Context-Aware Operation-Based Access Control for Internet of Things Applications

Didem Genç
Computer Engineering
İzmir Institute of Technology
Gülbahçe Urla,İzmir
Email: didemgenc@iyte.edu.tr

Emrah Tomur
Ericsson Research
Ericsson
İstanbul
Email: emrah.tomur@ericsson.com

Yusuf Murat Erten
Computer Engineering
İzmir Institute of Technology
Gülbahçe Urla,İzmir
Email: muraterten@iyte.edu.tr

Abstract—**Recently, interest of both the academic and industrial world in Internet of Things (IoT) has been increasing and this trend requires development of new security approaches addressing potential weaknesses in this domain. Despite the presence of many studies directed towards security of IoT applications, they are mostly adoption of current methods to IoT scenarios. Yet, IoT applications are comprised of various kinds of different entities including computers, processes, people and services. Therefore, it is inadequate to detect malicious attempts by using conventional security methods, which apply fixed security policies and do not take interaction of things, that is context information, into account. In this study, by considering new security requirements of next generation IoT applications, we propose a fine-grained, dynamic and easily manageable access control model, which is called context-aware operation-based access control.**

I. INTRODUCTION

Advances, particularly in sensor, actuator and network technologies pave the way for a new paradigm that is called Internet of Things (IoT), which encompasses ubiquitous/pervasive computing. Ubiquitous computing refers to commonly used, inch-foot-yard scale wireless computing devices like tablet computers, smart watches etc. which mostly interact just with their owners. Their use can often be described as one-to-many type of interaction. For example, in a smart home, the refrigerator can send a notification to the householders regarding lack of eggs. The interaction is only between the refrigerator's sensor and the householders, which refers to one-to-many type of interaction. However, IoT has a more extensive scope involving communication among various entities, hence called many-to-many type of interaction. For example, the refrigerator (in the above scenario) can make an order from the market and notifies the householders to pick up the given order from the market at a suitable time determined by the householders' schedule, traffic conditions and route to home. As can be seen, the refrigerator interacts with many services such as the market e-commerce service, navigation service, and calendar service.

As the scenario indicates, an essential part of the IoT phenomenon is being ”context aware”, which implies the ability of changing and adapting the behaviours of smart devices according to the context. Nowadays, different domains are converted into smarter environments by adapting and integrating context usage in their applications, such as smart homes, smart vehicles, smart healthcare, smart cities, and smart farms. One of the most significant concerns about these applications is how they comply with the privacy and security requirements.

The challenge of securing the IoT domain arises from its main characteristics, namely that it has unpredictable and spontaneous interaction among the various types and the vast amount of things. Therefore, the usage of conventional security approaches, not considering context information, and that using predefined security policies, which need to be written for each object and each subject separately, are inadequate to ensure security of the new generation of IoT applications. New security approaches addressing this field specifically are needed.

Although Role Based Access Control (RBAC) [1] is the most prevalently used access control model, it is insufficient to use it directly in the IoT domain due to role-explosion and permission-explosion drawbacks in implementation. Also, the role engineering process can be cumbersome for such an environment, which includes dynamic and complex interaction of things. Attribute Based Access Control (ABAC) [2] is another efficient access control model. However, this model is also inadequate for IoT when used alone since it is not easy to administrate. Related to this, NIST (National Institute of Standards and Technology) had a recommendation regarding merging the best features of RBAC and ABAC to achieve effective, dynamic, granular and flexible access control. In [3], they discussed the limitations of both RBAC and ABAC, and to overcome these, three main approaches, namely, dynamic roles, attribute centric and role centric approaches, were introduced to integrate roles with attributes. A remarkable decrease in the number of security policies can be achieved by using static attributes (subject's title, skill etc.) in creation of roles and dynamic attributes (time, location, temperature etc.) to form the rules. This approach, aiming to combine rule-based and attribute-based methods to utilize the advantages of

This work was funded by The Scientific and Technological Research Council of Turkey, under 1515 Frontier R&D Laboratories Support Program with project no: 5169902.

