IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 13, 2021, accepted October 31, 2021, date of publication November 8, 2021, date of current version November 12, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3126201

Activity Control Design Principles: Next Generation Access Control for Smart and Collaborative Systems

JAEHONG PARK[1], (Member, IEEE), RAVI SANDHU[2], (Life Fellow, IEEE), MAANAK GUPTA[3], (Member, IEEE), AND SMRITI BHATT[4]

[1]Department of Management, Marketing, and Information Systems, University of Alabama in Huntsville, Huntsville, AL 35899, USA
[2]Institute for Cyber Security, The University of Texas at San Antonio, San Antonio, TX 78249, USA
[3]Department of Computer Science, Tennessee Technological University, Cookeville, TN 38505, USA
[4]Department of Computer and Information Technology, Purdue University, West Lafayette, IN 47907, USA

Corresponding author: Jaehong Park (jae.park@uah.edu)

ABSTRACT Traditionally, access control solutions have focused on how to utilize a specific type of decision parameter for access control decisions. While these ''decision parameter''-focused approaches have been well accepted, they typically consider access control with centralized administration. Smart and collaborative computing systems (SCSs) such as online social networks, the Internet of Things (IoT) and connected cyber-physical systems (CPSs) require a disparate approach to meet their unique and complex access control requirements primarily because there are multiple participants who create, share, manage and protect resources (e.g., files, smart devices) individually, collaboratively or even competitively. A distinct feature of SCSs is the diffuse nature of control activities and their complex influence on other activities. Activity control (ACON) extends the scope of traditional access control models and considers how multiple administrative authorities (including users) can manage complex and interacting usage, service and control activities. In this paper, we articulate key characteristics and limitations of various existing access control models and highlight the significance and necessity of activity control in smart collaborative ecosystems. We then propose an extended ACON framework for catering to the needs of dynamic SCSs. Furthermore, we compare existing access control design principles and propose a set of activity control design principles for smart and collaborative computing systems. The proposed ACON framework and design principles will provide a solid foundation for secure SCS design and development.

INDEX TERMS Access control, activity control, smart system, connected system, collaborative system, cyber physical systems, online social networking, Internet of Things, security, privacy.

I. INTRODUCTION

With the advancement of smart and collaborative computing technologies and their applicable security solutions, various applications for smart and collaborative information sharing and management have emerged and gained increasing popularity. Unlike other computing domains, smart and collaborative computing systems (SCSs) facilitate complex interactions among users/devices/organizations to share resources and administer activities among different participating entities, such as smart objects, users, clouds or edge computers. In such SCSs, multiple participants collaboratively create, share, manage and protect digital contents and other resources [1], [2]. These complex interacting activities demand a sophisticated access control system. Specifically, access decisions are made collectively based on policies that are administered by multiple administrative authorities such as device owners/administrators, users whose information is included in contents, cloud service providers, security administrators and 3rd-party applications. This smart and connected ecosystem offers endless opportunities in terms of safety, health care, accessibility, and economic growth. It changes the overall quality of life due to the power of data it produces and unimaginable applications it enables. When converged with physical world devices such as machines, sensors, buildings etc., such technologies create

The associate editor coordinating the review of this manuscript and approving it for publication was Derek Abbott.

an intelligent and automated system that is more efficient, effective and real time. With billions of connected devices, the cloud provides foundation services to support Internet of Things(IoT) devices, which are usually resource-constrained, along with IoT applications for sustainable success and wide adoption of the IoT in the real world.

However, traditional access control models are not designed for this type of complex and dynamic SCS environment. Several relationship-based access control (ReBAC) models [3]–[5] that utilize participants' relationship information for policy specification and access control decisions have been introduced. While utilizing user (or even resource) relationships for access control is a necessary and natural approach for the online social networking (OSN) system, this works best for a system in which social relationship information is available. Other SCSs, such as social recommendation/reputation systems, collaborative IoT systems and connected vehicle systems that do not maintain explicitly defined and persistent user/resource relationship information, need a different access control solution other than the ReBAC. Moreover, typical ReBAC systems do not facilitate complex and dynamic controls on various usage and control activities performed by multiple participants, such as users and 3rd party applications, which in turn may influence the acting participant's or other participants' activities. The notion of activity control (ACON) was first coined to capture access controls on these multiparty activities found in online social networking system environments [6], [7],[1] which we will extend to accommodate other smart and connected systems.

As new computing systems and applications emerge, security researchers have proposed various access control solutions to address the security requirements of these emerging systems. While recent academic research practice commonly focuses on a specific access control problem/issue found in a certain target system and proposes a partial solution with a proof-of-concept implementation, there has also been some notable research that identified key high-level design principles for security and access control systems. For instance, in 1975, Saltzer and Schroeder identified eight design principles for security system enforcement and implementation [9]. RBAC96 principles [10] and ASCAA principles [11] were proposed for role-based access control policy and model design. Later, Smith revisited and examined Saltzer and Schroeder's design principles for modern computing system security [12]. These security principles are crucial for designing and developing access control solutions for various types of computing systems and applications. While many of these existing principles are still valid, they are not sufficient to adequately capture the core aspects of activity control, as they present unique and complex controls that are not found in many existing access control models.

Instead of developing yet another access control model that utilizes one or more types of decision parameters, the main goal of this paper is to define the scope of activity control and develop its foundational knowledge base for smart and collaborative systems. Specifically, we discuss several noteworthy characteristics of existing access control solutions and their approaches that are significant and helpful for understanding activity control and its design principles. We propose an enhanced ACON framework and discuss its scope and key components. We then examine and compare the existing design principles for access control systems and discuss the limitations of these principles. Founded on this discussion, a set of new design principles for activity control is proposed. The proposed ACON framework and design principles provide a guideline for activity control system design and development and help researchers and practitioners understand modern access control systems for smart and collaborative computing systems.[2] While the proposed ACON framework and design principles are developed mainly to support some of the unique characteristics found in smart and collaborative systems, we believe they are applicable to other computing systems that involve multiple participants who may want/need to control other participants' access.

Here, we present the key contributions of this paper:

- We present discrete motivations about the need for activity control which are fundamentally different than conventional access controls.
- We propose an extension of the activity control framework to accommodate smart collaborative systems.
- We propose next-generation access control design principles for future smart communities.
- We present use cases to highlight the need and support our conceptual framework and ACON principles.

This article is organized as follows. In section II, we discuss several related works and background. In section III, we analyze the characteristics and shortcomings of existing access control models in terms of several key criteria and compare them to activity control. In section IV, we present and discuss an extended ACON framework for smart and collaborative systems. We then discuss the evolution of design principles for security and access control solutions in section V. In section VI, we propose a set of design principles for activity control relevant for SCSs and compare them to the previous design principles. In section VII, we demonstrate how the proposed design principle can be applied to a smart healthcare use-case scenario and then conclude this paper in section VIII.

[1]The ACON framework proposed in [6], [7] utilizes attributes for the policy specification. In the framework, the term attribute is loosely defined to refer to any meaningful information of the corresponding users, objects, systems, etc. Examples include role, relationship and provenance. The attribute in ACON is different from the more tightly defined attribute found in attribute-based access control (ABAC) models [8] where attribute is a property expressed as a name:value pair.

[2]The term smart and collaborative computing system (SCS) is used in this paper to refer to a computing system that involves multiple participants such as users, IoT devices, and 3rd party applications who collaborate to create/share/manage desirable services and resources. It is not limited to just smart connected systems.

II. RELATED WORK

Access control for a connected and shared ecosystem presents unique challenges. Using adequate access control models and mechanisms, one can develop a connected smart system that allows control over participants' activity on various resources while still supporting other participants' privacy or preference. For example, it may allow control over which smart devices can create/share data with whom, and what applications can gather data from on-field devices. Here, controlling which user applications or devices can access other connected devices is a complex issue. Additionally, securing data in the cloud, local edge gateway or transit is an important concern that needs to be addressed. These issues can be even more complicated when data and resources are distributed and spread across different entities administered by different units.

Several approaches have been used in enforcing access control policies, including cryptographic mechanisms, capabilities, access control lists, and policy based solutions [13]–[20]. ABAC [8], [21], [22] supports fine-grained authorization capabilities for resources offering flexibility in a distributed multientity environment where the attributes of entities along with contextual information are used for access and communication authorization decisions. Several access control models [13], [23]–[26] and mechanisms [27], [28] have been proposed to address authorization needs in both edge and cloud assisted IoT and cyber-physical system (CPS) architectures. Ouaddah et al [29] presented a comprehensive review of IoT access control models, whereas survey-based studies [16] in smart home IoT have also highlighted the need for a novel perspective of access control based on the relationship between the device owner and the subject. Work by Fernández et al [15] proposed a novel data collection and sharing model for cloud-IoT architectures providing a plug-in module to support IoT application development. The convergent access control framework recently proposed by Bhatt and Sandhu [30] highlights the need for synergistic convergence of access control models at both policy and enforcement layers, which can address the evolving access control requirements of dynamic applications for future smart communities. In [31], the authors utilized both RBAC and ABAC in the proposed electronic health record (EHR) system for both coarse- and fine-grained access controls.

Using a capability-based access control (CAC) model for the IoT has been proposed because entities hold granted rights that support different levels of granularity with the possibility of delegation, while similar functionality is not feasible with access control lists(ACLs). However, two major drawbacks of using the capability approach are propagation and revocation [10]. The identity authentication and capability-based access control (IACAC) model [32] is proposed, where devices use an access point and the CAC model to be connected to each other. Moreover, the capability-based access control system (CapBAC) is used to control access to services and information. The authors describe use cases and argue that CapBAC supports rights delegation, the least privileges access principle, more fine-grained access control, fewer security issues, and fewer issues related to the complexity and dynamics of the subject's identities than ACLs, RBAC and ABAC. Bhatt et al developed a formal access control model for the Amazon Web Service (AWS) IoT platform and proposed ABAC enhancements for the AWS IoT, a real world cloud-enabled IoT platform [33]. The authors also recently proposed a convergent access control (CAC) framework that can converge access control features of different access control models (e.g., RBAC, ABAC, ReBAC, etc.) for enabling secure smart communities in the future [30]. Additionally, a simple, efficient, mutual authentication and secure key establishment based on elliptic curve cryptography (ECC), which has much lower storage and communication overheads, is proposed for the perception (object) layer of the IoT [29]. Gupta and Sandhu proposed a novel perspective by introducing activity-centric access control [34] for smart collaborative systems, assuming activity aka task as the prime notion to control new activities in the connected CPS systems.

The integration of the cloud and IoT has been extensively adopted in the industry by major cloud computing services providers - AWS,[3] Google Cloud IoT,[4] and Microsoft Azure[5] to enable IoT services and applications empowered by smart devices. These cloud service providers have dedicated IoT and CPS platforms catering to diverse applications and use cases supporting both cloud and real-time edge-based user applications and services [35], [36]. Several terminologies are used to refer to this integration, such as cloud-supported IoT, cloud-assisted IoT, and cloud-enabled IoT [33], [37], [38], more widely used as cloud-enabled IoT (CE-IoT). Such connected systems have the ability to support services in different domains, including agriculture [39], [40], transportation [38], [41], health care, energy, and manufacturing, offering data-driven and intelligent environments. In this paper, we utilize a use-case scenario that is based on cloud-enabled IoT.

While many of the related research discussed in this section focuses on developing an access control model solution for a specific application using specific decision parameter type(s), our main focus is on a novel activity control framework and its design principles for smart and collaborative systems that are not specific to a particular decision parameter type.

