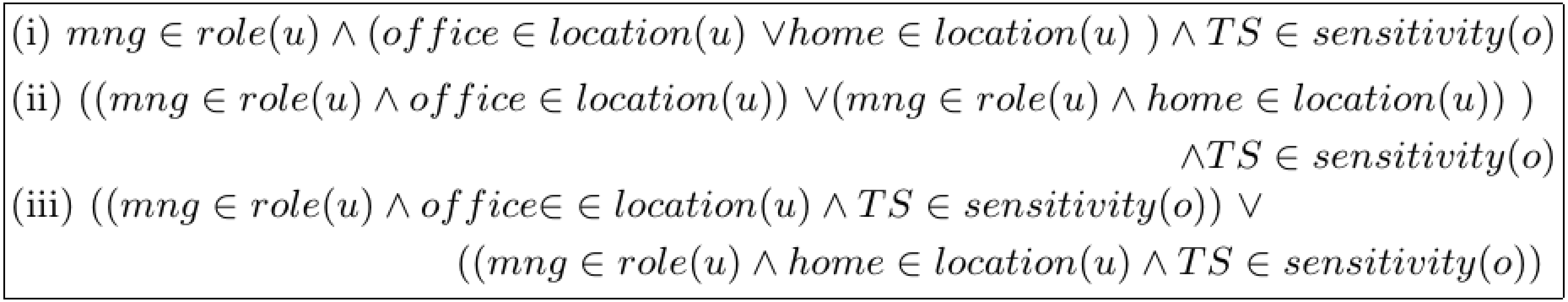
UTSA Engineering

I·C·S
The Institute for Cyber Security

# Auth Specification in LAP-ABAC

*Multiple ways to set up a policy*

(***Auth**$_{read}$ allows manager* to read *TS* objects from *home* or *office*).



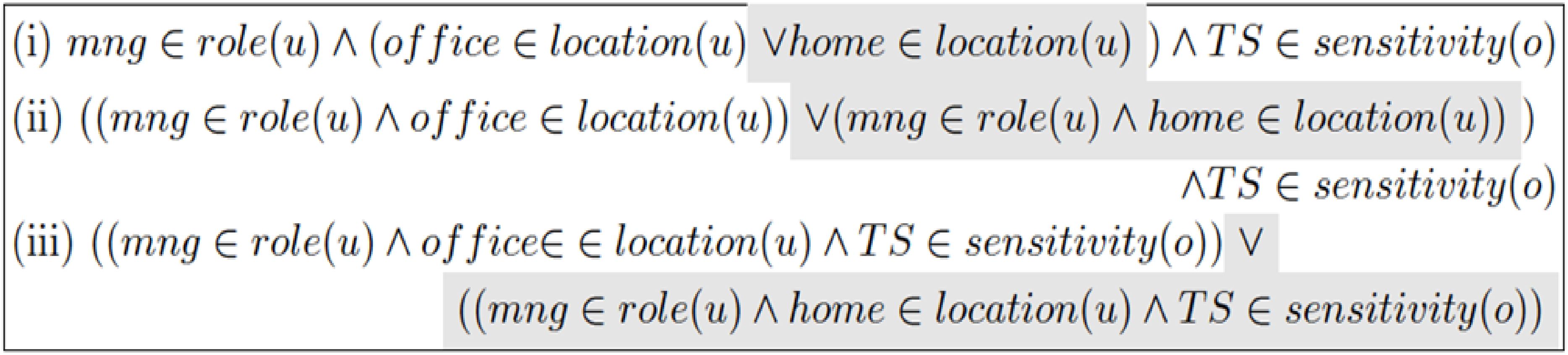
(i) $mng \in role(u) \wedge (office \in location(u)\ \vee home \in location(u)\ ) \wedge TS \in sensitivity(o)$

(ii) $((mng \in role(u) \wedge office \in location(u))\ \vee (mng \in role(u) \wedge home \in location(u))\ )$
$\wedge TS \in sensitivity(o)$

