I·C·S
The Institute for Cyber Security

C·SPECC
Center for Security and Privacy Enhanced Cloud Computing

Role-Based Administration of Role-Based Smart Home IoT

Mehrnoosh Shakarami
Ravi Sandhu

Institute for Cyber Security (ICS)
Department of Computer Science
University of Texas at San Antonio (UTSA)

UTSA Computer Science

Outline

Introduction and Background

- IoT Access Control Requirements
- Role-Base Access Control (RBAC)

RBAC Administrative Model for Smart Home IoT

- Operational Model for Smart Home IoT
  - EGRBAC Model Introduction
- Administrative Model for Smart Home IoT
  - RBAC Administrative Policy Model
  - Administrative Use Case
  - Proposed Model's Properties and Restrictions

Conclusion and Future Work

- What could be done?

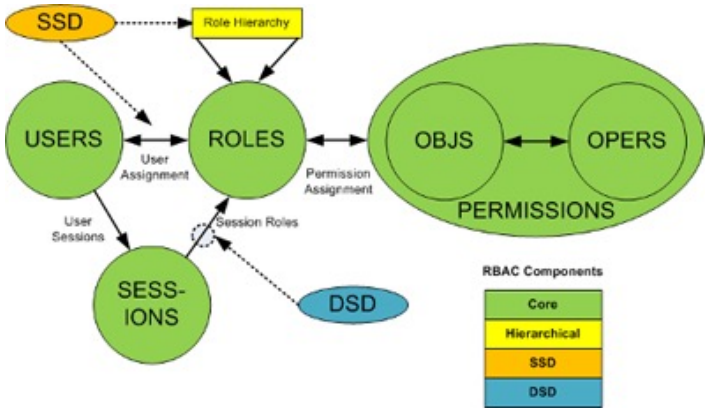
Internet of Things (IoT)

IoT Access Control

Policy Specification	Granularity	●	●	●	●	●
	Context Awareness	●	●	●	●	●
Policy Management	Handling the complexity of Environment	○	●	●	◑	●
	Usability	●	○	○	●	○
	Multi-domain Administration	○	●	●	◑	●
Policy Enforcement	Minimum user involvement	◑	●	●	●	●
	Light-weight	◑	◑	◑	●	●
	Reliability and Availability	●	●	●	●	●

Role-Based Access Control (RBAC)

- RBAC mediate permission assignment to users via the concept of a role.
- RBAC virtues include its policy neutrality, adherence to least privilege principle, and ease of management.
- Administration is facilitated by assigning different users to define roles or making changes to existing role sets of the system.



Outline

Introduction and Background

- IoT Access Control Requirements
- Role-Base Access Control (RBAC)

RBAC Administrative Model for Smart Home IoT

- Operational Model for Smart Home IoT
  - EGRBAC Model Introduction
- Administrative Model for Smart Home IoT
  - RBAC Administrative Policy Model
  - Administrative Use Case
  - Proposed Model's Properties and Restrictions

