CONTEXT AWARE ROLE BASED ACCESS CONTROL MODEL FOR INTERNET OF THINGS APPLICATIONS

A Thesis Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

in Computer Engineering

by
Didem GENÇ

December 2018
İZMİR

We approve the thesis of **Didem GENÇ**

Examining Committee Members:

Assoc. Prof. Dr. Yusuf Murat ERTEN
Department of Computer Engineering, İzmir Institute of Technology

---

Assoc. Prof. Dr. Tolga AYAV
Department of Computer Engineering, İzmir Institute of Technology

---

Assoc. Prof. Dr. Ahmet Tuncay ERCAN
Department of Computer Engineering, Yaşar University

28 December 2018

Assoc. Prof. Dr. Yusuf Murat ERTEN
Supervisor, Department of Computer Engineering
İzmir Institute of Technology

Dr. Emrah TOMUR
Co-Supervisor, BTTO
Yaşar University

Assoc. Prof. Dr. Tolga AYAV
Head of the Department of
Computer Engineering

Prof. Dr. Aysun SOFUOĞLU
Dean of the Graduate School of
Engineering and Sciences

To my lovely husband and son

ACKNOWLEDGMENTS

My deepest gratitude is to my advisor, Assoc. Prof. Dr. Yusuf Murat ERTEN for his excellent guidance, patience, continuous support and encouragement throughout this study.

I would like to thank to Dr. Emrah TOMUR for sharing his valuable knowledge without getting bored. He has been always there to listen and give advice. I am indebted to him for continuous help.

I also would like to send my special thanks to my husband, and parent due to their endless love. Without their motivation and encouragement I wouldn't have finished this thesis. Also, the most special thanks to my little sunshine, lovely son who let me write this thesis.

ABSTRACT

CONTEXT AWARE ROLE BASED ACCESS CONTROL MODEL FOR INTERNET OF THINGS APPLICATIONS

As the day goes on, both the academic and industrial studies related with IoT is increasing with the advance of technology, and this progresses require development of new security approaches aiming this domain. Despite the presence of many studies interested in security of IoT applications, they are just the implementation of currently security methods to IoT scenarios. IoT applications contain the interaction of different kinds of vast amount of thing(computer, process, people, service etc.). Therefore it is going to be inadequate and inefficient to try defining the interaction between these things, and providing security through execution of predefined static security policies. By considering these problems, we can conclude that new generation IoT needs an security mechanism which must offer fine-grained and dynamic access control. In the scope of this thesis, we design a context-aware role based access control model that provides dynamism by using attribute based access control model's attribute function, and fine-granularity with usage of context term, by considering the security needs of IoT domain.

ÖZET

NESNELERİN İNTERNETİ UYGULAMALARI İÇİN BAĞLAM-DUYARLI ROL TABANLI ERİŞİM DENETİMİ

Gün geçtikçe teknolojinin gelişmesi ile beraber, IoT alanındaki hem endüstriyel hem de akademik çalışmalar artmaktadır. Bu durum beraberinde IoT'ye özel, yeni güvenlik yaklaşımlarının geliştirilmesi gerekliliğini getirmektedir. IoT'de güvenliği amaçlayan birçok çalışma yapılmış olmasına rağmen, bunlar yalnızca hali hazırdaki güvenlik methodlarının IoT senaryolarına uygulanmasından oluşmaktadır. Fakat, IoT uygulamaları farklı çeşitlilikte(bilgisayar, süreç, kişi, servis vb.) birçok şeyin etkileşimini içermektedir. Bu sebeple, bu birçok şey arasındaki bütün etkileşimleri tanımlamaya çalışmak ve güvenliği önceden tanımlanmış güvenlik politikaları yardımı ile sağlamaya çalışmak verimsiz ve yetersiz olacaktır. Bu problemler göz önünde bulundurulduğunda, yeni nesil IoT için ihtiyaç duyulan güvenlik mekanizması dinamik ve granüler bir erişim denetimidir. Bu tez kapsamında, yeni nesil IoT uygulamalarının güvenlik ihtiyaçları göz önünde bulundurularak, öznitelik tabanlı erişim denetim modelinin, öznitelik fonksiyonunu kullanarak dinamikliği, bağlam bilgisi kullanımı ile de granülerliği sağlayan, bağlam farkında rol tabanlı bir erişim denetim modeli geliştirilmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

LIST OF ABBREVIATIONS

IoT . . . . . Internet of Things
PDA . . . . . Personal Digital Assistant
ABAC . . . . . Attribute Based Access Control
RBAC . . . . . Role Based Access Control
NIST . . . . . National Institute of Standards and Technology
CA-IRBAC . . . . . Context-Aware IoT Role Based Access Control
UCON . . . . . Usage Control Model
SA . . . . . Subject Attribute
OA . . . . . Object Attribute
CR . . . . . Context Rule
MCR . . . . . Multiple Context Rules
AAT . . . . . Attribute Assignment Table
SAOA . . . . . Subject Attribute Operation Assignment
OAOA . . . . . Object Attribute Operation Assignment
PEP . . . . . Policy Enforcement Point
PDP . . . . . Policy Decision Point
PCM . . . . . Policy Configuration Manager
CDB . . . . . Context Database
CA-RBAC . . . . . Context-aware role based access control

CHAPTER 1

INTRODUCTION

Advances particularly in sensor, actuator and network technologies pave the way for a new paradigm that is called the Internet of Things (IoT), which encompasses ubiquitous/pervasive computing. Ubiquitous computing refers to the commonly used, inch-foot-yard scales wireless computing devices like PDA, smart watch etc. which mostly interact just with their owner. That's why it is called as one-to-many type of interaction. However, IoT has more extensive scope like combining cloud, shroud and crowd technologies, which are called collective computing by (Abowd, 2016). This approach involves communication among various devices, hence called many-to-many type of interaction. An example can help to make the difference clear; in a hospital, the control of a patient's instantaneous pulse changes by using wearable sensors is in the scope of ubiquitous computing. As mentioned before, the scope of IoT is not limited to the given sample scenario. It also involves the integrated analysis of existing medical data and the collected data through cloud infrastructure. Data can be collected from wearable sensors or by analysing the speech or writings in real or virtual environments. As the example indicates, IoT phenomenon comprises of considerable amount of interaction among various types of smart things, which are communication skilled, context-aware, having computing capability devices like smart phones, tablets, autonomous vehicles, home appliances etc. These smart things are the most frequently used devices on a daily basis. That's why it is impossible to stay out of this developing technology, despite the privacy and security concerns.

The essential part of IoT is context awareness which implies the ability of changing and adapting the behaviours of these devices according to the ambient context information. 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, smart farms, assisted living etc. However, the most significant concern about these applications is the privacy and security requirements. Considering these concerns, the term context awareness is used for security issues particularly in access control. Although many of the researchers have conducted studies, which are

given in Chapter 3, related to the usage of context term in access control, they mostly address ubiquitous computing applications with one-to-many interaction, which does not exactly cover future IoT applications with more complicated interaction among things. Therefore, just addition of context usage to access control will not be sufficient to ensure the security requirements of this domain, that arise from complicated interaction of things. New security approaches addressing specifically this field are needed.

In this thesis, we propose a new access control model that is designed for especially IoT field. Requirements of IoT domain and the disadvantages of currently used access control models are considered in the design period of the proposed model. Therefore, by enhancing ABAC (Attribute based access control) with the advantageous features of RBAC (Role based access control) that is most used access control model, a novel access control mechanism is proposed.

1.1. Motivation

Development in technology, specifically the connection of things to internet, made human life easier, however, it brought new challenges regarding security and privacy concerns. Challenge of securing the IoT domain arises from the main characteristic of IoT that it has unpredictable and spontaneous interaction among the various types and the vast amount of things. Therefore, the usage of conventional security approaches, that does not consider context information and use predefined security policies which is written for each object and each subject separately, are inadequate to ensure security of new generation IoT technologies.

Although RBAC is the most prevalently used access control model, it is insufficient to use in IoT domain due to several reasons which are explained in Chapter 2 in detail. ABAC is another efficient access control model, however this model is also inadequate for IoT when used alone. Related to this, NIST (National Institute of Standards and Technology) had an announcement regarding merging the best features of RBAC and ABAC to achieve effective, dynamic, granular and flexible access control by (Kuhn et al., 2010). In this invitation, they discussed the limitations of both access control models, and in relation to that 3 main approaches were introduces to integrate roles with attributes:

1. **Dynamic Roles:**Attributes are used for determining subject's role.

2. **Attribute Centric:** Role name is used as one of the attributes of subject. Actually, this approach is not using the advantages of RBAC since it is not grouping the subject's permissions utilizing roles.

3. **Role Centric:** Attributes are added to RBAC in order to constrain the subject's permissions.