III. ACCESS CONTROL MODELS VS. ACTIVITY CONTROL

In this section, we discuss some of the existing access control models and their characteristics. Rather than reviewing these access control models individually, we compare them based on several criteria that highlight their unique characteristics and solution approaches as illustrated in Table 1. In Table 1, some of the noteworthy traditional and modern access control models are shown in columns and compared in terms of four

[3]https://aws.amazon.com/iot/
[4]https://cloud.google.com/solutions/iot
[5]https://azure.microsoft.com/en-us/overview/iot/

TABLE 1. Access control models and characteristics.

Characteristics	MAC	DAC	RBAC	ABAC	ReBAC	PBAC	UCON	ACON
Decision Parameters	Cle/cla	am	Role	Attr	Rel	Prov	Attr/oblig/cond	abs
Control Target E.g.	uc/sr	uc/sr	uc	uc	uc/up	uc	uc	u/uc/sr
Control Authorities	SO	ownr	SO	SO	ownr/usr	SO	SO/ownr/usr	SO/ownr/usr/app
Access vs Usage	Access	Access	Access	Access	Access	Access	Usage	Activity

cle: clearance, cla: classification, am: access matrix, attr: attribute, rel: relationship, prov: provenance, oblig: obligation, cond: condition/system-attribute, abs: decision parameter abstraction, uc: user contents, sr: system resource, up: user profile, SO: security officer, app: 3rd-party application

key characteristics listed in rows. Cells in the table show the relevant details of each access control model for each characteristic. For example, clearance and classification are the decision parameters used in MAC. Please note that the intention of this section is not to make airtight distinctions among access control models but rather to discuss some of the significant aspects of existing access control models that are crucial for understanding activity control and its design principles for smart and collaborative computing systems.

A. BASED ON DECISION PARAMETERS

Historically, many existing access control models have focused on how to utilize a specific decision parameter for desirable access controls. This is hinted at by the fact that many of the models are named using the decision parameter they utilize.[6]

Some of the notable decision parameters are clearance/classification, role, attribute, obligation, social and nonsocial relationships, data provenance and workflow. An access control model that utilizes one or more of these decision parameters may need to incorporate unique decision processes and address issues stemming from the unique nature or characteristics of the decision parameters it uses. For example, RBAC may need to use and manage role hierarchy and role assignments for access decisions. For typical online social networking systems, ReBAC is used, and user relationship graphs are maintained and utilized for access control decisions. Provenance-based access control (PBAC) [42]–[45] can be considered for an application that needs to utilize activity history or lineage of resources for access control decisions. In a PBAC system, the system needs to identify events and activities that need to be recorded. It also needs to maintain up-to-date provenance information in the form of a directed acyclic graph, and utilize meaningful patterns of the provenance graph for access decisions. Note that these decision parameters may represent actors, targets or the system.

As a computing system becomes more complex and requires finer and more sophisticated controls, so do the decision parameters associated with these controls. Unlike RBAC and other post-RBAC access control models such as ABAC, ReBAC and PBAC, activity control does not specify what kind of decision parameters have to be used but rather focuses on different types of activities interplayed among various participants and how to allow multiparty controls on these activities by collectively utilizing decision parameters that represent or are related to the participants and/or the system. Activity control requires that all of the decision parameters necessary for the system are readily available in the system. Ultimately, the decision parameters chosen for a target system must be adequately utilized and effectively managed for simple and easy-to-use/manage access controls.

B. BASED ON CONTROL TARGETS

Access control solutions focus on how to control user (or subject) access to a target. In typical access control models, this target is often *user-generated content* (e.g., text files, photos, program codes). In other cases, it can be a *system-provided resource* (e.g., CPU time, memory space). In SCS, in addition to controlling user access to these two types of targets, there are other control targets that need to be considered such as the *device, device-generated data, account profile, user-generated access control policies*, and the *user account* itself. For example, in social networks one may want to limit other participants' access to her profile (e.g., user name, gender, birth date, address, viewing history, friends list). A user may not want (her account) to be searched by or exposed to others. One may even want to update other users' activity policies (e.g., parental controls). In IoT systems, a user can control how connected devices may respond. For example, a homeowner can change her smart thermostat so it can be accessed by only her smart watch. Unlike many existing access control models, activity control deals with access to these unconventional targets.

C. BASED ON CONTROL AUTHORITIES

Generally, in an access control system, ''administrative activity'' is clearly distinguishable from users' (or subjects') typical ''usage activity'' as it is performed by dedicated security officers or administrators[7] In MAC, a security officer assigns

[6]Unlike RBAC and many other post-RBAC models, the names of mandatory access control (MAC) and discretionary access control (DAC) do not explicitly suggest what kind of decision parameters are used. Instead, the words mandatory and discretionary in the model names suggest that the focus is mainly on who administers the access control policies. Nevertheless, they also utilize certain types of decision parameters for access control policy specifications. Specifically, MAC primarily uses security clearance and classification labels that are assigned to subjects and objects respectively. In DAC, access rights such as ''read'' and ''write'' are assigned to sets of a subject ID and object ID pair.

[7]In this paper, we use the term ''usage activity'' to refer to users' access actions to objects that are available as part of the system's main service or application. Examples of usage activity include ''read a friend's profile'', ''deposit to/withdraw from a bank account'', ''read current room temperature'', etc. Additionally, we use the terms ''control activity'' and ''administrative activity'' interchangeably.

security clearances (or roles in the case of RBAC) to subjects/users to control what subjects/users can do in the system, which is clearly different from subjects' (or users') typical read or write actions on objects. In contrast, in other cases, distinguishing administrative activity from user usage activity is less intuitive, as it is often the users who perform both usage and administrative activities. Consequently, in such a case, both administrative activity and usage activity may need to be handled together in a single access control model. This is evidenced in DAC where administrative power is given to the owner of resource objects so that the owner is allowed to control other users' access to her own resource objects.

More complex cases are found in a collaborative system such as an online social networking system and a connected IoT system. In an OSN system, a user may post her own content or access content generated by her friends (or friends of friends). She may search for other users' profile information or manage her friend-relationship status with other users. She may also configure who can access photos she posted or was tagged in or who can search her profile information. She may also configure whether the OSN system can share her profile information with 3rd-party applications.

In a typical ReBAC-based OSN, access decisions are made using friend-relationship graphs as well as user-specific access control policies that specify what kind of relationships should exist to allow users access to certain content. Here, unlike DAC, a user may manage other users' friend-making policies as well as policies that govern access to content owned by other users. In other words, ReBAC-based OSNs need to address the administration of self-owned/other user-owned content as well as the administration of policies that govern other users' non-administrative and administrative activities. For example, parents may forbid their minor children from being a friend of an adult user unless the adult user is already a friend of the parents. Additionally, a user may control who can access a photo posted by another user if the photo includes the user's facial image in it. In a connected IoT system with a proximity-based advertisement feature, a user may control his smart watch to only obtain notifications for restaurant discount coupons during lunch hours.

To recap, users' access to targets can be controlled by *security officers, owners of the targets* or *3rd parties who are somehow related to the targets either directly or indirectly*. In addition, users' administration activities on policies can be controlled by *security officers, users to which the policies are applied* or *other users related to the users*. Most of the existing access control solutions do not consider these various control authorities. Activity control covers them in its scope.

D. BASED ON ACCESS VS. USAGE

In conventional access control models, access decisions are made before access is allowed/disallowed. Usage control expands this and allows continuous decisions throughout usage once access is made [46]. It also considers any changes to the attributes (decision parameters in usage control (UCON)) that are used for the access decision. It also facilitates usage decisions based on obligations and system conditions. These principles of decision continuity and attribute mutability as well as obligation- and condition-based decisions are key unique properties that differentiate usage control from conventional access control models (including recent ABAC models). Note that both access control and usage control focus on controlling users' usage activity while ignoring multiparty usage and administrative activities that can influence each other's activities. The scope of activity control includes controlling these activities as well. While activity includes usage as well as access, decision continuity and attribute mutability are not explicitly considered in this paper, as they are outside of the scope.

In usage control, the notion of session is not discussed explicitly. A session (aka subject in some literature) is activated by a user and performs actions on behalf of the user. While this is not part of the main focus of usage control, it is a crucial concept in activity control because a session may have different decision parameters from the user; hence, its activity request may obtain a different decision.

E. BASED ON POLICIES

Access control systems control access requests based on access control policies. Different access control models construct different policy formats. While traditional access controls such as MAC, DAC and RBAC specify access control policies in their own specific ways, such as in the form of a lattice, access matrix or role hierarchy/role assignment, many modern access control models such as ABAC specify their policies using some sort of policy language. Typically, these policies are managed centrally by security officers and must be flexible to address the needs of dynamic and collaborative smart systems.

Unlike these access control models, in activity control, there are other types of access control policies that are tied to actors and targets. These individualized policies may be self-managed or managed by other actors. In addition, these policies can be used to control either incoming actions against targets or outgoing actions of actors who requested the action [4].

IV. EXTENDING ACTIVITY CONTROL FOR SMART AND COLLABORATIVE SYSTEMS

In this section, we present an extended conceptual ACON framework as depicted in Figure 1. While the original ACON framework [6], [7] was developed mainly for OSN systems, the ACON framework presented in this paper is enhanced to capture other smart and collaborative computing systems such as the IoT and CPS. As pointed out in section III and further articulated below, there are several key access control requirements and characteristics of SCS that need to be supported in the ACON framework:

- Support various types of actors such as users, devices, 3rd party applications, administrators and the system itself.

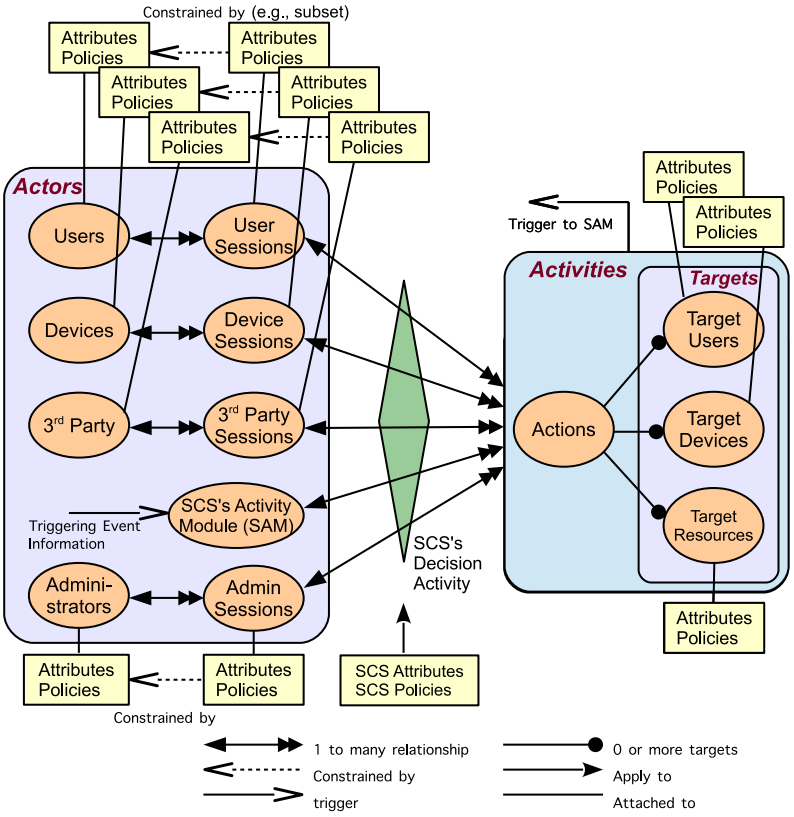

FIGURE 1. Conceptual Activity Control (ACON) framework for smart collaborative systems.