978-1-7281-1244-2/19/$31.00 ©2019 IEEE

both approaches, has the potential to meet the requirements of future IoT environments having dynamic, many-to-many interactions.

In this paper, we propose an access control method which has both context-aware features and also integrates RBAC and ABAC. Therefore, our proposed method is a hybrid of the context-aware role-based access control (CA-RBAC) [4] with attribute based access control (ABAC) method. We call this new model CA-OBAC: Context Aware Operation Based Access Control. It makes use of the efficiency of context awareness for dynamic and fine-grained access control and the easy administration property of RBAC's role approach. Additionally, ABAC's attribute usage approach is utilized to reduce the complexity that arises from the presence of vast amount of objects and subjects.

The rest of the paper is organized as follows. In Section II related studies are examined. Section III presents the methodology of the proposed access control model. Section IV presents a sample scenario and the example rule set for this scenario under the proposed model. In section V, the proposed model is compared with the CA-RBAC and ABAC models regarding the number of required policy rules under worst case. Finally, we present the conclusion in section VI.

II. RELATED WORK

There are two main approaches that we follow to obtain a flexible, fine-grained, usable, and scalable access control model. One of them is the utilization of context information in access control, and the other is to merge advantages of RBAC and ABAC. In this section, existing work of these two approaches will be summarized.

Use of context in access control ensures a fine-grained and dynamic access control mechanism. Therefore, many researchers have proposed different works that integrate context awareness with access control. However, among these studies, it appears that there are only a few studies addressing next generation IoT scenarios considering complicated interaction of things. The studies [5], [6], which address IoT applications, state that merely RBAC or ABAC usage is inadequate for IoT domain.

Studies in [7], [8] propose distinct role-based access control models with context enhancements, but they all addressed ubiquitous computing environments with one-to-many interactions. Also, the study conducted by Abdella et al. [4] uses a combination of context awareness and RBAC for providing security. However, their study only addresses Android-based mobile applications.

The study by Bai et al. [9], which aimed to develop access control for many-to-many type of interactions for Web of Things applications, is named ConUCON security module. ConUCON makes use of context awareness to provide security and privacy in usage control. The main difference of this work is that they build the system on top of the UCON (Usage Control) [10] model instead of RBAC. However, since it does not apply any grouping of entities, adoption of a UCON based model in ConUCON resulted in overly complicated security policy rules.

The first initiative proposing to integrate ABAC and RBAC as recommended by NIST is [11] by Jin et al. They call their model Role Attribute Based Access Control: RABAC, and this model is good at addressing the role-explosion problem. Also, the studies [12], [13], and [14] propose distinct models that extends properties of ABAC and RBAC by combining role and attribute concepts in a role-centric manner.

As can be seen from the reviewed papers, attribute-based (role-centric) and context-aware RBAC approaches are parallel in terms of limiting permissions based on related attributes or contexts. However, none of these works utilizing context-awareness, RBAC or ABAC in the literature are not designed for IoT and, hence, do not consider specific requirements, which are discussed earlier, of IoT applications.

III. METHODOLOGY

A. Formal Definitions

Operations (Op): Operations are defined actions that are allowed to perform on an object in a system. Subject attributes are grouped under specific operations.

Subjects(S): Subjects are the entities that have access rights for objects under specific situations. In an IoT environment; users, services, processes, applications, and devices can be the subjects of the system.

Subject Attributes (SA): Subject attributes capture the properties of each subject. Each subject should have at least one subject attribute. There are many-to-many mappings between SAs and subjects, which means a subject attribute can be assigned to many subjects and a subject can have more than one SA.

Objects (O): Objects are the resources that are protected by the security policies.

Object Attributes(OA): Object attributes are assumed to be given by the manufacturer or administrator of the system, and it defines the properties of the object. Each object should have at least one object attribute. There are many-to-many relationships between objects and OAs like the mapping between subject and SAs.

