

Usage Based Access Control for Web Based Applications



By
Um-e-Ghazia
2010-NUST-MS-CCS-25

Supervisor
Dr. Muhammad Awais Shibli
Department of Computing

A thesis submitted in partial fulfillment of the requirements for the degree
of Masters in Computer and Communication Security (MS CCS)

In
School of Electrical Engineering and Computer Science,
National University of Sciences and Technology (NUST),
Islamabad, Pakistan.

(January, 2013)

Approval

It is certified that the contents and form of the thesis entitled “**Usage Based Access Control for Web Based Applications**” submitted by **Um-e-Ghazia** have been found satisfactory for the requirement of the degree.

Advisor: **Dr. Muhammad Awais Shibli**

Signature: _____

Date: _____

Committee Member 1: **Dr. Zahid Anwar**

Signature: _____

Date: _____

Committee Member 2: **Dr. Abdul Ghafoor**

Signature: _____

Date: _____

Committee Member 3: **Muhammad Bilal**

Signature: _____

Date: _____

Abstract

Cloud computing is the latest computing technology that provides various on demand services to large variety of users. This technology helps large organizations and enterprises by executing large number of processes in order to reduce their computational overhead. Even though the Cloud offers significant benefits, there are still many security issues that avoid users to adopt this technology. Some of the major security issues include data confidentiality, trust establishment, access management and data integrity etc. Access control is one of the mandatory security requirements in Cloud environment that avoids the unauthorized usage of Cloud resources. In the current thesis, we have carried out research in two directions and one of them is the detailed study of access control models for Cloud environment. Based on this study, we have examined the viability of access control models for Cloud environment and their comparative analysis has been performed. Assessment criteria have been proposed that analyzes the Cloud based access control models according to NIST defined evaluation features for access control models. This analysis highlights the essential features that must be incorporated in access control models for Cloud dynamic environment. After the analysis, we have concluded that Usage Based Access Control Model (UCON) is the most appropriate model that can perform better according to specifications of Cloud environment.

Another research direction of our thesis is the comprehensive study of UCON model and its applicability in different applications and environment. Main distinguishing features of UCON model are attribute mutability and continuity of access decision that makes it far better than the traditional access control models. In order to increase the accuracy of access decision, UCON model has three main decision factors i-e authorization, obligation and condition. Despite of all these excellent features, UCON model is not being widely adopted by organizations in order to provide the controlled access for their resources. The major reason for this is that there is no proper specification available for UCON model in any policy specification language. There is a need to provide the specification of UCON model in

order to be used for different real world applications. We have proposed the UCON profile in eXtensible access control markup language (XACML) in order to address this issue. XACML is a generic policy language that offers the request response phenomenon in addition to the policy specification standard. The UCON profile has been formulated by the addition of newly created attributes and identifiers in XACML that enable organizations to deploy this model in different scenarios.

Certificate of Originality

I hereby declare that this submission titled **Usage Based Access Control for Web Based Applications** is my own work. To the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEECS or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis.

I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics which has been acknowledged.

Author Name: Um-e-Ghazia

Signature: _____

Acknowledgment

First and foremost, I would like to thank ALLAH ALMIGHTY for giving me courage and motivation during the thesis to cater all the difficulties and problems in amicable manner. This motivation helps me to perform tasks with hard work and full dedication and devotion.

I offer sincere gratitude to my supervisor Dr. Muhammad Awais Shibli who has put his great effort throughout the thesis phase with his knowledge, expertise and valuable suggestions. He has provided full support, mentorship and continuous assistance that enable me to learn new concepts of related domain and develop an understanding of how to perform research. I wish to thank my committee members Dr. Zahid Anwar, Dr. Abdul Ghafoor and Muhammad Bilal for their kind support, availability and useful ideas that help me to refine the research work.

I am very blessed to have my mother with me whose great company adds significant contribution to my life and education. I would specially acknowledge my mother for her prayers, patience and support without which I could not be able to achieve all this. I would also appreciate my friends and colleagues Rahat Masood, Hirra Anwar and Syeda Zahra Ali for sharing their knowledge, insights and providing feedback. Our combined interactive sessions and discussions come up with the result that helps me to resolve the critical issues during the research phase.

I would extend my appreciation towards KTH lab administrators who provide me useful resources in order to perform my work smoothly. In the end, I thanks to all my friends and everyone who supported me in any manner for the completion of my thesis.

Um-e-Ghazia

To my everloving Mother who has provided me every possible support to make this dream come true and Nana Abbu that has always been a source of inspiration for me.

Table of Contents

1	Introduction	1
1.1	Access Control	1
1.1.1	Access Control Matrix	3
1.1.2	Role Based Access Control	5
1.1.3	Attribute Based Access Control	6
1.2	Access Control Policy Languages	7
1.2.1	A P3P preference exchange language (APPEL)	7
1.2.2	Customer Profile Exchange (CPEXchange)	8
1.2.3	Enterprise Privacy Authorization Language (EPAL)	8
1.2.4	Declarative Privacy Authorization Language (DPAL)	9
1.2.5	Extensible Access Control Markup Language (XACML)	9
1.3	Web Application Security	9
1.4	Conclusion	10
2	Research Methodology	12
2.1	Research Methodology	12
2.1.1	Theory	13
2.1.2	Hypothesis	13
2.1.3	Observation	14
2.1.4	Confirmation	14
2.2	Specific Contribution	14
2.2.1	Survey Paper	14
2.2.2	Conceptual Paper	15
2.2.3	Access Control for Web Based Applications	16
2.3	Validation of Results	16
2.4	Conclusion	17
3	Related Research	18
3.1	Overview of Literature Survey	18
3.1.1	Security Issues of Cloud Computing and Collaborative Environment	19

3.1.2	Access Control Models on Cloud	23
3.1.3	Usage Control Model	30
3.2	Conclusion	37
4	Assessment Criteria for Cloud Based Access Control Systems	39
4.1	NIST Assessment Criteria for Access Control Systems	39
4.1.1	Least Privilege	40
4.1.2	Separation of duty	40
4.1.3	Management Complexity	40
4.1.4	Performance of access control enforcement mechanism	41
4.1.5	Policy Conflicts	41
4.1.6	Horizontal Scope	41
4.1.7	Configuration Flexibility	41
4.2	Comparative Analysis of Access Control Systems on Cloud	42
4.2.1	Role Based Access Control	43
4.2.2	Task-Role Based Access Control	43
4.2.3	Attribute and Role Based Access Control	44
4.2.4	Attribute Based Encryption Fine Grained Access Control	44
4.2.5	Hierarchical Attribute Based Encryption Access Control	45
4.2.6	Capability Based Access Control	45
4.3	Conclusion	46
5	A Brief Overview of UCON and XACML	47
5.1	Usage Control Model	47
5.1.1	Model Features	48
5.1.2	Model Processes	49
5.2	XACML	57
5.3	Conclusion	59
6	UCON Access Control Framework	61
6.1	UCON Access Control Framework	61
6.1.1	Components	62
6.2	Proposed Extensions in XACML	64
6.3	UCON Policy Builder	67
6.3.1	Authorization Process	67
6.3.2	Obligation Process	71
6.3.3	Condition Process	75
6.4	Conclusion	77

7	Validation of UCON Model Features	78
7.1	Evaluation Methodology	78
7.1.1	UCON Policy Ontology	79
7.1.2	Web Service Usage (Scenario 1)	80
7.1.3	Voucher Limit (Scenario 2)	82
7.2	Conclusion	83
8	Conclusion and Future Research	85
8.1	Conclusion	85
8.2	Future Research	86

List of Figures

1.1	Access Control Models	7
2.1	Deductive Research Approach	13
3.1	Complexity of Security in Cloud Environment	22
3.2	RBAC at API Level	24
3.3	Task-Role Based Access Control	25
3.4	Attribute and Role Based Access Control	26
3.5	Attribute Based Encryption Fine Grained Access Control . . .	27
3.6	Attribute Based Encryption Fine Grained Access Control . . .	28
3.7	Hierarchical Attribute Based Encryption Access Control . . .	29
3.8	Capability Based Access Control	30
3.9	Nego-UCON(ABC) Model	33
3.10	UCON Framework for Cloud Based Multimedia Content . . .	34
3.11	UCON Policy Enforcement Engine	36
5.1	UCON Classification of Subject, Object and Rights	48
5.2	XACML Access Control Framework	58
6.1	UCON Access Control Framework in XACML	62
6.2	UCON Policy Builder	68
6.3	Authorization Process	69
6.4	Subject Attributes	69
6.5	Resource Attributes	69
6.6	Model Parameters	70
6.7	Policy Target	70
6.8	Subject Attribute Designator	72
6.9	Resource Attribute Designator	73
6.10	Model Parameters	73
6.11	Condition Parameters	75
6.12	Model Parameters	76

LIST OF FIGURES

xi

7.1	UCON Policy Ontology	79
7.2	First Access Request	80
7.3	Usage Limit Reached	81
7.4	Voucher of amount 2000	82
7.5	Voucher Limit Reached	83

List of Tables

4.1	Comparison of Access Control Systems on Cloud	42
-----	---	----

Chapter 1

Introduction

In this chapter, we briefly describe the history and significance of some of the well-known access control models. These access control models need to formulate the access control policies in policy specification languages in order to provide the controlled access. We discuss access control policy languages in this chapter and their pros and cons are mentioned. Further we highlight the motivation for web applications security by describing well known standards for web application and web services security. Our primary focus is the security of web based applications that must be provided with an access control mechanism in order to provide enhanced protection for web resources.