- Support various types of targets, such as users, resources and devices.
- Support various types of activities such as usage, control, service and decision activities.
- Support various participants' control over their own activity, other participants' activity, and activities performed on resources that they own or have certain stakes in.
- Support collective decisions even with conflicting control policies of multiple participants.

In Figure 1, attributes are attached to actors or targets and hold certain properties of the actors or targets. These attributes are used to specify policies so that the system can make a decision based on the attributes as well as the policies. The policies attached to an actor are used to control the corresponding actor's activities, while policies attached to a target are used to control any activity against the attached target. When an actor attempts to perform an action on a certain target, the system evaluates the request based on all the relevant attributes and policies and makes an access decision on the request. Below, we discuss the extended ACON framework and their unique characteristics in detail.

A. ACTORS AND TARGETS IN ACON

In a smart and collaborative system (SCS), multiple parties create, share, view, control and manage digital contents and services. In the ACON system for SCS, we identify five main actors whose activities may need to be controlled. They are *Users, Devices, 3rd-Party Applications, SCS*, and *Security Administrators*. As shown in Figure 1, ACON also considers sessions of users, devices, 3rd-party applications and administrators. The user and admin sessions are the subjects who logged into the SCS and conducted actions on behalf of the associated users or administrators. The device and 3rd-party application sessions are active processes that run in the system. We also identify target users, target devices and target resources. [8] In SCS, a user can be a target of certain actions. For example, if Alice sends a friend request to Bob or pokes Chris, Bob and Chris are target users for the friend-request action and poke action, respectively. Examples of target devices include smart healthcare devices, connected cars with on-board sensors and other IoT devices. The target resource could be content (that user and device created and shared in the system) and services available in the system.

B. ACTIVITIES IN ACON

An *Activity* is a meaningful abstraction of actions performed on targets. An Action is a meaningful abstraction of operations defined in SCS. For example, pokes and deer threat warnings are some instances of actions in social computing

[8] In traditional access control, the target is often referred to as "object".

and vehicular IoT, respectively. In this case, poking a user, or sending deer threat notifications to a car in proximity are activities in the system. Actors in ACON conduct various types of activities. We identify four main activities of *usage, control, service* and *decision*. Usage activity refers to actions that access or use targets. Control activity refers to actions that control future usage/control/service actions. This is mainly done by changing the attributes and policies of the targets. Service activity refers to actions that provide service or functionality to the system and its participants. Decision activity refers to the system's decisions on usage/control/service activity requests. Table 2 shows who exercises these activities on targets. Columns in the table reflect different actors in the smart and collaborative systems such as *users*, *device*, *3rd-party*, *SCS* and *admin*. Rows define different activities that can be performed by these actors, and the cells in the table reflect the targets of these activities that are performed by the actors in the system. Below we discuss these activities in detail.

1) USER ACTIVITIES

In SCS, a user performs usage and control activities on different targets.

a: USER's USAGE ACTIVITIES

A user can perform usage activities on a target user, device and resource. For example, a user can view or ''like'' a photo on a social platform. A user may poke another user, recommend a friend, or invite another user as a friend. Similarly, in remote patient monitoring of SCS, a primary doctor may check the status of a patient's insulin pump.

b: USER's CONTROL ACTIVITIES

If allowed in a system, a user can control other users' or devices' activities on her own device/resource (or any device/resource she has a control authority over) by changing attributes and policies of the device or resource. One distinctive characteristic found in SCS is a user's capability to manage activities on a target that is not owned by the user. For example, a user may control who can view a group picture shared by a friend, though this may require collective control [47]. Another example is that a connected vehicle driver may be allowed to control a car behind her to keep a certain distance by setting a minimum distance value. In addition, a user can control certain activities of a specific user without identifying a specific resource. For example, in an OSN, a mother may want to prohibit her son from uploading any content with personally identifiable information. This kind of activity is rarely discussed in traditional access control policies and models but needs to be addressed in ACON.

A session represents an active actor who has logged into the system and carries attributes and policies that may be different from those of the corresponding actor. By default, a session inherits all attributes and policies of the corresponding actor. However, if allowed, an actor can disable certain attributes or policies in a particular session or add some temporary attributes or policies for a session.

In SCS, a user does not perform any service or decision activity. If a user provides a certain service to the system, we consider the user a 3rd-party service provider. Additionally, in SCS, only the system makes the decisions on activity requests.

2) DEVICE ACTIVITIES

In the ACON framework, a device refers to a device that is managed by the system only. If a device is provided by a 3rd-party application to support a service the app provides, it is viewed as a part of the 3rd-party app and hence not explicitly captured in the system. In SCS, a device can be an actor or a target. As an actor, a device may perform usage and service activities on other targets. However, a device does not perform any control activity in the ACON framework. Unlike users or 3rd parties, the device itself does not have a system-independent ''mind'' and only acts as designed to provide the intended services.

Device's Usage and Service Activities: A device can perform usage activities on target devices and target resources. It may also perform service activities on target users and target devices. For example, a smart thermostat (actor) may read (usage activity) current temperature data (target resource) of each room and then activate the AC (service activity) or close blinds in a room with high temperature. In another case, when a smart watch (acting device) detects nearby restaurants that participate in a proximity-based discount event (usage activity), it pushes a message to the user (service activity to target user). In this case, the user may want to control this service by modifying the discount rate threshold so she only receives events with an over 20% discount. An example in a smart health scenario would be a wearable device accessing data from a patient's body sensors to obtain the patient's vital parameters. Similarly, in an industrial IoT domain, a smart device can request various connected sensors' data and use it to monitor employees' health and machinery performance efficiency.

3) 3rd-PARTY APPLICATION ACTIVITIES

In SCS, a 3rd-party application may perform usage, service and control activities on different targets.

a: 3rd PARTY USAGE AND SERVICE ACTIVITIES

In SCS, a 3rd-party application provides certain services while using necessary resources. When a user uses a 3rd-party application, it may access the user's profile information or friends list. In a smart home system example, there could be a 3rd-party proximity-based garage door opening service. In this case, the service app may want to open a garage door (service activity) when certain requirements are met.[9] To do

[9]Note that while this includes a device and can be viewed as a device's service activity, we assume the device is part of a 3rd-party system in this particular example.

TABLE 2. Actors & activities & targets in smart and collaborative system.

Activities \ Actors	User	Device	3rd-Party	SCS	Admin
Usage Activity	TU,TD,TR	TD,TR	TU,TD,TR	TU, TR, TD	
Control Activity	TU,TD,TR,S		TU,TD,TR,S	TU,TD,TR,S	TU,TD,TR,S
Service Activity		TU,TD	TU,TD	TU,TD	
Decision Activity				UA,CA,SA	

TU: Target User, TD: Target Device, TR: Target Resource, S: Session,
UA: Usage Activity, CA: Control Activity, SA: Service Activity

so, it may want to access the home owner's location data and the security camera's video data (usage activity). A home owner may want to control the application's access to her location data while still allowing the service simply based on the security camera data (user's control activity). This can be done by changing the user's policy on the application in the system.

b: 3rd PARTY CONTROL ACTIVITIES

We recognize that there could be a case in which an SCS may allow certain control capability to a 3rd-party application. For example, the proximity-based garage opening application may be allowed to control who can see the history of its operation, so only premium members can see the full operation history. Additionally, this application may limit who (user, device, or other 3rd party) can activate the garage door opener device to open the door. Additional discussion is provided in section VI.

In ACON, a 3rd-party application is viewed as a type of target resource that a user can access, as well as a type of actor who can access user information or other resources (usage activity) to provide certain services. The usage activity of the application may need to be controlled by users who use the service or someone who has control authority (which is captured as user's control activity on target resource).

4) SCS's (AUTOMATED) ACTIVITIES

In SCS, the system may perform usage, service, control and decision activities that are triggered either by user or device activities or other system conditions (which are defined using attributes such as time, location/platform of accessing device, and other system statuses). A location-based coupon or a friend recommendation can be an example of an automated activity that provides a service. It can also include automated creation of value-added resources such as most viewed video and best rated products that are computed based on user activities. Traditional access controls rarely recognize this. Below we discuss these activities.

a: SCS's USAGE AND SERVICE ACTIVITIES

In popular OSN systems such as Facebook, the system provides functionality and information to promote users' social interactions and information sharing. To do so, the system often utilizes users' shared resources (SCS's usage activity) and generates value-added resources and services. For example, when a user logs in, an OSN may collect information about nearby friends and show who is in close proximity to the user. As this service activity utilizes user information, control over this activity needs to consider the preference of users who receive the services as well as the nearby friends who are included in the provided list.

b: SCS's CONTROL ACTIVITIES

SCS may perform automated control activities by managing the attributes and policies of users, devices, other resources and sessions. These attributes and policies are updated based on the activities related to the users, devices, resources or sessions that then influence future activities. For example, Amazon's product recommending system may demote a product's reliability based on user reviews, and then remove it from a recommendation list. Additionally, SCS may add a user's current device status information in the session attribute list, so critical activities can be performed only in a trusted system.

c: SCS's DECISION ACTIVITIES

Unlike many access control systems where only users' usage activities are considered and evaluated for access decisions, in SCS, the system evaluates usage, service and control activity requests and decides whether the requests should be allowed.

5) SCS ADMINISTRATOR's ACTIVITIES

Similar to traditional access control systems, in SCS, the security administrator performs manual control activities. Controlling the administrator's activities is important, especially for highly distributed computing systems. In SCS, most research has focused on controlling user activities; hence, further studies on how administrators' activities should be managed are necessary. Similar to the approach found in administrative RBAC [48], applying user activity control approaches to control SCS administrators' activity could be an intuitive approach and may lead to interesting research. However, it is beyond the scope of this paper and is not explored here.

C. ATTRIBUTES AND POLICIES IN ACON

Similar to usage control, ACON collectively utilizes actor and target attributes to make a control decision. However, while UCON updates mutable attributes of a user to reflect the result of that user's activities as a side effect of the user activities,

TABLE 3. Protection principles by saltzer and schroeder.

Principles	Descriptions
Economy of mechanism	Keep the design simple and small.
Fail-safe default	Based on permission, not exclusion.
Complete mediation	Check every access to every object.
Open design	No secret design.
Separation of privilege	Use two keys to unlock.
Least privilege	Do not use unnecessary privileges.
Least common mechanism	Minimize common shared mechanism.
Psychological acceptability	Ease of use

ACON further allows a user to deliberately update her own or other users' attributes and policies.

In ACON, one of the unique characteristics is **policy diffusion**. Specifically, ACON incorporates both system-wide SCS policy and *individualized policies* of actors and targets. Policies for targets include *incoming action policies* that control access to the targets. Policies for actors include *outgoing action policies* that administer the actors' action on certain targets. Consequently, user and device policies may include both incoming and outgoing action policies, as they can be both actors and targets.

When an activity request is made, the ACON system collectively utilizes all the necessary attributes and policies and makes an access decision. Formal details and more information about the ACON decision can be found in [7].

V. ACCESS CONTROL PRINCIPLES: EVOLUTION AND LIMITATIONS

As new computing applications and security solutions have emerged over the last few decades, researchers and practitioners have identified various design principles that are crucial for understanding and developing access control solutions for such systems. In this section, we discuss some of the notable studies that identified these access control principles. Using these principles as a starting point, we will develop a new set of design principles for activity control in section VI. Figure 2 shows these early works and how the principles evolved over time. The arrows in the figure reflect how some of the principles in each work are subsumed/reused in later works.