NIST announcement suggested researchers to head towards the role-centric approach by stating the separation of attributes as static (subject's title, skill etc.) and dynamic (time, location, temperature etc.). A remarkable decrease can be achieved in the number of security policies by using static attributes in creation of roles and dynamic attributes to form the rules. As a numerical example, it can be considered to set up an access control with 10 different attributes (assume 4 of them are static and others are dynamic). Such an example access control requires to create 2^{10} rules in ABAC or 2^{10} roles in RBAC for each combination of attributes. However, by combining these two access control models we can decrease numbers to 2^4 roles which are based on static attributes and 2^6 rules that uses dynamic attributes. As it is seen, there is a significant reduction of role and rule numbers that should be handled.

Accordingly, it can be said that, guidance of NIST has motivated us to design an access control model which utilizes the advantages of both RBAC and ABAC for rapidly changing, dynamic environments such as IoT.

1.2. Aim of Thesis and Contributions

Considering the reasons described in section 1.1, the objective of the thesis is to design a new access control model that specifically meets the requirements of future IoT applications. Our method integrates the context aware role based access control with attribute based access control method. We call this model as CA-IRBAC; Context-Aware IoT Role Based Access Control. It makes use of the efficiency of context awareness for dynamic and fine-grained access mechanism, and the easy administration property of RBAC's role approach. Additionally, ABAC's attribute usage approach is used to reduce the complexity that arises from the presence of vast amount of objects and subjects.

Main contributions of the thesis are listed below:

- We propose a new access control model, that integrates the best features of RBAC

and ABAC, in line with NIST announcement by considering IoT environment requirements.

- RBAC's role-explosion problem is handled by grouping the subjects according to operations that they are allowed to perform.
- RBAC's role-permission explosion problem is also handled through attribute assignment to objects.
- Being administratively easy of ABAC is provided by extending this model with the addition of grouping of subjects according to operations.

1.3. Thesis' Outline

The rest of the thesis is organized as follows. In Chapter 2, firstly the environment characteristics of the ubiquitous and collective computing are given in sections 2.1 and 2.2. After that, IoT domain is explained briefly in section 2.3. Main terms of context-awareness and context are described in subsections 2.1.1 and 2.1.1.1. Lastly, basic models of RBAC and ABAC are explained under section 2.4 as background information. Chapter 3 summarizes the related work in 2 aspects. Access control models which utilize context term is examined under section 3.1, and the combined models of ABAC and RBAC are given in section 3.2 respectively. Chapter 4 presents the methodology of proposed CA-IRBAC model by giving the formal model and working principle. Afterwards, in section 4.6 on a basic example scenario, the proposed model is explained. Chapter 5 is divided into 2 sections. In section 5.1, a sample scenario that the proposed model is implemented on is given, and the comparison of complexity analysis of CA-RBAC, CA-IRBAC and ABAC models are shown in section 5.2. Finally, we conclude the thesis with the advantages of designed model and discuss the future work in Chapter 6.

CHAPTER 2

BACKGROUND

In this chapter, the main ideas that the thesis is based on are explained separately.

2.1. Ubiquitous Computing

Mark Weiser has introduced the term "Ubiquitous Computing" as the 3rd generation of computing in 1991 (Weiser, 1991). According to Weiser, there was a need to create a new relationship between people and computers. This relationship should contain continuous interaction between hundreds of wirelessly connected computers and people. The objective was to develop a technology that is invisible to users and they use these computing devices unconsciously to accomplish everyday tasks. Therefore, to achieve this goal, the size and the shape of computing devices are required to vary, and they defined inch-foot-yard scale devices which are called as tabs, pads and boards respectively. This type of communication between people and inch-foot-yard scales device identifies one-to-many interaction.

2.1.1. Context-awareness

Context-awareness concept has became more popular with the introduction of ubiquitous computing term, and it is firstly put forward by (Schilit et al., 1994). They defined it as the application that can adapt themselves to context. Unfortunately, according to this definition, such a scenario that the applications, which just displays context of the user but not adjusts its behaviour accordingly can not be considered as context-aware although they are in the scope of context-aware computing.

Afterwards many researchers offered different definitions, but also those definitions are too specific. Most general definition that consists of both using context and adapting to context is asserted by (Abowd et al., 1999): ***"the system is context-aware if it uses context to provide relevant information and/or services to user, where rele-***

vancy depends on the user's task (Abowd et al., 1999)". Context-awareness notion is the essential part of IoT domain.

2.1.1.1. Context

Several researchers have defined the context notion, however, the most comprehensive and accepted explanation is also introduced by Abowd et al. in 1999. According to them, ***"context is any information that can be used to characterize the state of an entity. An entity can be a person, object or place that is considered relevant to the interaction between a user and an application, including the user and application itself (Abowd et al., 1999)"***. It means that any information which is acquired by wearable or environmental sensors, or extracted through cloud or social media data can be used as context if that information is efficient to characterize your system/environment. Location, brightness, temperature, blood pressure, time can be enumerated as an example to most used context types in applications.

Studies that use context information in their applications are divided into two for the purpose of context usage. Some of them are aimed to facilitate the human life as in the example of an application that adjusts the contrast of the smart phone or tablet's screen according to the brightness of environment. The other context usage objective is to ensure security. For instance, the authentication mechanism that is currently used by the user to withdraw money can be changed in the case of changing time or location they usually withdraw money, or an application can alter its encryption mechanism used for communication according to network trust level. This thesis is also an example of the use of context information in order to provide security.

2.2. Collective Computing

In 2016 Gregory D. Abowd has come up with a new notion "Collective Computing" which is called as 4th generation of computing (Abowd, 2016). This technology has moved ubiquitous computing forward. Abowd realized that ubiquitous computing did not support to enhance the interaction among people, since it just considers the interaction between device and its user (one-to-many). In reality, many people are interacting with

another through many devices, and there is a many-to-many relationship between them. Therefore, collective computing arose to fill this gap by using cloud computing, crowd sourcing, and shroud which is a new notion defined by Abowd that means the digital technology that adapts the physical properties of people to the digital domain. According to this, many things should communicate with other through these cloud-crowd-shroud technologies to share their data. This kind of new generation of computing which includes billion numbers of heterogeneous device interactions is the basis of IoT environment.

2.3. IoT Environment

Kevin Ashton is coined the term "Internet of Things" as the title of his presentation that he made at Procter & Gamble (P&G) in 1999. And now, IoT is the today's one of the hot topics, and buzzword that is frequently used in the world. This notion basically means that any device with an on/off switch can be connected to, and controlled through the Internet (Morgan, 2014). In contrast to Weiser's Ubiquitous computing phenomena, this technology includes the interactions between things-things and people-people. According to a survey, which is held by Gartner, there will be 26 billion connected devices in 2020. This means that IoT domain hosts vast amount of many-to-many things interaction that brings in many challenges. Most important ones of these challenges are the security and privacy concerns. In the area of security, the use of access control model plays an important role.

2.4. Related Access Control Models

Access control is the authorization mechanism for limiting the access based on predefined rules/permissions or security policies. Although there are various types of access control models designed for the same purpose that offer different advantages, in the scope of this thesis we exploited 2 of them which are "Role based access control model" and "Attribute based access control model". The basic definitions and structures of both models are given in the following subsections.

2.4.1. RBAC: Role Based Access Control

Role Based Access Control has widespread use in industry, business, government etc. After significant studies, NIST has proposed a nearly completed RBAC model in 2003 under a document named NIST solution (Weber, 2003). In this document, 4 versions of RBAC are explained in detail. The variations originated from used features like; hierarchy and constraints. The common aspect of these different versions is that the base model depends on 3 core elements in its nature; users, roles and permissions, and this version is called as Flat RBAC (Sandhu et al., 2000). Users, roles and permissions, and their relationships are explained separately below.

Users: The users are the entities who can gain access on any object in the system. A user can be a person or non-person like computer or process.

Roles: Function of the role can be considered as grouping operation of users who have same permissions. On the other hand, it also enables grouping the permissions. Therefore, a role which is the main element of RBAC, is a bridge between both the group of users and their permissions. Title, department or job can be an example to roles.

Permissions: A permission is composed of operations and objects. An object can be a file, document or anything that need to be protected. Operation is an action that is allowed to be performed on a specific object like read, write, update etc. The diagram that illustrates the relationship between these elements is given below in Figure 2.1.

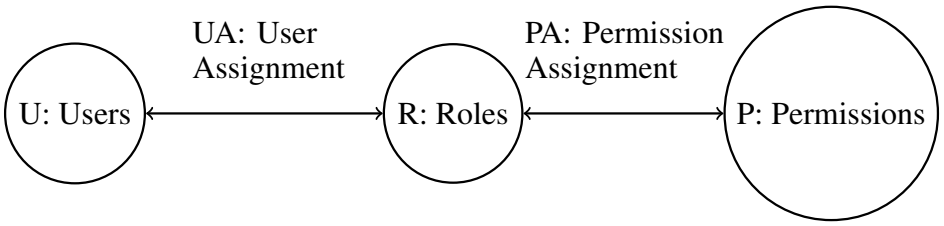

Figure 2.1. Flat Role Based Access Control(Sandhu et al., 2000).