1.1 Access Control

In early days of computing, there were large machines called as mainframes that has complex operations and performing inter process communication is a trivial task for them. Organizations used to consider these mainframe systems as an important asset because they are used for heavy data processing such as consumer statistics, transactions processing and enterprise resource planning. These mainfarme systems now have been evolved to personal computer (PC) and personal digital assistant (PDA) that require transfer of digital information and sharing of computation resources to perform collaboration. It is also important to protect these assets from malicious users to avoid the leakage of confidential information which causes damage to organization reputation in the business world. Therefore, providing controlled access for organization resources and information is essential to avoid the consumption of resources of illegal users.

Organizations use access control mechanism as one of the important security features that restricts the access of unauthorized users for confidential

data and sensitive resources. This security feature can be referred to either as physical security or computer security. Physical security includes devices such as biometric devices, metal detector, door locks, while computer security can be achieved with the help of authentication, authorization and accountability of customers. When request is generated for the required resource or data, users credentials are checked for identification and authentication after which access is either permitted or denied. Authorization handles access rights and permissions for particular user and can be obtained by access control policies that are formulated on the principle of who can access what. Digital systems having access control procedures must also undergo the process of auditing in order to be sure that resources are being utilized in accordance with access rights defined by policies.

First access control policies are formulated that define access permissions of individual for particular resource and then these policies are enforced by the system administrator in order to permit or deny the user request. Access control policies formulate the access criteria in order to secure the system processes and resources from unauthorized or malicious users. These policies define standard for consumption of critical resources according to system requirements and specification to avoid the leakage and illegal usage of confidential information. Access rules are defined that specify how consumer with particular credentials can get access for the required resource, there can be single or multiple rules that combines to form the access control policy. The level of access granularity for policy depends upon the rules or the criteria used to define these access rules.

Access control models define the mechanism for the formulation of access control policies and also highlight the main features that must be considered for specification of access restrictions. These models range from simple to more complex that make use of different technologies in order to address the changes in organizational structure, needs and capabilities. Each access control model has different requirements and performs well in corresponding scenarios and applications to provide the secure access mechanism for system resources. Increasing complex data access and sharing requirements of organizations demand for more sophisticated access control models that can provide flexible, granular and dynamic access restrictions in order to enhance the performance.

Until now large number of access control models have been proposed that implement the access control policies and each of these models have defined access restrictions according to different criteria. There are mainly two types of access control model; discretionary and non-discretionary access control models. Access permissions are defined by the owner of objects and resources in discretionary models while in non- discretionary models access rights are

specified by the system rather than individual owner of objects. Some of the well-known access control models are access matrix, role based access control and attribute based access control models. Initially access matrix was proposed which is the discretionary access control model to provide secure access in the context of operating systems, role based and attribute based are non-discretionary access control models that offers mediated access as per system administrator policy specifications. Brief description of these access control models are provided below.

1.1.1 Access Control Matrix

Access matrix was first introduced in 1971 (access-matrix, 2012) that provides the discretionary protection framework in order to define the access rights of subjects over the objects. As it depicts the matrix representation for the given authorization state so it is called as access control matrix. Access matrix is composed of three main entities; subjects, objects, rights and represents the system state by the tuple (S, O, A). S indicates the set of subjects, an active entity that is the consumer of resources or objects. O is the set of objects that is the passive entity for which access mediations are defined and has to be secured. A represents the set of actions or rights that are the allowable privileges or operations a subject can perform on particular object. The access matrix represents the access rights in a table in which a single row represents the subject and the objects are placed in columns, each cell of the table is the intersection of one particular subject and object and contains the action value that is allowable for that subject to perform on object. In order to reduce the complexity and improve the performance of systems having large number of entries, access matrix is divided into two variants; access control list and capability list.

Access Control List

Access control list (ACL) is referred to as column of access matrix containing the list of subjects that are allowed to perform specified action for that particular object. Each object has its own access control list that can be represented as follows:

```
ACL for File 1: (Subject 1, read), (Subject 2, not allowed),  
(Subject 3, read, execute)
```


ACL for Process 1: (Subject 1, read, write), (Subject 2, read, write, execute), (Subject 3, write, execute)

ACL for Process 2: (Subject 1, read, execute), (Subject 2, write, execute), (Subject 3, read, write)

Initially ACL has been implemented on UNIX operating system, later it is also used by multiuser systems where there is a need to restrict access for shared data and files. ACLs are flexible enough to be used for modern operating systems and networks where resource resides on remote location. It can also be used for relational database management systems to implement the data views that determine which subject is able to view the certain data elements.

Since ACL has a very simple concept and does not use much underlying technology, it can be easily deployed by organizations to define access restrictions. Instead of wide adoption there is a scalability issue related to ACL that degrades the performance of systems. ACLs have to perform the look up operation for each file, object and process access to entertain the user request that makes it inefficient.

Capability List

Second variant of access control matrix is capability list that represents the row of matrix having list of objects for related subject. Each capability list specifies the objects that are attached to a particular subject and represents the capability of each subject in terms of objects that can be accessed by that subject.

Capability List of Subject 1: (File 1, read), (Process 1, read, write), (Process 2, execute)

Capability List of Subject 2: (File 1, not allowed), (Process 1, read, write, execute), (Process 2, write)

Capability List of Subject 3: (File 1, read, execute), (Process 1, write, execute), (Process 2, read, write)

Capability lists can be also be used by the large enterprises to define the list of objects accessible by the specific user, but it cannot be used to define the different levels of access permissions for each user. Capability lists do not work for organizations having large number of users and each of them have different access requirements. Dynamic creation or deletion of objects that handles the changes of organizational structures are also not handled by these capability lists.

1.1.2 Role Based Access Control

Role based access control (RBAC) model was proposed in 1992 (role-based-access-control, 2012) that provides access restrictions according to the individual role defined by job functions. RBAC is non-discretionary access control in such a way that access permissions are assigned to individuals by the central authority; it can also implement the discretionary access control in case of file system permissions. Role assignment, role authorization and permission assignment are three main steps that are defined in RBAC. For example doctor can perform the operations of patient diagnosis, view the medical records in health care system while the nurse can only view some particular details of patient. Roles can be created or destroyed dynamically according to job functions, relationships and requirements that enhance the flexibility and performance of RBAC. Role hierarchy defines the inheritance relationship between roles that is also supported by RBAC increasing the granularity level for access mediations.

Access matrix model define access rights for objects that remain static while RBAC model has the ability to dynamically manage the creation and deletion of roles as per enterprise requirement. RBAC formulates the access permissions for operations that are useful to the organization rather than assigning access rights to low level objects in access matrix. For example, access matrix model allow write access to file but cannot explain how this file can be changed whereas RBAC defines the access permissions for operations such as creation of bank voucher and verification of transactions in financial applications. RBAC model supports scalability for large enterprises which is the main drawback of ACL.

RBAC is mostly implemented at application level, where this model is the main component of enterprise middleware. This single middleware component then can then be used to restrict access to large number of system and resources. Assignment of roles by dividing individuals into groups does not allow creating granular access permissions, specific mechanisms can be used in order to exclude or include particular roles or groups. This technique still cannot provide the granular access control which can be achieved based

on selected attributes.

1.1.3 Attribute Based Access Control

Attributes are the basic building blocks according to which access permissions are defined in attribute based access control (ABAC). Attributes are the labels or properties that can be used to describe the authorization entities and access requests. ABAC is next generation authorization model that is widely used to provide the dynamic, context aware access control for different applications. Attributes are assigned to subject, resource, action and environment in order to validate the access request in ABAC model that improves the accuracy of access decision than the other models. Subject attributes include the user ID, role, age and object attributes include accessing list, usage limit and resource validity etc. Action are different operations that subject performs on objects like read, write, create, delete, modify and their corresponding attributes are action type, action status whose value depend upon the context in which it is used. Distinguishing feature of ABAC model are the environment attributes that handles the system related parameters and restrictions such as ip address, location, time, protocol etc.

ABAC model handles the dynamic environment conditions along with subject and object attributes that makes it more reliable and flexible. RBAC model can be considered as the portion of ABAC model in such a way that it only considers the role attributes while assigning access permissions. Increasing the number of attributes for subject, object, action or environment defines the more refined access control rules that cover the different aspects of access request before granting permission to user. ABAC is well suited for distributed environment to provide the access mediations and does not require centralized authority as in RBAC model to handle the role and permission assignments. Another advantage of ABAC model is that there is no need for the consumer to be known in advance, access is granted to users as long as they meet the certain criteria of attributes. This ability of ABAC model makes it useful for organizations where people leave or join arbitrarily in order to manage the changes in organization structure.

Usage Control Model

Usage Control Model (UCON) is a specific form of attribute based access control that handles the usage restriction for digital objects and generates the access decision based on multiple factors. Three access decision factors for UCON model are authorization, obligation and condition that cover the multiple aspects of access decision. UCON model is comprehensive enough

to incorporate the traditional access control models such as RBAC, trust management and digital rights management managing the access control scenarios for different applications and computing environments (Lazouski et al., 2010).

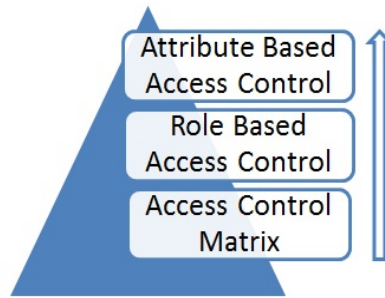


Figure 1.1: Access Control Models

1.2 Access Control Policy Languages

With the growing number of users for online web services, there must be procedures to restrict the access to critical assets of organization in order to maintain privacy and secrecy. The best way for avoiding any security breach within organizations is to formulize the policies regarding the critical infrastructure and customers confidential information. In fact policies are the best source to define the access and usage criteria for each and every element according to which required information is available to users in a uniform manner. There are languages that present the user defined policies in a precise way and allow users to communicate their privacy preferences in a specific format such as A P3P preference exchange language (APPEL), Customer profile exchange (CPExchange), Enterprise privacy authorization language (EPAL), Extensible access control markup language (XACML) and Declarative privacy authorization language (DPAL) (Han and Lei, 2011; Kumaraguru et al., 2007).