A. PROTECTION MECHANISM DESIGN PRINCIPLES BY SALTZER AND SCHROEDER

One of the formative contributions in identifying access control principles is made by Saltzer and Schroeder [9]. As shown in Table 3, they identified **eight** key design principles for protection systems. They are economy of mechanism, fail-safe default, complete mediation, open design, separation of privilege, least privilege, least common mechanism, and psychological acceptability. Below, we briefly discuss these design principles and later compare them to other design principles identified in other research. More discussion on these design principles can be found at [9], [12].

Although most of these principles are still applicable, some may be viewed as rather subjective or obsolete in today's computing systems design and development practices. For example, complete mediation suggests that every access to every object must be checked for authority. However, a system may choose to facilitate random or partial screening on some access requests for better performance [12]. The least common mechanism suggests minimizing common mechanisms that are shared by multiple users. However, code reuse in the software engineering discipline suggests the opposite for better efficiency. The open design principle suggests that a system mechanism should not rely on ignorance of potential adversaries. A large number of security researchers and practitioners support the open design principle. However, there are many groups and organizations that practice the opposite.

While these eight principles are mainly applicable to enforcement- and implementation-level design and development, only some of them (such as fail-safe default, separation of privilege and least privilege) could also be applied for policy-level design. **Fail-safe default** suggests that the default should be a condition where no one is allowed without a proper privilege. This ''white list'' approach has been considered a safer approach since an accidental omission causes no information leakage and an accidental inclusion causes unwanted access of the included users only. While this may be true, not all computing applications should take the ''white list'' approach unconditionally. For example, it makes more sense to use the ''black list'' approach to block certain IP addresses in a firewall. Therefore, the fail-safe principle should be considered as an optional design decision for individual access control applications. Our proposed principles do not consider fail-safe default, but one could see it as a part of the containment principle.

Separation of privilege (or duty) is a time-honored practice and has been a key principle for RBAC. This suggests that certain jobs should be committed by multiple users to reduce the possibility of fraud or security breaches. Popular examples include two-person required missile launch, separation of purchase request and payment, etc. This principle is captured in our proposed containment principle.

Least privilege is also a long-established principle of access control. It suggests that users should use the least set of privileges necessary for the job. A well-known example for this principle is the ''need-to-know'' security rule in the military system. The principle of least privilege is captured in our proposed access control principles as part of the containment principle.

Both the economy of the mechanism and psychological acceptance principles are issues of design philosophy for access control enforcement and implementation. In our proposed principles, these issues of design philosophy are implicitly incorporated in different principles such as abstraction, controllability, containment and automation.

B. RBAC96 PRINCIPLES

In RBAC96 [10], there are four foundational principles, namely, abstraction, separation of administrative functions, least privilege and separation of duty. Unlike the protection

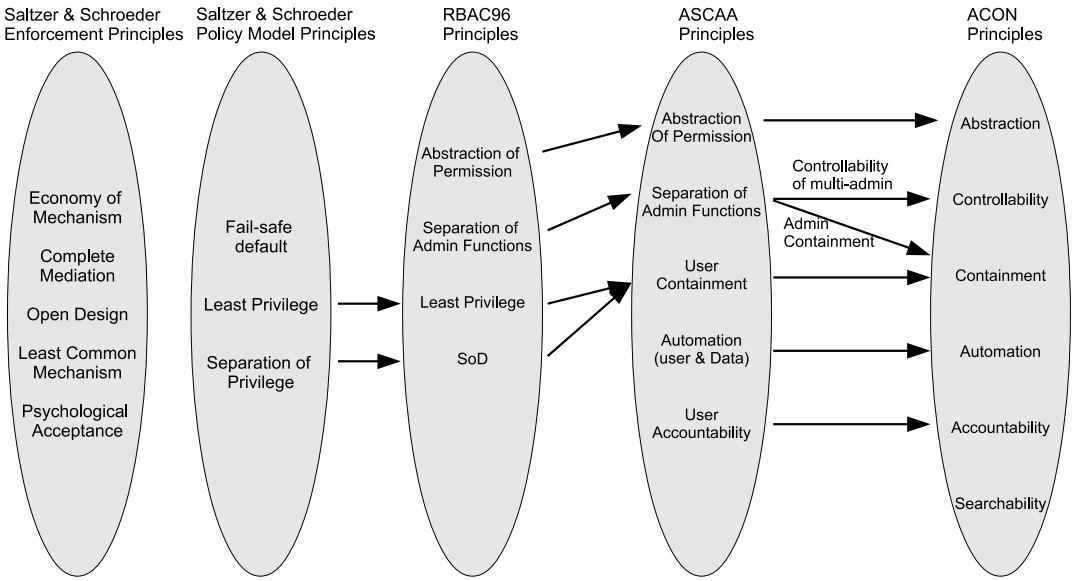

FIGURE 2. ACON design principles and previous principles.

principles in [9], RBAC96 principles are focused on access control policy and model design. Nevertheless, the RBAC96 model itself does not enforce or require these principles. One can design an RBAC system that violates all of these principles. These principles merely provide a guideline for better design and development of an RBAC system.

In RBAC96, **abstraction** means abstraction of permission. Examples for permission are system-level operations such as read, write, select and delete. The main benefit of permission abstraction is that it enables controls on abstract operations specific to computing system applications that are not possible with bare permissions. Examples for permission abstraction include credit and debit operations in a banking system application. It also provides management ease and simplicity by allowing meaningful and manageable units for access control. Permission abstraction is identified in our proposed abstraction principle, which also includes several newly identified abstractions (see Figure 2).

The second principle is **separation** of administrative functions. This recognizes different administrative functions that require different skills and knowledge and are distinct in operation. In RBAC96 user-role assignment and permission-role assignment are recognized separately. In our proposed principles, this principle of separation of administrative functions is subsumed by the newly expanded administrator containment principle and controllability of the multiple administrator principle as shown in Figure 2.

The main ideas of **least privilege** and **separation of duty** in RBAC96 are not changed from the Saltzer and Schroeder's principles.

C. ASCAA PRINCIPLES

The ASCAA principles [11] are identified mainly for RBAC, although the authors also claim they are mostly applicable to other access control policy models as well. ASCAA stands for abstraction, separation, containment, automation and accountability. These principles encompass RBAC96 principles and further identify additional principles that can be found in other access control systems such as usage control and digital rights management applications.

The **abstraction** and **separation** principles are not changed from RBAC96 and refer to abstraction of permission and separation of administrative functions, respectively.

The **containment** principle is newly identified to unify previously identified principles of least privilege and separation of duty and to further encompass other constraints and usage control elements such as cardinality, usage limits and rate limits. The main purpose of containment is to minimize the detriment users can perpetrate either deliberately or accidentally. RBAC96 introduced an open-ended notion of constraints to recognize various restrictions including separation of duties and cardinality. Usage control [46], [49] points out various user activity-based consumable restrictions such as usage limits and rate limits. Our proposed ACON principles capture this as a user containment principle and include additional containments on security officers/administrators and 3rd-party applications.

The **automation** principle has been newly identified in ASCAA principles to suggest automation of access control administration. This largely makes sense considering that today's computing applications usually handle a large number of users and data, and manually managing and controlling these users' access to the data is not a trivial task. In traditional access control systems, assignment and revocation have been treated as administrative actions, which usually require human involvement. In RBAC, this assignment process is significantly simplified by assigning roles (rather than individual permissions) to users. Sandhu and Bhamidipati claim that automation of user-role assignment can be achieved with self-assignment. Here, self-assignment means

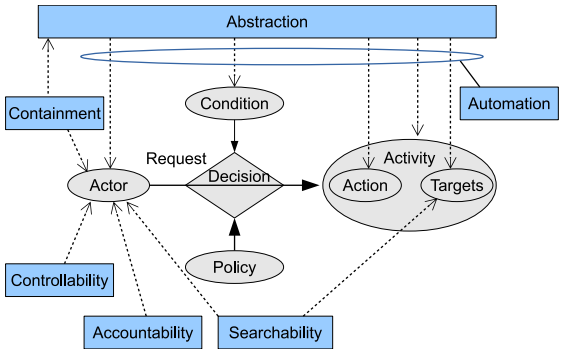

FIGURE 3. Activity control design principles.

that the system provides users with the ability to assign roles to themselves when necessary. An example of the automatic revocation of a role is self-expiring role assignment, which allows the revocation of a role based on a duration or even a user abuse of the role. A similar self-assignment can be found in modern OSN systems where users are allowed to establish their social relationships with other users and to specify access rules based on these relationships.

In ASCAA principles, automation on both assignment and revocation mainly focuses on automation of user and data abstractions. Automation of user and data abstraction principles are captured in our proposed automation principle. Our proposed automation principle encompasses automation of actors, targets (including resources and services), conditions, and action/activity abstractions. This will be further discussed in Section VI.

The **accountability** principle is identified to emphasize user responsibility for their actions in a system. This principle suggests using adjustment to achieve accountability in a sense that not all authorized actions are identical and some sensitive operations require more stringent auditing, notification or authentication. Access to highly sensitive information should alert some users or may require additional authentication. The main idea of the accountability principle is essentially unchanged in our proposed principles but has been expanded to incorporate the accountability of administrators, users, and other participants.

VI. ACTIVITY CONTROL DESIGN PRINCIPLES

In this section, we propose several design principles for activity control for smart and collaborative computing systems. While activity control can accommodate various conventional access control models such as RBAC, ABAC, ReBAC, PBAC and UCON, the focus of this paper is not on specific decision parameter based-access control. Rather, the goal of the paper is to properly identify and understand access control aspects that are crucial for SCS but rarely identified or discussed in the existing access control models. Figure 3 shows these principles and where these principles are applicable in the ACON framework. How each of these principles can be applied in the ACON framework is discussed below. The ACON framework shown in Figure 3 is a simplified view of the one shown in Figure 1.

While these principles are general enough to be applicable to different computing environments or applications, our focus in this paper is mainly on smart and collaborative computing systems. Additionally, the focus of our proposed principles lies in activity control policies and model designs. Any enforcement- and implementation-related principles are out of our scope and not discussed in this paper. In addition, note that the proposed principles are not necessarily complete or always applicable. Instead, they should be viewed as guidelines for the development of activity control systems for smart and collaborative computing systems.

A. ABSTRACTION

In ASCAA principles, abstraction is unchanged from RBAC96 principles. In both ASCAA and RBAC96 principles, abstraction means abstraction of permission such as credit and debit. The **abstraction of permission** principle suggests that rather than controlling system-level operations (or permissions) such as read, write, execute, delete, etc., one should use a more meaningful and manageable abstraction of permissions as a decision parameter that is often specific to an application. This is especially true in an application with a large number of users and data where management of user access requires significant administrative efforts. Additionally, as the computing system environment becomes more complex and diverse, it is crucial to make access control solutions and administration as streamlined and manageable as possible while still meeting the security requirements of today's feature-rich smart and collaborative computing systems. The same benefits can be achieved by abstracting other information used in the access control decision process. Below, we further articulate abstraction principles and expand the scope of abstraction beyond permission abstraction.

