Role-Based Administration of Role-Based Smart Home IoT

Mehrnoosh Shakarami and Ravi Sandhu
mehrnoosh.shakarami@my.utsa.edu,ravi.sandhu@utsa.edu
Institute for Cyber Security (ICS) & NSF Center for Security and Privacy Enhanced Cloud Computing (C-SPECC)
Department of Computer Science, University of Texas at San Antonio

ABSTRACT

Using role-based access control (RBAC) to manage RBAC is among RBAC's attractive benefits, contributing to its long-standing dominance in practice. Administrative models facilitate management of (mostly configuration) changes in the underlying operational models. Overall system security is crucially dependent on both the administrative and operational models.

In this paper, we develop an RBAC administrative model to manage authorization assignments in the EGRBAC (enhanced generalized role-based access control) operational model for smart home IoT. We design the administrative model based on pairwise disjoint Administrative Units, each of which contains a uniquely assigned administrative role and a set of administrative tasks. Administrative tasks determine the administrative permissions available to manage the operational model assignments. We begin with a model containing a single administrative unit and then extend it to include additional units. Multiple administrative units enable decentralized administration which could be adapted to provide scalability in inherently distributed and large-scale environments beyond smart home, such as smart buildings or smart campuses. We provide formalism of our proposed model and illustrate it by specifying operational and administrative use cases. Although, the model is proposed based on a specific smart home operational model, our approach could be applied to environments with similar dynamics.

KEYWORDS

RBAC Administrative Model, Decentralized Administration, Smart Home

ACM Reference Format:
Mehrnoosh Shakarami and Ravi Sandhu. 2021. Role-Based Administration of Role-Based Smart Home IoT. In *Proceedings of the 2021 ACM Workshop on Secure and Trustworthy Cyber-physical Systems (SAT-CPS'21), April 28, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3445969.3450426

1 INTRODUCTION

Classic Role Based Access Control (RBAC) approach has been proposed to mediate permission assignment to users via the concept of a role. RBAC improves on its predecessor models because of its policy neutrality, ease of management and adherence to least privilege principle. Moreover, it provides built-in support for Static and Dynamic Separation of Duty (SSoD and DSoD). One of the RBAC benefits is after an operational RBAC model has been established, the administration is facilitated by assigning different users to define roles or making changes to existing role sets of the system. However, the notion of an administrative model is not included in the NIST standard [11] nor the seminal RBAC [25] models. Administrative RBAC (ARBAC) has been proposed as an approach to use RBAC itself to manage different aspects of RBAC [23].

We consider operational models to be dynamic as we are aiming for a smart home environment and recognize the need to develop administrative models in order to govern access changes in such a system. Moreover, as RBAC has been widely utilized in large-scale environments, the ever-changing nature of operational models has to be considered in order to efficiently perform administration. Although the need of an administrative model is independent of the size of the operational environment or its notion of centralization, the decentralization, growing size and dynamic nature of operational models do make the administration more complicated and challenging. Many administrative models for RBAC have been proposed in literature [5, 6, 19, 23, 24, 31], in either centralized or decentralized ways.

There are many situations in which it would not be reasonable to make access control decisions only based on roles assigned to individuals. Instead, other contextual information such as environmental conditions and location should be involved in access control. One rising example is the integration of smart home IoT devices into people's everyday lives, which raise the need for specific access control models. Even with one IoT device in the smart home, some dynamics are essential to be considered. Moreover, it is quite possible for access objects to be added/deleted to the smart home environment. To address this requirement, several access control models tailored to smart home IoT environments have been proposed. Recently, Ameer et. al. [1] proposed EGRBAC (extended generalized RBAC), an access control model applicable in smart home environments, which is a dynamic and fine-grained RBAC model, and provides access to legitimate users considering different situational conditions. EGRBAC provides access at the permission-level granularity instead of device-level, which is a requisite for smart home users in many situations [13, 29].

In this paper we propose a role-based administrative model corresponding to the EGRBAC operational model to govern authorization functionalities, through management of important assignments in the operational model. We introduce the concept of Administrative Units (AU) which uniquely associates an administrative role to an administrative task. Our first model has been designed to manage RPDRA (role pair to device role) assignment in EGRBAC, which determines the access policy (i.e., which role pair has the

SAT-CPS'21, April 28, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8319-6/21/04…$15.00
https://doi.org/10.1145/3445969.3450426

access to what device role).[1] We then extend our model to govern other assignment relations in EGRBAC. So, similar to what has been initiated in ARBAC'97 [23], which separates user-role and permission-role assignments, our proposed model also adopts the notion of separation in administration of different assignment relations of the operational model. We augment our proposed model by providing use case scenarios for both operational and administrative models.

The remainder of this paper is organized as follows. Section 2 summarizes previous studies on both topics of RBAC administration and smart home access control. Moreover, a brief description of EGRBAC as our operational model is provided. In Section 3, we present our proposed administrative model to manage policy definitions in underlying operational model (EGRBAC), which is followed by a formal representation of the model. To illustrate our model, we provide operational and administrative use case scenarios in a smart home environment in Section 4. Section 5 presents an extension to the previously proposed administrative model, extending it in order to enable it to manage multiple assignments of operational model. Corresponding changes/extensions to the preceding model formalism and use cases are presented. Some salient features of the proposed administrative models along with its limitations are discussed in Section 6. Section 7 concludes the paper.

2 BACKGROUND AND RELATED WORK

In this section we outline previous related works in two parts. First prior work on RBAC administrative models is summarized. Then we give a brief statement of some work in the area of smart home, including an explanation of EGRBAC [1] which is the operational model we build our administrative model upon.

2.1 RBAC Administration

Popularity of RBAC to a large degree has its origins in its ease of administration. Several research works have been done aiming to propose role-based administrative models, based on different administrative assumptions and principles while offering different levels of permissiveness.

ARBAC97 [23] is the pioneering work in role-based administration, in which RBAC is used for administration of RBAC itself. ARBAC97 has a distinct set of administrative roles/permissions and includes three base components as independent sub-models for user-role assignment (URA), role-permission assignment (PRA), and role-role assignments (RRA). These components use the notions of role range and prerequisite roles in order to define restrictions for exercising administrative permissions including granting and revocation. Authors in [26] analyzed the ARBAC97 model. In another research [19], an accountability mechanism for execution of access rights in ARBAC97 has been presented to enhance its security.

The concept of mobile/immobile users has been introduced in [24] which made enhancements over URA and PRA in ARBAC97. Another related study is ARBAC02 [17], in which a bottom-up administration approach has been proposed in contrast to the top-down approach of ARBAC97. Authors try to overcome shortcomings of ARBAC97 which result from unnecessary integration of user and permission pools and role hierarchy. So, it assumes users and user pools to be independent from role hierarchy. Researchers in RHA (Role Hierarchy Administration [6]) tried to improve the role hierarchy management in ARBAC97 by providing a scoped administrative model. Authors used the notion of administrative scope, as a unit of administration, which dynamically changes according to role hierarchy manipulation. Authors then proposed SARBAC to construct role hierarchies in a decentralized way.