Context Expression (Con): In this model, the constraints limiting access to an object is called a context since the restriction is based on context information, which contains subject, object or environmental attributes. Context information is stored in a database in the following format:

Con_i=<*ContextName, Operator, Value*> where $Con_i \in Con$. For comparison operators; equal (=), not equal ($\neq$), greater than (>), less than (<), greater than and equal ($\geq$), less than and equal ($\leq$), not ($\neg$) can be used.

Context Rule (CR): Context rules are the expressions that are evaluated to decide whether access to the associated object will be allowed or denied. CR is composed of two terms; context and action. Action can be allow or deny. CR is represented as follows:

CR_i=<*(ContextName, Operator, Value), Action*> where $CR_i \in CR$.

Multiple Context Rules (MCR): For some of the cases, access to an object can be related to many contexts which need to be checked, so many context rules are required. Multiple context rules enable writing combined context rules by using logical operators like and (∧), or (∨). MCR is stored in the form of:

MCR_i=<((CR_1 ∧ CR_2) ∨ CR_3), *Action*> where $MCR_i \in MCR$ and $CR_j \in CR$.

B. Access Control Model

The role component of RBAC enables grouping users and permissions so that an administrator can easily review the permissions of a specific user. However, IoT environments have dynamic interactions of vast amounts of things, and it is not possible to define the interactions among the things exactly and a priori. Therefore, it is inefficient to try to predefine exact access control policies for each entity in the environment. Besides, forcing the assignment of a role to each subject will lead to the creation of an excessive number of roles due to the vast amount and different types of subjects. Considering the existence of the role-explosion problem of RBAC, it is obvious that a radical solution is needed to remove these problems for such a complicated environment. As a solution, we propose to; use operations on their own in access control rather than grouping operations and objects as permissions as done in classical RBAC. In other words, our access policy rules refer to individual operations such as read, write, turn on/off, etc. instead of referring to permissions like read file and turn on light. What is more, we group subject attributes under operations as if subject attributes are subjects and operations are roles in RBAC. The reasoning behind this is that in IoT applications, the number of operations that can be executed is generally limited and much less compared to the number of subjects, which can be any ”thing”. In this way, it is aimed to prevent the role-explosion problem, which is highly likely to occur in IoT environments. Since it is hard and inefficient to define a relationship between each subject and operation, subject and object attributes are used. Therefore, instead of subjects, subject attributes are assigned to operations to provide flexibility and dynamism. Also, object attributes are used to group the objects in order to prevent the role - permission explosion. Figure 1 illustrates the relationship between entities in our proposed model.

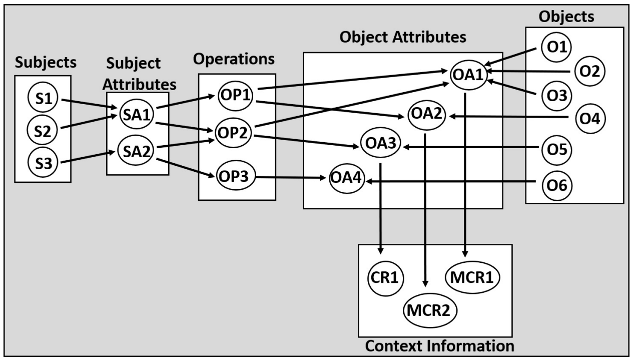

Fig. 1. CA-OBAC Access Control Model Schematic

As it is seen in the Figure 1, S1, S2 and S3 are the subjects. S1 and S2 have SA1 subject attribute and S3 is assigned to SA2 subject attribute. Subject attributes have authorized operations. SA1 can perform OP1 and OP2, while SA2 is allowed for OP2 and OP3. Objects have following attributes; O1, O2 and O3 are labelled with OA1, O4 with OA2, O5 with OA3, and O6 OA4 respectively. Object attributes are assigned to operations, which means that an operation can be performed only on the object attribute(s) (thus, on the object(s)) assigned to it. Finally, the context information is assigned to object attributes.

The architectural design of the proposed model is shown in Figure 2, and the component descriptions are given below.