(iii) $((mng \in role(u) \wedge office\in \in location(u) \wedge TS \in sensitivity(o)) \vee$
$((mng \in role(u) \wedge home \in location(u) \wedge TS \in sensitivity(o))$

# Auth Update in LAP-ABAC

*Update* ***Auth***$_{\textbf{\textit{read}}}$ *so that*
*manager* can no longer read *TS* objects from *home*



(i) $mng \in role(u) \wedge (office \in location(u) \vee home \in location(u)\ ) \wedge TS \in sensitivity(o)$

(ii) $((mng \in role(u) \wedge office \in location(u)) \vee (mng \in role(u) \wedge home \in location(u))\ )$
$\wedge TS \in sensitivity(o)$

(iii) $((mng \in role(u) \wedge office\in \in location(u) \wedge TS \in sensitivity(o)) \vee$
$((mng \in role(u) \wedge home \in location(u) \wedge TS \in sensitivity(o))$

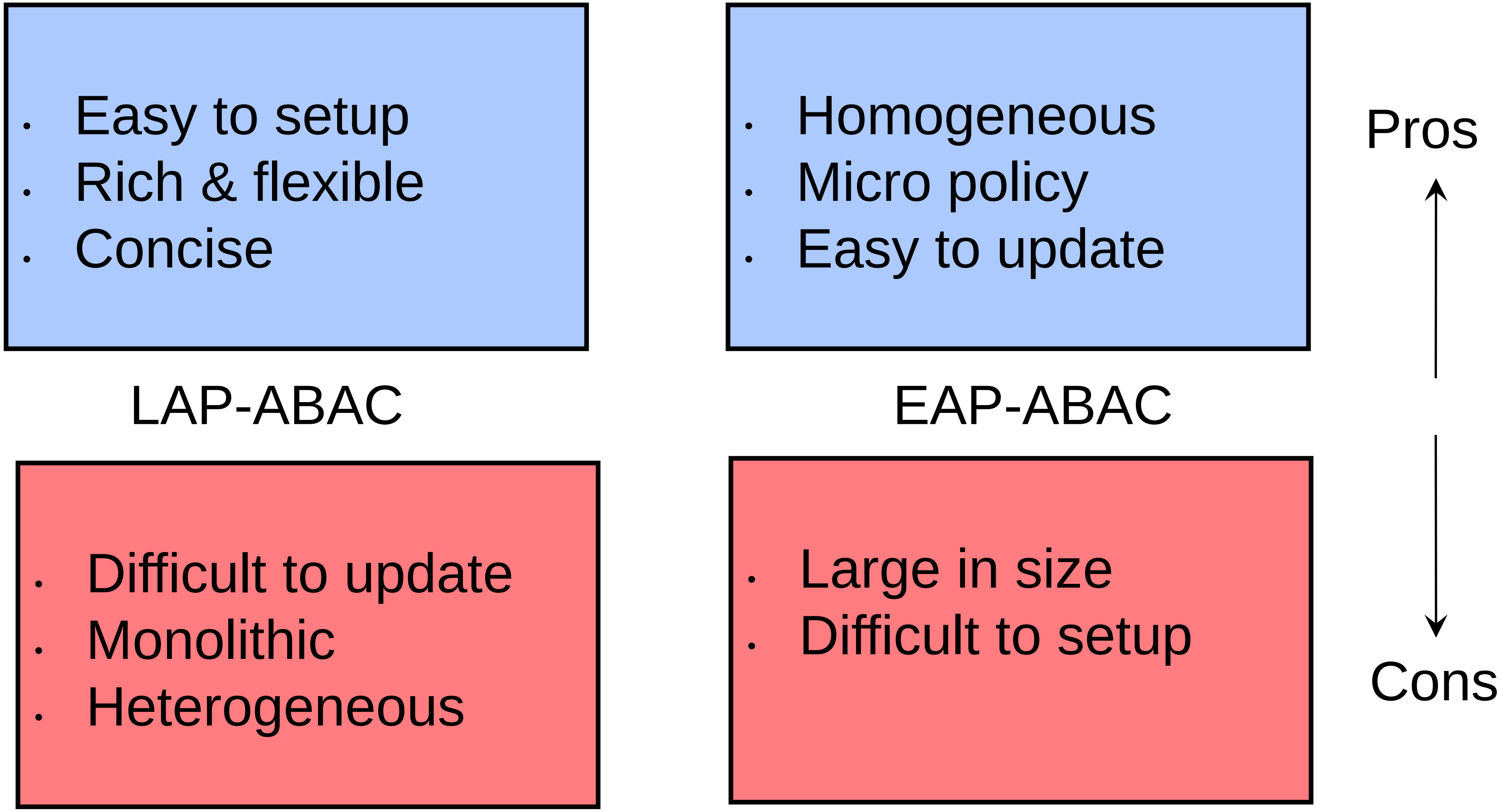
# Auth Update in EAP-ABAC

- ❑ $\text{Auth}_{\text{read}} \equiv$ {(mng, home, TS), (mng,office,TS)}
- ❑ $\text{Auth`}_{\text{read}} \equiv$ { ~~(mng, home, TS),~~ (mng,office,TS)}

# Canonicalization of EAP-ABAC

- Suppose $\text{Auth}_{\text{write}}$ = {({mgr},{TS}), ({mgr,Dir},{TS})}