One example of RBAC administration is proposed by Moffet and Sloman [16] in which the domain concept is used to refer to administrative domains in distributed systems. In that work authority is not controlled by a single administrator, rather it is negotiated between a group of independent administrators who have limited trust to each other. X-GTRBAC Admin [5] is another research in which the domain concept refers to distributed administrative domains. X-GTRBAC Admin proposed an XML-based administrative model to address the requirements of a dynamic multi-domain environment with partially ordered administrative domains.

Another related research is reported in [28] which is a formal administrative model, namely AMTRAC, which was designed for temporal RBAC and analyzed in [14]. There are a great number of research works focused on RBAC administration in distributed environments [7, 30, 34] recognizing multi-domain decentralized access control management as an important administration issue, which are orthogonal to our focus in this paper.

2.2 Smart Home IoT Access Control

There is a rich body of research on security of IoT [2, 3, 12, 15, 32]. Authors in [18, 20–22] review the access control requirements and approaches to protect security and privacy of IoT. Security of a smart home environment, as a specific application of IoT, has been investigated in [8, 33]. Common to all of these studies, access control has been recognized as a critical requirement to build a secure IoT environment.

A context-sensitive access control approach for smart home has been proposed in [9] in which policies are focused to control access to users' personally identifiable information (PII). Authors use semantic network knowledge graphs to define the context in a smart home environment and supplement their work with an anomaly detection sub-system to inform users about suspicious activities. Another related research is reported in [4] in which standalone ABAC model was proposed for smart home environments, considering the NIST Next Generation Access Control (NGAC) [10] specifications to specify ABAC requirements. Both of these works lack in presenting a specific operational model for their proposed approaches.

In this paper, we adopt the EGRBAC model presented in [1], as the operational model for a smart home environment. EGRBAC takes into account requirements and challenges of the access control in a smart home environment and enhances over traditional RBAC in order to satisfy the required properties. These characteristics along with a formal model proposed in the work inspired us to consider it as our operational model of choice. This model is briefly described as follows.

2.2.1 EGRBAC Model. EGRBAC has been proposed by Ameer et. al. [1], to provide a fine-grained access control model for smart

[1]The details are described in the next section.

Table 1: EGRBAC Model Formalization [1]

Users, Roles and Sessions $-U$, R and S are sets of users, roles and sessions respectively $-UA \subseteq U \times R$, many to many users to role assignment (homeowner specified) $-SU \subseteq S \times U$, many to one sessions to user relation that assigns each session to a single user who controls the session $-SR \subseteq S \times R$, many to many session to roles relation that assigns each session to a set of roles that can change under user control, where $(s_i, r_j) \in SR \Rightarrow (\exists u_k \in U)[(s_i, u_k) \in SU \wedge (u_k, r_j) \in UA]$; by definition of SU, u_k must be unique
Devices, Operations, Permissions and Device Roles $-D$, OP, P and DR are sets of devices, operations, permissions and device roles respectively $-P \subseteq D \times OP$, every permission is a device, operation pair (device manufacturer specified) $-$ $PDRA \subseteq P \times DR$, a many to many permissions to device roles assignment (homeowner specified)
Environment Roles and Environment Conditions $-ER$ and EC are sets of environment roles and environment conditions respectively $-EA \subseteq 2^{EC} \times ER$, many to many subsets of environment conditions to environment roles assignment (homeowner specified)
Role Pairs $-RP \subseteq R \times 2^{ER}$, a set of role pairs specifying all permissible combinations of a user role and subsets of environment roles (homeowner specified); for every $rp = (r_i, ER_j) \in RP$, let $rp.r = r_i$ and $rp.ER = ER_j$ $-RPRA \subseteq RP \times R$, many to one role pairs to role association induced by RP, where $RPRA = \{(rp_m, r_n) \mid rp_m \in RP \wedge rp_m.r = r_n\}$ $-RPEA \subseteq RP \times 2^{ER}$, many to one environment roles to role pairs association induced by RP, where $RPEA = \{(rp_m, ER_n) \mid rp_m \in RP \wedge ER_n = rp_m.ER\}$
Role Pair Assignment $-RPDRA \subseteq RP \times DR$, many to many role pairs to device roles assignment (homeowner specified)
Authorization Predicate $-$ The authorization predicate takes 4 inputs: session s_i, device d_j, operation op_k and set of active environment conditions EC_l; a session s_i can access device d_j with operation op_k when the set of environment conditions EC_l is active iff the following predicate is true: $(\exists (rp_m, dr_n) \in RPDRA)$ $[((d_j, op_k), dr_n) \in PDRA \wedge$ $(s_i, rp_m.r) \in SR \wedge$ $rp_m.ER \subseteq \{er \in ER \mid (\exists EC'_l \subseteq EC_l)$ $[(EC'_l, er) \in EA]\}]$

home environments. Authors provide finer grained RBAC model, compared to existing models, in that the scope of control has been defined to be at device-operation level. Instead, other RBAC models in the same context commonly provide the device level granularity of control.

Different Device Roles (DR) have been created based on categorizing available manufacturer-specified operations in a device. It is also possible to put pairs of (device, operation) in the same DR for different devices. Then, permissions would be assigned to device roles instead of devices themselves, making the model permission-centric. As a result, it is possible in EGRBAC to grant partial access to a device for different users, for instance a DR called Dangerous Devices could contain on/off operation for the oven as well as turning smoke detector on/off.

On the other hand, EGRBAC captures environmental context such as time and location using Environment Conditions (EC) which subsequently would activate/deactivate Environment Roles (ER). For instance, light sensors would capture the daylight and determine whether it is daytime or nighttime. Multiple subsets of ECs could be grouped together as an ER, which would later be coupled by regular roles to create Role Pairs (RP). EGRBAC assigns Device Roles (DR) to Role Pairs (RP) to establish the access policy, by defining RPDRA relationship. Formal definition of EGRBAC is given in Table 1.

EGRBAC is an operational model of our choice upon which we would build our administrative model, in that its provided granularity along with context-awareness make it a suitable choice for access control in smart home environments. However, this model is limited to govern only user to device accesses and leaves device to device access control for future investigation. Correspondingly, our administration model would also inherit the same constraint.

3 ADMINISTRATIVE MODEL

Our model specifically addresses the administration of EGRBAC [1]. However, it could be simply extended to manage other more sophisticated access control models with similar dynamics. The use of RBAC for RBAC administration enables us to separate governing of different assignments in corresponding operational model. In case of EGRBAC (as our operational model), we have different relations to be administered including assigning users to roles, defining new environmental conditions, introducing new role pairs and assignment of device roles to role pairs, each of which could be a component of administration.

We classify possible changes in smart home environment into three classes which need to be administered.