As Figure 2.1 illustrates, in flat RBAC the users are assigned to roles, and roles have permissions assigned to them. The relationship between users and roles is called as user-to-role assignment (UA: user assignment). Also, there exists a similar connection between roles and permissions that is called as role-to-permission (PA: permission assignment) assignment. There is a many to many relation between both 'users and roles', and

'roles and permissions', which means a user can be assigned to many roles, and a role can include many users, and similarly, a role can have many permissions and a permission can be assigned to many roles. Permissions are granted to the users according to their roles. Therefore, when a user request a permission, access control model first checks its roles, then the connected permissions. If the user does not have any role, or the requested permission is not included in the list of permissions relevant to the roles assigned to the user, the access request is denied.

In summary, RBAC is the procedure of assigning a role to each user and having its own permissions for each role. Benefit of this method is to enable administratively easy review of security policies regarding users. Therefore, it offers really simple and understandable access control, and this has made this method user friendly and efficient.

Advantage of RBAC is given below:

- It has the ease of management for the administrator. It is clear to manage the permissions of a specific user.

Limitations of RBAC are given below:

- **Role Explosion Problem**: In large organizations or domains, it is needed to create many roles to be able to represent the users. This large number of roles causes a high storage requirements and high computing requirements leading to high costs for the operation of the security system. Furthermore, this large number of roles make the management of the system difficult.
- **Role Permission Explosion Problem**: There can be many objects that need to be protected. This large number of objects requires individual permissions for each of them separately, thus causing to write many permissions regarding a role.
- **Role Engineering Requirement**: Extracting roles and designing a model with access control policies is a costly and time-consuming task.

2.4.2. ABAC: Attribute Based Access Control

Attribute based access control (ABAC) is a relatively new model which is more flexible and fine-grained than RBAC. In this method, the users have attributes, and the

attributes are used to decide granting access instead of roles (Jin et al., 2012). Attributes are any information that is used to identify a specific person or object. There are 3 kinds of attributes:

Subject Attributes: Title, job, data of birth, home city can help to identify a person or group of people. Therefore, these are given as an example of subject attributes.

Object Attributes: Like subject attributes, the object attributes are used to define an object. The object status, type, name can be assigned as an attribute to the objects.

Environmental Attributes: Environmental attributes defines the frequently changing information like time, location, ip address etc. In access decisions, consideration of these environmental attributes enable a fine-grained access control.

ABAC is a flexible access control model that controls access to an object by evaluating rules against the attributes of subjects, objects and environment. Access decisions are based on rules which specify the conditions under which access is granted or denied. The drawing regarding the ABAC model is given in Figure 2.2.

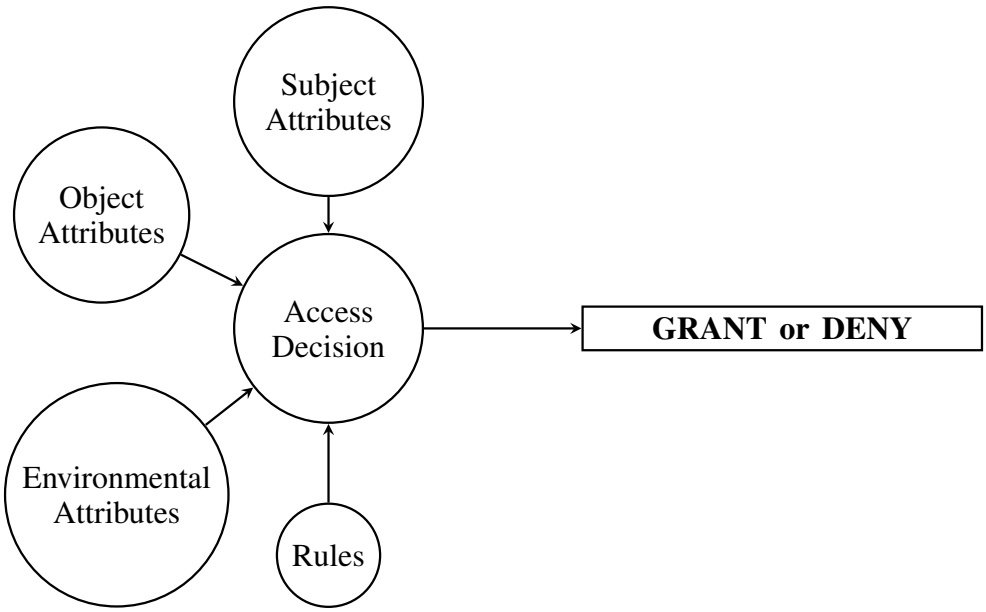

Figure 2.2. Attribute Based Access Control

Advantages of ABAC are given below:

- ABAC does not require any role structuring or role engineering step since it does not use any role function. Therefore, it is an easy design access control. This

reduces the implementation time and cost of the access control model.

- It has high granularity in access control since it uses attributes.
- ABAC is a flexible access control model that is easy to use in dynamic environments.

Limitation of ABAC is given below:

- ABAC does not provide ease of management for administrator. Analysing and managing the security policies are cumbersome.

CHAPTER 3

RELATED WORK

In this field there are two lines of thought that we can make use of, since both of them are aimed to obtain flexible, fine-grained, easy to manage, usable and scalable access control models. One of them is the usage of context term in access control, and the other one is to merge advantages of RBAC and ABAC.

Using context awareness in access control ensures a fine-grained and more dynamic access control mechanism. Therefore, many of the researchers proposed different works that integrate context awareness with access control, but the proposed models in these studies targeted the pervasive computing applications that involve simple scenarios with less number of communicating components (1-to-many) such as smart home, smart office etc. The term IoT, as we use it here, refers to many-to-many type of communication with different members of application (person, process, service, device) through the cloud infrastructure and social networks. For such scenarios, access control for IoT applications should be dynamic and more fine-grained requiring the use of context information wisely.

RBAC is the most adopted access control model by business environment, because it has the advantage of easy administration through role function usage. However, dealing with predefining all interactions between things is not an easy process when considering IoT environment with complex communications. Additionally, role creation phase is also troublesome since the environment has too many subjects with different rights, which cause the creating of excessive number of roles leading to role-explosion problem. For this reasons, it is accepted that the usage of RBAC model alone is inadequate to ensure the security of IoT applications. ABAC is a more convenient access control model for use in environments with dynamic interaction, because it provides flexibility with attribute usage. Despite this feature, the policy review in terms of administration is cumbersome since any grouping function like role is not used in this model. Consequently, it is required to generate a new unified access control model that combines the flexibility of ABAC with RBAC's ease of administration. These two approaches are reviewed separately below.

3.1. Context-Aware Access Control Models

(Covington et al., 2001) has widened the the general role based access control approach by adding environmental roles that capture the security relevant information from the environment for fine-grained access control. Context information is defined as a role in their study. According to the context information acquired by sensors, the related environmental roles are activated. When a person request access to a particular resource, the participant's subject roles determine the allowed resources and the related environmental roles state the constraints. In other words, the permissions are associated with both subject roles and environmental roles. For example; in an aware home project, the person assigned to the babysitter role can reach the intercom service if the weekday and working hour roles are active. However, this approach will increase the number of roles by assigning role to the environmental conditions like the day of the month, time of the day, particular location in the house etc. This is an undesirable situation for IoT applications that already contain many components. Implementation of this proposed model is performed by (Covington et al., 2002).

(Al-Muhtadi et al., 2003) presented an authentication mechanism and context-aware access control model that is called Cerberus. The authentication mechanism part composed of two modules; one included the communication protocols like SESAME, username-password, Kerberos etc. and the other contained the authentication devices such as fingerprint scanner, PDA, smart badge etc. Separation of these two modules provided flexible authentication mechanism. The context information acquired by a pretty simply designed context infrastructure which uses first order predicates and boolean algebra. The access control part of this project is controlled by the inference engine. According to the authentication device that the principal is authenticated with, different confidence values are assigned to different principals, and the inference engine is considered the confidence values when granting access to the principle. In addition, the system is offered a dynamic access control that implies the continuous checking of context related to the principal. In case of detection of any violation in context information, access to a particular device or resource is terminated. Within the scope of this project, the designed security module Cerberus is implemented on a simple scenario that falls under the ubiquitous computing. This study differs from ours in two aspects; one of them is not using the role based access control, and the second one is the designed system is not for the IoT applications as it is dawned on proposed scenario.

Another work that combines RBAC with context awareness, developed for pervasive computing applications is presented by (Kulkarni and Tripathi, 2008). In this study the context information is also used for admitting a user to a role and to continue being a member of that role. For a patient information system given in the paper, the temporal constraints can be used to admit a nurse to the 'NurseOnDuty' role only within the working hours. When the constraints are no longer hold, the nurse's membership in NurseOnDuty role is terminated. Also the roles are dynamically created in the application design period and activated only during related application's execution time. Actually the context information is used for different tasks such as role permission assignment, permission activation and resource access. For the same patient information system scenario, a nurse can only access to the patient's medical documents only if the nurse is in the same ward with a doctor (permission activation). Also, he/she can reach the documents of patients only in her/his ward, not all patient's (resource access). As mentioned above, this study is developed for pervasive computing applications that do not use the interactions of many things like person, process, device etc. This CA-RBAC model is not suitable for IoT applications that contain many components, as dynamic role creation for every application will lead to redundancy of roles.