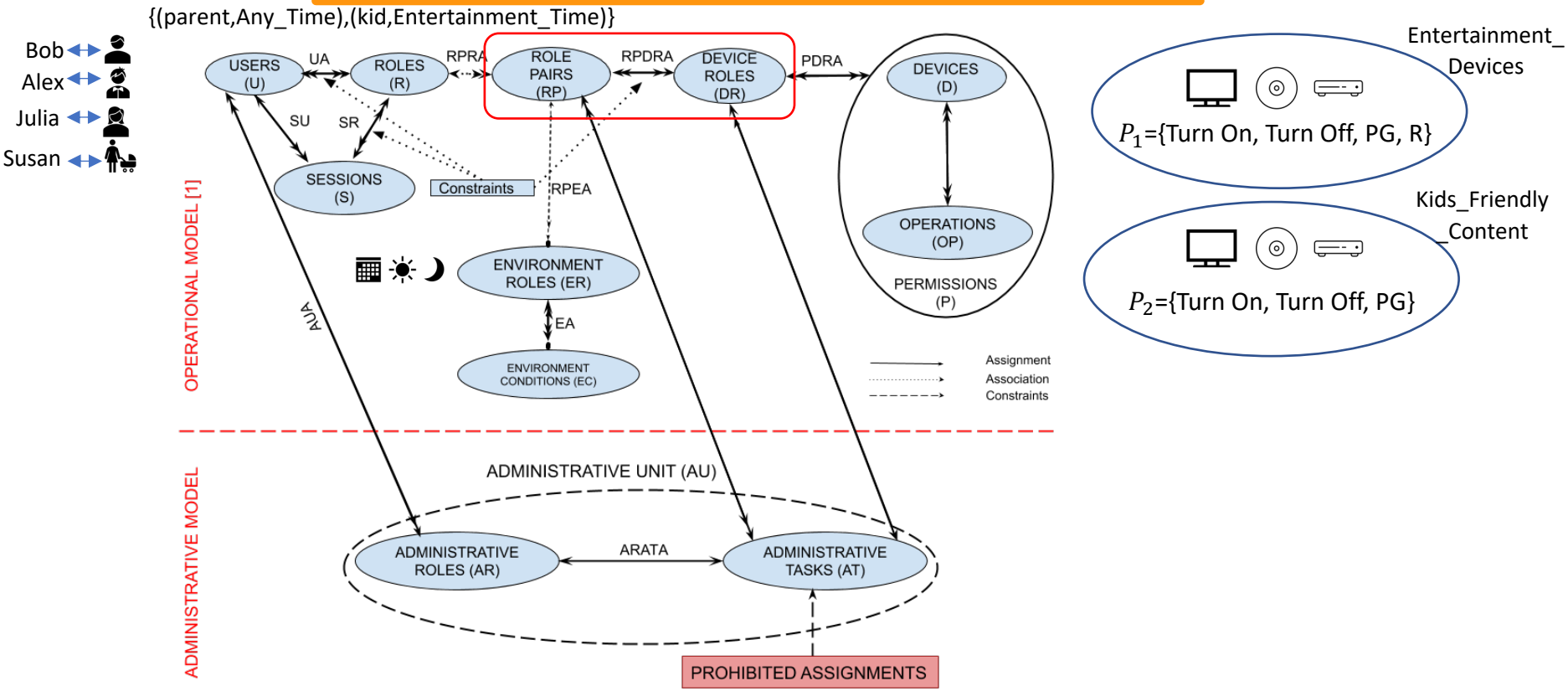
Conclusion and Future Work

- What could be done?

Operational Model for Smart Home IoT

- Many operational models have been proposed for Smart Home IoT.
- EGRBAC (Extended Generalized RBAC) is a generalized RBAC model for user-to-device operational access [Ameer et al., 2020]
  - Provides granularity by defining Device Roles (DR)
    - It is permission-centric, instead of device-centric
  - Capture the environmental conditions through Environmental Roles (ER)
  - Contextuality has been provided via Role Pair (RP) definition

Basic Administrative Model



- Administrative Unit (AU) is the core component of decentralization.
- Each AU contains a unique specific Administrative Role (AR) and a set of Administrative Tasks (AT).
- Authorization is scoped as a set of administrative tasks defined to manage corresponding assignments in an operational model.

RBAC Administrative Model for Smart Home

- Our model addresses the administration of EGRBAC.
- Our model is decentralized:
  - Avoid single point of failure
  - Improve users' privacy
- Potential changes to the dynamic environment of smart home include:
  - Add a new user
  - Add a new device ☑
  - Modification of current assignments in the model ☑

Smart Home Use Case

au_1 = Entertainment_Management

ar_1 = Entertainment_Manager

at_1 = { DR={ENTERTAINMENT_DEVICES, KIDS_FRIENDLY_CONTENT} RP={(KID, ENTERTAINMENT_TIME), (PARENT,ANY_TIME), (GUEST, ANY_TIME)} }

au_2 = Adult_Management

ar_2 = Adult_Manager

at_2 = { DR={ADULT_CONTROLLED} RP={(PARENT,ANY_TIME), (BABY_SITTER, ANY_TIME)} }

((KID,ENTERTAINMENT_TIME), ENTERTAINMENT_DEVICES)