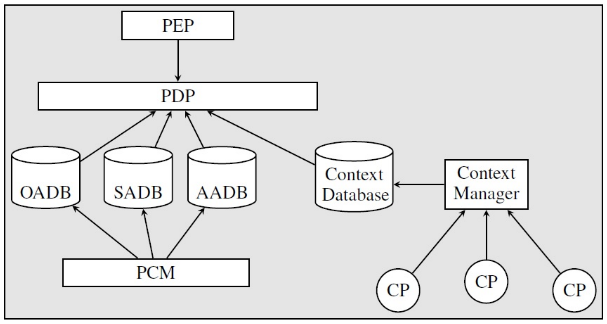

Fig. 2. Access Control Model Architecture

Policy Enforcement Point (PEP): The requests coming from subjects are met by this unit and are forwarded to the PDP unit for evaluation. Reply of the request is also conducted to the subject over PEP.

Policy Decision Point (PDP): The main security decisions are made by this unit and the results are transmitted to the PEP. Making the required checks associated with the coming requests to allow access is the responsibility of PDP unit which performs it by activating the OADB, SADB, AADB, and CDB peripheral units, which are explained below.

Policy Configuration Manager (PCM): This unit facilitates the configuration of the subject attribute to operation, and object attribute to operation assignments by using the PCM interface.

Context Manager: To check whether the related context policies are fulfilled by the subject, context information is acquired by the Context Manager unit through the context providers, and stored in a context database (CDB). Context providers can be different kinds of sensors like temperature, proximity, RFID etc., or the services getting data from Web or social networks etc. The context manager has the responsibility of continuously updating the context information to provide the current context in the CDB. As a result, the PDP unit is able to access these up-to-date context information of related subjects through the CDB in evaluation of access request.

Assignment Databases: The model includes 3 different databases which are object assignment database (OADB), subject assignment database (SADB), and attribute assignment database (AADB). As the names imply, OADB stores the object attribute-operation-authentication-context relation, SADB stores the SA-operation pairs, and AADB is used to store the subjects and objects with their corresponding attributes.

The requests come in the form of:

$Rq < Subject\_id, Object\_id, Operation, AuthType >$

PEP handles the request and forwards it to PDP for evaluation of access. PDP first checks that the operation is defined in SADB. If there is no such an operation defined in the system, it sends a deny reply to the PEP. If it finds the operation, it retrieves the subject attribute list that is assigned to requested operation from SADB. Also, the subject attributes assigned to the requestor subject are retrieved from AADB, and PDP compares these values with the subject attribute list of the operation. If PDP cannot find any subject attributes of the subject within the SA list of the operation, access is denied. Otherwise, according to the requestor subject's authentication type, allowed object attributes, and corresponding context that are assigned to the requested operation are listed from OADB. Requested object's attributes are also retrieved from AADB. If any of the object attributes do not match with the elements of the list, access will be denied. If there is a match, the corresponding context will be checked from CDB. In case the context requirements are satisfied, PDP sends a grant or allow message to PEP. However, if the context is not satisfied, PDP sends deny message. In addition, there can be no context defined for the related object attributes, in which case PDP always sends an allow message. If the requested object has more than one object attribute with different contexts associated with them, the subject should satisfy all the contexts in order to get access. Finally, PEP transmits the reply to the user.

IV. SAMPLE SCENARIO

Integration of context awareness with the RBAC model is proposed by different researchers, but the scope of IoT applications we consider is such that smart things interact in a complex way through the cloud infrastructure or social networks. Therefore, the scenario that we choose in order to validate the efficiency of our proposed model includes many-to-many interactions of entities in different smart environments like smart home and smart car. The complexity of CA-RBAC, ABAC and CA-OBAC models are evaluated using this scenario. Entities of the scenario are given below.