1.2.1 A P3P preference exchange language (APPEL)

P3P is W3C standard that is used to express the privacy policies for web sites, web browsers and different applications in a machine readable format. Websites express in P3P why user personal data is needed, how long this data is kept, what data is collected and who uses this data. These elements are encoded in XML format by web browsers in order to determine the websites

privacy policies by the web users. APPEL is a collection of users privacy policies that are compared with websites P3P policies to carry out the decision by the user agent of either accepting or rejecting the websites policies. There are inherent problems with the design of APPEL that creates complexities and contradictions in the matching pattern. Since P3P encodes the same policy differently, APPEL can accept the one encoding and may reject the other encoding of the same policy. Both P3P and APPEL does not provide any technical mechanism to check the user request against defined policies to provide the access control for the data and resources.

1.2.2 Customer Profile Exchange (CPExchange)

Customer profile exchange is a standard that enables the exchange of customer privacy related profile information and define the metadata to attach the privacy controls with this profile information. This standard defines the data model that not only represents the user of particular application but also provides the comprehensive view that helps to determine how customer interacts with enterprise. It provides the accumulative history and records of customer activities that improves the relationship of customer with enterprise and also service and support as per customer requirements. CPExchange integrates both online and offline customers in XML based data model to be used in various enterprise on and off web applications. Despite all these benefits, it is not as sophisticated or generic to handle the changing organizational requirements in order to meet the challenging issues.

1.2.3 Enterprise Privacy Authorization Language (EPAL)

Enterprise privacy authorization is an XML based language that enable organization to formulate their authorization policies for inter and intra enterprise for business to business privacy control. It can interpret the human readable policies into machine language to manage the data access for authorization decision. Privacy policies formulated by EPAL contains subject, resource, action with the help of identifiers that can be directly mapped to actual objects in order to support the directly enforceable policies. EPAL mainly concentrates on core privacy policies of enterprises rather than data model and user authentication details. It mainly targets the data access preferences according to enterprise specifications and does not handle the general data access control policies. Therefore it is not widely deployed as XACML to be used as general access control policy language.

1.2.4 Declarative Privacy Authorization Language (DPAL)

Declarative policy authorization is a formal language that allows creating the authorization policies for both customers and enterprise. DPAL policy is interpreted by taking into consideration of all applicable statements by enforcing all the rules of policy unlike in EACL. This feature introduces local reasoning and policy combination and concatenation that mitigate the risk of not knowing the authors intention. Since DPAL does not terminate the evaluation of mid policy, it creates inconsistency that must be detected through some technique in order to improve performance. Moreover DPAL policy expression is dependent on EPAL and can only represent those safe policies that are expressible in EPAL.

1.2.5 Extensible Access Control Markup Language (XACML)

XACML is an OASIS standard of policy specification language based on extensible markup language (XML) representing two way communication mechanism between requestor and responder. It is widely used as policy language standard both in industry and academia that handles the access management in distributed systems. It includes components like tags of different elements and attributes, functions, combining algorithms that can help to create desired criterion corresponding to scenario. Distinguishing feature of XACML is target element that specifies the corresponding subject, resource and action for which policy must be valid. Policy sets, policy and rule can have their individual targets to define the hierarchy of user request evaluation. Request verification is carried out first by matching the policy set target components with access request, after that with policies target and finally with rules target (Abi Haidar et al., 2006).

1.3 Web Application Security

There are different standards and technologies for web applications security that mainly addresses the integrity and confidentiality of request and response messages. OASIS (Organization for the Advancement of Structured Information Standards) has proposed different standards for web based applications in order to provide the security functions such as confidentiality and integrity of messages. OASIS web services security (WSS) SOAP message security (Nadalin et al., 2004) specifies the integrity and confidentiality

for SOAP messages with the help of XML encryption and XML signatures in order to provide end to end security. This specification also explains the mechanism to attach the security token with the message and their binary encoding. OASIS web services security username token profile (OASIS Website, 2006) discuss the phenomenon how web user can be authenticated with username in order to verify the request. In addition to this, secret passwords can also be used along with username token to validate the identity for web service producer. OASIS web services security X.509 Certificate Token (OASIS Website, 2006) describes the authentication framework with X.509 certificates that are used to identify the user public key. This public key is then further used to verify the encrypted SOAP message. OASIS web services security SAML token profile (OASIS Website, 2006) explains the use of SAML (Security Assertion Markup Language) assertions as a security token with SOAP messages. OASIS web services security Kerberos token profile suggest the authentication mechanism of SOAP messages with Kerberos tickets. This profile also explains how to encode the Kerberos (OASIS Website, 2006) tickets in order to perform encryption and add signature with SOAP messages. OASIS web services security Rights Expression Language (REL) profile (OASIS Website, 2006) describes how to use the REL with the web services. These web services standards do not explain how the usage of web services can be restricted to avoid misuse by unauthorized users.

NIST has provided guidelines to secure web services (Anoop Singhal, 2007) which discuss the authorization models such as RBAC and ABAC that are suitable for distributed nature of web services. RBAC can provide access restrictions for web services operation by the administrator, developers and any other privileged role in a coarse grained manner. On the other hand ABAC provides fine grained access control policies for web services that are more flexible and suitable for distributed environment as compared to RBAC.

1.4 Conclusion

In ninety decades, more work has been done in the access control domain and number of access control models has been proposed. We have observed currently ABAC model is most widely used to provide the fine grained access control in order to offer the mediated access to critical resources. UCON model is one of the attribute based access control model that has the distinguishing features of attribute mutability and decision continuity which is the major requirement of today's dynamic and distributed environment. We have worked out on this model to be used for web based applications to satisfy their dynamic authorization requirements. In order to achieve this, UCON

profile has been proposed in XACML i-e OASIS standard of policy specification and also acts as request/response phenomenon. Also the UCON access control framework has been suggested that can be integrated with different applications to avoid the unauthorized access of resources.

Chapter 2

Research Methodology

In this chapter, we explain the conceptual methodology of our research in terms of theory, hypothesis, observation and confirmation. We discuss briefly our specific contribution in terms of research publications and practical implementation of Usage Based Access Control Model for web based applications. Finally, we provide the methodology for the validation of research results.

2.1 Research Methodology

There are two basic methods through which research can be carried out; inductive and deductive research approach. Deductive research approach works in top down fashion where research is carried out from general to more specific, researcher formulates hypothesis from the theory and design strategy to validate the formulated hypothesis. Inductive approach works opposite to that of deductive approach, also called as bottom up approach in which the researcher formulates the hypothesis from specific observation and ends up with the theory or conclusion as per research problem. In deductive approach, conclusion follows from available facts while inductive approach draws conclusion based on some assumptions that introduce uncertainty. As research problem has been identified from the already available laws and observations in deductive approach, it is more reliable and authentic than inductive approach. Deductive approach is mostly used to test and analyze the initially defined hypothesis and observe the conclusion. The current research is also based on deductive approach which has theory, hypothesis, observation and confirmation phases. Our research has been carried out by formulating the hypothesis from literature survey and then observations have been made to support this hypothesis which is verified at the end.

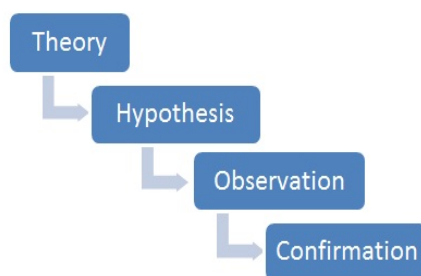


Figure 2.1: Deductive Research Approach

2.1.1 Theory

First the problem domain has been identified in which literature survey has been conducted. There are two directions in which survey has been done; Cloud computing security issues and access control models. Cloud computing security issues have been identified out of which we pursue our research in access control that is one of the vital security issues. On the other hand, different access control models have been studied that have been proposed until now. Detailed survey of UCON model has been done in which model features have been highlighted that can be useful for various applications.

2.1.2 Hypothesis

Based on literature survey in access control domain, following hypothesis has been formulated:

Is Usage Based Access Control Model feasible to provide dynamic authorization for web based application(s)

We have proved this hypothesis during our research by the evaluation of UCON model policies for web based application. Two example scenarios have been simulated in which UCON access control policies continuously monitor the dynamically changing attribute values throughout the access phase. This continuous monitoring improves the accuracy of dynamic access decision for web based applications that verifies the above proposed hypothesis.

2.1.3 Observation

At this phase following observations have been made in order to support the hypothesis:

- UCON model access decision depends on multiple factors that improves the authorization decision
- UCON model handles the update in attribute values to cope the dynamic changes
- UCON model evaluates the decision throughout the access phase that improves the access decision for web based applications
- UCON model incorporates the traditional access control models that can be used to provide the access control for web services of web based applications

2.1.4 Confirmation

The UCON policy specification module has been developed that can create the UCON policies having all the required attributes and identifiers. This module is able to formulate the policies of all UCON processes that are based on authorization, obligation and condition factors. These defined policies are then evaluated with the help of two scenarios for web based applications that conforms the above defined hypothesis. UCON model features are fully satisfied in these formulated scenarios that are the achieved results of current research.