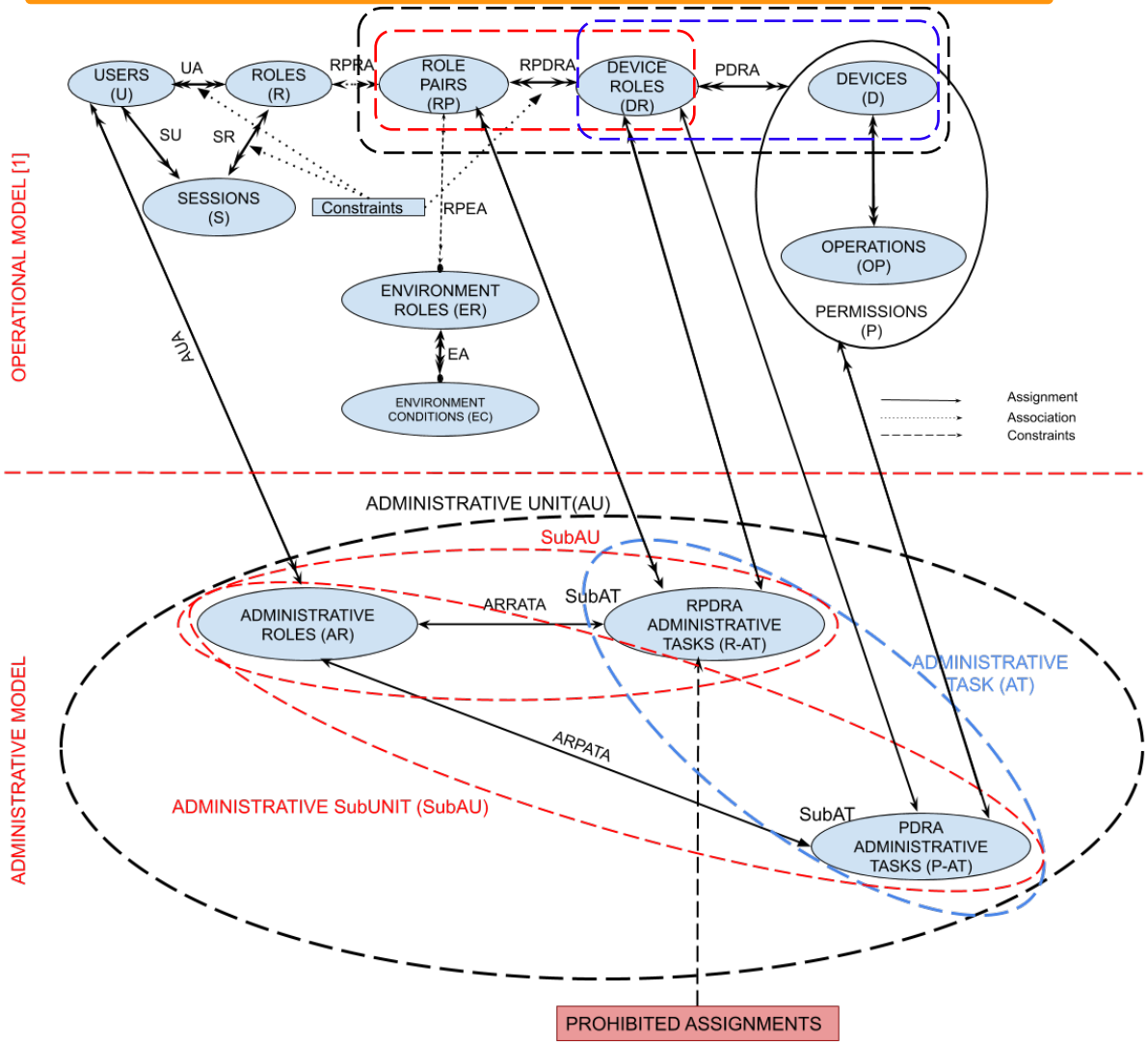
Prohibited Assignments

au_3 = Ownership_Control

ar_3 = Home_Owner

at_3 = { DR={OWNER_CONTROLLED} RP={(PARENT,ANY_TIME)} }

Extended Administrative Model



Extended Administrative Model Formalization

Core Components

– $AUser \subset U$ is a set of administrator users.

– AR is a set of administrative roles, authorized to manage a specified subset of RPDRA.

– $AUA \subset AUser \times AR$ is a many to many administrator user to administrative role assignment.

– $AU = \cup_{\forall i} SubAU_i$,is a set of administrative sub-units (SubAU).

– AT is a set of administrative sub-tasks (SubAT), i.e. $AT = P-AT \cup R-AT$.

– $R-AT \subseteq (2^{RP} \times 2^{DR}) \backslash ProhibitedAssignment$ is a set of administrative tasks related to RPDRA assignment, which contains all pairs of cross product of a subset of RP, and a a subset of DR, but a set of Prohibited Assignments has to be excluded.