1) ABSTRACTION AS DECISION PARAMETER

In an access control system, an access request generally includes information about the requesting user, requested action, target resource and sometimes other system or environmental information. A bare form of this information may include user ID, system operation (e.g., read or write), and object ID. For better control and simplified administration, more meaningful abstractions have been used as decision parameters in many access control models. When designing an access control solution for a computing system, it is crucial to identify what kind of abstractions an access decision process needs to use to meet the security goals of the system. For example, in many business applications, a notion of role is used instead of user ID. In addition, permission abstractions such as credit and debit are used instead of system-level operations. In military systems, security clearances and classifications are used. In modern access control, other abstractions that are often dynamically constructed and managed may need to be adopted. Examples of dynamic abstractions include social relationships, provenance, etc.

In ACON systems, control policies utilize abstractions as decision parameters. For each abstraction type, a set of

available abstractions and their definitions are identified, and access control policies are specified using these abstractions. For example, the abstraction type ''relationship'' comprises friends, coworkers, family, local friends and local family. Other abstraction types, such as blood sugar level, body temperature and heart rate, can be used as decision parameters.

Abstractions can be used to form other abstractions. For example, ''friend relationship'' and ''hometown'' are two different abstractions that can be *grouped* to form a ''hometown friends'' abstraction. ''Blood sugar level'' and ''heart rate'' can be further abstracted as a ''body condition''. A location-based abstraction such as ''city'' and ''county'' can be *nested* into another location-based abstraction such as ''state''. Furthermore, abstractions can be *linked* (forming a graph) based on their relationships or dependencies to define another abstraction. For example, a file can have a declassified status if it has been redacted by a specialist, and then the redaction process is verified/signed by an authority. Here, the ''declassified'' abstraction is defined using two linked abstractions of ''redact'' and ''verify'' in that order.

In ABAC, an attribute (with attribute value(s)) can be viewed as an abstraction; a collection of attribute-value pairs can be viewed as an abstraction. The notion of ''category'' in [50], [51] is another way of viewing this collection, hence an example of abstraction.

2) TARGETS OF ABSTRACTION

As shown in Figure 3, ACON extends the abstraction principle found in ASCAA principles to include abstraction of other ACON components such as actors, targets, actions/activities, services and environmental conditions. These abstractions are discussed below.

An **abstraction of actors** groups actors based on specific values of certain actor properties such as user role, user age range, user gender, user location, device location, device type, etc. While actor abstraction was not one of the identified design principles in ASCAA principles, role is still the underpinning decision parameter in RBAC. In RBAC, actor (more precisely, user) abstraction is realized using roles that are also used to assign permission abstractions. Using the same abstraction type for both users and permissions, access control policy is configured by simply using user-role and permission-role assignments and an optional role hierarchy. In ABAC, a set of user attributes (or device attributes in the case of an IoT system) can be used to identify a meaningful user/device group in a system. Such an attribute set is collectively meaningful and significant for the system and can be effectively utilized as a control parameter in ABAC solution.[10]

Other user-related information, such as provenance and social relationships, can also be used to construct actor abstractions. However, user provenance is different from typical attributes, as it includes information about certain actions on objects, forms a chronological ordering and can form certain patterns. Additionally, (social) relationships are different from attributes or provenances since they include target (user or object) and form a relationship graph. Furthermore, unlike RBAC where actor (aka, user) abstraction and activity (aka, permission) abstraction are set in advance by the administrator, in PBAC (and in ReBAC for OSN), actor abstraction is rather dynamic and computed at each access request.[11] Therefore, one should consider various actor information to construct meaningful actor abstractions for fine-grained control as well as simplified administration.

Abstraction of targets suggests using meaningful abstractions of targets. Similar to actor abstraction, target information, such as target attributes, target provenance, and relationships to users (or other targets), can be utilized to define target abstractions. Additionally, some target information forms a graph, and meaningful abstraction may need to be defined based on specific graph formations or patterns. For example, if a certain workflow is required to have been performed on a target resource for a user to access the resource, a specific provenance pattern of the resource must be verified.

In ACON, a service feature is viewed as a special type of target resource. Access to a service means that the actor may exercise those activities that are available in the service. For example, a user who is granted an online survey service can access a survey building template, sample survey questions, survey analysis tools and a list of email addresses to send survey invitations to. Controls on these multilevel activities may need to be addressed together or individually.

Abstraction of activities suggests using meaningful abstractions of activities. In ACON, an activity itself is an abstraction that is defined using a set of actions on certain targets. Here, targets can be an empty set, meaning that an activity can be defined without any target, which is equivalent to an action abstraction. In trust negotiation [52], [53], a negotiation process can be abstracted to a low-level or high-level trust dependent on what kind of credentials are shared between a user and a system during the negotiation phase; thus, the abstraction can be used for access control decisions. In usage control, a set of obligation requirements can be captured as an abstraction. This abstraction has to be verified during the decision process whether it was fulfilled or not [46]. In a smart health system, a set of actions such as ''check body temperature'' and ''check heart rate'' can be combined as an abstracted activity called ''check patient condition''.

Abstraction of conditions suggests utilizing abstraction of system or environmental conditions such as the current CPU usage meter or memory usage meter, current location

[10]Note that an attribute set may not have an explicit abstraction name, but it could be defined in ABAC policy to capture a meaningful user group who shares the same access privileges. An attribute set can be defined in various formats but is not discussed in this paper, as it is beyond the scope of the paper. In ABAC, unlike RBAC, actor abstraction and permission abstraction may be named separately; hence, the access control policy needs to be defined using both actor and permission abstraction.

[11]This means that the security architect should consider how to automate the abstraction process. Automation is discussed later in the paper.

or timezone, current time, system under attack, etc. One may define an abstraction called the ''emergency mode'' for certain system conditions or environmental conditions and utilize different access control policies for different situations. It could also be considered for a location-based mobile system or an IoT system where environmental conditions are a key factor for access control decisions. In usage control, condition-based usage decision is an example abstraction of condition [46].

3) ABSTRACTION AND ATTRIBUTE

Abstraction is similar to the notion of attributes in ABAC in that both capture the properties and statuses of users, objects and the system. However, abstraction is a more generalized construct in its meaning and structure, and captures any meaningful semantics that can be used for the policy specification and decision-making processes. In this paper, we treat attribute as an instance of abstraction.

While not all of these abstractions may need to be used in a single system, we believe these abstractions are crucial for simplified and more usable/manageable access control systems and must be considered at the time of access control system design and development.

B. CONTROLLABILITY

The controllability principle is newly identified in our proposed principles and suggests considering the *controllability of multiple participants*. The proposed controllability principle emphasizes that there are participants (such as users and 3rd-party applications) other than the SCS provider who may want or need to exercise access control administration in the system. It is recognized to suggest that activity control systems should accommodate effective and collective control capability while embracing or resolving possibly conflicting access control policies of the participants.

In ACON, certain participants can control the activities of other participants or even themselves by modifying decision parameters and policies that correspond to actors or targets. Changing a user's decision parameter allows controls on the user's activity or others' activity against the user. Changing a target's decision parameter allows controls on activity against the target. Below, we discuss who may need to have control capability in the ACON system. Please note that we do not consider device controllability as discussed in Section IV.

1) ADMINISTRATOR CONTROLLABILITY

Administrator controllability is the most common and typical capability in an access control system. In a MAC system, administrators (or security officers) control user activity based on a strict information flow policy, while users do not have any choice other than following the decision made by the system. In RBAC, administrators control user activity through user-role assignments and permission-role assignments. In a typical business application, a policy maker establishes general policy rules, while a policy implementer implements user access privilege based on the policy rules provided by the policy maker [54]. In some commercial applications such as digital rights management applications, the administrator's (or application provider's) control capability reaches even further to the content stored in users' own computer systems.

The result of ''separation of admin functions'' found in the ASCAA principles is captured as part of the administrator controllability, while the rule of such separation is captured in the containment principle shown later in this section.

2) USER CONTROLLABILITY

Unlike administrator controllability, it is rarely the case that a typical access control system incorporates the user's control capability. This notion of user controllability gains importance in today's computing environment, such as online social computing and cloud computing systems where users' data are stored and computed in a service provider's system and where the data could be used by yet another application provider. User controllability is somewhat similar to the discretionary access control policy, which allows users to have full control over their own data, except that in ACON, access to the data can be controlled by nonowners with a certain authority. In trust negotiation, a user can configure what kind of information can be shared with a server for access authorization [52], [53]. In today's social computing applications, it is often the case that a system provides users some control capability on user-created data mainly for privacy purposes. In other words, users are allowed to create their own access control policy rules on their own data, while the system enforces the users' access control policies on behalf of the users. In a typical cloud computing service, a user's data are stored in a service provider's data center, and the user may want to have certain control capability on the stored data or even want to take the data with him when the service is terminated [55]. In a smart home system, a homeowner may want to control who can change her smart thermostat's setting or which smart device can access it to retrieve the setting information.

Furthermore, the user controllability principle includes controllability of the originator or other stakeholders and suggests that an access control system should consider a user's ability to control his own data and copies of the data throughout the lifetime of the data regardless of the location or possession of the data. The originator control system allows an originator to retain controls over the disseminated or copied data. In some social computing applications, a user may make a copy of a photo from his friend's profile and post it in his photo album. If the photo is allowed to be seen only by friends of the owner, those who are not the friends of the originator should not be able to see the photo even though they are friends of the user who copied it.

However, one should not misunderstand what users can do with their control capability. Although user controllability allows a user's control over others' access to the user's data, it is not usually the case that the user can achieve privacy protection against the system unless the system provides

such a protection mechanism. In a social networking service, sharing users' private information with others is often the service provider's business model. Therefore, this type of user controllability may or may not be available depending on the system, and the degree of controllability may vary.

3) THIRD-PARTY APPLICATION CONTROLLABILITY

The controllability principle also recognizes the possibility of 3rd-party application controllability in SCS. In an SCS environment, it is not uncommon to see mashup applications where multiple applications are incorporated to provide a seamless service. An SCS may allow these 3rd-party applications to collect certain user/device data and control users' access to the collected data. For example, in a social networking service such as Facebook, 3rd-party plug-in applications provide services to users by utilizing developer APIs (Facebook Platform). Suppose a 3rd-party provides a survey service app and a user initiates a survey at his social network site. Although the app collects survey data from the user's friends (and/or other participants) and shows a basic result to the user or the participants, it may want to control the user's access to additional analysis results and only share them with paid users.[12]

If an SCS provider, users and 3rd-parties can configure their access control policy rules, it is inevitable to have conflicts in the control policies. This means that the system should be able to resolve conflicting decisions through a careful design of the activity control system and conflict management solutions. If allowed by the system, the conflicts could be resolved by the interest parties or individuals. In a cloud computing environment, the cloud provider, service provider and administrator of the service users want to exercise their control capability against other parties [55]–[57]. In today's computing environment, access control systems should strive to achieve an effective and collaborative control capability of multiple participants.

C. CONTAINMENT

In ASCAA principles, the main purpose of the containment principle was to minimize the damage users can make, whether it was deliberate or accidental. The separation of duties and least privilege are some of the policies that can be utilized in an access control system in this regard. In activity control, in addition to this ''user containment'', additional containments on administrators, abstractions and other participants need to be considered.

1) USER CONTAINMENT

User containment means containment applied to a user. In social networking applications, the previously recognized policies such as separation of duty, least privilege, cardinality, usage limit, and rate limit are all applicable. For example, in a voting service, anyone administering a poll may not be allowed to vote in the poll (separation of duty). A user with expert status in a crowdsourcing application such as Wikipedia should be able to login without expert privilege for certain cases (least privilege). In a social networking application, a user may be allowed to send SMS messages 5 times per day (rate limit) or until account balance reaches zero (usage limit). Another user containment policy in a social networking system is user-relation graph-based control policies such as ''friends-only'' and ''friends-of-friends'' accesses.