(1) New User Added: A new individual could join to the set of smart home users any time, which consequently needs administrative changes to be done such as defining a new role, an environment role or a role pair. We recognize adding a new user to be an infrequent event. So, we consider this case orthogonal to central focus of this paper. Its administration would be centralized, say, in the homeowner.
(2) New Device Added: Adding a new device is likely to happen increasingly frequently, considering the surge in smart home devices to be available nowadays. This change should be reflected in access control model by defining new device roles, making changes to current PDRA assignment or new assignments of permission to device roles through adding new PDRA relations. Establishment of new access control policies through managing RPDRA, is also a plausible administration requirement. In this paper, we focus on governing RPDRA and PDRA to address these requirements. We assume making changes in an existing device role or defining a new device role is centrally managed in some way.
(3) Modify Current Assignments: Sometimes it is required to change current assignments in a smart home, even if there is

no change in the set of users or devices. For instance, adding a new constraint for assigning a device role to a role pair (modify RPDRA), changing the set of (device, permission) pairs which have been assigned to a device role (modify PDRA). Modifying current PDRA sometimes is required as a result of adding a new device to the system, by adding new (device, permission) pairs to current PDRA. We focus on PRDA and RPDRA administrative modifications. Although other assignment changes are plausible to be required, e.g. change a user's role (UA modification), we consider those changes out of scope.

We recognize administration to be best done if it is decentralized. Centralized administration generates a single point of failure. Moreover, even in a small environment like a smart home, decentralized administration is worthy to consider. Suppose one of the administrator users are not available to manage/delegate the access control authorizations. A decentralized approach would bring the benefit of presence of another assigned administrator user who could do the task. Decentralized administration also helps to improve user's privacy by defining all permissions to manage a user's privacy zone contained in a separate administrative unit, and specify that user as the only possible user who could be assigned to the correspondent administrative role. In this paper, decentralization has been applied on two assignment relations (PDRA and RPDRA) in EGRBAC. We develop a formal description of administrative concepts and constraints in the following.

3.1 Model Description

Access control is embodied in different authorization assignments of the EGRBAC model, including UA, RPRA, PDRA, RPDRA, etc. These components collectively would establish the access control policy of the system. In this paper, we first focus on the RPDRA assignment through which device roles would be assigned to role pairs and considered as the central step of access policy establishment. Our administrative model focuses on managing the operational access control model in a way that any legitimate user in the smart home environment only has access to what s/he is authorized to access. In other words, insider threats are limited such that our system observes the least privilege principle[2] while managing the authorization assignments.

In order to design our administrative model to be decentralized, we use the abstract of Administrative Unit (AU), which is a core component of decentralization in our model. As indicated in [25], it highly matters how to scope the administrative authority conferred to administrative roles. In our model, each administrative unit contains a unique specific Administrative Roles (AR) and a set of Administrative Tasks (AT). In other words, each administrative role is authorized to manage the administrative tasks within a given administrative unit. This authorization is scoped as a set of administrative tasks defined to manage corresponding assignments in an operational model.

The introduced concept of administrative unit in our work is comparable to the abstract of administrative scope introduced in ARH [6], which *"informally associates each role in the role hierarchy to the set of roles over which it has control"*. However, there is a twofold distinction between these concepts: first, similar to ARBAC97 [23], we assume administrative roles are separate from regular roles, while in ARH administrative roles are a set of regular roles in the system augmented with administrative authorities. Second, ARH is focused on role hierarchy administration. It considers Role-Role relation in RBAC model, in contrast to our administrative model which has a dissimilar underlying operational model and designed to manage different kinds of assignments.

We propose a basic administrative model to manage RPDRA in the operational model, and then extend it to a more generic model which is able to also manage PDRA. This extension could be generalized to construct a comprehensive administrative model which is able to manage all assignments in the operational model. We define one administrative unit per operational assignment to be managed, which includes a unique administrative role and a set of administrative tasks, as follows. The set of Administrative Tasks reflects the scope of control which is potentially available to each AU's administrative roles.

RPDRA Administration. In order to manage RPDRA, each Administrative Task is defined as a set which itself contains two sets: a set of Device Roles (DR), which is a subset of available device roles defined in the system and a set of Role Pairs (RP) which is a subset of available RPs in the system.

PDRA Administration. For managing PDRA relation, each Administrative Task is defined as a set which includes two sets: a subset of Device Roles (DR) and a subset of permissions (P).

3.2 Formal Definition of Proposed Model

In this section, we present most notable features of our model via formalism. Formal definitions have been also presented in Table 2. `Core components` include the concepts of Administrator Users (AUser), Administrative Roles (AR), Administrative Unit (AU), Administrative Task (AT) and Administrative User Assignment (AUA).

Administrator Users (AUser) are a subset of regular users, with administrative authorizations. Administrator users would be recognized by their assignment to Administrative Roles (AR). Administrative User Assignment (AUA) is a relation which assigns administrator users to administrative roles. Administrative Unit (AU) is an abstraction to represent a unit of administration, which contains the scope of management of its contained AR. Each Administrative Unit (AU) includes two components, a uniquely associated AR and a subset of possible authorization assignments, namely Administrative Tasks (AT). Any AR included in an AU is permitted to manage any of the possible authorization assignments included in its corresponding AT. For instance, if a `Homeowner` assigned to be the AR of an AU and `scheduling the thermostat` is included in the AT included in the same AU, it implies that any user with `Homeowner` role would be authorized to manage `thermostat schedule`.

We define `Administrative Constraint` as a set of prohibited assignments which indicate denial of access instead of conferring it. That is negative permissions are modeled as constraints in our system. For instance, babysitter does not need and should not be granted access to the thermostat's schedule. `Administrative Authorizations` indicate the relation defined in order to assign of AT