– $P-AT \subseteq (2^{P} \times 2^{DR})$ is a set of administrative tasks related to PDRA assignment.

– $SubAU \subset AR \times \{R-AT, P-AT\}$ is a administrative sub-unit.

Administrative Constraint

– $ProhibitedAssignment$ is a set of prohibited (rp, dr) pairs each of which is a member of possible pairs of assignment but specified to be forbidden to be added to RPDRA by design, ($Constratints \subset RP \times DR$).

Administrative Authorization

– $ARRATA \subseteq AR \times R-AT$, is a one to one AR to R-AT assignment determining the scope of administrative control for a given AR on RPDRA.

– $ARPATA \subseteq AR \times P-AT$, is a one to one AR to P-AT assignment determining the scope of administrative control for a given AR on PDRA.

– $ARAUA \subseteq AR \times AU$ is a one to one AR to AU assignment, determines which AU is under control of a given AR.

Derived Administrative Relations

– $AR_{at \in SubAT} \subset SubAT \in AT \times AR : AR_{AT}(at) = ar \in AR : at \in ARRATA(ar) \vee at \in ARPATA(ar)$: many to one administrative subtask to administrative role function which determines which AR can manage this AT.

– $RolePair_{at \in AT} \subseteq 2^{RP}$ determines which role pairs are included in a given administrative task.

– $DeviceRole_{at \in AT} \subseteq 2^{DR}$ discovers the device roles which are included in a given administrative task.

– $InclusiveTask((st, dr)) \subseteq (\{st \in RP\} \vee \{st \in P\}, dr \in DR) \times \{AT \cup FALSE\}$ determines the association of a (st, dr) to an administrative task if this pair is currently defined as a member of that administrative task, if no inclusive administrative task found, it returns FALSE.

Authorization Functions

– $\textsc{assignRPDR}(auser \in AUser, ar \in AR, rp \in RP, dr \in DR) \equiv$
$((\text{at} = \text{InclusiveTask(rp,dr)} \wedge ar = AR_{at} \wedge (rp, dr) \notin RPDRA)) \Rightarrow RPDRA' = RPDRA \cup (rp, dr)$

– $\textsc{revokeRPDR}(auser \in AUser, ar \in AR, rp \in RP, dr \in DR) \equiv$
$((\text{at} = \text{InclusiveTask(rp,dr)} \wedge ar = AR_{at} \wedge (rp, dr) \in RPDRA)) \Rightarrow RPDRA' = RPDRA \backslash (rp, dr)$

– $\textsc{assignPDR}(auser \in AUser, ar \in AR, p \in P, dr \in DR) \equiv$
$((\text{subat} = \text{InclusiveTask(p,dr)} \wedge ar = AR_{subat} \wedge (dr, p) \notin PDRA)) \Rightarrow PDRA' = PDRA \cup (p, dr)$

– $\textsc{revokeRPDR}(auser \in AUser, ar \in AR, p \in P, dr \in DR) \equiv$
$((\text{at} = \text{InclusiveTask(p,dr)} \wedge ar = AR_{at} \wedge (p, dr) \in PDRA)) \Rightarrow PDRA' = PDRA \backslash (p, dr)$

Model Properties and Restrictions

- Model Properties:
  - Decoupled Assignment and Revocation
  - Symmetric Assignment and Revocation
  - Generalizability

- Model Restrictions:
  - Continuous Usage Control
  - Quota-based Access Enforcement
  - Conflict of Interest

Outline

Introduction and Background

- IoT Access Control Requirements
- Role-Base Access Control (RBAC)

RBAC Administrative Model for Smart Home IoT

- Operational Model for Smart Home IoT
  - EGRBAC Model Introduction
- Administrative Model for Smart Home IoT
  - RBAC Administrative Policy Model
  - Administrative Use Case
  - Proposed Model's Properties and Restrictions

Conclusion and Future Work

- What could be done?

Conclusion and Future Directions

- In Conclusion:
  - We proposed an RBAC administrative model based on EGRBAC operational model in smart home environments.
  - We introduced the concept of administrative unit, which consists of a unique administrative role and a set of administrative tasks.

- Future Directions:
  - Address Model Constraints (continuous usage control, quota-based management
  - Device to Device (D2D) communication helps with providing an autonomous intelligence in IoT environments.

Acknowledgement

This work is partially supported by NSF CREST Grant 1736209.

Thank You
Any Questions?