2.2 Specific Contribution

In this research we have solved the following issues and created the following contributions in the access control domain:

2.2.1 Survey Paper

In our paper entitled "*Comparative Analysis of Access Control Systems on Cloud*", authors Um-e-Ghazia, Rahat Masood, Muhammad Awais Shibli, published in IEEE proceedings of 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, Japan, August 8-10, 2012, we have analyzed existing Cloud based access control systems and evaluated those using NIST defined

access control systems evaluation criteria. Based on our analysis we have proposed future research direction in the domain of access control systems for Cloud based environments, which will eventually pave the way towards Cloud adoption. Motivation behind this idea is that access control mechanism is considered as an obligatory part of security in order to prevent sensitive data from unauthorized access of malicious users. Since Cloud environment has to offer services to various users, number of access control challenges are involved depending upon the varying security requirement of access requests. Literature does not highlight the assessment features for access control systems in Cloud environment, so there is a need to specify some factors according to which access control systems are evaluated for Cloud environment. In this paper, we have selected these NIST evaluation parameters; least privilege, separation of duty, management complexity, enforcement mechanism, policy conflicts, horizontal scope and configuration flexibility. Low, medium and high level has been assigned to access control models according to the satisfaction level of NIST evaluation features in the selected model (Masood et al., 2012b).

2.2.2 Conceptual Paper

In our paper entitled "*Usage Control Model Specification in XACML Policy Language*", authors Um-e-Ghazia, Rahat Masood, Muhammad Awais Shibli, published in the Springer LNCS series of 11th International Conference on Industrial Management and Information Systems, Italy, September 26-28, 2012, we have suggested the interpretation of UCON model in extensible access control markup language (XACML) which is an OASIS standard of access control policies. We also highlight UCON model features by explaining its core processes and characteristics with respect to the case study of financial application. UCON is suitable for the distributed environment of grid and Cloud computing platforms however the proper formulation of this model does not exist in literature in any policy specification standard. It is for this reason that UCON is not widely adopted as an access control model by industry, though research community is now paying attention to make standard policy specification for this model. Since UCON can facilitate the diverse range of applications like digital rights management (DRM), health care systems and social networking, it is highly encouraged to provide the formal specification of model in generic policy language like XACML. There is a need to define the separate profile of UCON in XACML that will enable organizations to adopt this flexible model. Also to guarantee the accurate access decision in different deployment scenarios, it is mandatory to propose the required alterations and additions in generic policy language

of XACML which is not developed so far. We wish to propose the implementation of UCON model in XACML by incorporating additional elements and specify the information flow of different UCON processes in this paper (Masood et al., 2012a).

2.2.3 Access Control for Web Based Applications

Web based applications provide user-friendly platform having sophisticated and interactive applications that are accessed from central server. They provide quick access to corporate resources, massive collaboration and refined interface to interact with customers in order to enhance the business productivity and efficiency. Since the web applications are easy to access so they can be used by any unauthorized or malicious user in order to get access to backend database. SANS institute have highlighted different security issues of web application and provides the checklist for security features that must be catered in web application design phase (Bayse, 2004). Authorization and access control has also been mentioned in this checklist that is required to provide access restrictions and avoid the unauthorized usage of web application resources. Web based applications offer different services to customers varies form online data storage or retrieval to dynamic creation of images, therefore an access control model is required to handle the dynamic nature of these services and user requests as well. Since web based applications provide easy access to data and resources, access control model for these applications must generate the access decision that is based on multiple factors rather than single factor. Therefore we wish to propose the UCON authorization model that evaluates the user request for web based applications to handle the dynamic changes in access decision throughout the access phase. In this way access decision factors are monitored continuously during the whole resource usage phase that prevents the unauthorized user to access the application resources.

2.3 Validation of Results

To determine the impact and interpretation of ones research, validity is one of the main factors that must be considered throughout the research phase. Usually, validation is performed to prove that the identified question has been solved and is either useful for community. There are different types of validity that has been defined as part of research methodology; content validity, statistical conclusion validity, constructs validity, internal validity and external validity. Research is generally validated in order to verify the

truthfulness of achieved results and determine whether research measures what it intended to measure. Through this validation, research worth is calculated in such a way that how it plays an important role to broadly addresses the issues of related domain.

One of the main goals achieved by conducting this research is the assessment of Cloud based access control systems based on NIST defined evaluation features. There are few access control models that have been proposed until now for Cloud environment, but none of them can fully satisfy all the requirements for Cloud distributed environment. There is a need to highlight those features that must be catered in access control models to handle the dynamic authorization requirements of Cloud computing. We have concluded that usage control model can be considered as suitable access control mechanism for Cloud environment that has the ability to generate the accurate decision based on multiple factors and can also handle the updated attribute values.

Another aspect of this research is how to interpret the UCON model in any policy specification standard so that it can be widely used for different applications. Some of the additions have been proposed in XACML policy language in order to support the specification of UCON model policies. UCON policy specification module has been developed that creates the policies of UCON processes according to user defined parameters. These policies have been evaluated with the help of two scenarios that clearly reflects the UCON model features.

2.4 Conclusion

We have observed that the inductive and deductive research methodologies are both useful in carrying out a valuable research. The selection of these methodologies depends upon the research phenomenon that is used to carry out the research. It can also be possible to evaluate the research with the help of both inductive and deductive methodologies in order to validate the results. It is also observed that the research impact mainly depends upon arguments and research methodology that are used to verify the expected research results.

Chapter 3

Related Research

In this chapter we categorize our literature survey into three main categories (a) security issues of Cloud computing and collaborative environment (b) access control models on Cloud (c) Usage Based Access Control Model (UCON). First we highlight the security issues of Cloud computing and different challenges of collaborative environment in this chapter. Then we discuss the access control requirements for Cloud computing and detail description of Cloud based existing architectures of access control models. In the end, we give the overview of UCON model based applications and systems and the further enhancements proposed for this model.

3.1 Overview of Literature Survey

Cloud computing is a logical platform providing wide variety of services to individuals as well as large organizations. Instead of having significant advantages, there are large numbers of security concerns identified in the literature that avoid customers in adopting this platform. These security concerns are covered in the first category of security issues of Cloud computing and collaborative environment. One of the major security concerns for Cloud environment is that the customers data resides in third domain so access control model is required allowing data owners to manage access for their resources. Access control models for Cloud that have been proposed until now are discussed in second category of access control models on Cloud. These models have been critically analyzed to highlight their pros and cons in the context of Cloud and collaborative environment. Based on this analysis, UCON model is considered as suitable for any collaborative environment because of its significant advantages over other models. The third category of literature survey comprises the detail discussion of UCON model features

and its applications in Cloud and collaborative environment.

3.1.1 Security Issues of Cloud Computing and Collaborative Environment

Background

Today Cloud computing is providing features like flexibility, unlimited storage capacity, easy and quick way to access resources and cost reduction. Despite all these benefits, providing secure and trusted environment is still major issue in this paradigm. There are various security issues and challenges involved that must be catered to provide the secure and trusted platform for customers (Calero et al., 2010; Olden, 2011; Kaefer, 2010).

Related Work

In paper (Sengupta et al., 2011), major security concerns and vulnerabilities are highlighted in order to specify the future research directions in this domain. These issues are discussed from three perspectives; Cloud provider, Cloud consumer and third party authorities which are responsible for auditing and forensic tasks on Cloud. Security issues are categorized into four types; Cloud infrastructure platform, data, access and compliance. First category of Cloud infrastructure involves security concerns associated with storage networking and security vulnerabilities of physical data center of Cloud. Second category of data includes the concerns like data confidentiality, data integrity, tracing of data origin and its representation etc. Access category contains the security problems of AAA (Authentication, Authorization, Access Control), access management and encrypted communication of confidential Cloud data. Compliance category includes the regulatory issues of Cloud based activities like auditing, tracing of different operations and their compliance concerns. Security concerns are highlighted in the form of question statements from the point of view of Cloud consumers and third party authority, and then their technical implications are discussed from Cloud provider perspective. Four security concerns are explained which are as follows:

Concern C1: Is my Cloud-services providers physical and software infrastructure secured? Implication: Secure physical computing, storage and network access environment

Concern C2: What happens to my data in Cloud? Implication: Ensure effective data management including integrity, confidentiality and privacy

Concern C3: Are users accessing Cloud services really mine and can all my genuine users get seamless and secure accessibility? Implication: Ensure proper access control and identity management

Concern C4: Are Cloud providers compliant with regulation? Implication: Ensure proper regulatory compliance

Other than all these issues, advanced security challenges of Cloud computing are highlighted which are abstraction, lack of execution controls, third party control of data and multiparty processing of Cloud data. Domains of trusted computing, information centric security and privacy preserving models are discussed in the context of Cloud for building secure model. In the end three steps security assessment framework are proposed which can further be extended by the incorporation of automatic security tools.

In paper (Subashini and Kavitha, 2011), Cloud computing security risks and breaches are discussed in each service layer of Cloud that results from the complex underlying architecture of Cloud. The deployment layer of architecture comprises of Cloud deployments models; public, private, hybrid and community Cloud. The layer above deployment layer includes three service delivery models for Cloud; Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) (Rimal et al., 2009). These service delivery models exhibit different characteristics such as on-demand self-service, ubiquitous network, measured service, multi-tenancy and rapid elasticity which are present in top layer of architecture.

SaaS layer offer large number of services to customers in order to perform their process execution tasks and operations. At this layer, Cloud providers must guarantee the availability of services, secure migration of data, isolation of different users data and enhancement of security functionalities as per customers requirements. There is also customer concern of how their data is being stored at different locations and level of control provided by the Cloud providers. Following is the list of security threats that have been highlighted on SaaS layer:

- Data Security
- Network Security

- Data Locality
- Data Integrity
- Data Segregation
- Data Access
- Authentication and Authorization
- Data Confidentiality
- Web Application Security
- Data Breaches
- Virtualization Vulnerability
- Availability
- Backup
- Identity Management and Sign on process

PaaS provide platform to customers in order to deploy their applications but providers still have to ensure the security requirements of network intrusion prevention and data unavailability between applications. How malicious users can react with these deployed applications and can exploit the vulnerabilities of application architecture is one of the main security concerns of this layer. IaaS mainly considers the secure virtualization of hardware components, physical and environmental security to provide the secure and trusted Cloud environment. In addition to it, IaaS layer security mostly depends on Cloud deployment model through which services are delivered to customers. Infrastructure does not only includes the hardware components but also consider the transfer of data through different paths, so there must be proper encryption algorithms, policies and protocols that ensures the secure data transmission within Cloud.