The most similar study to ours is the ConUCON security module that was developed for the Web of Thing applications by (Bai et al., 2014). ConUCON make use of the context awareness to provide security and privacy in usage control. The main difference of this work is that it is based on UCON (Usage Control Model) instead of RBAC. However, UCON based model, ConUCON, is a complicated security model since it does not utilize the flexibility of grouping users under roles. For example; granting access right to printer for five research assistants who are in the same status, five number of permission assignment should be written (5 subjects X 1 object) in the ConUCON model. In RBAC, one permission assignment that enable access to a printer will be enough by grouping these five people under a role name of "Research Assistants". In addition, the context modelling that ConUCON's proposed is quite sophisticated.

(Abdella et al., 2016) used combination of context awareness and RBAC for providing security to Android based mobile applications. A quite simple and understandable model is developed in which the users are assigned roles and the roles are in a relation with permissions. The context information is associated with the permissions. To gain an access to a resource, defined context information should be ensured. Since the Android based systems do not enable continuous checking of context information, their model

does not offer a dynamic access control. In addition, their scope of work is different than ours in several aspects like created role types, principals (users) and the context information types that need to be checked. Our aim is to develop their approach in order to adapt for the IoT applications.

3.2. RBAC with ABAC Extension

According to the words of NIST, the first integrated access control model initiative called RABAC (Role-Centric Attribute Based Access Control) is proposed by (Jin et al., 2012). In this model, the subjects are assigned to roles and permissions are grouped by roles as the base of RBAC and the subjects and objects are associated with attributes. To be able to constrain the permissions according to the subject and object's attributes, they added a new module, permission filtering policy (PFP). PFP used filters in the form of Boolean expressions consisting of subject and object attributes. The available permissions associated with roles in the subject's session is limited by checking the conformance of the subject and object's attributes with the filters. If one of the filter's result is FALSE, the related permission is removed from the available session permission. At the end, the access decision is made based on the obtained final available permissions. Proposed model is good at addressing the role-explosion problem since it uses constraints/filters on permissions, however they did not use environmental attributes like time of day or location. However, in IoT systems, it can be crucial to consider frequently changing environmental attributes in access decision. Another study by (Barkha and Sahani, 2017) enhanced the RABAC model by adding revocation of granted access in case of violation of constraints. Also, they used context based role activation mechanism that first checks the related condition to activate the roles rather than checking the condition related with permission. Since the model first checks whether the subject has any role, if the role activation condition does not fit the desired ones, the role will not be activated and the system will safely deny the access regardless of monitoring each related permissions and conditions in the session. In this way, they aimed to decrease the computation time. Both of these enhancements are efficient for access control.

Another access control model proposed by (Rajpoot et al., 2015) is AERBAC. AERBAC provided both content based authorization and context aware access control. By utilizing conditions associated with permissions they overcame the role explosion

problem. Also, they addressed the role-permission explosion problem by using object attributes in permissions. They obtained object set by grouping the objects that have same attributes, and the permissions are consisted of object set and operation. Thus, single permission can include an object set which have a few objects with common attributes instead of a unique object. Since it is not needed to write separate permission rules for each object, this approach reduces the number of related permissions with roles. Since model uses RBAC's properties it is easy to manage the policy modifications. However, as any increment in object attributes cause to exponential grow in number of object groups, attribute based object grouping can cause another managing problem in an environments having huge amount of objects.

(Hasiba et al., 2017) also proposed a model that extends properties of ABAC and RBAC by combining role and attribute concepts in a role-centric manner. As in RBAC, subjects are assigned to roles and roles have permission rules that limited with constraints. The rules are divided into two according to the object type; private object rules and shared object rules. This approach enables restricting each user to access only his/her own data without needing separate role or rule. Also, they grouped the objects based on the action/access type (write,read etc.) rather than object attributes as in AERBAC, and for each access type there is a separately defined permission rule including related object groups. This model addressed both the role-explosion and role-permission explosion problems. Also, grouping objects based on access mode is solved managing problem of huge number of object groups.

(Rath and Colin, 2017) developed an architecture for risk-aware ABAC model in their study. Actually, in access decision they make use of risk values obtained by risk-estimation engine, as well as context information.

(Qi et al., 2015) proposed a model that is created the roles based on static attributes and formed the rules considering dynamic attributes as in the suggestion of NIST to merge ABAC and RBAC. Therefore, they utilized attributes in both user to role assignment and role to permission assignment. Thus, in case of having huge amount of attributes, number of roles in RBAC and number of rules in ABAC are significantly reduced.

As can be seen from the reviewed papers, actually the attribute based (role-centric) and context-aware RBAC approaches are parallel in terms of limiting permissions according to the related attributes or contexts, since both of them aim at handling security issues. In other words, they serve for the same purpose. However, it should not be forgotten that none of the revised papers in this section are addressing IoT applications.

CHAPTER 4

DESIGNED MODEL: CA-IRBAC

In a sense, this model can be considered as combination of CA-RBAC and ABAC in order to enhance ABAC model to be able to have easy administration feature like RBAC. Role component of RBAC enables to group the users and permissions, and so administrator can easily review the permissions of a specific user. Therefore, grouping purpose of RBAC's role component is used functionally in the designed model. However, IoT environment have dynamic interaction of vast amount of things, and it is not possible to define the interaction among these things exactly and priorly. So, it is inefficient to write predefined policies that is going to used in access decisions. Besides, forcing to assign a role to each subjects will lead to creation of excessive number of roles due to the variety of vast amount and different types of subjects. Considering the existence of 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, in this thesis we propose that; operations like read, write, surveillance etc. is used to group the subjects since the number of operations that can be executed in a system is always limited. In this way, it is aimed to prevent the role-explosion problem which have high occurrence probability for IoT environment. Additionally, in such an environment, it is too hard and inefficient to define a relationship between each subject and operation. Therefore, subject attributes are assigned to operations instead of subjects to provide flexibility. Also, object attributes are used to group the objects in order to enable easy administration of objects and to prevent the role - permission explosion.

4.1. Formal Model

Operations (Op): Operations indicate defined actions that are allowed to perform on an object in a system, and the subjects are grouped under specific operations.
Subjects(S): Subjects are the entities that have access right on objects under particular situation. In IoT environment; users, services, processes, applications, devices can be the subjects of the system.

Subject Attributes (SA): Subject attributes capture the properties of each subjects. Each subject should have at least one subject attribute. There are many-to-many mapping between SA's and subjects which means a SA 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 relationship between objects and OA's like the mapping between subject and SA's.

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

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 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 needs to be checked, so many context rules are required. Multiple context rules enable to write combined context rules by using logical operators like and ($\wedge$), or ($\vee$). MCR is stored in the form of:

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

4.1.1. Attribute Assignment

Attribute Assignment Table (AAT) includes mapping between attributes and subjects/objects. For simplicity, both SA assignment to subjects and OA assignment to objects are held in the same table. Both of the subjects and the objects can have multiple attributes. The records of the AAT consist of; *(*S_j*,* Att_i*)* or *(*O_j*,* Att_i*)* where $Att_i \in SA$ or OA and $S_j \in S$, similarly $O_j \in O$. Figure 4.1 given below shows the relation between the entities and the attributes.

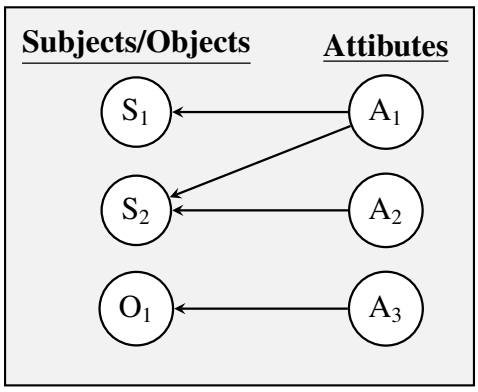

Figure 4.1. Attribute Assignment to Subjects and Objects

4.1.2. Subject Attribute - Operation Assignment

Aim of the subject attribute assignment to operation is to get easy administration by grouping the subject attributes (implicitly subjects) under allowed operations. Subject attribute - operation assignment (SAOA) table stores data as follows; (Op_k, SA_t) where $SA_t \in SA$ and $Op_k \in Op$. Figure 4.2 shows the relationship between the subject attributes and operations.