[2]https://us-cert.cisa.gov/bsi/articles/knowledge/principles/least-privilege

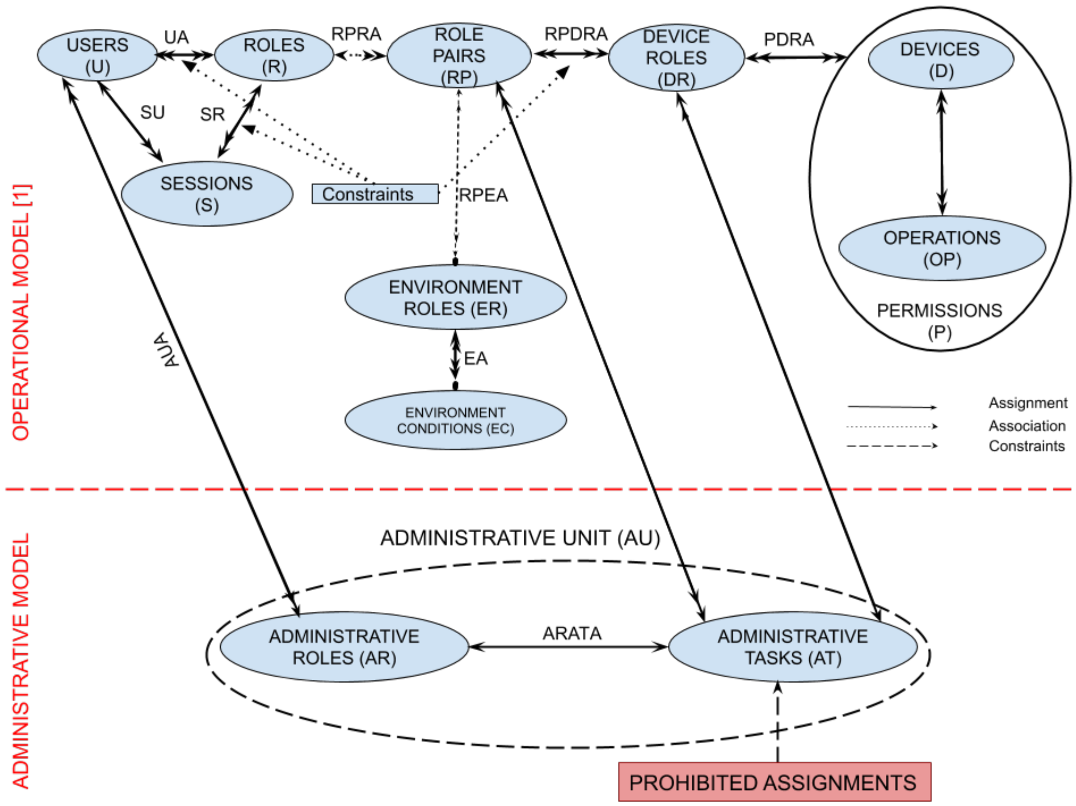

Figure 1: Administrative Model

to AR (defining the scope of control of AR) and AR to AU (indicating the Administrator Role in an Administrative Unit). ARATA is Administrative Role to Administrative Task assignment, which is a one to one relation, which means only one AR could be authorized to activate authorizations included in corresponding AT. ARAUA is Administrative Role to Administrative Unit Assignment, which is a one-to-one relation, that means no more than one AR can be assigned to an AU. So, both AT and AU are uniquely associated to an AR. It is notable that it is always possible to assign more than one user to an AR.

`Derived Administrative Relations` are a set of functions used to retrieve administrative relations between different components of the model. These functions could be later utilized to evaluate a constraint which should be sustained in all assignments/ revocations. $AR_{at \in AT}$ indicates the AR which has control over specified at. To determine role pairs and device roles which are included in an administrative task, functions $RolePair_{at \in AT}$ and $DeviceRole_{at \in AT}$ could be used correspondingly. $InclusiveTask((rp, dr))$ function finds out the administrative task within which the given pair of device role and role pair are included. Our model components have been depicted in Figure 1.

`Authorization Functions` which are represented in bottom part of Table 2, determining the conditions that qualify an administrator user to do assignments/revocation which completes the operational model's access policy. Proposed authorization functions decouple assignment and revocation of a specific (rp, dr), which means there is no requirement for the revoking user to be the same user who granted a specific access.

Function ASSIGNRPDR$(auser \in AUser, ar \in AR, rp \in RP, dr \in DR)$ enables a user *auser* with *ar* role to add the (rp, dr) to the set of RPDRA of operational model. This means the device role *dr* would be assigned to the role pair *rp*, which consequently adds a new rule to the set of policies of EGRBAC. To qualify the requesting user, the assignment function finds the including AT of given (rp, dr) set as well as the AR which is in charge for that specific task. The model checks if the requesting administrator user has the AR which controls the retrieved AT and add the (rp, dr) to the set of RPDRA provided that the rule has not been previously created.

Similarly, function REVOKERPDR$(auser \in AUser, ar \in AR, rp \in RP, dr \in DR)$ would authorize an administrator user *auser* with *ar* role to revoke a device role from a role pair by checking similar preconditions as ASSIGNRPDR, unless in case of revocation, the intended (rp, dr) should has been previously assigned by a legitimate administrator. As a result, the (rp, dr) pair would be deleted from the set of RPDRA of EGRBAC.

4 USE CASE DEFINITION

In this section we will discuss a case study of smart home in two parts of operational and administrative cases. Proposed operational

Table 2: 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.
$-AU$ is a set of administrative units.
$-AT \subseteq (2^{RP} \times 2^{DR}) \backslash ProhibitedAssignment$ is a set of administrative tasks, 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.

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 by design, ($Constratints \subset RP \times DR$).

Administrative Authorization
$-ARATA \subseteq AR \times AT$ is a one to one AR to AT assignment determining the scope of administrative control for a given AR.
$-ARAUA \subseteq AR \times AU$ is a one to one AR to AU assignment, determines which AU is under control of a given AR.

Derived Administrative Relations
$-AR_{at \in AT} \subset AT \times AR : AR_{at} = ar \in AR : at \in ARATA(ar)$: many to one administrative task 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.
$-InclusiveTask((rp, dr)) \subseteq (rp \in RP, dr \in DR) \times \{AT \cup FALSE\}$ determines the association of a (rp, dr) to an administrative task, at, if this pair is currently defined as a member of that administrative task, if no inclusive administrative task found, it returns FALSE.

Authorization Functions
$-$ASSIGNRPDR$(auser \in AUser, ar \in AR, rp \in RP, dr \in DR) \equiv$ (((auser,ar) $\in AUA) \wedge (at =$ InclusiveTask(rp,dr) $\wedge\ ar = AR_{at}) \wedge ((rp, dr) \notin RPDRA)) \Rightarrow RPDRA' = RPDRA \cup (rp, dr)$
$-$REVOKERPDR$(auser \in AUser, ar \in AR, rp \in RP, dr \in DR) \equiv$ (((auser,ar) $\in AUA) \wedge (at =$ InclusiveTask(rp,dr) $\wedge\ ar = AR_{at}) \wedge ((rp, dr) \in RPDRA)) \Rightarrow RPDRA' = RPDRA \backslash (rp, dr)$

use case is an extension to what has been presented in [1]. Then the corresponding administrative use case would be discussed.

4.1 Operational Use Case

Presented use case aims to make a representation of a smart home environment in which users' accesses are granted to parts of functionalities of given devices, a.k.a. device roles. Parents want children to have access only to the kids_friendly_content on entertainment devices (TV, DVD, and PlayStation). It should not be possible for kids to access to some functionalities of devices, which should be specifically controlled by an adult, for example turn the oven on/off, controlling the thermostat or garage door functionalities and so on.

Table 3: 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. $AT = P\text{-}AT \cup R\text{-}AT$.
$-R\text{-}AT \subseteq (2^{RP} \times 2^{DR}) \backslash ProhibitedAssignment$ 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 ProhibitedAssignments has to be excluded.
$-P\text{-}AT \subseteq (2^{P} \times 2^{DR})$ is a set of administrative tasks related to PDRA assignment, which defines permission assignments to device roles.
$-SubAU \subset AR \times \{R\text{-}AT, P\text{-}AT\}$ 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, ($ProhibitedAssignment \subset RP \times DR$).

Administrative Authorization
$-ARRATA \subseteq AR \times R\text{-}AT$, is a one to one AR to R-AT assignment determining the scope of administrative control for a given ar on RPDRA.
$-ARPATA \subseteq AR \times P\text{-}AT$, 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
$-AR_{at \in SubAT} \subset SubAT \in AT \times AR : AR_{at} = ar \in AR : at \in ARRATA(ar) \vee at \in ARPATA(ar)$: 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.
$-InclusiveTask((rpp, dr)) \subseteq (\{(rpp \in RP) \vee (rpp \in P)\}, dr \in DR) \times \{AT \cup FALSE\}$ determines the association of a (st, dr) to an administrative task (either R-AT or P-AT) if this pair is currently defined as a member of that administrative task, if no inclusive administrative task found, it returns FALSE.