Cloud computing is a platform that offers various services and sharing of resources in order to perform collaboration. This collaboration has become challenge when there is no central authority to handle the communication between domains. The paper (Shehab et al., 2005) presents a framework that introduces the idea of secure paths to make the access decision locally instead of having global view. Access decision depends on users access history that represents the sequence of roles user acquired in current session. Framework

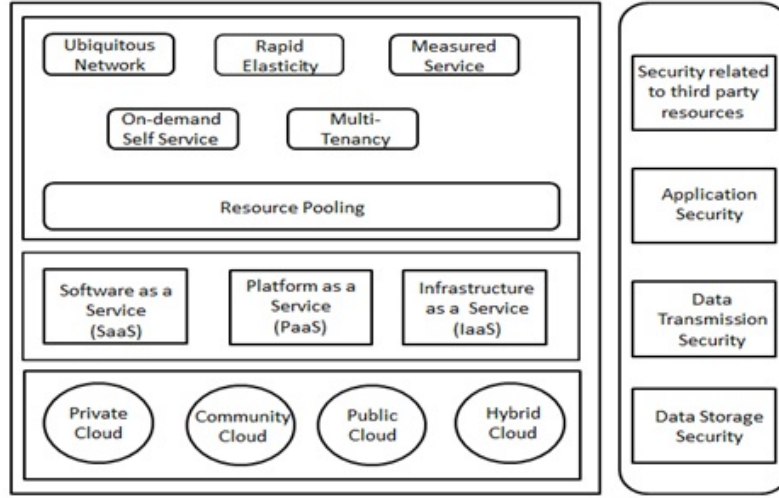


Figure 3.1: Complexity of Security in Cloud Environment

implies Chinese wall policy for making access decision and this policy is based on idea; user can access information if it does not have any conflict with the already accessed information. Further, this framework has been analyzed from security point of view and has been proven that this framework is resistant to path corruption, path replay, denial of service and violation of restricted relation attacks. Following are three main modules in the proposed framework that are used to calculate the localized access control decisions in collaborative environment:

Request processing module: It is responsible for evaluating the user request from another domain and accordingly making access control decision. Access request generated from another domain includes information of user access history and requested role that are used to create the access decision. Since each domain has a knowledge of its access control policies, decision is formulated by taking into account that access control policies of corresponding domain has not been violated.

Path Authentication module: This module authenticates the user path that is transferred along with user request through path signature and verification. On demand and proactive path discovery algorithms has been used by this module to enable multi hop collaborations in distributed environment. These algorithms allow domains to discover paths for roles in other domain and determine either there is path through intermediate domains.

Path Selection: Multiple paths have been returned by the path discovery algorithms out of which one path is selected by this module. Selection criteria have been defined based on selected features that helps home domain to select

the best suitable path. Path length, visited domains and composite domain reputation collectively combines to form the path selection criteria.

Analysis

Cloud offers services to organizations in same environment which raises security issues like: protection of resources from unauthorized users (Cloud owners, other Cloud consumers), data placement in third party platform makes it available outside the organization. These issues can best be resolved by access control mechanism in terms of mediating access. Further, Cloud computing provides platform that allows to share resources and collaborate with each other introducing access control challenges. There is a need of access control mechanism that can handle the dynamic and collaborative environment of Cloud in order to provide the secure usage of Cloud resources or data.

3.1.2 Access Control Models on Cloud

Background

Access control is one of the vital security concerns that allow customers to manage the access for their Cloud hosted applications. It allows data owners to create the required access control policies in order to restrict which users can access what. There are a number of challenging issues involved in providing authorization and access management in Cloud environment. Since Cloud offering services to various users; individual user, application, corporate user of organization, their access level must be differentiated in order to have controlled data access.

Related Work

Access control requirements for Cloud computing has been identified in accordance with the conceptual categorization of Cloud architecture defined in (Gouglidis and Mavridis, 2010). Cloud computing has been divided into four conceptual layers; entropy, asset, management and logic layers. Entropy layer is concerned with the distribution of objects among multiple organizations. Asset layer controls the fine grained access to the set of object that are being shared or provided as a service to customers. Management layer deals with the policy administration and management of policy violations at higher corporate level. Logic layer caters the quality of service factors that helps the customer to obtain the agreed levels of quality. The two most prominent access control models RBAC and UCON have been compared according to

these four conceptual layers of Cloud. As a result, UCON model satisfies most of the requirements that are specified by the conceptual Cloud layers as compared to RBAC.

Application programming interfaces (APIs) are responsible for monitoring and provisioning of resources in Cloud environment and they act as an interface between customers and services hosted on Cloud. A. Sirisha and G. Kumari have proposed Role based access control (RBAC) at API level in Cloud (Sirisha and Kumari, 2010). RBAC has been suggested for Cloud environment because it is mostly used for commercial organizations and enterprises. Access control is provided at two stages: user attributes for authentication and corresponding role validation. Initially, the user is authenticated through credentials, then roles are identified and corresponding access rights are assigned to the user. If user belongs to an organization which has already been registered, then authentication takes place by validating attributes in database through identifier. Same is the case in second stage where permissions are granted based on roles assigned to users.

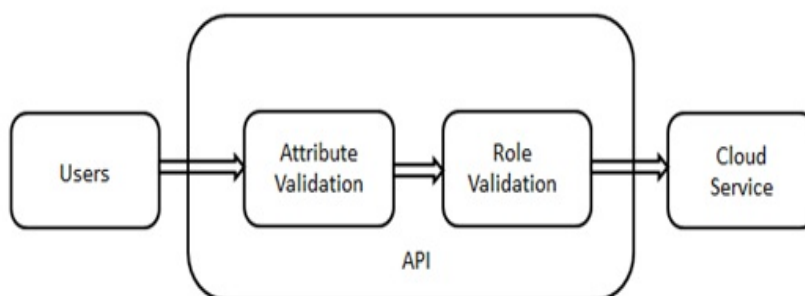


Figure 3.2: RBAC at API Level

Health care systems are multi-tenant including tenants such as hospitals, clinics, insurance companies and pharmacies so there are large number of users involved like patients, doctors, nurses, lab technicians, receptionists and IT professionals. There must be suitable access control mechanism for health care systems preventing unauthorized users to access data and important private details of patient. Traditional access control mechanisms are not appropriate for this environment as it consists of large number of resources and users that have to be managed. In RBAC (Ferraiolo and Kuhn, 2009), roles and tasks are not separated so the combination of these two parameters; Task Role based Access control (TRBAC) has been adopted for these systems by H. Andal and M. Hadi (Narayanan and Gunes, 2011; Thomas and Sandhu, 1998). Classification of tasks and activities has been done on the basis of

active and passive access control and inheritable and non-inheritable tasks. Tasks that are part of workflow requiring active access control and that are not part of workflow require passive access control. There are four classes in this way:

Passive access control: Private (non-inheritable), Supervision (inheritable)

Active access control: Workflow (non-inheritable), Approval (inheritable)

TRBAC models work in such a way that the users are assigned roles, roles are assigned to both workflow and non-workflow tasks and tasks are assigned to permissions as shown in Figure 3.3. Health care provider creates administrator that is a part of tenant and performs these assignments, manages relational database of roles, tasks, permissions, resources, policies and performs authentication and authorization of their respective domains. For password verification, MD-5 hashes have been stored in database and to avoid collision, random salt values are also considered. When user login into the system, verification is performed by matching passwords in database and corresponding rights are assigned related to these roles and tasks.

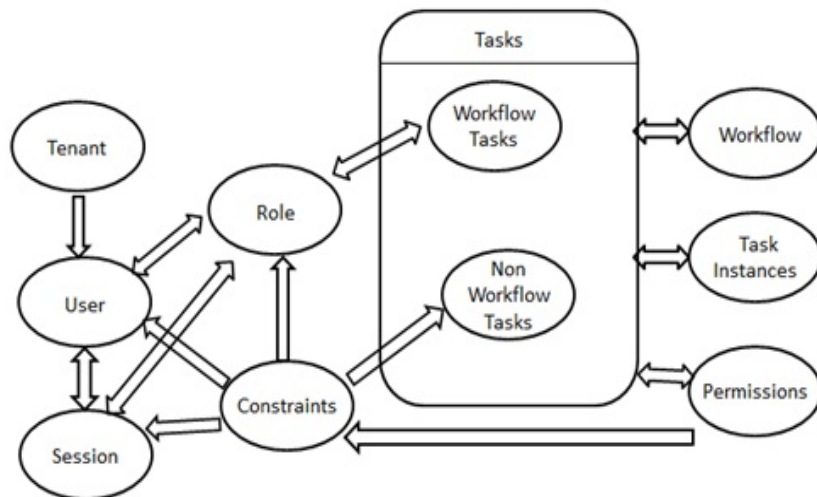


Figure 3.3: Task-Role Based Access Control

Privacy aware access control system (ARBAC) is proposed for Cloud that

is composed of two models; role based access control (RBAC) and attribute based access control (ABAC) (Mon and Naing, 2011). It provides secure access to personal identifiable information (PII). The system consists of the data owners, data users, Cloud providers and privacy managers. Data owners use virtual machines instances to host their data according to organizational permissions and specify the privacy preferences of data. User access the Cloud based services and data hosted by other data owners according to the defined access rights and policies. Cloud providers perform different operations on servers and their management tasks as defined by the data owner specified rules for Cloud users. Privacy manager is the essential component of the system, responsible for the specification of privacy policies based on user and data classification levels. In proposed ARBAC system, user requests to access data and provides corresponding subject, resource and environment attributes that are required for the service. Cloud service provider verifies the given attributes according to defined privacy policy in order to return the response of either permit or deny.