Subjects: Katie (mother), John (father), James (child), Joe (child), Sue (child), Jessica (babysitter), Smart Home Application, Smart Healthcare Application

Subject Attributes: Parent, Child, Babysitter, Healthcare App, Home App

Objects: Smart Door, Oven, Washing Machine, Dish Washer, Camera, Wearable Insulin Pump

Object Attributes: Smart Door, Camera, Household Appliances, Wearable Devices

Environmental Attributes: Authentication (Mobile Device, Biometric), Time (school hours, working hours), Location (inside house, outside house), Distance, Parent's Approval (yes, no), Somebody in front of door (yes, no), Emergency (yes, no)

Each object and subject should have at least one attribute. There are 7 environmental attributes defined to achieve fine-grained access control. The distance attribute is defined to be atomic valued, while the others are defined as set type attributes. Principals can either use biometric or mobile device authentication. Subjects included can be either person or application. In the scope of this paper, context modelling is not handled. Therefore, it is assumed that all contexts in the context database are already stored and continuously updated. The scenario is explained in detailed below:

Smart Door: The smart door can be opened in given cases by predefined subjects via different authentication type usage:

- In all cases, mother and father can open the smart door by using biometric authentication.
- Mother and father can also open the smart door by using their mobile device if their smart car, that is defined in the system, is within 10 meters of the house, and the time is outside working hours.
- If the children are outside the house, in all situations, they can open the smart door by using biometric authentication. However, if they are inside the house, to open the smart door some adult (at least one of the parents, or babysitter) should be inside the house, or emergency situation should be enabled.
- When the school bus is closer than 10 meters to the house, and the time is outside the school hours, the door can be opened by children using their mobile devices.
- A babysitter can open the smart door using only biometric authentication if he/she is outside the house and time is working hours for her. When a babysitter is inside the house and there is somebody in front of the door, parent's approval is required to be able to open the door. However, in the same situation, if nobody is outside the house, parent's approval is not needed to open the door.
- A smart home application can also send request to open the smart door if an emergency situation context is asserted, and the ambulance is less than 10 meters away from the house.

Household Appliances:

- The mother and father can turn the oven, dish washer and washing machine on using their mobile devices when they are not inside the house.
- The children are not allowed to turn any electrical household appliances on or off.
- The babysitter can turn the electrical household appliances on by using his/her mobile device if he/she is inside the house, and the time is within the working hours.
- The mother, father and smart home application can turn the electrical household appliances off, if 30 minutes passed since a request to turn them on has been made, and a parent or babysitter is not inside the house.

Camera:

TABLE I
RULE SET

Operation	Auth.	Object Attribute	Context	Access
OPEN: **Parent** **Babysitter** **Home App** **Child**	Biometric	Smart Door	SA= parent	ALLOW
			SA=child ∧ loc(requestor)=outside house	ALLOW
			SA=child ∧ (loc(requestor)=inside house ∧ loc(parent) ∨ loc(babysitter)=inside house) ∨ context=emergency	ALLOW
			SA=babysitter ∧ loc(requestor)=outside house ∧ time=working hour	ALLOW
			SA=babysitter ∧ loc(requestor)=inside house ∧ sb. in front of door=yes ∨ parent's approval=yes	ALLOW
			SA=babysitter ∧ loc(requestor)=inside house ∧ sb. in front of door=no	ALLOW
	Mobile Device	Smart Door	SA=Home app ∧ distance(ambulance)<10m ∨ emergency=yes	ALLOW
			SA=parent ∧ time>working hour ∧ distance(car)<10m	ALLOW
			SA=child ∧ time>school time ∧ distance(schoolbus)<10m	ALLOW
		Household Appliances	SA=parent ∧ loc(requestor)=¬inside house	ALLOW
			SA=babysitter ∧ loc(requestor)=inside house ∧ time=working hour	ALLOW
DATA READ: **Parent** **Home App.** **Healthcare App.**	Biometric	Camera	SA=parent	ALLOW
	Mobile Device	Camera	SA=Home app. ∧ context=emergency	ALLOW
			SA=parent ∧ context=emergency	ALLOW
		Wearable Devices	SA=Healthcare app.	ALLOW
			SA=Home app. ∧ context=emergency	ALLOW
TURN OFF: **Home App.** **Parent**	Mobile Device	Household Appliances	(time(req-open the household appliances) - time(current))>30 min ∧ (loc(requestor) ∨ loc(babysitter)=¬inside house)	ALLOW

- The mother and father can access the camera using their mobile devices if the emergency situation context is enabled.
- The smart home application can also access the camera in case of emergency.