Authorization Functions
$-$ASSIGNRPDR$(auser \in AUser, ar \in AR, rp \in RP, dr \in DR) \equiv$ (((auser,ar) $\in AUA) \wedge (r\text{-}at =$ InclusiveTask(rp,dr) $\wedge\ ar = AR_{r-at}) \wedge ((rp, dr) \notin RPDRA)) \Rightarrow RPDRA' = RPDRA \cup (rp, dr)$
$-$REVOKERPDR$(auser \in AUser, ar \in AR, rp \in RP, dr \in DR) \equiv$ (((auser,ar) $\in AUA) \wedge (r\text{-}at =$ InclusiveTask(rp,dr) $\wedge\ ar = AR_{r-at}) \wedge ((rp, dr) \in RPDRA)) \Rightarrow RPDRA' = RPDRA \backslash (rp, dr)$
$-$ASSIGNPDR$(auser \in AUser, ar \in AR, p \in P, dr \in DR) \equiv$ (((auser,ar) $\in AUA) \wedge (p\text{-}at =$ InclusiveTask(p,dr) $\wedge\ ar = AR_{p-at}) \wedge ((p, dr) \notin PDRA)) \Rightarrow PDRA' = PDRA \cup (p, dr)$
$-$REVOKERPDR$(auser \in AUser, ar \in AR, p \in P, dr \in DR) \equiv$ (((auser,ar) $\in AUA) \wedge (p\text{-}at =$ InclusiveTask(p,dr) $\wedge\ ar = AR_{p-at}) \wedge ((p, dr) \in PDRA)) \Rightarrow PDRA' = PDRA \backslash (p, dr)$

Table 4: Operational Use Case

U = {Alex, Bob, Susan, James, Julia}
R = {kid, parent, babySitter, guest}
UA = {(Alex,kid), (Bob,parent), (Susan,babySitter), (James,guest), (Julia,parent)}
D = {TV, DVD, PlayStation, DoorLock, Oven, SurveillanceCamera, BurglarAlarm, GarageDoor, Thermostat}
OP = {On, Off, PG, R, Lock, Unlock, Activate, Deactivate, $\mathrm{On}_{\mathrm{Oven}}$, $\mathrm{Off}_{\mathrm{Oven}}$, StartRecording, StopRecording, $\mathrm{Open}_{\mathrm{GarageDoor}}$, $\mathrm{Close}_{\mathrm{GarageDoor}}$, $\mathrm{On}_{\mathrm{Thermostat}}$, $\mathrm{Off}_{\mathrm{Thermostat}}$, $\mathrm{Schedule}_{\mathrm{Thermostat}}$}
$P_1 = \{TV, DVD, PlayStation\} \times \{On, Off, PG, R\}$
$P_2 = \{TV, DVD, PlayStation\} \times \{On, Off, PG\}$
$P_3 = \{Oven\} \times \{On\_Oven, Off\_Oven\}$
$P_4 = \{FrontDoor\} \times \{Lock, Unlock\}$
$P_5 = \{SurveillanceCamera\} \times \{StartRecording, StopRecording\}$
$P_6 = \{BurglarAlarm\} \times \{Activate, Deactivate\}$
$P_7 = \{GarageDoor\} \times \{Open_{GarageDoor}, Close\_GarageDoor\}$
$P_8 = \{Thermostat\} \times \{On\_Thermostat, Off\_Thermostat,$ Schedule_Thermostat}
$P_9 = \{Thermostat\} \times \{On\_Thermostat, Off\_Thermostat\}$
$P_{10} = \{OutdoorCamera\} \times \{On_{OutdoorCamera}, Off_{OutdoorCamera}\}$
$P = \bigcup_{i=1..10} P_i$
DR = {Entertainment_Devices, Adult_Controlled, Owner_Controlled, Kids_Friendly_Content }
PDRA = {$P_1 \times Entertainment\_Devices\} \cup \{P_2 \times Kids\_Friendly\_Content\} \cup \{\{P_3 \cup P_4 \cup P_9\} \times Adult\_Controlled\} \cup \{\{P_5 \cup P_6 \cup P_7 \cup P_8\} \times Owner\_Controlled\}$
EC = {weekends, evenings, vacation, TRUE}
ER = {Entertainment_Time, Any_Time, Not_At_Home}
EA = {({weekends, evenings},Entertainment_Time), ({vacation},Not_At_Home), ({TRUE},Any_Time)}
RP = {(kid,{Entertainment_Time}),(parent,{Any_Time}), (babySitter,{Any_Time}),(guest,{Any_Time}), (parent,{Not_At_Home})}
RPDRA = {((parent,{Any_Time}),Adult_Controlled),
((parent,{Any_Time}),Owner_Controlled),
((parent,{Any_Time}),Entertainment_Devices),
((kid,{Entertainment_Time}),Kids_Friendly_Content),
((babysitter,{Any_Time}),Adult_Controlled),
((guest,{Any_Time}),Entertainment_Devices)}

Table 5: Administrative Use Case

AUser = {Bob, Julia}
AR = {Entertainment_ Manager, Home_Owner, Adult_Manager}
AUA = {(Bob, Home_Owner), (Julia, Home_Owner), (Julia, Adult_Manager), (Bob, Entertainment_ Manager)}
AU = {Entertainment_Management, Ownership_Control, Adult_Management}
ProhibitedAssignment = {((kid,{Entertainment_Time}), Entertainment_Devices)}
AT = {at_1, at_2, at_3}
$at_1 = \{(parent, \{Any\_Time\}), (babysitter, \{Any\_Time\})\} \times \{Entertainment\_Devices, Kids\_Friendly\_Content\} \setminus$ {ProhibitedAssignment}
$at_2 = \{(parent, \{Any\_Time\}), (babysitter, \{Any\_Time\})\} \times \{Adult\_Controlled\} \setminus \{ProhibitedAssignment\}$
$at_3 = \{(parent, \{Any\_Time\})\} \times$
{Owner_Controlled} $\setminus \{ProhibitedAssignment\}$
RolePair(at_1) = $\{(parent, \{Any\_Time\}), (guest, \{Any_Time\}),$
(kid,{Entertainment_Time})}
DeviceRole(at_2) = $\{Adult\_Controlled\}$
InclusiveTask((kid,{Entertainment_Time}), Kids_Friendly_Content) = at_1
ARATA = {(Entertainment_Manager,at_1),(Adult_Manager,at_2), (Home_Owner,at_3)}
$AR_{at_1} = \{Entertainment\_Manager\}$
$AR_{at_2} = \{Adult\_Manager\}$
$AR_{at_3} = \{Home\_Owner\}$
assignRPDR(Bob, Entertainment_Manager,
({(kid,{Entertainment_Time}), Kids_Friendly_Content})) $\Longrightarrow$ RPDRA = RPDRA $\cup$ {((kid,{Entertainment_Time}), Kids_Friendly_Content)}
revokeRPDR(Bob, Entertainment_Manager,
({(kid,{Entertainment_Time}), Kids_Friendly_Content})) $\Longrightarrow$ RPDRA = RPDRA $\setminus$ {((kid,{Entertainment_Time}), Kids_Friendly_Content)} $\Longrightarrow$
$RPDRA = \emptyset$
assignPDR(Julia, Home_Owner, P_{10}, $Owner\_Controlled) \Longrightarrow$
PDRA = PDRA $\cup$ {(P_{10}, $Owner\_Controlled)\}$
revokePDR(Julia, Home_Owner, P_3, $Adult\_Controlled) \Longrightarrow$
PDRA = PDRA $\setminus$ {(P_3, $Adult\_Controlled)\}$

Furthermore, we want babysitter to access the required adult-controlled functionalities, such as turning the oven/thermostat on/off and lock/unlock the front door. However, we do not want to grant an unnecessary access to babysitter, e.g. modifying the thermostat schedule. The most permissive users would be the parents, to whom all functionalities of smart home are available.