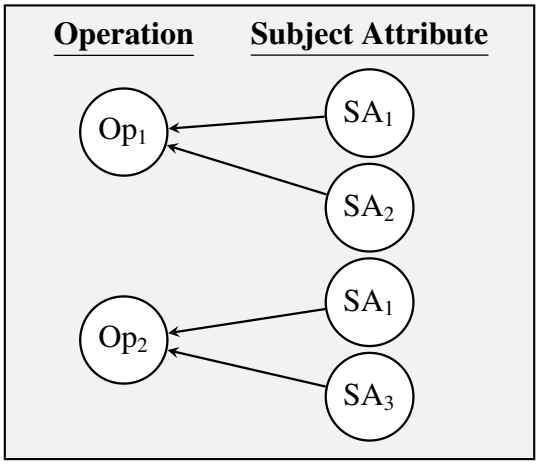

Figure 4.2. SA - Operation Assignment

As seen in the Figure 4.2 a many-to-many relationship exist between subject attributes and operations: SAOA $\subseteq$ *SA* X *Op*. This means that, a SA can be allowed to

perform more than one operation, so can be assigned to different operations. Similarly, one operation may contain many subject attributes.

4.1.3. Object Attribute - Operation Assignment

This access control model designed to facilitate the use of different authentication methods. Some objects may be accessed by using different authentications, whereas accessing to an object requiring high security can be limited with only a specific authentication method. Therefore, authentication methods are related with object attributes and operations. Object Attribute - Operation Assignment (OAOA) table is consisted of 4 columns which are operations, object attributes, context and authentication. Each record should include 4 elements; (Op_u, $Auth_z$, OA_v, Con_y) where $Op_u \in Op$, $Auth_z \in Auth$, $OA_v \in OA$ and $Con_y \in Con$. Figure 4.3 illustrates the relationship between operation, OA, authentication and context rule.

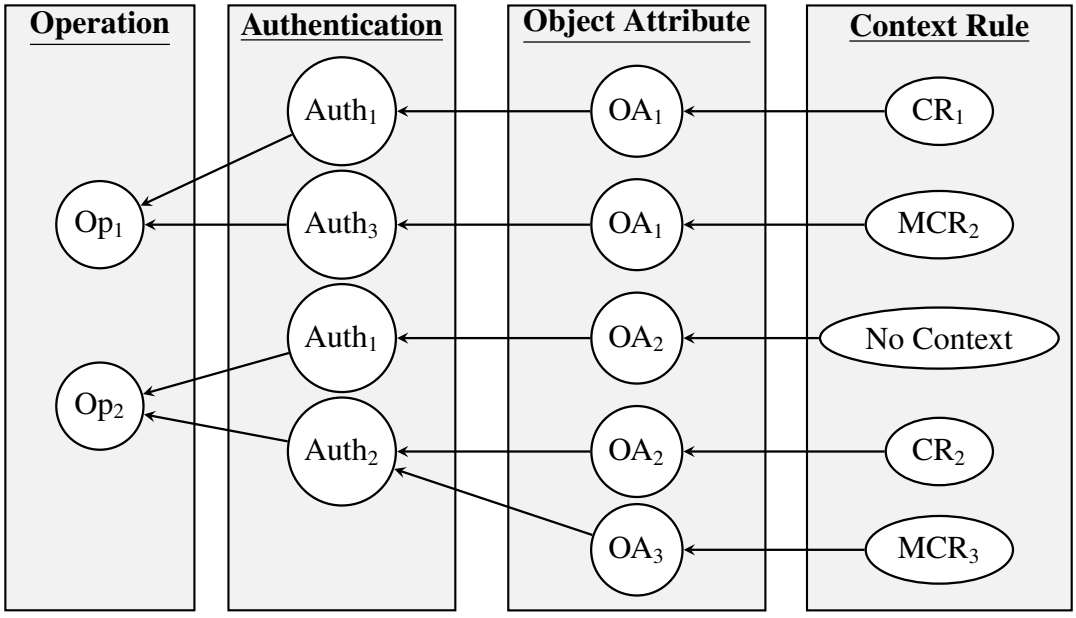

Figure 4.3. OA - Operation Assignment

As the Figure 4.3 illustrates, Op_1 is allowed to access objects that only have OA_1, however these objects can be reached by using different authentication methods with each having distinct contexts. Op_2 can not be performed on objects in group OA_1, but it can

access to objects having OA_2 and OA_3. While OA_1 and OA_2 accept the access requests by using different authentication methods associated with various contexts, access to OA_3 can only be allowed through a specific authentication. Some of the object attributes have no context associated with it, which means that all subjects having SA that are assigned to related operation are granted to perform that operation on requested OA. Allowing to utility of various kinds of authentication methods is enabled to access control being flexible and fine-grained.

4.2. Framework Architecture

Architectural design of the proposed model is shown in Figure 4.4, and the component descriptions are given below.

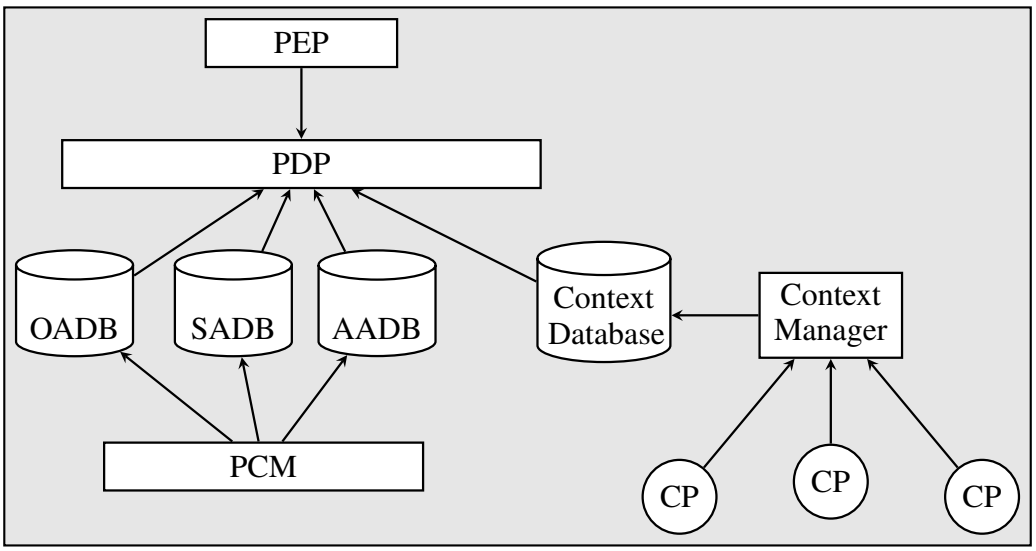

Figure 4.4. Framework Architecture

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, CDB peripheral units.

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 is fulfilled by the subject, context information is acquired by the Context Manager unit through the context providers, and stored in 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 instant context in CDB. Therefore, PDP unit is able to access these current context information of related subjects over CDB in evaluation of access request.

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

4.3. Working Principle of Model

In this section the working principle flow diagram of the designed model is given in Figure 4.5 to show the general idea of the execution sequence.

The requests are come in the form of:

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

PEP handles this request and forward it to PDP for evaluation of access. PDP first checks the operation in SADB whether it is defined or not. If there is not such an operation defined in the system, it sends 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, assigned subject attributes to the requestor subject is retrieved from AADB, and PDP compares these values with the subject attribute list. If PDP can not find any subject attributes of the subject within the SA list, access is denied. Otherwise, according to requestor subject's authentication type allowed object attributes and corresponding context that are assigned to requested operation is listed from OADB. Requested object's