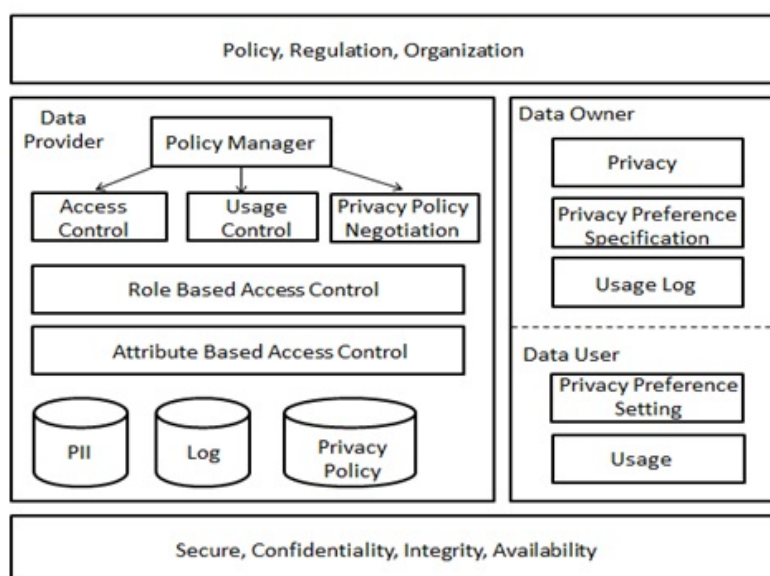


Figure 3.4: Attribute and Role Based Access Control

In order to provide fine grained access control for Cloud, attribute based encryption (ABE) has been suggested in (Li et al., 2010). Two main issues that are resolved by this mechanism are user accountability and efficient user revocation. There are two kinds of ABE; key policy ABE (KP ABE) and cipher text policy (CP ABE). In KP ABE, access policy is given in private key which is assigned to users and can decrypt only those files whose

attributes match with this policy. On the other hand, in CP ABE, access policy is defined in cipher text with each file and user key having different attributes, where access structure is defined over attributes assigned to each file. The user is granted file access only when its attributes list matches with the structure.

This system considers a system that consists of data owners, data users, Cloud servers and third party auditors. Data owners store encrypted data on Cloud servers due to large storage capacity and computational power, whereas, attributes have been assigned to users and third party performs auditing of all the events. Attribute authority is responsible for assigning keys to the attributes of newly entered user in the system which is then used for attributes verification. In the proposed system, broadcast encryption is performed by the data owner on user groups by selecting a random number and then uploading the cipher text on Cloud. Policy is defined in a way such that the user can decrypt cipher text if attribute list matches with the policy and intersection of dummy attributes of the list and cipher text contains at least number of attributes which is user defined threshold value. Tracing for illegal device is done through black box tracing algorithm with a defined detection procedure in case the suspicious user set is either small or large.

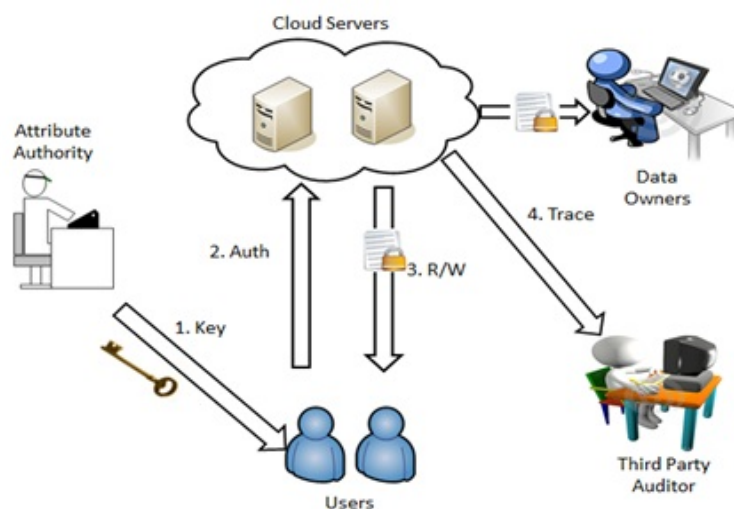


Figure 3.5: Attribute Based Encryption Fine Grained Access Control

G.Wang, Q.Liu and J.Wu presented a hybrid access control model in (Yu et al., 2010) involving attribute based encryption (ABE), proxy re encryption and lazy encryption. Each file consists of attributes and public keys corresponding to these attributes. Access structure of each file is defined in terms of logical expressions over attributes of public keys and corresponding

data file sets are defined against each user, achieving fine-grainedness. In this system, files are encrypted using symmetric keys that are further encrypted with key policy attribute based encryption. Health care scenario of this system is shown in Figure 3.6.

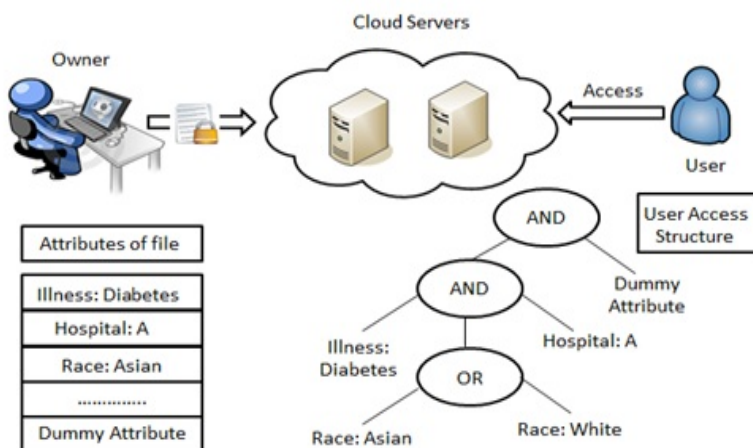


Figure 3.6: Attribute Based Encryption Fine Grained Access Control

Two stages of user revocation have been described in this proposed model; in first stage data owner determines minimal set of attributes modifies public key and master key for related attributes and generates proxy re encryption keys. Then sends user ID, minimal attribute set, proxy re encryption keys and public keys with his signature to Cloud servers and go offline. After this Cloud server revokes that user from user list and stores updated keys of corresponding attributes in attribute history list. In the second stage, Cloud server first verifies each user request to see whether the user is valid or not by checking user list. Users request is further processed in case the user is valid.

Hierarchical attribute based encryption combining hierarchical identity based encryption (HIBE) and cipher text policy based attribute based encryption (CP ABE) has been proposed on Cloud for access control in (Wang et al., 2010). Hierarchical structure consists of a root master (RM) and domain masters (DM), where RM corresponds to private key generator and is responsible for generation and distribution of keys and other important parameters. On the other hand, DM is like attribute authority in CP ABE and domain master in HIBE which handles delegation of keys and their distribution to users at next level. Users are allotted ID and other attributes, whereas unique identifier is assigned to each DM and attribute. Each users

position is defined by its own ID and public key of DM administrating the user. Mathematical algorithm for processes like create user and DM, encryption and decryption of files are prescribed. User revocation phenomenon has been specified according to this hierarchical structure in which processes of keys modification and re-encryption has been transferred to Cloud servers.

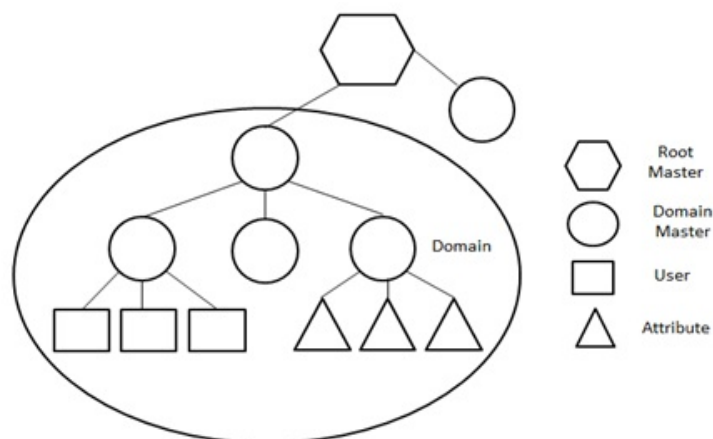


Figure 3.7: Hierarchical Attribute Based Encryption Access Control

There are three parties involved in offering Cloud services to customers who are data owner (DO), Cloud service provider (CSP) and user. Capability based access control system along with cryptographic techniques has been proposed for Cloud platform in (Sanka et al., 2010). Capability list is the row based decomposition of access control list which describes the list of objects accessible by a particular subject. It consists of user ID (UID), file ID (FID) and their corresponding access rights. Values for access rights are assigned as: 0 for read, 1 for write, 2 for both read and write. DO computes the MD5 hash of data files; encrypt it with the private key of himself and public key of CSP. CSP stores these encrypted data files and capability lists for users but the contents of data files are not revealed to them. Diffie Hellman algorithm is used to generate the symmetric keys which are shared between CSP and user for the purpose of secure communication. Symmetric key and its hash value are encapsulated with file to provide strong authentication and data integrity between user and CSP. In this proposed model, new user first performs the registration by DO sending UID, FID, nonce, timestamp and the required access rights. DO send the capability list, intended encrypted content and corresponding decryption keys to CSP after the user verification. CSP updates the capability list accordingly and also send registration confirmation to newly added user. After that user directly requests to CSP for data access and get encrypted response which

is then decrypted to get the session key and hash value. User calculates the hash value which is compared with original digest attached with message to confirm the data integrity.