The operational use case can be configured as illustrated in Table 4. There are five users `Alex, Bob, Susan, James, Julia` who are correspondingly assigned to roles `kid, parent, babysitter, guest and parent`. The set of devices include `TV, DVD, PlayStation, DoorLock, Oven, SurveillanceCamera, BurglarAlarm, GarageDoor, Thermostat` each of which has been associated with a set of operations defined by the manufacturer.

We defined a set of permissions including 9 different permission sets. We designed the set of permissions based on available operations for each device, as well as the desired access control regulations we previously mentioned. For example, we come up with two different permissions P_8 and P_9 for thermostat. This design aims for implementing the least privilege principle, as in next steps we can assign these permission sets to different device roles, e.g., assign P_8 to `Adult_Controlled` device role. Then, we assign babysitter to this device role, it is possible to turn the thermostat on/off, but excessive access to thermostat's schedule would not be provided. Same consideration has been taken in designing separate permission sets of P_1 and P_2 for entertainment devices, so it would be possible to define a device role, `Kids_Friendly_Content`, which would provide kids with least required permissions necessary for their access. Four Device Roles have been introduced and different permission sets have been assigned to them using `PDRA`.

A set of Environment Conditions, `EC`, has been assigned to different Environment Roles, `ER`, which would be later coupled by Roles to create Role Pairs, `RP`. Coupling device roles with role pairs through `RPDRA` completes the set of access rules in the system. As an instance, the `((parent,Any_Time),Adult_Controlled)` pair communicates that parent can access to `Adult_Controlled` device role, which includes access to turn the oven and thermostat on/off and lock/unlock the front door, at any time.

4.2 Administrative Use Case

Table 5 depicts the administrative use case based on our proposed model and corresponds to the operational use case discussed in

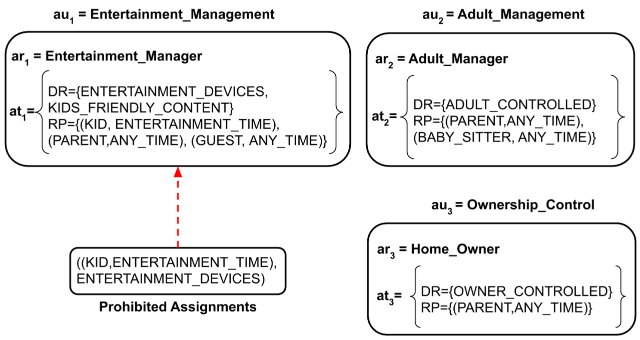

Figure 2: Administrative Units

previous section. Administrator users are a subset of regular users in the operational use case, and include Bob and Julia. As illustrated in Table 5, each of administrator users has been assigned to two different administrative roles. Bob has been assigned to Home_Owner and Entertainment_Manager and Julia has been assigned to Home_Owner and Adult_Manager. Note that the Home_Owner administrative role has both Bob and Julia as administrator, which addresses the single point of failure associated with centralized administration.

There are three Administrative Units (AU) defined in our smart home use case including Entertainment_Management, Ownership_Control and Adult_Management. There is one AR uniquely associated with each AU, so an admin unit cannot have more than one AR in charge of it, but more than one administrator user could be assigned to that AR. There is one Administrative Task (AT) in each AU, which includes a set of DR and a set of RP. AU can grant any subset of $RP \times DR$ using the ASSIGNRPDR authorization functions or revoke by using REVOKERPDR.

There are some samples of administrative relations depicted in Table 5. For instance, AR_{at_1} indicates the AR which is in charge of at_1. Authorization functions have also been shown, for example ASSIGNRPDR = (Bob, Entertainment_Manager, ({(kid,{Entertainment_Time}), Kids_Friendly_Content})), would first find the inclusive task of the given (rp,dr) which is at_1. It then checks if Entertainment Manager is the AR assigned to ({(kid,{Entertainment_Time}), Kids_Friendly_Content}), which is true in this use case. Lastly, if the requested access pair is not previously defined, it would add it to the access rules in RPDRA of the operational model.

It is noteworthy to see the ((kid,{Entertainment_Time}), Entertainment_Devices) has been defined as ProhibitedAssignment, so the authorization function (ASSIGNRPDR) would not let any administrator to grant access to Entertainment_Devices to the kids at their entertainment time. The illustration of discussed usecase has been provided in Figure 2.

5 ADMINISTRATIVE MODEL EXTENSION

In this section, we extend our model to support PDRA assignments as well. So, when a new device added to the smart home environment, its permissions could be assigned to an existing/newly created device role/s by adding PDRA assignments. Also, an administrator might decide to rearrange permissions associated to a device role, which is accomplished through making changes to PDRA. Succinctly, we extended previous administrative model by defining different administrative sub-units, each of which includes an AR and an administrative sub-task. Consequent changes to the model formalization has been proposed, that we will review in this section. Proposed extended administrative model is illustrated in Figure 3.

5.1 Formal Definition of Extended Model

The formal definition of previous administrative model has been extended as depicted in Table 3. We adopt the same administrative functional categories as our first model. The same concept of Administrative Unit (AU) exists in the extended model. Here, AU would encompass two sub-units (SubAU), one for governing PDRA and another for managing RPDRA. Similarly, the set of Administrative Tasks (AT) includes two Administrative SubTask (SubAT), naming P-AT which includes the subAT which corresponds to PDRA and another subAT named R-AT which contains ATs correspond to RPDRA.

R-At is the same as AT in previous version of our model and contains two sets, a subset of role pairs (RP) and a subset of device roles (DR). P-AT includes two sets, one is the subset of permissions (P) and another is a subset of device roles (DR). There is a unique set of Administrative Roles (AR). Each Sub-AU includes a SubAT and an AR which has been uniquely assigned to manage that SubAT. This Assignment would be a one-to-one relationship, however it is possible for one AR to be administrator for different SubATs.

Corresponding administrative authorizations have been added to the model formalism, as ARRATA is a one to one relationship which uniquely assigns an administrative role (AR) to a R-AT administrative subtask. Likewise, ARPATA assigns a unique administrative role (AR) to a P-AT administrative subtask via a one to one relation.