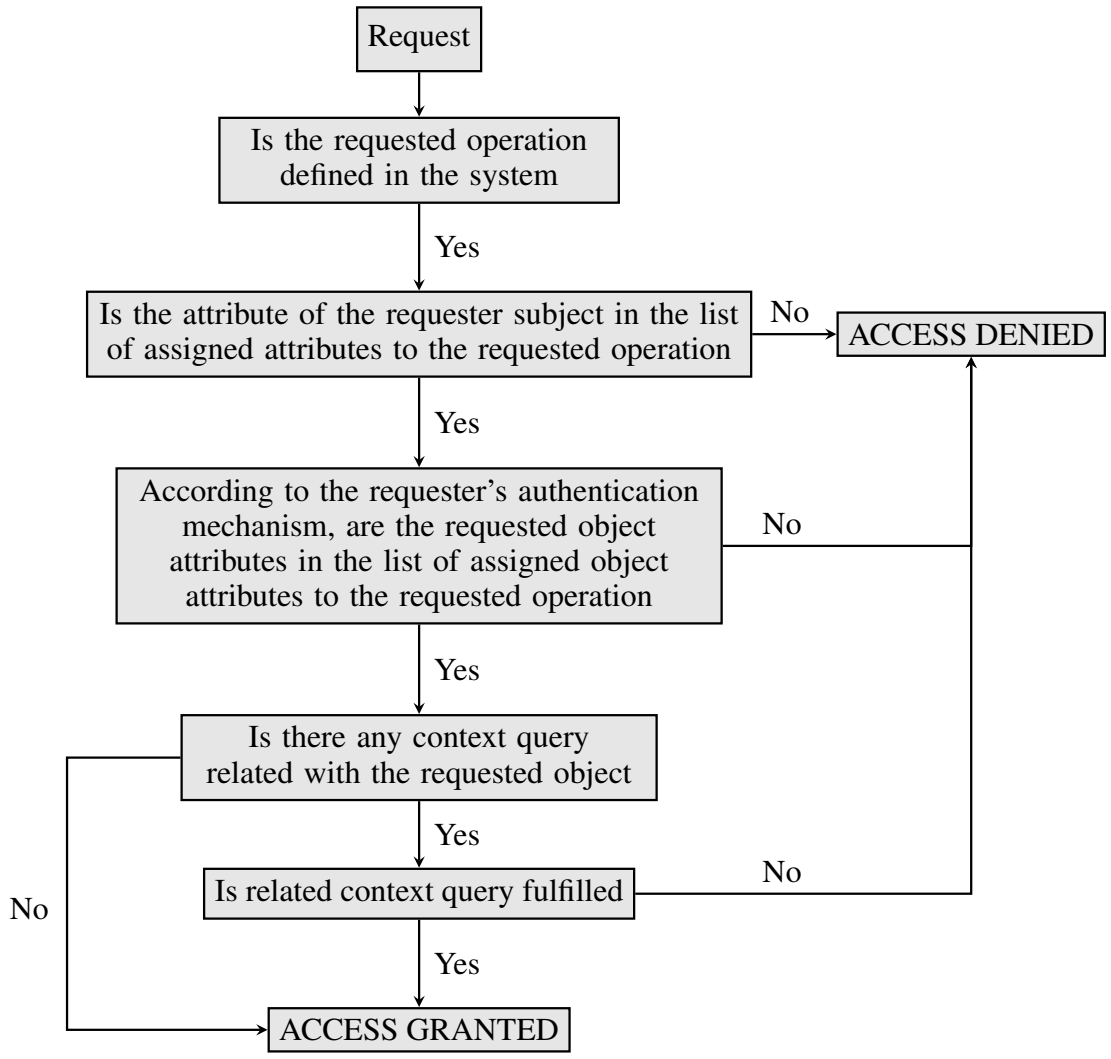

Figure 4.5. Flow Diagram of Working Principle

attributes are also retrieved from AADB. If the object attribute does not match with one of the element of the list, the access will be denied. If it matches, the corresponding context will be checked from CDB. In case of context requirements satisfaction, PDP sends 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 this situation PDP always sends 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. The algorithm of explained procedure is given below.

Algorithm 1

Input: Access_ Request (Rq<Sid, Oid, Op, Authenticated>)
Output: Access

$Operations \leftarrow List\_Operations$
if (Op) in list Operations **then**
    $SA \leftarrow get\_attributes(Sid)$
else
    $Access = Deny$
end if
$SA\_List \leftarrow List\_Subject\_Attibutes(Op)$
if (SA) in list SA_ List **then**
    $OA \leftarrow get\_attributes(Oid)$
else
    $Access = Deny$
end if
$OA\_List \leftarrow List\_Object\_Attibutes(Authenticated, Op)$
if ((OA) in list OA_ List) **then**
    $Context \leftarrow context(Authenticated, Op, OA)$
else
    $Access = Deny$
end if
$Result \leftarrow evaluate(Context)$
if Result = False **then**
    $Access = Deny$
    return$(RequestDenied)$
else
    $Access = Grant$
    return$(RequestGranted)$
end if

4.4. Example Scenario

To illustrate the proposed model the example scenario below is used. This scenario is related with an online entertainment store which is also used as an example by (Rajpoot et al., 2015) and (Hasiba et al., 2017). Actually, this example is not related with an smart environments that is covered by IoT domain. However, since it is a simple and easy to understand scenario, it is included.

The system contains 2 types of users as adult and juvenile, also the users can be assigned as premium or regular user. The movies are separated into two; G-rated and R-rated movies. Juvenile users can just view the G-rated movies while adult users can view all kinds of movies. Also, newly released movies can be watched by premium users, but during promotion times regular users can also access these movies. Premium users are also allowed to download the movies except the newly released ones. For this

scenario, subject attributes are assigned as *Adult, Juvenile, Premium, Regular*, and the object attributes are selected as *G-rated movie, R-rated movie, Newly released movie*. This system allows to perform *download, view* operations. Figure 4.6 shows the sample scenario graphically.

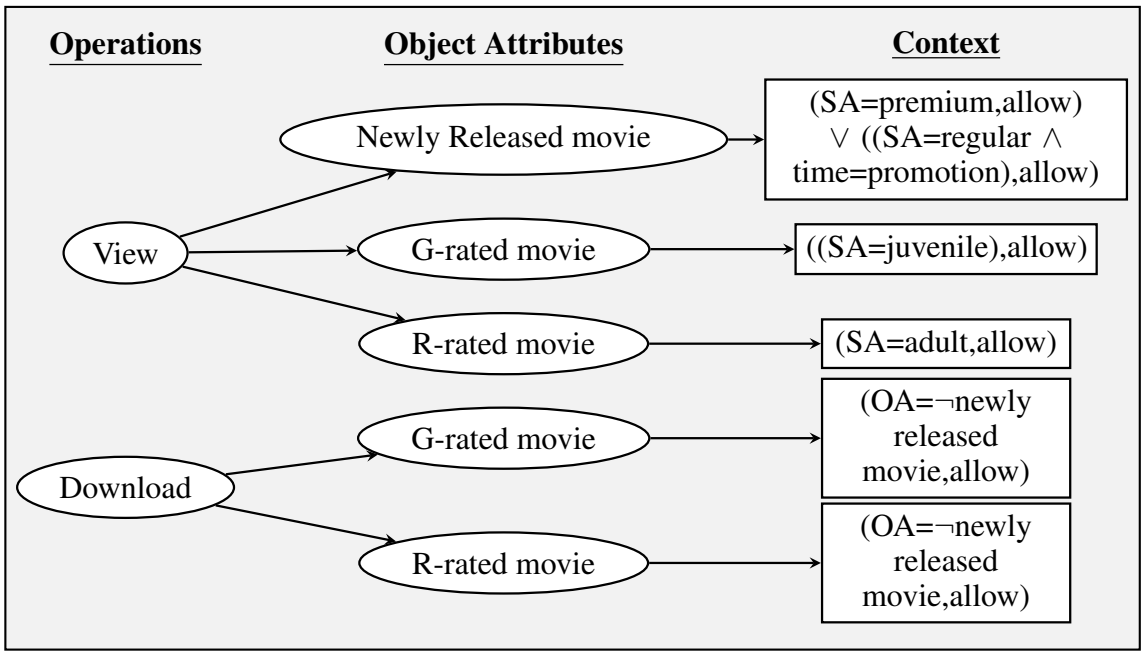

Figure 4.6. Online Entertainment Store Scenario

If we use flat RBAC without context, we need to create 6 roles as "*Adult-premium*, *Adult-regular*, *Juvenile-premium*, *Juvenile-regular*, *Adult-promotion*, *Juvenile-promotion*". The newly released movies will be assigned to the *Adult-premium* and *Juvenile-premium* roles according to the rating they are allowed to view, and for the promotion periods *Adult-promotion* and *Juvenile-promotion* roles will be activated which have permissions regarding permission to view the newly released movies. It is shown that such a basic scenario require 6 role creations. As we highlighted from the beginning, IoT domain contains complex interaction of things that is not possible to be predefined. Therefore, it is needed to create many roles which lead to role-explosion problem. Another problem is related with the objects. It can be imagined that there are many movies/objects stored in the database. Each object has its own permission which cause many permissions being assigned to each role, that is called as role-permission explosion problem.

Enhancing RBAC with context has reduced the number of roles needed. For this

scenario, by using context aware-RBAC, role numbers will be reduced from 6 to 2 which are "*Adult* and *Juvenile*". This can be a solution to role-explosion problem. However, context usage is not a solution for the role-permission explosion problem, it requires attribute usage for objects.

CHAPTER 5

EVALUATION

5.1. Sample Scenario

Integration of context awareness with RBAC model is proposed by different researchers but the term IoT we consider implies the communication of (many x many) complexity smart things through the cloud infrastructure or social networks. The scenario that we choose in order to investigate the efficiency of our proposed access control model includes the interaction of different smart environments like smart home and smart car.

Complexity of CA-RBAC (Context aware role based access control), ABAC and CA-IRBAC models are evaluated using this scenario. According to this, access control of 8 subjects to 6 objects is performed under various contexts. The comparison of the rule sets of the models and complexity results are given separately in section 5.2. First, the entities regarding the scenario, then the interpretation of security policies are given below.
Subjects:

1. Katie (mother)
2. John (father)
3. James (child)
4. Joe (child)
5. Sue (child)
6. Jessica (babysitter)
7. Smart Home Application
8. Smart Healthcare Application

Subject Attributes:

- Title: Parent, Children, Babysitter, Healthcare App, Home App

Objects:

1. Smart Door
2. Oven
3. Washing Machine
4. Dish Washer
5. Camera
6. Wearable Insulin Pump

Object Attributes:

- Object Type: 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

Object type and *title* are the defined attributes for objects and subjects respectively. Each object and subject should have at least one attribute. There are 7 environmental attributes defined to achieve fine-grained access control. Distance attribute is defined as atomic valued, and the others are defined as set type attributes. Principals are free to use biometric or mobile device authentication. Subjects included are both person and application. In the scope of this thesis context modelling is not handled. Therefore, it is assumed that all contexts in 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 at less than 10 meters to the house, and the time is outside the 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.
- 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.
- 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 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.
- Smart home application can also send request to open the smart door if emergency situation context is asserted, and the ambulance is close to the house less than 10 meters away.