2) DEVICE CONTAINMENT

Device containment means containment applied to a device. As devices perform usage and service activities, various containment policies found in user containment are applicable to minimize possible harm or damage in the system. For example, in a connected vehicle system, a vehicle that reported an accident should not be able to confirm or re-report the accident.

3) ADMINISTRATOR CONTAINMENT

The administrator containment principle suggests considering possible restriction policies that could be applied to administrators or service providers to minimize the damage administrators can make. Similar policies identified in user containment can be found in administrator containment. The separation of administrative functions in RBAC96 and ASCAA principles is an example of administrator containment. It suggests separation of user-role assignment from permission-role assignment because of their different sets of required knowledge bases and operational distinctions. Bauer et al identified the separation of policy makers and policy implementers [54]. One may also consider workload-based assignment of administrative functions to balance administrators' workload evenly, which can reduce possible mistakes that might be caused by overloaded duties. Furthermore, an administrator could log in with minimum privilege that is required for a certain job.

4) 3rd-PARTY CONTAINMENT

Containment on 3rd-party applications recognizes that there could be some access control policies that 3rd-party applications should be bounded by. We believe the similar containment policies discussed above are likely to be applicable to this type of containment.

5) ABSTRACTION CONTAINMENT

Abstraction containment suggests applying a restriction to maximize the benefit of abstraction. In general, abstraction simplifies the management of access control and provides more intuitive and human-friendly controls. Overly fine-grained abstraction may diminish this benefit of abstraction. However, overly coarse abstraction may fail to have effective control ability and may not support sufficient least privilege. Therefore, abstraction should be detailed enough to support a meaningful and manageable control unit.

[12]Note that SCS can manage the 3rd parties' control over the participants' access to the targets (analysis results) only if the targets are under the control of the SCS. If the targets belong to the 3rd party's management, the SCS is unlikely to be able to manage or limit the 3rd-party's controllability.

In our proposed principles, we leave the containment principle as an open-ended principle that is not limited to the containment instances discussed here.

D. AUTOMATION

In our proposed design principles, the automation principle focuses on automation of abstraction to achieve automation of access control administration. Automated abstraction also provides stale-safe control through a dynamic and continuous modification of abstraction. In this section, we discuss the automation of actor, target, condition, and activity abstractions.

1) AUTOMATION OF ACTOR ABSTRACTION

The automation of the actor abstraction principle suggests that automated assignment or revocation of actor abstraction is necessary to minimize administrative intervention, especially for a large system. Note that the automated assignment means no administrator action is required to assign abstractions to an actor. Instead, this is accomplished through other ways. For example, automated assignment or revocation of actor abstraction can be performed by either the actor, the system or both. In a social networking system, a user may accept a friendship invitation from another user and become a friend of the inviting user. A system can assign or revoke user abstraction when a certain condition has been met. This can be accomplished based on changes in other related systems. For example, a user can be assigned to (or revoked from) a certain role when she is included in (or removed from) the human resource system of a company with a particular job description [54]. Additionally, it can be performed based on time, location or user activity. A user can be assigned to a certain project only during business hours (time-based assignment/revocation). A reviewer is promoted to an expert reviewer when her reviews receive more than a certain number of high recommendations from other users for a given period of time (activity-based assignment) or degraded to a normal reviewer if the reviewer does not post regularly (activity-based revocation). If a smart watch detects an available Wi-Fi connection, it may set itself into hi-speed mode, so it can perform certain activities that need a fast Internet connection.

2) AUTOMATION OF TARGET ABSTRACTION

The automation of the target abstraction principle is newly identified in our proposed principles. Similar to automation of actor abstraction, automated assignment or revocation of target abstraction is necessary for the efficient administration of access control systems, especially in a system where there are a large number of actors who create and/or manage data that can be activity targets. Automated assignment or revocation of target abstraction can be done by either the user or system. In a social networking system, a user may tag a photo and identify who is in the photo. A system can also assign or revoke a target abstraction to or from the target if a certain condition has been met. For example, when a project is initiated, a set of data that is related to a particular company becomes a part of the project data during the project duration, so it can be accessed by users in the project (time-based assignment/revocation). When a review receives more than a certain number of recommendations, the review becomes a part of the best reviews and can be presented at a more visible place until a better review is allocated (activity-based assignment/revocation). In a home security system, security cameras with more frequent motion detection events are marked as high-traffic areas and could be listed on the first page of the monitoring app window.

3) AUTOMATION OF CONDITION ABSTRACTION

The automation of condition abstraction principle suggests that automated changes of system status should be made when a certain system or environmental condition has been met. For example, if a system detects an abnormal activity, it sets itself into a possible attack mode, so no access to critical data can be allowed. Unlike automation of actor abstraction and target abstraction, only the system is allowed to perform automation of condition abstraction.

4) AUTOMATION OF ACTIVITY ABSTRACTIONS

The automation of the activity abstraction principle suggests that an abstraction can be assigned to activities without any administrative actions. It could be a user who identifies a user-defined abstraction for a set of activities. When a certain condition has been met, a system can modify the abstractions assigned to activities. While automation of all of these abstractions may not be necessary in a single access control system, it is crucial to reduce manual administration of these abstractions in a large and complex system such as a smart and collaborative computing system for scalability concerns.

E. ACCOUNTABILITY

In a world of insecure cyber systems, accountability can improve security by discouraging misbehavior [55]. Unfortunately, today's cyberspace is neither secure nor accountable. Carl Landwehr called for an accountable internet infrastructure to remedy today's insecure cyberspace [58]. He also recognized that the internet should remain open for innovative applications and services. While this must be true and pursued by researchers and practitioners, accommodating accountability at an application level is also a must-have for improved security in cyberspace. Today's cyber world is shifting computing resources to a centralized system such as the cloud. One of the greatest concerns of cloud users will be the security and trustworthiness of the cloud and the services provided by the cloud. Hence cloud and service providers must incorporate mechanisms that can provide users with improved accountability and trustworthiness. Assuming perfect security is not feasible in the near future, holding both service providers and participants accountable for their activities is likely to increase the security and trustworthiness of the system hence leading to a more reliable system.

In our proposed activity control principles, user accountability is primarily unchanged from ASCAA principles but

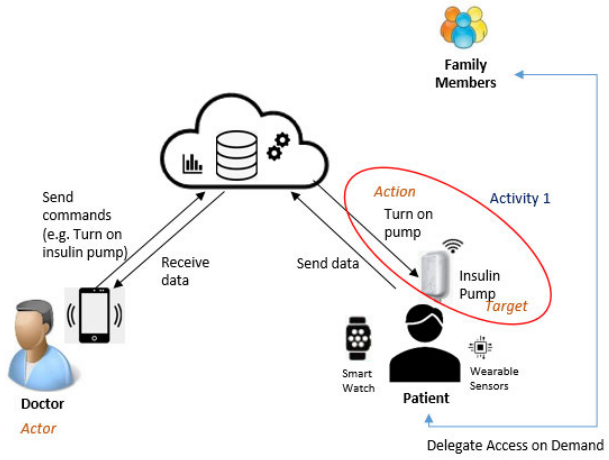

FIGURE 4. Smart health use case.

is suggested to further include the accountability of other participants, such as administrators and 3rd-party application providers. In a smart and collaborative system, the participation of users, devices, system providers, and other 3rd-party application providers is crucial for the success of the system. Naturally, facilitating accountability of these participants in access control systems will provide better security and trust to all participants.

F. SEARCHABILITY

Searchability includes both the ability to search what an actor wants and to control being searched by actors such as users, devices and 3rd-party applications. We believe searchability is one of the most crucial principles in a modern access control system. In a large system such as Facebook or an enterprise-wide smart and collaborative system, being a participant does not guarantee that one can find other participants or data that the participant wants [5]. In such a system, the access control system should consider incorporating participants' ability to search.

On the other hand, with this search ability in a system, participants in the system can now be searched by anyone in the system. To restrict this unlimited searchability, the system may need to allow its participants an option to opt out from a certain search list or an option to choose who can search the participants in the system. Of course, this is within the capacity given by the service provider. Many of today's social networking applications provide users with a global search capability as well as a capability to traverse the relationship graph to find friends. However, they are often reluctant to allow users to completely opt-out from a search list because their business model relies heavily on user connectivity and searchability.

VII. A SMART HEALTH USE CASE

In this section, we present a smart health use case and briefly demonstrate how the ACON principles are applicable in this use case.

Figure 4 depicts a smart health use case where a doctor is monitoring a patient who is diabetic and needs insulin when her sugar level reaches a threshold. The patient has a wearable sensor that measures her sugar level and a wearable smart watch monitoring her heart rate, pulse, and other vital parameters. The patient also has an attached insulin pump that can be remotely turned ON or OFF by the doctor through a mobile app. The doctor receives an alert when the sensor senses that the patient's sugar level is above a threshold value and turns the insulin pump ON through the app. The patient can also delegate access to his/her family members on demand. For example, in an emergency situation, the patient may have defined an access delegation policy for enabling access to family members. This policy would only be triggered when such a situation arises.

This use-case scenario utilizes a cloud-enabled IoT system where all the devices send data to the cloud and participants can receive data and send commands through the cloud. We understand that one may argue to use edge computing instead of cloud IoT, as it can minimize network latency. While our paper utilizes a cloud IoT-based case to discuss the proposed framework and design principles, the proposed activity control framework and the design principles are deployment agnostic and can be applied to different computing environments such as edge computing.[13]

Figure 4 shows an example activity where the doctor receives a high sugar level alert of a patient. Then, the doctor sends a command to perform *Activity 1*, where the action is *"turn on pump"* and the target is *insulin pump*. Several ACON principles can be found in this use-case scenario. For example, abstractions of participants such as a primary doctor (role), family member (relationship), sugar level (attribute) and heart rate (attribute) are used as decision parameters.

The *controllability* principle is also found in the scenario. For example, a patient can control who can access her health data and smart devices by modifying access control policy that is defined using decision parameters such as attributes (e.g., heart rate) and user relationships (e.g., parents, spouse). If the patient's heart rate is very high and no movement is detected at the patient's location, then certain family members can access the patient's device and send notifications to emergency services and the doctor. The patient may change her access policy and allow a designated family member such as a spouse to also have control capability over her devices so the spouse can decide who can access her devices. Additionally, there could be third-party applications that may have control capability over the patient's device. For example, the insulin

[13]Please note that in practice, the proposed framework and design principles can be used as guidelines that system architects may reference at the high-level design phase. Developing an activity control system for an SCS application requires extensive design and implementation details that are specific to the application system. For example, developing an activity control solution for an SCS application that utilizes edge computing requires solutions for implementation-specific issues that are unique to edge computing, such as resource constraints, device synchronization, real-time access control decisions, etc. However, these are outside of this paper's scope and hence not discussed.

pump manager app is allowed to change a list of authorized 3rd party apps that can access certain device data so that other 3rd party apps can provide additional services (that are user approved) to the doctor or the patient.

The *containment* principle can be applied to different participants (e.g., users, devices, etc.) in smart and collaborative systems. As discussed, the main idea is to minimize any potential security and privacy threats. In a healthcare use case, a doctor may not be allowed to prescribe medical marijuana to herself (separation of duty). Additionally, the insulin pump should not inject more than the maximum daily dose per day (usage limit).

The *automation* principle means automation of abstraction without administrator action. A doctor may gain a specialized doctor role for a patient when the patient visits the doctor for special treatment. This is an example of ''automation of actor abstraction'' as the role is assigned without administrator action. The automation of other abstractions can also be implemented. In a smart health care system, the *accountability* principle should be a key requirement for a reliable and trustworthy system. The *searchability* principle can also be important for the system to allow a patient to find a doctor.