Wearable Insulin Pump:

- Smart healthcare application can read data from wearable insulin pump at all times since no context is assigned to this object.
- Smart home application can also read data from wearable devices in emergency situations to verify the context.

Under this scenario, access control rule set given in Table I is sufficient to enforce desired access control policy as defined by permitted operations of entities.

V. COMPLEXITY ANALYSIS

In this section, we compare the maximum number of rules required to enforce desired access control policy for CA-OBAC, CA-RBAC and ABAC models.

For CA-RBAC model, the number of created roles, permissions and contexts are given in Table II.

Using these, it is possible to write $(permission \times context)$ distinct permission-context relations for one role. Context refers to the environmental attributes, thus, we have 13 contexts. Permissions consist of object-operation pairs, so $(objects \times operations)$ gives the permission set. When we take into consideration that different roles should have distinct permissions, we get the total number of security policies as $[role \times context \times (object \times operation)]$. For our scenario this means;

$Permissions = (6\ objects) * (3\ operations) = 18$

$SecurityPolicies = (5\ roles) * (18\ permissions) * (13\ context) = 1170$

TABLE II
COMPLEXITY ANALYSIS OF CA-RBAC MODEL

Number of Created Roles	5	Parent, Babysitter, Child, Smart Home App., Smart Healthcare App.
Number of Operations	3	Data Read, Open, Turn off
Number of Objects	6	Smart door, Camera, Washing Machine, Dishwasher, Oven, Insulin Pump
Number of Contexts	13	Authentication Distance Parent's Approval Sb. in front of door Location Time Emergency

ABAC model does not include any group of entities. Thus, each rules that are related with all possible interactions are written separately. Under the scenario of this paper, in order to implement the ABAC model required number of attributes to be created are given in Table III.

TABLE III
COMPLEXITY ANALYSIS OF ABAC MODEL

Number of Subject Attributes	5	Parent, Babysitter, Child, Smart Home App., Smart Healthcare App.
Number of Object Attributes	4	Smart door, Camera, Household Appliances, Wearable Devices
Number of Environmental Attributes	13	Authentication Distance Parent's Approval Sb. in front of door Location Time Emergency
Number of Operations	3	Data Read, Open, Turn off

According to this approach, it is possible to write up to $(OA \times SA \times EA \times Operation)$ distinct rules. Thus, we get the total number of security policies as:

$SecurityPolicies = (4\ OA) * (5\ SA) * (13\ EA) * (3\ Op) = 780$

CA-OBAC model implementation entities are given in Table IV. The difference in terms of the number of contexts stems from that we also use subject attributes as constraints. Therefore, number of contexts is evaluated as $(subject\ attributes + environmental\ attributes)$.

TABLE IV
COMPLEXITY ANALYSIS OF CA-OBAC MODEL

Authentications	2	Biometric, Mobile Device
Number of Object Attributes	4	Smart door, Camera, Household Appliances, Wearable Devices
Number of Contexts	16	Subject Attributes: Parent, Child, Babysitter, Home app., Healthcare app. Distance Parent's Approval Sb. in front of door Location Time Emergency
Number of Operations	3	Data Read, Open, Turn off

According to this approach, it is possible to write $(Operation \times Authentication \times OA \times Context)$ distinct access control rules. Thus, we get the total number of security policies as;

$Number\ of\ Context = (11\ EA) + (5\ SA) = 16$

$Security\ Policies = (3\ Op) * (2\ Auth) * (4\ OA) * (16\ Context) = 384$

Finally, Table V shows the number of maximum possible security policies for each of the models for the same scenario. It is obvious that, the complexity (number of security policy) is nearly reduced by 3 times by using CA-OBAC model. Also, the role-explosion problem of CA-RBAC is removed by using operations instead of roles. Additionally, the policy review in terms of subjects is made quite easily since the operations are assigned allowed subject lists. This model will perform better under more complex scenarios that consist of interactions of different smart environments like smart hospital, smart car, and smart office.