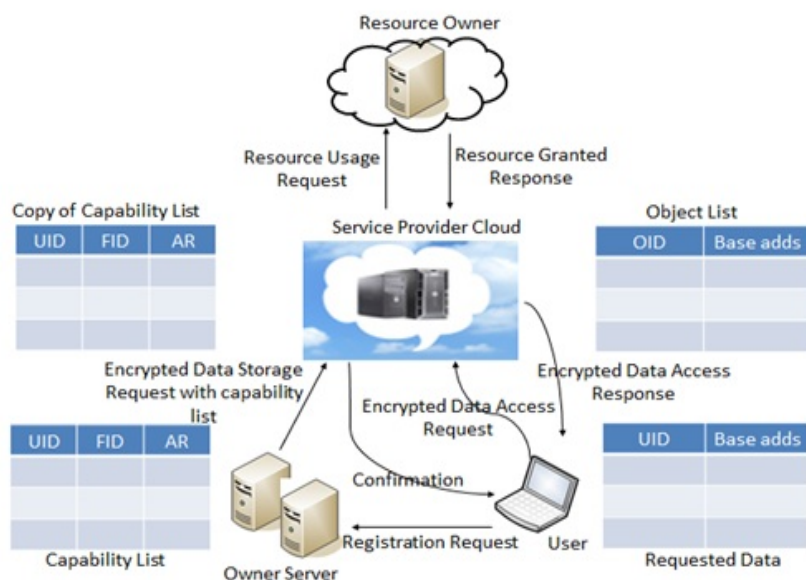


Figure 3.8: Capability Based Access Control

Analysis

Based on our literature, the existing access control systems target different aspects of Cloud authorization. They either target a specific scenario or provide a solution to any one of the problems such as efficient user revocation and delegation of rights. None of the proposed access control systems for Cloud include the features that are essential to provide the restricted access for Cloud services and resources. Therefore, access control model is required that is reliable and scalable for Cloud environment and target the security as well as usability aspects. Further, access control framework is highly encouraged to satisfy the user requirements and dynamic nature of Cloud.

3.1.3 Usage Control Model

Background

Traditional access control models do not completely satisfy the new requirements and challenges of modern systems. They provide controlled access to

resources until the access is granted to user and does not support access control during and after the request phase. With the advancement of systems according to organizational requirements, access control model is required to address requirements such as dynamic resource allocation and access decision in distributed environment.

Related Work

Usage Based Access control (UCON) model is an attribute based access control model that has been proposed by Park and Sandhu (Park and Sandhu, 2002). UCON is flexible and comprehensive model supporting access decision continuity and attributes mutability. Decision continuity means that access decisions are checked and enforced before or during the usage session called pre decisions and ongoing decisions respectively. Some attributes can be changed as a result of access known as attribute mutability which is of three types: pre updates, ongoing updates and post updates. There are three access decision factors in UCON model; authorizations, obligations and conditions. Authorizations are predicates (conditional statements) defined on subjects and objects; obligations are tasks that have to be performed before or during granting access and conditions are environmental properties that have to be satisfied. There are different UCON models considering mutable, immutable attributes, authorization, obligation and conditions which can be applied in corresponding scenarios (Park and Sandhu, 2004).

UCON model identifies three types of subjects; consumer, provider and identifye. Consumers are the subjects who make request to perform certain action on object. For example subscribers who want to access Cloud services are regarded as consumers. Providers are the individuals who own services and issue the rights to the requesting party. Distributor offering online services of shopping is considered as providers. Identifye is the entity whose confidential information is incorporated within digital object. Patient whose health records are stored in health care system is an example of identifye subject. It is an optional group of subjects which may or may not be present depending on system requirements however it is always present in case of systems having users confidential information. Depending on the job functions of subjects, three types of rights (actions) are specified namely consumer, provider and identifye rights which indicates the set of actions or privileges on digital objects (Park and Sandhu, 2002).

UCON model also classify the objects as privacy sensitive and privacy non sensitive objects that determines whether the object contains critical information of identifye subject or not. Improper management of privacy sensitive objects cause security breaches which results in data disclosure to

unauthorized users and compromising data integrity. There is another phenomenon of UCON model called as reverse UCON in which the position of consumers and providers are inverted depending on the scenario. For example while listening music, log file is maintained that accounts the customers usage information. This log file which is created as a result of exercising rights on object is called derivative object, which is also one of the category of UCON object. Derivative object also has the same level of protection as the original object and needs to be secured; consumer takes the place of provider and provider acts as a consumer for these objects (Park and Sandhu, 2002).

Formal models based on logic and process algebra of UCON model has also been proposed in (Lazouski et al., 2010). Further architecture of UCON model has been discussed in order to provide the protection for digital resources. Main components of this architecture include virtual organization (VO) and reference monitor (RM). RM is a trusted entity that allows specific subject to access particular object. So it must be tamper resistant and reliable. It consists of two main components access enforcement facility (AEF) or policy enforcement point (PEP) or usage enforcement facility (UEF) and access decision facility (ADF) or policy decision point (PDP) or usage decision facility (UDF). PDP consists of three modules authorization module, condition module and obligation module and PEP consists of customization module, monitoring module and update module. RM can be on either client side or server side or on both sides. Different enforcement mechanisms (which assures only authorize users can access resources and prevents them from malicious users) have been proposed. UCON has been implemented in different computing environments like collaborative environment (GRID), operating systems, database management systems and mobile systems. Most demanding implementation is in collaborative environment due to large number of its real time applications.

UCON model has been proposed in Grid systems in order to provide protection to shared resources in collaborative environment (Zhang et al., 2008). Grid systems have components like virtual organization (VO) and resource provider (RP). VO provides resources to users and RP manages how resources are assigned according to defined access control mechanism. PEI (policy, enforcement and implementation) methodology has been used in this framework of collaborative systems. In policy layer, UCON policies have been defined at both VO and RP level. In enforcement layer, hybrid model of push and pull mode of attribute acquirement has been used. In last layer, prototype implementation has been done through policy specification in Extensible access control markup language (XACML).

Cloud is one of the most commonly used collaborative and distributed

platforms nowadays; implementing usage based access control on Cloud can provide additional features of security and protection . C. Danwei, H. Xiuli, and R. Xunyi have proposed the UCON model for Cloud based services with the main focus on negotiation module; this model is named as Nego UCON (ABC) model (Danwei et al., 2009).

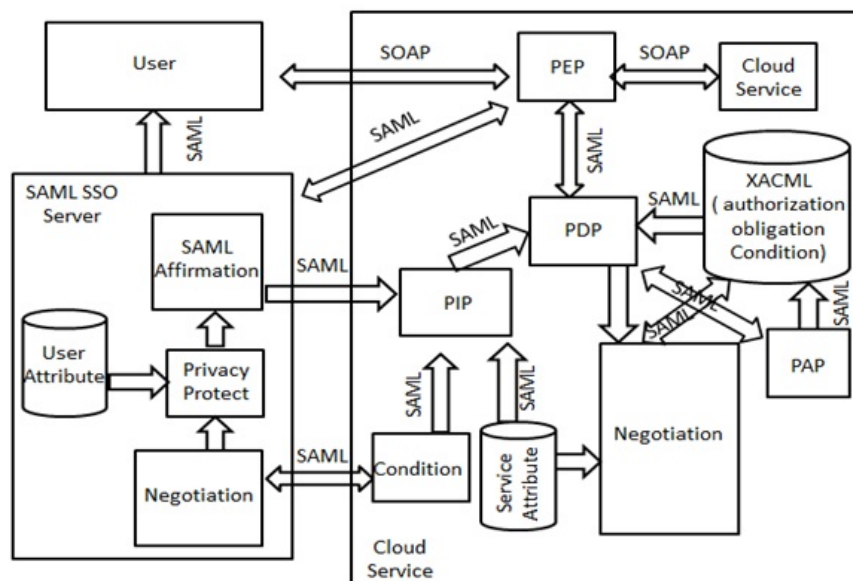


Figure 3.9: Nego-UCON(ABC) Model

Negotiation module enhances the flexibility of model in such a way that if user attributes mismatch then they can get a chance to access through negotiation by changing certain parameters. This Nego- module has three levels of negotiation; attributes query, attributes automatic negotiation and artificial negotiation. Attributes query asks for user attributes if the required number of attributes is not obtained otherwise automatic negotiation will occur that get the required attributes according to privacy policies. If automatic negotiation even has not got any result then Cloud service asks user to join artificial negotiation. This negotiation gives suggestions to users in order to change the privacy conditions so that they can get access to Cloud services. The proposed model consists of three main components; Cloud user, SAML server and Cloud service. Cloud user initiates the access request for Cloud service. SAML server comprises three modules; SAML assertion module, sensitive attributes protection module and negotiation module. SAML server issues assertions and generate responses to these assertions, its negotiation module is used to communicate with Cloud server for attributes, obligations and conditions. Attribute protection module protects the users sensitive attributes

when SAML assertions are issued and expose these attributes as per their privacy policies. Moreover, Cloud service component contains seven parts; Cloud service, XACML policy database, policy enforcement point (PEP), policy decision point (PDP), policy information point (PIP), policy administration point (PAP) and negotiation module. Cloud service is the service provider, PEP accepts user request and forwards to PDP, PDP evaluates access decision based on UCON policies, PIP obtains attributes, obligations and conditions and make them available to PDP for access decision and PAP creates and manage UCON policies.