- This can be reduced to $\text{Auth}_{\text{write}}$ = {({mgr},{TS})}
- EAP-ABAC auth policies can be efficiently canonicalized as per policy semantics

# Comparison



LAP-ABAC

- Easy to setup
- Rich & flexible
- Concise

- Difficult to update
- Monolithic
- Heterogeneous

EAP-ABAC

- Homogeneous
- Micro policy
- Easy to update

- Large in size
- Difficult to setup

Pros

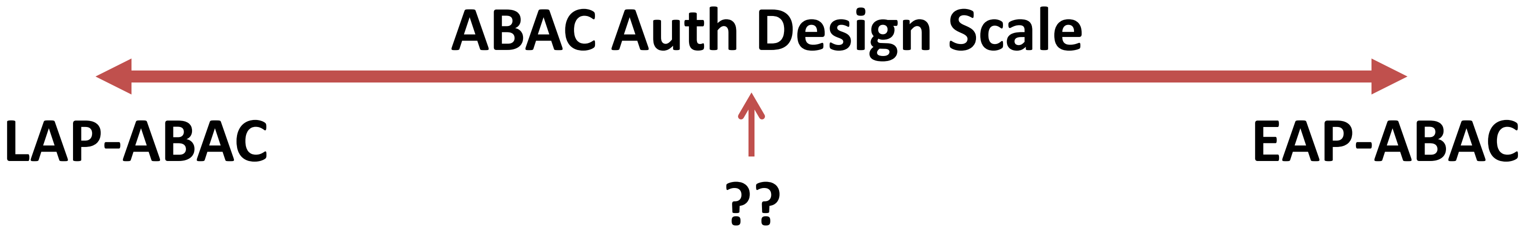
Cons

UTSA Engineering

I·C·S The Institute for Cyber Security

# Conclusion

- ❑ ABAC should be designed with objectives that go beyond expressive power
  - ➢ E.g.: Administration of authorization policy
    - Setting up new policies, update existing policies, etc



ABAC Auth Design Scale

# Q&A

Consider submitting your work to ACM CODASPY ’16
Submission deadline: Sept 15, 2016
http://www.codaspy.org/

7th ACM Conference on Data and Application Security and Privacy (ACM CODASPY 2017)

March 22-24, 2017, Scottsdale, Arizona, USA

# Thank you!