Household Appliances:

- Mother and father can turn the oven, dish washer and washing machine on using their mobile devices when they are not inside the house.
- Children are not allowed to turn any electrical household appliances on or off.
- 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.

- Mother, father and smart home application can turn the electrical household appliances off, if 30 minutes passed since turn them on request has been made, and parent or babysitter is not inside the house.

Camera:

- Camera can be viewed by people who have parent attribute via biometric authentication.
- Mother and father can view camera using their mobile devices if the emergency situation context is enabled.
- Smart home application can also view 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.

5.2. Complexity Analysis

In this section the rule sets of given scenario is shown regarding to CA-RBAC in Table 5.1, ABAC in Table 5.4, and CA-IRBAC in Table 5.6. The comparison is made based on complexity of the models which is related with the number of security policies. Cartesian product of the component's sets give us the total possibility of interaction between things with different context. Therefore, the results are shown in terms of the cartesian product results. According to this; for RBAC model we have $(Permissions \times Context)$ number of possible interactions for one role, and in total $(Role \times Permission \times Context)$ number of security policies can be written. However, in CA-IRBAC we have $(Operations \times ObjectAttributes \times Authentications \times Contexts)$ of different combinations. This number can be interpreted as the maximum possible interaction of things under different context that should be considered. As it is mentioned before, since new generation IoT environment have vast amount of things, it can not be

easy to identify all the interactions among them. Therefore, it is aimed to reduce the number of security policies as far as possible in order to enable easy administrated access control.

5.2.1. CA-RBAC implementation of the scenario

In this section access to objects, which are in the scope of the given scenario, are controlled by using context aware RBAC. Each subject have roles, and the roles have corresponding permission sets. Role members are given below the role names in the Table 5.1. The permissions consist of object and the operation. The context terms are assigned to the permissions in this model. The 'Access' column shows the results of evaluation in case of fulfilling the context. Rule set is given in Table 5.1 and Table 5.2.

Table 5.1. Rule set of CA-RBAC

Roles	Permission	Context	Access
PARENT **Katie** **John**	Open the smart door	Authentication=biometric	ALLOW
	Open the smart door	Authentication=mobile device ∧ time>working hour ∧ distance(car)<10m	ALLOW
	Surveillance camera	Authentication=biometric	ALLOW
	Surveillance camera	Authentication=mobile device ∧ emergency=yes	ALLOW
	Turn the oven on	Authentication= mobile device ∧ loc(requestor)= ¬inside house	ALLOW
	Turn the oven off	Authentication= mobile device ∧ (loc(requestor) ∨ loc(Jessica-babysitter))= ¬inside house ∧ time(req-turn the oven on) - time(current)>30min	ALLOW
	Turn the washing machine on	Authentication= mobile device ∧ loc(requestor)= ¬inside house	ALLOW
	Turn the dish washer on	Authentication= mobile device ∧ loc(requestor)= ¬inside house	ALLOW
CHILDREN **James** **Joe** **Sue**	Open the smart door	Authentication=biometric ∧ loc(requestor)=outside house	ALLOW
	Open the smart door	Authentication=biometric ∧ loc(requestor)=inside house ∧ (loc(Katie-mom) ∨ loc(John-dad) ∨ loc(Jessica-babysitter)=inside house ∨ emergency=yes)	ALLOW

The CA-IRBAC scenario rule set continues in Table 5.2 below.

Table 5.2. Rule set of CA-RBAC (cont.)

CHILDREN	Open the smart door	Authentication=biometric ∧ loc(requestor)=outside house	ALLOW
BABYSITTER **Jessica**	Turn the dish washer on	Authentication= mobile device ∧ loc(requestor)=inside house ∧ time=working hour	ALLOW
	Turn the oven on	Authentication= mobile device ∧ loc(requestor)=inside house ∧ time=working hour	ALLOW
	Turn the washing machine on	Authentication= mobile device ∧ loc(requestor)=inside house ∧ time=working hour	ALLOW
	Open the smart door	Authentication=biometric ∧ loc(requestor)=inside house ∧ sb. in front of door=no	ALLOW
	Open the smart door	Authentication=biometric ∧ loc(requestor)=outside house ∧ time=working hour	ALLOW
	Open the smart door	Authentication=biometric ∧ loc(requestor)=inside house ∧ sb. in front of door=yes ∧ parent's approval=yes	ALLOW
SMART HOME APPLICATION **Home app.**	Turn the oven off	Authentication=mobile device ∧ time(req-turn the oven on) - time(current))>30 min	ALLOW
	Open the smart door	Authentication=mobile device ∧ distance(ambulance)<10m ∧ emergency=yes	ALLOW
	Surveillance camera	Authentication=mobile device ∧ emergency=yes	ALLOW
	Data read from insulin pump	Authentication=mobile device ∧ emergency=yes	ALLOW
SMART HEALTHCARE APPLICATION **Healthcare app.**	Data read from insulin pump	No context	ALLOW

Considering the subjects of the system and their allowed rights on objects, it is required to create 5 distinct roles for this scenario. It is defined 6 distinct permissions that control the access to 6 different objects. 7 distinct context types are used to provide fine-grained access control. Contexts are assigned to the permissions. Therefore, if a subject wants to access an object in order to perform the allowed action specified in the permission, it must satisfy the related context. The roles, operations, objects and the

contexts are shown in Table 5.3 below.

Table 5.3. Complexity Analysis of CA-RBAC Model

Number of Created Roles	5	Parent, Babysitter, Children, 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: Biometric, Mobile Device Distance Parent's Approval: yes, no Sb. in front of door: yes, no Location: inside house, outside house Time: working hour, school hour Emergency: yes, no

According to this, it is possible to write $(permission \times context)$ distinct permission-context relation for one role. Context refers to the environmental attributes, so we have 13 number of contexts. Permissions are consisted 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 $[role \times context \times (object \times operation)]$. If we calculate this for our scenario;

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

$SecurityPolicies = (6\ roles) * (18\ permissions) * (13\ context)$ = 1404

5.2.2. ABAC implementation of the scenario

In this section the ABAC model is used to provide application level security. ABAC rule set consists of rules indicating the security policies. Rules form is as follows;

<Context, Operation, Access>

The contexts include the subject, object and environmental attributes to limit the access to specific objects. In case of matching between the subject's context and the specified context within the evaluated rule, the result which is stated in 'Access' parameter, that can be allow or deny, will be returned to the subject. For example, the rule engine searches the rules until finding the contexts that are matching with the subject's. When it finds such a match between the contexts, if the access parameter is 'deny', it denies the request,

or if the access parameter is 'allow', it allows the request. However, if it does not find any context match, just denies the request. Subjects and objects are represented with the attributes instead of their id's. This attribute usage procedure makes ABAC flexible and dynamic. Rule set of access control model is given below in Table 5.4.

Table 5.4. Rule set of ABAC

RULES
<(SA=parent ∧ OA=smart door ∧ auth=biometric), open, allow>
<(SA=children ∧ OA=smart door ∧ auth=biometric ∧ ((loc(requestor) ∧ (loc(parent) ∨ loc(babysitter)))=inside house ∨ emergency=yes), open, allow>
<SA=children ∧ OA=smart door ∧ auth=biometric ∧ loc(requestor)=outside house, open, allow>
<SA=babysitter ∧ OA=smart door ∧ auth=biometric ∧ loc(requestor)=outside house ∧ time=working hour, open, allow>
<SA=babysitter ∧ OA=smart door ∧ auth=biometric ∧ loc(requestor)=inside house ∧ sb. in front of door=yes ∧ parent's approval=yes, open, allow>
<SA=babysitter ∧ OA=smart door ∧ auth=biometric ∧ loc(requestor)=inside house ∧ sb. in front of door=no, open, allow>
<SA=home app. ∧ OA=smart door ∧ auth=mobile device ∧ distance(ambulance)<10m ∧ emergency=yes, open, allow>
<SA=children ∧ OA=smart door ∧ auth=mobile device ∧ time>school hour ∧ distance(schoolbus)<10m, open, allow>
<SA=parent ∧ OA=household appliances ∧ auth=mobile device ∧ loc(requestor)=¬inside house, open, allow>
<SA=babysitter ∧ OA=household appliances ∧ auth=mobile device ∧ loc(requestor)=inside house ∧ time=working hour, open, allow>
<SA=parent ∧ OA=camera ∧ auth=biometric, data read, allow>
<SA=parent ∧ OA=camera ∧ auth=mobile device ∧ emergency=yes, data read, allow>
<SA=home app. ∧ OA=camera ∧ auth=mobile device ∧ emergency=yes, data read, allow>
<SA=home app. ∧ OA=wearable devices ∧ auth=mobile device ∧ emergency=yes, data read, allow>
<SA=healthcare app. ∧ OA=wearable devices ∧ auth=mobile device, data read, allow>
<(SA=home app. ∨ parent) ∧ OA=household applicances ∧ auth=mobile device ∧ (time(req-open the household appliances) - time(current))>30min∧ loc(requestor) ∨ loc(babysitter)=¬inside house, turn off, allow>
<SA=parent ∧ OA=smart door ∧ auth=mobile device ∧ time>working hour ∧ distance(car)<10m, open, allow>

As the Table 5.4 shows, ABAC model does not include any group of entities. All the rules which are related with the all possible interactions are written separately. For our

scenario, ABAC model implementation uses 4 object attributes for grouping the objects, 5 subject attributes for representing subjects and 13 environmental attributes to control the access to 6 objects. Name and the number of attributes used in complexity calculation can be seen in Table 5.5.