TABLE V
COMPARISON RESULTS REGARDING COMPLEXITY OF ACCESS CONTROL MODELS

	CA-RBAC	**ABAC**	**CA-OBAC**
Number of Roles	5	-	3
Number of Security Policies	1170	780	384

VI. CONCLUSION

In this paper, we propose a new access control model that is designed particularly for future IoT applications. Requirements of IoT domain and the disadvantages of currently used access control models are considered in the design of the proposed model. Therefore, by augmenting ABAC (Attribute based access control) with context-awareness and also with the advantageous features of RBAC (Role based access control), a novel access control mechanism is proposed. The features of the proposed model are listed below:

Operation-Based: The main difference of the proposed model compared to RBAC and ABAC is that grouping is done based on the operations instead of roles. The model first checks the requested operation to decide whether the requester has access permission or not. Additionally, since the number of allowed operations is usually less compared to number of subjects, this approach helps to reduce the number of roles required, hence, role - explosion problem can be prevented accordingly.

Reduced Number of Security Policies: Decrease in number of roles, and usage of object attributes also leads to the reduction of maximum number of possible security policies.

REFERENCES

[1] H. A. Weber, "Role-based access control: the nist solution," *SANS institute InfoSec Reading Room*, 2003.

[2] X. Jin, R. Krishnan, and R. Sandhu, "A unified attribute-based access control model covering dac, mac and rbac," in *IFIP Annual Conference on Data and Applications Security and Privacy*, pp. 41–55, Springer, 2012.

[3] D. R. Kuhn, E. J. Coyne, and T. R. Weil, "Adding attributes to role-based access control," *Computer*, vol. 43, no. 6, pp. 79–81, 2010.

[4] J. Abdella, M. Özuysal, and E. Tomur, "CA-ARBAC: privacy preserving using context-aware role-based access control on Android permission system," *Security and Communication Networks*, vol. 9, no. 18, pp. 5977–5995, 2016.

[5] R. Xu, Y. Chen, E. Blasch, and G. Chen, "A federated capability-based access control mechanism for internet of things (iots)," in *Sensors and Systems for Space Applications XI*, vol. 10641, p. 106410U, International Society for Optics and Photonics, 2018.

[6] Y. Zhang and X. Wu, "Access control in internet of things: A survey," *arXiv preprint arXiv:1610.01065*, 2016.

[7] M. J. Covington, W. Long, S. Srinivasan, A. K. Dev, M. Ahamad, and G. D. Abowd, "Securing context-aware applications using environment roles," in *Proceedings of the sixth ACM symposium on Access control models and technologies*, pp. 10–20, ACM, 2001.

[8] D. Kulkarni and A. Tripathi, "Context-aware role-based access control in pervasive computing systems," in *Proceedings of the 13th ACM symposium on Access control models and technologies*, pp. 113–122, ACM, 2008.

[9] G. Bai, L. Yan, L. Gu, Y. Guo, and X. Chen, "Context-aware usage control for web of things," *Security and Communication Networks*, vol. 7, no. 12, pp. 2696–2712, 2014.

[10] J. Park and R. Sandhu, "The ucon abc usage control model," *ACM Transactions on Information and System Security (TISSEC)*, vol. 7, no. 1, pp. 128–174, 2004.

[11] X. Jin, R. Sandhu, and R. Krishnan, "Rabac: role-centric attribute-based access control," in *International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security*, pp. 84–96, Springer, 2012.

[12] P. Barkha and G. Sahani, "Flexible attribute enriched role based access control model," in *Information, Communication, Instrumentation and Control (ICICIC), 2017 International Conference on*, pp. 1–6, IEEE, 2017.

[13] Q. M. Rajpoot, C. D. Jensen, and R. Krishnan, "Attributes enhanced role-based access control model," in *International Conference on Trust and Privacy in Digital Business*, pp. 3–17, Springer, 2015.

[14] B. A. Hasiba, L. Kahloul, and S. Benharzallah, "A new hybrid access control model for multi-domain systems," in *Control, Decision and Information Technologies (CoDIT), 2017 4th International Conference on*, pp. 0766–0771, IEEE, 2017.