VIII. CONCLUSION

In this paper, we present an enhanced next generation activity control framework that incorporates collective controls on usage, service and control activities performed by multiple participants in smart and collaborative systems. We first compared existing access control models based on several key characteristics that highlight the necessity of new activity control. We then proposed an ACON framework and a set of design principles for activity control comprising abstraction, controllability, containment, automation, accountability and searchability followed by their applicability in a smart health use case. We believe the proposed ACON framework and design principles are critical for novel access and activity control systems. This perspective paper will provide a foundation and reference for the design and development of activity control solutions, including security models, architectures and deployment prototypes required in futuristic fast-evolving smart and collaborative computing systems.

REFERENCES

[1] Y. Cheng, J. Park, and R. Sandhu, ''Preserving user privacy from third-party applications in online social networks,'' in *Proc. 22nd Int. Conf. World Wide Web (WWW Companion)*, L. Carr, A. H. F. Laender, B. F. Lóscio, I. King, M. Fontoura, D. Vrandecic, L. Aroyo, J. P. M. de Oliveira, F. Lima, and E. Wilde, Eds., 2013, pp. 723–728.

[2] F. Paci, A. Squicciarini, and N. Zannone, ''Survey on access control for community-centered collaborative systems,'' *ACM Comput. Surveys*, vol. 51, no. 1, pp. 1–38, Apr. 2018.

[3] Y. Cheng, J. Park, and R. S. Sandhu, ''A user-to-user relationship-based access control model for online social networks,'' in *Proc. 26th Annu. Data Appl. Secur. Privacy (IFIP)*, in Lecture Notes in Computer Science, vol. 7371, N. Cuppens-Boulahia, F. Cuppens, and J. García-Alfaro, Eds., Paris, France. Berlin, Germany: Springer, Jul. 2012, pp. 8–24.

[4] Y. Cheng, J. Park, and R. Sandhu, ''An access control model for online social networks using user-to-user relationships,'' *IEEE Trans. Dependable Secure Comput.*, vol. 13, no. 4, pp. 424–436, Jul./Aug. 2016.

[5] P. W. L. Fong, M. M. Anwar, and Z. Zhao, ''A privacy preservation model for facebook-style social network systems,'' in *Proc. 14th Eur. Symp. Res. Comput. Secur. Comput. Secur. (ESORICS)*, in Lecture Notes in Computer Science, vol. 5789, M. Backes and P. Ning, Eds., Saint-Malo, France. Berlin, Germany: Springer, Sep. 2009, pp. 303–320.

[6] J. Park, R. Sandhu, and Y. Cheng, ''ACON: Activity-centric access control for social computing,'' in *Proc. 6th Int. Conf. Availability, Rel. Secur.*, Aug. 2011, pp. 242–247.

[7] J. Park, R. Sandhu, and Y. Cheng, ''A user-activity-centric framework for access control in online social networks,'' *IEEE Internet Comput.*, vol. 15, no. 5, pp. 62–65, Sep. 2011.

[8] X. Jin, R. Krishnan, and R. S. Sandhu, ''A unified attribute-based access control model covering DAC, MAC and RBAC,'' in *Proc. 26th Annu. Conf. Data Appl. Secur. Privacy (DBSec)*, vol. 7371, N. Cuppens-Boulahia, F. Cuppens, and J. García-Alfaro, Eds., Paris, France. Berlin, Germany: Springer, Jul. 2012, pp. 41–55.

[9] J. H. Saltzer and M. D. Schroeder, ''The protection of information in computer systems,'' *Proc. IEEE*, vol. 63, no. 9, pp. 1278–1308, Sep. 1975.

[10] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, ''Role-based access control models,'' *Computer*, vol. 29, no. 2, pp. 38–47, 1996.

[11] R. Sandhu and V. Bhamidipati, ''The ASCAA principles for next-generation role-based access control,'' in *Proc. 3rd Int. Conf. Availability, Rel. Secur. (ARES)*, Jan. 2008, pp. 27–32.

[12] R. E. Smith, ''A contemporary look at Saltzer and Schroeder's 1975 design principles,'' *IEEE Security Privacy*, vol. 10, no. 6, pp. 20–25, Nov. 2012.

[13] Y. Tian, N. Zhang, Y. Lin, X. Wang, B. Ur, X. Guo, and P. Tague, ''SmartAuth: User-centered authorization for the Internet of Things,'' in *Proc. 26th USENIX Secur. Symp. Secur. (USENIX)*, E. Kirda and T. Ristenpart, Eds., Vancouver, BC, Canada, Aug. 2017, pp. 361–378.

[14] Z. B. Celik, G. Tan, and P. McDaniel, ''IoTGuard: Dynamic enforcement of security and safety policy in commodity IoT,'' in *Proc. 26th Annu. Netw. Distrib. Syst. Secur. Symp.*, San Diego, CA, USA, 2019, doi: 10.14722/ndss.2019.23326.

[15] M. Fernández, A. F. Tapia, J. Jaimunk, M. M. Chamorro, and B. Thuraisingham, ''A data access model for privacy-preserving cloud-IoT architectures,'' in *Proc. 25th ACM Symp. Access Control Models Technol.*, J. Lobo, S. D. Stoller, P. Liu, Eds., Barcelona, Spain, Jun. 2020, pp. 191–202.

[16] W. He, M. Golla, R. Padhi, J. Ofek, M. Dürmuth, E. Fernandes, and B. Ur, ''Rethinking access control and authentication for the home Internet of Things (IoT),'' in *Proc. 27th USENIX Secur. Symp. USENIX Secur.*, W. Enck and A. P. Felt, Eds., Baltimore, MD, USA, Aug. 2018, pp. 255–272.

[17] W. He, V. Zhao, O. Morkved, S. Siddiqui, E. Fernandes, J. Hester, and B. Ur, ''SoK: Context sensing for access control in the adversarial home IoT,'' in *Proc. IEEE Eur. Symp. Secur. Privacy (EuroS P)*, Sep. 2021, pp. 37–53.

[18] M. Yahyazadeh, S. R. Hussain, M. E. Hoque, and O. Chowdhury, ''PATRIOT: Policy assisted resilient programmable IoT system,'' in *Proc. 20th Int. Conf. Runtime Verification (RV)*, in Lecture Notes in Computer Science, vol. 12399, J. Deshmukh and D. Nickovic, Eds., Los Angeles, CA, USA, Cham, Switzerland: Springer, Oct. 2020, pp. 151–171.

[19] M. Yahyazadeh, P. Podder, E. Hoque, and O. Chowdhury, ''Expat: Expectation-based policy analysis and enforcement for appified smart-home platforms,'' in *Proc. 24th ACM Symp. Access Control Models Technol.*, F. Kerschbaum, A. Mashatan, J. Niu, and A. J. Lee, Eds., Toronto, ON, Canada, May 2019, pp. 61–72.

[20] E. Bertino, ''IoT security a comprehensive life cycle framework,'' in *Proc. IEEE 5th Int. Conf. Collaboration Internet Comput. (CIC)*, Los Angeles, CA, USA, Dec. 2019, pp. 196–203.

[21] M. Gupta and R. S. Sandhu, ''The GURA_{G} administrative model for user and group attribute assignment,'' in *Proc. 10th Int. Conf. Netw. Syst. Secur. (NSS)*, in Lecture Notes in Computer Science, vol. 9955, J. Chen, V. Piuri, C. Su, and M. Yung, Eds., Taipei, Taiwan. Cham, Switzerland: Springer, Sep. 2016, pp. 318–332.

[22] M. Gupta and R. S. Sandhu, ''Reachability analysis for attributes in ABAC with group hierarchy,'' *CoRR*, vol. abs/2101.03736, pp. 1–15, Jan. 2021.

[23] A. Alshehri and R. Sandhu, ''Access control models for cloud-enabled Internet of Things: A proposed architecture and research agenda,'' in *Proc. IEEE 2nd Int. Conf. Collaboration Internet Comput. (CIC)*, Pittsburgh, PA, USA, Nov. 2016, pp. 530–538.

[24] I. Bouij-Pasquier, A. A. Ouahman, A. A. El Kalam, and M. O. de Montfort, ''SmartOrBAC security and privacy in the Internet of Things,'' in *Proc. IEEE/ACS 12nd Int. Conf. Comput. Syst. Appl. (AICCSA)*, Marrakech, Morocco, Nov. 2015, pp. 1–8.