Table 5.5. Complexity Analysis of ABAC Model

Number of Subject Attributes	5	Parent, Babysitter, Children, Smart Home App., Smart Healthcare App.
Number of Object Attributes	4	Smart door, Camera, Household Appliances, Wearable Devices
Number of Environmental Attributes	13	Authentication: Biometric, Mobile Device Distance Parent's Approval: yes, no Sb. in front of door: yes, no Location: inside house, outside house Time: working hour, school hour Emergency: yes, no
Number of Operations	3	Data Read, Open, Turn off

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

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

5.2.3. CA-IRBAC implementation of the scenario

CA-IRBAC example of the scenario is explained in this section. In CA-IRBAC model, roles are replaced with operations. Each subjects, who are allowed to perform an operation, have attributes and these attributes are assigned to corresponding operations. This process implicitly defines the subject's authorized operations. Object attributes are assigned to operations that indicate the actions allowing to perform on related object. However, assigning a subject to an operation does not mean that the subject can perform the operation on all objects which are in the allowed objects list of the operation.

Different authentications can be used accessing same object attributes. Thus, object attributes are grouped according to authentication that are allowed to use in access. Contexts limit the subject's access to an object attribute using subject attributes and environmental attributes. The security policies of CA-IRBAC model is given in Table 5.6.

Table 5.6. Rule set of CA-IRBAC

Operation	Auth.	OA	Context	Access
OPEN: **Parent** **Babysitter** **Home App** **Children**	Biometric	Smart Door	SA= parent	ALLOW
			SA=children ∧ loc(requestor)=outside house	ALLOW
			SA=children ∧ (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

CA-IRBAC model implementation uses 4 object attributes, 11 environmental attributes, 5 subject attributes and 2 authentication types to control the access. 3 different operations are defined to represent the actions which are allowed to perform on objects. The difference in terms of the number of contexts is that the context number includes the summation of subject and environmental attributes since we use subject attributes as constraints. So, number of context is evaluated as $(subject\ attributes + environmental\ attributes)$. The related attributes, contexts and operations are shown in Table 5.7.

Table 5.7. Complexity Analysis of CA-IRBAC 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, Children, Babysitter, Home app., Healthcare app. Distance Parent's Approval: yes, no Sb. in front of door: yes, no Location: inside house, outside house Time: working hour, school hour Emergency: yes, no
Number of Operations	3	Data Read, Open, Turn off

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

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

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

5.2.4. Comparison

Figure 5.8 shows the number of maximum possible security policies for each of the models on the same scenario. It is obvious that, the complexity is nearly reduced by 4 times by using CA-IRBAC model. Also, the role-explosion problem of RBAC is removed by using operation instead of role. Even on this simple scenario it is achieved to reduce the number of roles by half. Additionally, the policy review in terms of subjects is made quite easily since the operations are assigned allowed subject lists. This model will benefit

more complex scenarios that consists of interactions of different smart environments like smart hospital, smart car, smart office.

Table 5.8. Comparison Results Regarding Complexity of Access Control Models

	CA-RBAC	**ABAC**	**CA-IRBAC**
Number of Roles	6	-	3
Number of Security Policies	1404	780	384

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1. Conclusion

The main aim of this thesis is to provide fine-grained and dynamic access control mechanism for the next generation IoT applications, that includes many-to-many interaction complexity of things, by combining the role based access control approach with attribute based access control model through adding role function to ABAC. According to this, CA-IRBAC model that integrating role function with attribute usage is proposed. The features of the proposed model is listed below:

- ***Operation-Based***: The main difference of the proposed model is the grouping approach is followed for the operations instead of roles. The model first checks the requested operation to decide whether the requester has access permission or not.
- ***Reduced Number of Security Policies***: Decrease in role number lead to reduction of maximum possible security policy number. Also, there is a contribution of object attribute usage to this achievement too. Since no more need to write separate security policies for each of the objects by using object attributes, number of security policies is decreased.
- ***Context Aware Role Based Access Control***: By using context awareness with the role based approach it is aimed to provide more fine-grained access control. Also, the access request results are not static, it depends on the required context conformance. Thus, context usage provides dynamism.
- ***Attribute Usage***: By using attributes, it is aimed to get dynamic access control model. Also, attribute usage prevents the role - permission explosion problem by assigning objects an attribute that enable to group them.
- ***Flexibility in Authentication***: Proposed model enable the use of different authentication types on the same object. By limiting the access to an specific object, it is

provided more granular access control.

- ***Decreased number of roles***: In any environment, the number of operations that are allowed to perform is limited. In this design, it is benefited from this limitation. Therefore, by using operations instead of roles, it is provided to reduce the created role number, and the role - explosion problem is prevented accordingly.

- ***Easily Manageable***: Aim of to design this model by including operation function is to get easy manageable access control model. It can be easily reviewed that which operations the subject are authorized to perform on which object groups.

6.2. Future Work

According to NIST announcement regarding the enhancement of ABAC with role function, there are many models proposed up to now. Our model is also designed considering this announcement, and we achieve good result in terms of reducing the complexity as it is shown in Chapter 5. Now, we plan to implement this scenario on real life environment using FIWARE open source IoT platform. We expect to get efficient performance parameters. In addition, both the verification of whether the security rules are executed properly in practice and the success of the system in providing security will be investigated especially under attack scenarios.

REFERENCES

Abdella, J., M. Özuysal, and E. Tomur (2016). Ca-arbac: privacy preserving using context-aware role-based access control on android permission system. *Security and Communication Networks 9*(18), 5977–5995.

Abowd, G. D. (2016). Beyond weiser: From ubiquitous to collective computing. *Computer 49*(1), 17–23.

Abowd, G. D., A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles (1999). Towards a better understanding of context and context-awareness. In *International symposium on handheld and ubiquitous computing*, pp. 304–307. Springer.

Al-Muhtadi, J., A. Ranganathan, R. Campbell, and M. D. Mickunas (2003). Cerberus: a context-aware security scheme for smart spaces. In *Pervasive Computing and Communications, 2003.(PerCom 2003). Proceedings of the First IEEE International Conference on*, pp. 489–496. IEEE.

Bai, G., L. Yan, L. Gu, Y. Guo, and X. Chen (2014). Context-aware usage control for web of things. *Security and Communication Networks 7*(12), 2696–2712.

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

Covington, M. J., P. Fogla, Z. Zhan, and M. Ahamad (2002). A context-aware security architecture for emerging applications. In *Computer Security Applications Conference, 2002. Proceedings. 18th Annual*, pp. 249–258. IEEE.

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

Hasiba, B. A., L. Kahloul, and S. Benharzallah (2017). 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.

Jin, X., R. Krishnan, and R. Sandhu (2012). 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.

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

Kuhn, D. R., E. J. Coyne, and T. R. Weil (2010). Adding attributes to role-based access control. *Computer 43*(6), 79–81.

Kulkarni, D. and A. Tripathi (2008). 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.

Morgan, J. (2014). A simple explanation of'the internet of things'. *Retrieved November 20*, 2015.

Qi, H., H. Ma, J. Li, and X. Di (2015). Access control model based on role and attribute and its applications on space-ground integration networks. In *Computer Science and Network Technology (ICCSNT), 2015 4th International Conference on*, Volume 1, pp. 1118–1122. IEEE.

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

Rath, T. A. and J.-N. Colin (2017). Adaptive risk-aware access control model for internet of things. In *2017 International Workshop on Secure Internet of Things (SIoT)*, pp. 40–49. IEEE.

Sandhu, R., D. Ferraiolo, R. Kuhn, et al. (2000). The nist model for role-based access control: towards a unified standard. In *ACM workshop on Role-based access control*, Volume 2000, pp. 1–11.

Schilit, B., N. Adams, and R. Want (1994). Context-aware computing applications. In *Mobile Computing Systems and Applications, 1994. Proceedings., Workshop on*, pp. 85–90. IEEE.

Weber, H. A. (2003). Role-based access control: the nist solution. *SANS institute InfoSec Reading Room*.

Weiser, M. (1991). The computer for the twenty-first century (pp. 94-100). *Scientific American, September Issue*.