Authorization functions which control over manipulating RPDRA remain the same. We also added analogous authorization functions for PDRA management, which have been shown at the bottom of Table 3.

Function ASSIGNPDR($auser \in AUser, ar \in AR, p \in P, dr \in DR$) enables a user *auser* with *ar* role to add the (p, dr) to the set of PDRA of the operational model. As an illustration, suppose a smart outdoor camera has been added to smart home, with the permissions set to be $\{OutdoorCamera\} \times \{On, Off\}$. So, an authorized administrator user, any of homeowners, should be able to assign this new permission to previously/newly defined device roles. In this example, the new permission could be assigned to *Owner_Controlled* device role in the system. Required changes to represent this usecase have been color-coded in Tables 4 and 5.

Equivalently, required changes in Function REVOKEPDR($auser \in AUser, ar \in AR, p \in P, dr \in DR$) enables the user *auser* with *ar* role to remove the (p, dr) from the set of PDRA of operational model. So, any role pair which has been coupled with device role *dr* would consequently lose the permission *p*. For instance, as depicted in Table 4, Julia as the an administrative user with the administrative role Home_Owner can remove the permission $P_3 = \{Oven_{On}, Oven_{Off}\}$ from the set of permissions of *Adult_Controlled* device role. So, any user with that role, e.g., babysitter, would no longer has the permission to turn the oven on/off. She might

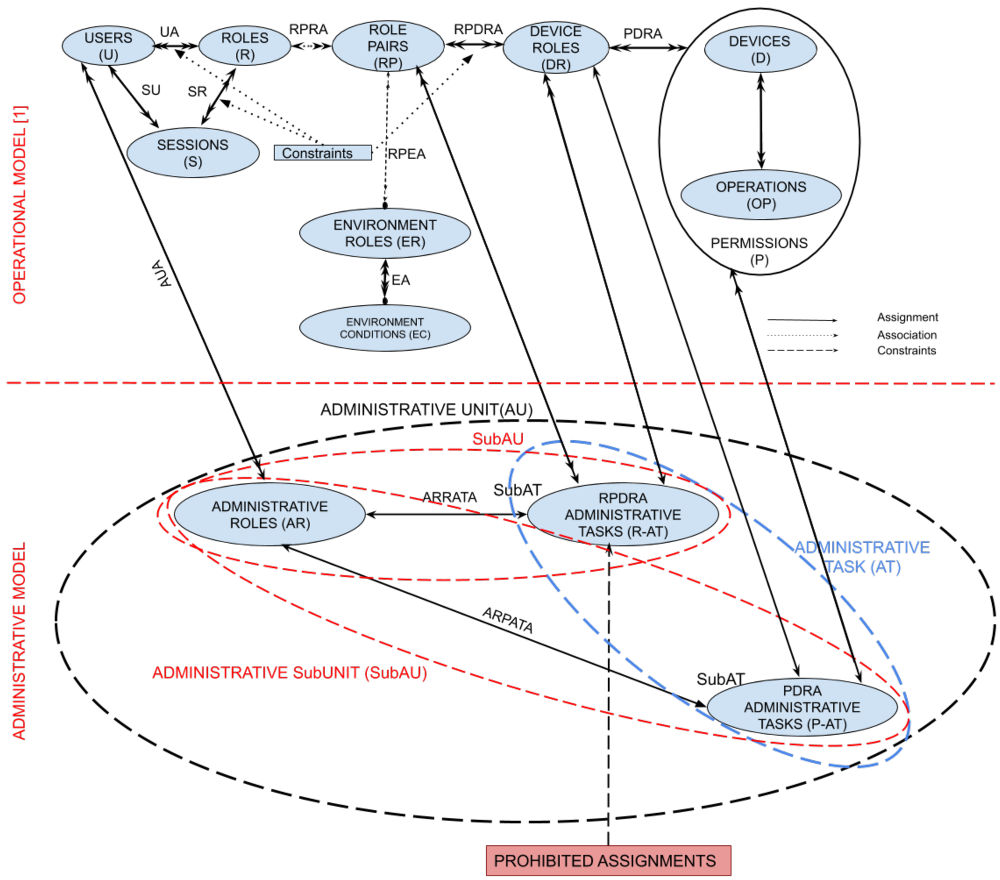

Figure 3: Extended Administrative Model for Managing both RPDRA and PDRA

want to add that permission later to another device role, e.g. *Owner_-Controlled*. This example has been added in red to the bottom of Table 5.

6 DISCUSSIONS

Proposed models in this paper use RBAC to design a decentralized administrative model for managing an operational RBAC model in a smart home environment. We focused on administration of assignments which are more dynamic due to the inherent characteristics of a smart home environment. We assume the set of regular user roles, administrative roles and device roles have been centrally managed in some way. Our administrative models have been built upon EGRBAC as an underlying operational model.

6.1 Model Properties

Decoupled Assignment and Revocation. Proposed authorization functions in our models decouple assignment and revocation. Therefore, any administrator user can conduct authorized grant/revoke assignments, provided that the function's preconditions are satisfied. This means there is no need that granting and revocation of a permission to be done by the same administrator.

Symmetric Assignment and Revocation. Even though grant and revoke are decoupled as stated above, our authorization functions enable an administrator user to revoke a permission, which has been conferred previously by him/her, from a subject. Similarly the same administrator who revoked a permission is able to re-grant it in the future, as long as the administrator user holds the same administrative role.

Generalizability. Although our model manages two of assignments in underlying operational model, it could be easily generalized to govern other assignments by defining extra administrative units, each of which would cover a new scope of administration defined as an administrative sub-task.

6.2 Model Restrictions

Continuous Usage Control. Considering usage as practicing granted access rights by subjects on objects, it is required for dynamic environments to have continuous control over it. In many cases, enforcement of the access control necessitates an immediate change in permissions, e.g., when administrator revokes the access from a user who is currently utilizing it. There are administrative models proposed to enforce administrative decisions in a session-aware manner to satisfy this requirement [31].

Quota-based Access Enforcement. Some access control requirements in a smart home environment may call for access quota. Access quota is a consumable amount of resource usage which is non-refundable. For instance, we may want to limit kids access to entertainment devices to one hour per day. Such requirements are irrefutable evidences that operational and administrative access control models should be able to handle quotas. Authors in [27] proposed a quota-based approach to address the consistency problem in ABAC environments.

Conflict of Interest. Although there is a single administrative role assigned for each administrative unit/sub-unit, dissension among different administrator users in that role is possible. So, the permission granted by one administrator user may be revoked shortly after by another administrator user. Therefore, it is required to incorporate a conflict resolution policy in the model.

7 CONCLUSION

In this paper, we propose 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. Each administrative task corresponds to one of the assignments in the operational model. The model has been extended to enclose another assignment relation of the model by introducing administrative sub-units and administrative sub-tasks. Following the same approach, it is possible to extend the model in order to manage other model assignments by adding corresponding sub-task and sub-units of administration. We outlined the formal specification of proposed model and consolidate the ideas presented in the paper by proposing smart home case studies.

ACKNOWLEDGEMENT

This work is partially supported by NSF CREST Grant 1736209.