[25] N. Ye, Y. Zhu, R.-C. Wang, R. Malekian, and L. Qiao-Min, "An efficient authentication and access control scheme for perception layer of Internet of Things," *Appl. Math. Inf. Sci.*, vol. 8, no. 4, pp. 1617–1624, Jul. 2014.
[26] M. Gupta, J. Benson, F. Patwa, and R. S. Sandhu, "Secure V2V and V2I communication in intelligent transportation using cloudlets," *CoRR*, vol. abs/2001.04041, pp. 1–12, Sep. 2020.
[27] R. Schuster, V. Shmatikov, and E. Tromer, "Situational access control in the Internet of Things," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, D. Lie, M. Mannan, M. Backes, and X. Wang, Eds., Toronto, ON, Canada, Oct. 2018, pp. 1056–1073.
[28] Y. J. Jia, Q. A. Chen, S. Wang, A. Rahmati, E. Fernandes, Z. M. Mao, and A. Prakash, "ContexIoT: Towards providing contextual integrity to appified IoT platforms," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, San Diego, CA, USA, 2017, doi: 10.14722/ndss.2017.23051.
[29] A. Ouaddah, H. Mousannif, A. A. Elkalam, and A. A. Ouahman, "Access control in the Internet of Things: Big challenges and new opportunities," *Comput. Netw.*, vol. 112, pp. 237–262, Jan. 2017.
[30] S. Bhatt and R. Sandhu, "Convergent access control to enable secure smart communities," in *Proc. 2nd IEEE Int. Conf. Trust, Privacy Secur. Intell. Syst. Appl. (TPS-ISA)*, Atlanta, GA, USA, Oct. 2020, pp. 148–156.
[31] M. Al-Zubaidie, Z. Zhang, and J. Zhang, "PAX: Using pseudonymization and anonymization to protect Patients' identities and data in the healthcare system," *Int. J. Environ. Res. Public Health*, vol. 16, no. 9, p. 1490, Apr. 2019.
[32] P. N. Mahalle, B. Anggorojati, N. R. Prasad, and R. Prasad, "Identity authentication and capability based access control (IACAC) for the Internet of Things," *J. Cyber Secur. Mobility*, vol. 1, no. 4, pp. 309–348, Feb. 2013.
[33] S. Bhatt, F. Patwa, and R. S. Sandhu, "Access control model for AWS Internet of Things," in *Proc. 11th Int. Conf. Network Syst. Secur. (NSS)*, Lecture Notes in Computer Science, vol. 10394, Z. Yan, R. Molva, W. Mazurczyk, and R. Kantola, Eds., Helsinki, Finland. Cham, Switzerland: Springer, Aug. 2017, pp. 721–736.
[34] M. Gupta and R. Sandhu, "Towards activity-centric access control for smart collaborative ecosystems," in *Proc. 26th ACM Symp. Access Control Models Technol.*, J. Lobo, R. D. Pietro, O. Chowdhury, and H. Hu, Eds., Madrid, Spain, Jun. 2021, pp. 155–164.
[35] S. Bhatt, F. Patwa, and R. Sandhu, "An access control framework for cloud-enabled wearable Internet of Things," in *Proc. IEEE 3rd Int. Conf. Collaboration Internet Comput. (CIC)*, San Jose, CA, USA, Oct. 2017, pp. 328–338.
[36] D. Gupta, S. Bhatt, M. Gupta, O. Kayode, and A. S. Tosun, "Access control model for Google cloud IoT," in *Proc. IEEE 6th Int. Conf. Big Data Secur. Cloud (BigDataSecurity), Int. Conf. High Perform. Smart Comput., (HPSC), IEEE Int. Conf. Intell. Data Secur. (IDS)*, Baltimore, MD, USA, May 2020, pp. 198–208.
[37] M. Gupta and R. Sandhu, "Authorization framework for secure cloud assisted connected cars and vehicular Internet of Things," in *Proc. 23nd ACM Symp. Access Control Models Technol.*, E. Bertino, D. Lin, and J. Lobo, Eds., Indianapolis, IN, USA, Jun. 2018, pp. 193–204.
[38] M. Gupta, J. Benson, F. Patwa, and R. Sandhu, "Dynamic groups and attribute-based access control for next-generation smart cars," in *Proc. 9th ACM Conf. Data Appl. Secur. Privacy*, G. Ahn, B. M. Thuraisingham, M. Kantarcioglu, and R. Krishnan, Eds., Richardson, TX, USA, Mar. 2019, pp. 61–72.
[39] M. Gupta, M. Abdelsalam, S. Khorsandroo, and S. Mittal, "Security and privacy in smart farming: Challenges and opportunities," *IEEE Access*, vol. 8, pp. 34564–34584, 2020.
[40] S. Sontowski, M. Gupta, S. S. L. Chukkapalli, M. Abdelsalam, S. Mittal, A. Joshi, and R. Sandhu, "Cyber attacks on smart farming infrastructure," in *Proc. IEEE 6th Int. Conf. Collaboration Internet Comput. (CIC)*, Atlanta, GA, USA, Dec. 2020, pp. 135–143.
[41] M. Gupta, F. M. Awaysheh, J. Benson, M. Alazab, F. Patwa, and R. Sandhu, "An attribute-based access control for cloud enabled industrial smart vehicles," *IEEE Trans. Ind. Informat.*, vol. 17, no. 6, pp. 4288–4297, Jun. 2021.
[42] J. Park, D. Nguyen, and R. Sandhu, "A provenance-based access control model," in *Proc. 10th Annu. Int. Conf. Privacy, Secur. Trust*, N. Cuppens-Boulahia, P. Fong, J. García-Alfaro, S. Marsh, and J. Steghöfer, Eds., Paris, France, Jul. 2012, pp. 137–144.
[43] L. Sun, J. Park, and R. Sandhu, "Engineering access control policies for provenance-aware systems," in *Proc. 3rd ACM Conf. Data Appl. Secur. Privacy (CODASPY)*, New York, NY, USA, 2013, pp. 285–292.
[44] D. Nguyen, J. Park, and R. Sandhu, "Adopting provenance-based access control in OpenStack cloud IaaS," in *Network and System Security*, M. H. Au, B. Carminati, and C.-C. J. Kuo, Eds. Cham, Switzerland: Springer, 2014, pp. 15–27.
[45] L. Sun, J. Park, D. Nguyen, and R. Sandhu, "A provenance-aware access control framework with typed provenance," *IEEE Trans. Dependable Secure Comput.*, vol. 13, no. 4, pp. 411–423, Jul. 2016.
[46] J. Park and R. S. Sandhu, "The UCON_{ABC} usage control model," *ACM Trans. Inf. Syst. Secur.*, vol. 7, no. 1, pp. 128–174, 2004.
[47] A. C. Squicciarini, M. Shehab, and F. Paci, "Collective privacy management in social networks," in *Proc. 18th Int. Conf. World Wide Web (WWW)*, J. Quemada, G. León, Y. S. Maarek, and W. Nejdl, Eds., Madrid, Spain, 2009, pp. 521–530.
[48] R. Sandhu, V. Bhamidipati, and Q. Munawer, "The ARBAC97 model for role-based administration of roles," *ACM Trans. Inf. Syst. Secur.*, vol. 2, no. 1, pp. 105–135, Feb. 1999.
[49] J. Park and R. Sandhu, "Towards usage control models: Beyond traditional access control," in *Proc. 7th ACM Symp. Access Control Models Technol. (SACMAT)*, Monterey, CA, USA, 2002, pp. 57–64.
[50] M. Fernández and B. Thuraisingham, "A category-based model for ABAC," in *Proc. 3rd ACM Workshop Attribute-Based Access Control*, New York, NY, USA, Mar. 2018, pp. 32–34.
[51] M. Fernández, I. Mackie, and B. Thuraisingham, "Specification and analysis of ABAC policies via the category-based metamodel," in *Proc. 9th ACM Conf. Data Appl. Secur. Privacy*, G. Ahn, B. M. Thuraisingham, M. Kantarcioglu, and R. Krishnan, Eds., Richardson, TX, USA, Mar. 2019, pp. 173–184.
[52] A. J. Lee, M. Winslett, J. Basney, and V. Welch, "The traust authorization service," *ACM Trans. Inf. Syst. Secur.*, vol. 11, no. 1, pp. 1–33, Feb. 2008.
[53] W. H. Winsborough, K. E. Seamons, and V. E. Jones, "Automated trust negotiation," in *Proc. DARPA Inf. Survivability Conf. Expo. (DISCEX)*, vol. 1, 2000, pp. 88–102.
[54] L. Bauer, L. F. Cranor, R. W. Reeder, M. K. Reiter, and K. Vaniea, "Real life challenges in access-control management," in *Proc. Conf. Hum. Factors Comput. Syst. (SIGCHI)*, D. R. Olsen, Jr., R. B. Arthur, K. Hinckley, M. R. Morris, S. E. Hudson, and S. Greenberg, Eds., Boston, MA, USA, Apr. 2009, pp. 899–908.
[55] B. Hayes, "Cloud computing," *Commun. ACM*, vol. 51, no. 7, pp. 9–11, Jul. 2008.
[56] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A Berkeley view of cloud computing," Dept. EECS, Univ. California Berkeley, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2009-28, 2009.
[57] T. Jaeger and J. Schiffman, "Outlook: Cloudy with a chance of security challenges and improvements," *IEEE Secur. Privacy Mag.*, vol. 8, no. 1, pp. 77–80, Jan. 2010.
[58] C. E. Landwehr, "A national goal for cyberspace: Create an open, accountable internet," *IEEE Security Privacy*, vol. 7, no. 3, pp. 3–4, May 2009.

JAEHONG PARK (Member, IEEE) received the B.B.A. degree in information management from Dongguk University, Seoul, South Korea, the M.S. degree in information systems technologies from George Washington University, and the Ph.D. degree in information technology from George Mason University. He is currently an Associate Professor of information systems, supply chain and analytics with the University of Alabama in Huntsville. Previously, he was a Research Associate Professor at the Institute for Cyber Security, The University of Texas at San Antonio. He pioneered the area of usage control (UCON), and his seminal works on UCON models have been well-regarded and received over 2000 Google Scholar citations. He published research papers in leading cybersecurity journals and conference proceedings and his research papers were cited over 4000 times according to the Google Scholar Citation Index. His research interests include data and application security and privacy, access and usage control, secure collaboration, cloud computing security, smart and connected system security, the IoT security, secure CPS, secure provenance, and social computing. He is a member of ACM and ACM SIGSAC. He has served as a General Chair, a Program Chair, and a Program Committee Member for the ACM Conference on Data and Application Security and Privacy (CODASPY). He has also served as program committee members for numerous conferences and workshops.

RAVI SANDHU (Life Fellow, IEEE) received the B.Tech. degree from IIT Bombay, the M.Tech. degree from IIT Delhi, and the M.S. and Ph.D. degrees from Rutgers University. He is a fellow of ACM and AAAS. He worked with the Faculty at George Mason University, from 1989 to 2007, and The Ohio State University, from 1982 to 1989. He is currently a Professor of computer science and the Executive Director of the Institute for Cyber Security and a Lead PI of the NSF Center for Security and Privacy Enhanced Cloud Computing, The University of Texas at San Antonio, where he holds the Lutcher Brown Endowed Chair in Cyber Security. He has received numerous awards from IEEE, ACM, NSA, NIST, and IFIP, including the 2018 IEEE Innovation in Societal Infrastructure Award for seminal work on role-based access control (RBAC). A prolific and highly-cited author, his research has been funded by NSF, NSA, NIST, DARPA, AFOSR, ONR, AFRL, ARO, and private industry. His seminal articles on role-based access control established it as the dominant form of access control in practical systems. His numerous other models and mechanisms have also had considerable real-world impact. He has consulted for leading industry and government organizations, and has lectured all over the world. He is an inventor on 31 security technology patents and has accumulated over 45,000 Google Scholar citations for his articles. At UTSA, his team seeks to pursue world-leading research in both the scientific foundations of cyber security and their applications in diverse 21st century cyber technology domains, including cloud computing, the Internet of Things, autonomous vehicles, big data, and blockchain. Particular focus is on foundations and technology of attribute-based access control (ABAC) as a successor to RBAC in these contexts. He was the Chairman of ACM SIGSAC, and founded the ACM Conference on Computer and Communications Security, the ACM Symposium on Access Control Models and Technologies and the ACM Conference on Data and Application Security and Privacy. He has served as a general chair, a steering committee chair, a program chair, and a committee member for numerous security conferences. He served as the Editor-in-Chief for the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, and previously as the Founding Editor-in-Chief of *ACM Transactions on Information and System Security*. More information can be found at: www.profsandhu.com.

MAANAK GUPTA (Member, IEEE) received the B.Tech. degree in computer science and engineering, India, the M.S. degree in information systems from Northeastern University, Boston, MA, USA, and the M.S. and Ph.D. degrees in computer science from The University of Texas at San Antonio (UTSA). He worked as a Postdoctoral Fellow at the Institute for Cyber Security (ICS), UTSA. He is currently an Assistant Professor of computer science at Tennessee Technological University, Cookeville, TN, USA. He worked in developing novel security mechanisms, models and architectures for next generation smart cars, smart cities, intelligent transportation systems, and smart farming. He is also interested in machine learning-based malware analysis and AI assisted cyber security solutions. His research has been funded by the U.S. National Science Foundation (NSF), NASA, U.S. Department of Defense (DoD), and private industry. His research interests include security and privacy in cyber space focused in studying foundational aspects of access control and their application in technologies including cyber physical systems, cloud computing, the IoT, and big data.

SMRITI BHATT received the Ph.D. degree in computer science from The University of Texas at San Antonio. She did her Ph.D. research at the Institute for Cyber Security (ICS) and the Center for Security and Privacy Enhanced Cloud Computing (C-SPECC). She is currently an Assistant Professor of computer and information technology with Purdue University. Her current research projects focus on developing secure access control and communication control models for cloud-enabled IoT architecture applicable to various IoT domains, such as smart home, smart health, and the wearable IoT. Her research work also expands into deep learning for the IoT security with applications in access control and anomaly detection. Her research interests include cyber security, mainly focused on access control and communication control models, and security and privacy in cloud computing and the Internet of Things (IoT). She has been actively publishing her work on well-regarded conferences and journals in the field, and also continually serves as an Expert Reviewer for journals, such as the IEEE TRANSACTIONS ON CLOUD, IEEE ACCESS, and *Transactions on Dependable and Secure Computing*; and a technical program committee on several conferences and workshops.