REFERENCES

[1] Safwa Ameer, James Benson, and Ravi Sandhu. 2020. The EGRBAC Model for Smart Home IoT. In *Information Reuse and Integration for Data Science (IRI)*. IEEE.

[2] Mahmoud Ammar, Giovanni Russello, and Bruno Crispo. 2018. Internet of Things: A survey on the security of IoT frameworks. *Journal of Information Security and Applications* (2018).

[3] Orlando Arias, Jacob Wurm, Khoa Hoang, and Yier Jin. 2015. Privacy and security in internet of things and wearable devices. *Transactions on Multi-Scale Computing Systems* (2015).

[4] Bruhadeshwar Bezawada, Kyle Haefner, and Indrakshi Ray. 2018. Securing home IoT environments with attribute-based access control. In *Workshop on Attribute-Based Access Control*. ACM.

[5] Rafae Bhatti, Basit Shafiq, Elisa Bertino, Arif Ghafoor, and James BD Joshi. 2005. X-gtrbac admin: A decentralized administration model for enterprise-wide access control. *Transactions on Information and System Security (TISSEC)* (2005).

[6] Jason Crampton and George Loizou. 2003. Administrative scope: A foundation for role-based administrative models. *Transactions on Information and System Security (TISSEC)* (2003).

[7] MAC Dekker, Jason Crampton, and Sandro Etalle. 2008. RBAC administration in distributed systems. In *Symposium on Access control models and technologies*. ACM.

[8] Tamara Denning, Tadayoshi Kohno, and Henry M Levy. 2013. Computer security and the modern home. *Communications Journal* (2013).

[9] Sofia Dutta, Sai Sree Laya Chukkapalli, Madhura Sulgekar, Swathi Krithivasan, Prajit Kumar Das, Anupam Joshi, et al. 2020. Context Sensitive Access Control in Smart Home Environments. In *Big Data Security on Cloud (BigDataSecurity 2020)*. IEEE.

[10] David Ferraiolo, Ramaswamy Chandramouli, Rick Kuhn, and Vincent Hu. 2016. Extensible access control markup language (XACML) and next generation access control (NGAC). In *Workshop on Attribute Based Access Control*. ACM.

[11] David F Ferraiolo, Ravi Sandhu, Serban Gavrila, D Richard Kuhn, and Ramaswamy Chandramouli. 2001. Proposed NIST standard for role-based access control. *Transactions on Information and System Security (TISSEC)* (2001).

[12] Vikas Hassija, Vinay Chamola, Vikas Saxena, Divyansh Jain, Pranav Goyal, and Biplab Sikdar. 2019. A survey on IoT security: application areas, security threats, and solution architectures. *Access Journal* (2019).

[13] Abhiditya Jha, Jess Kropczynski, Heather Richter Lipford, and Pamela J Wisniewski. 2019. An Exploration on Sharing Smart Home Devices Beyond the Home. In *IUI Workshops*.

[14] Sadhana Jha, Shamik Sural, Jaideep Vaidya, and Vijayalakshmi Atluri. 2014. Security analysis of temporal RBAC under an administrative model. *Computers & security* (2014).

[15] Francesca Meneghello, Matteo Calore, Daniel Zucchetto, Michele Polese, and Andrea Zanella. 2019. IoT: Internet of Threats? A survey of practical security vulnerabilities in real IoT devices. *Internet of Things Journal* (2019).

[16] Jonathan D Moffett and Morris S Sloman. 1991. Delegation of authority. *Integrated Network Management II* (1991).

[17] Sejong Oh and Ravi Sandhu. 2002. A model for role administration using organization structure. In *Seventh ACM symposium on Access control models and technologies*.

[18] Aafaf Ouaddah, Hajar Mousannif, Anas Abou Elkalam, and Abdellah Ait Ouahman. 2017. Access control in the Internet of Things: Big challenges and new opportunities. *Computer Networks* (2017).

[19] Rajkumar PV and Ravi Sandhu. 2016. POSTER: security enhanced administrative role based access control models. In *Conference on Computer and Communications Security*. ACM.

[20] Jing Qiu, Zhihong Tian, Chunlai Du, Qi Zuo, Shen Su, and Binxing Fang. 2020. A survey on access control in the age of internet of things. *Internet of Things Journal* (2020).

[21] Sowmya Ravidas, Alexios Lekidis, Federica Paci, and Nicola Zannone. 2019. Access control in Internet-of-Things: A survey. *Journal of Network and Computer Applications* (2019).

[22] Indrakshi Ray, Ramadan Abdunabi, and Rejina Basnet. 2019. Access Control for Internet of Things Applications. In *Workshop on Cyber-Physical System Security*. ACM.

[23] Ravi Sandhu, Venkata Bhamidipati, and Qamar Munawer. 1999. The ARBAC97 model for role-based administration of roles. *Transactions on Information and System Security (TISSEC)* (1999).

[24] Ravi Sandhu and Qamar Munawer. 1999. The ARBAC99 model for administration of roles. In *Annual Computer Security Applications Conference (ACSAC'99)*. IEEE.

[25] Ravi S Sandhu, Edward J Coyne, Hal L Feinstein, and Charles E Youman. 1996. Role-based access control models. *Computer* (1996).

[26] Amit Sasturkar, Ping Yang, Scott D Stoller, and CR Ramakrishnan. 2006. Policy analysis for administrative role based access control. In *Computer Security Foundations Workshop (CSFW'06)*. IEEE.

[27] Mehrnoosh Shakarami and Ravi Sandhu. 2019. Safety and Consistency of Mutable Attributes Using Quotas: A Formal Analysis. In *Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*. IEEE.

[28] Manisha Sharma, Shamik Sural, Jaideep Vaidya, and Vijayalakshmi Atluri. 2013. AMTRAC: An administrative model for temporal role-based access control. *Computers & security* (2013).

[29] Madiha Tabassum, Jess Kropczynski, Pamela Wisniewski, and Heather Richter Lipford. 2020. Smart Home Beyond the Home: A Case for Community-Based Access Control. In *CHI Conference on Human Factors in Computing Systems*. ACM.

[30] William Tolone, Gail-Joon Ahn, Tanusree Pai, and Seng-Phil Hong. 2005. Access control in collaborative systems. *ACM Computing Surveys (CSUR)* (2005).

[31] Min Xu, Duminda Wijesekera, Xinwen Zhang, and Deshan Cooray. 2009. Towards session-aware RBAC administration and enforcement with XACML. In *International Symposium on Policies for Distributed Systems and Networks*. IEEE.

[32] Yuchen Yang, Longfei Wu, Guisheng Yin, Lijie Li, and Hongbin Zhao. 2017. A survey on security and privacy issues in Internet-of-Things. *Internet of Things Journal* (2017).

[33] Eric Zeng and Franziska Roesner. 2019. Understanding and improving security and privacy in multi-user smart homes: a design exploration and in-home user study. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*.

[34] Zhixiong Zhang, Xinwen Zhang, and Ravi Sandhu. 2009. Towards a Scalable Role and Organization Based Access Control Model with Decentralized Security Administration. In *Handbook of Research on Social and Organizational Liabilities in Information Security*. IGI Global.