Following detailed architecture has been proposed in (Ali et al., 2010) that provides fine grained access control to multimedia content residing on Cloud based on UCON model.

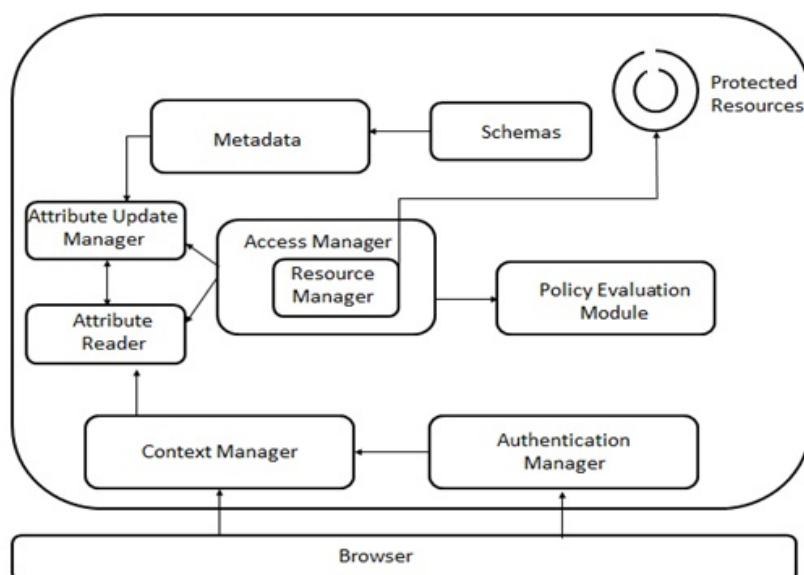


Figure 3.10: UCON Framework for Cloud Based Multimedia Content

Authentication Module authenticates different stakeholders and provides the interface that separates out the authentication mechanism from the rest of architecture. *Context Manager* entertains and manages the user request and forwards them to other UCON constructs. It supports the continuity feature of UCON model by keeping record of subject and object attributes. In addition to it, this module communicates with access manager to perform state transitions and receive notifications of access decisions from access manager. *Access Manager* evaluates the defined UCON policies according to corresponding states as per request of context manager. It also manages the attribute update manager and attribute reader to retrieve and change the

attributes. Policy parsing and predicates evaluation has been carried out by *Policy Evaluation Module* on the request of access manager. This module also updates attribute values passed by access manager that are then synchronized with the usage of digital content. *Schema* provides the standard XML schema according to which policies are specified and subject and object attributes are defined. *Resource Manager* protects Cloud resources through trusted platform module (TPM) that allows the secure transfer of objects between different Cloud virtual machines.

UCON model has been proposed for secure interoperation in multidomain environment in order to provide the flexible access control policies (Lu et al., 2009). There is a big issue of interpreting users attributes across multiple domains that has been resolved by the clear classification of attributes. Attributes have been classified into six different types:

- Local Domain Attributes
- Multidomain Attributes
- Temporary Attributes
- Persistent Attributes
- Mutable Attributes
- Immutable Attributes

There is a need of translating foreign attributes in local domain and these attributes must also have some knowledge of local entities. UCON policies have been formulated for LPM (local persistent mutable attributes) and LPI (local persistent immutable) attributes for dynamic attribute mapping in order to make interoperation across multiple domains. Attribute Mapping technique is based on five entities $\{a1, D1, a2, D2, m\}$, $a1$ is an attribute of domain $D1$, $a2$ is an attribute of domain $D2$ and m is the mapping mode that either LPM attributes of foreign domain $D1$ will be mapped to local domain $D2$ or LPI attributes foreign domain $D1$ will be mapped to local domain $D2$. This proposed mapping technique can resolve the security issues such as violation of separation of duty and cyclic inheritance that arises due to different parameters of multiple domains.

UCON model considers multiple decision factors in access decision which makes it better choice for access control mechanism as compared to other traditional access control models. UCON has pre and ongoing models of authorization, obligations and conditions that performs policy evaluation before

or during the access phase. In some cases, there are some of the tasks or actions required to be completed after the access phase which is not handled by UCON model. In paper (Katt et al., 2008), post obligations are introduced in order to increase the effectiveness of model or further increase the accuracy of access decision. Initially UCON state transition diagram has states of initial, requesting, accessing, end, denied and revoked. Since UCON continuously monitors the access decision during the access phase, there must be some state in transition diagram to represent the ongoing monitoring and transitions. Ongoing check state with two transitions ongoing request or ongoing permit has been suggested in transition diagram to represent the ongoing evaluation of rules. Moreover UCON enforcement engine has been proposed in XACML, which is based on security framework approach of PEI (policy, enforcement, implementation). In policy layer of this framework, UCON model with post obligations and extended state transition diagram has been suggested. Usage control enforcement model has been developed for enforcement layer of framework, which consists of three modules: enforcement point, decision point and session management point. Enforcement point accepts the user request and forwards it to session management point which is responsible for managing usage sessions. Session management point is further connected to decision point that has two sub components of attribute decision function and obligation decision function.

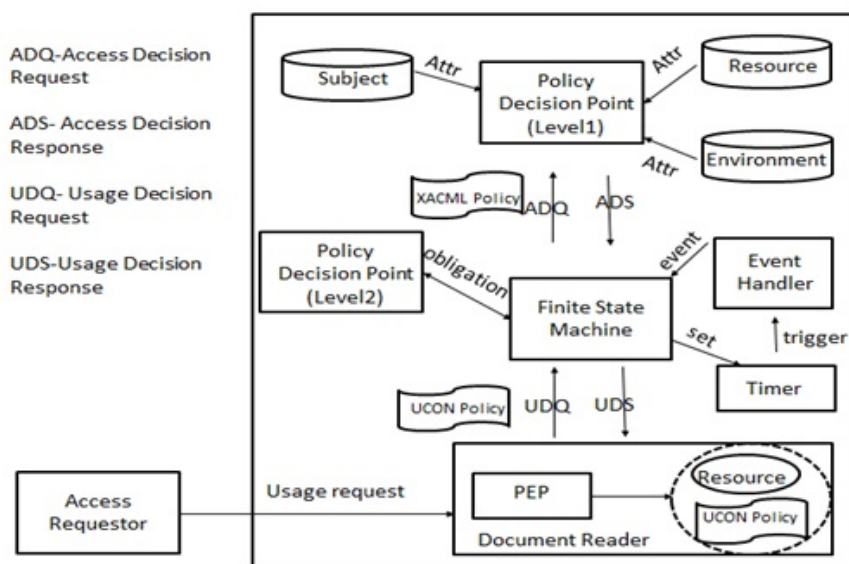


Figure 3.11: UCON Policy Enforcement Engine

Prototype architecture of this enforcement model has been developed in

XACML by the addition of modules like finite state machine, event handler, timer, document reader and an additional PDP along with existing one. *Finite state machine* represents the session management point of enforcement model and handles the access continuity feature of UCON model. *XACML PDP* will perform the functionality of attribute decision function of enforcement model and makes the authorization decision based on attributes. *Additional PDP* has been introduced which manages the obligation monitoring and obligation decision function of enforcement model. *Event handler* will handle the events of state transition and generates alerts to finite state machine with each transition. *Timer* is used to manage the time constraints, conditions and obligation re-evaluation. PEP is integrated with *document reader* application that is used to retrieve the corresponding UCON policy and required data. This developed architecture and UCON policy schema has been evaluated for health care scenario in order to check its functionality.

Analysis

UCON model still needs to cover the requirements of real world applications to provide improved protection for digital resources. UCON model must be flexible enough to create and destroy objects, subjects and their attributes in order to enhance model expressiveness. Comprehensive security analysis of UCON model must be performed to avoid security violation between administrative tasks that is the least explored area till now. Implementation of this model for real world applications are highly appreciated to provide the feasible enforcement mechanism that can support the UCON features of decision continuity, attributes management and obligation fulfillment. There is no proper specification provided in any policy specification language that can incorporate the UCON model core features. This is the major hindrance in model adoption for real world applications.

3.2 Conclusion

It has been examined that several security issues for Cloud computing are considered as one of the major hindrances in its adoption. Access control is also one of these security issues that need to be covered so that customers can manage access on their data residing in Cloud. Access control models for Cloud have been mentioned in literature but none of them incorporates the essential features to provide the dynamic authorization. This problem motivates us towards an access control model that can completely satisfy the dynamic authorization requirements of distributed environment. In this

regard, UCON model is considered as the most suitable authorization model that provides the continuous monitoring helps to achieve the dynamic access decision.

Chapter 4

Assessment Criteria for Cloud Based Access Control Systems

This chapter explains our proposed assessment criteria for Cloud based access control system that is based on NIST defined evaluation features for access control systems. These features are briefly described in this chapter according to authorization requirements for Cloud Computing. The evaluation of proposed assessment criteria is discussed with the help of comparative analysis for different Cloud based access control systems. Comparative analysis is done by assigning high, medium and low level for the assessment features according to their satisfaction level in corresponding access control systems which is shown in the form of table.

4.1 NIST Assessment Criteria for Access Control Systems

Cloud Computing, a relatively new concept and has gained an immense attention of research community in the past few years. Research and Development organizations and industry are investing a lot in Cloud based research and applications. Similarly on the consumers side, organizations are moving their business on Cloud to provide flexibility and conceive ever increasing computational power requirements. In spite of significant advantages, and its demand, different stakeholders are still reluctant to migrate to Cloud. A major hindrance is the absence of reliable and comprehensive access control mechanism for Cloud resources (Brunette and Mogull, 2009; Jansen and Grance, 2011). We have analyzed existing Cloud based access control systems and evaluated those using NIST defined access control systems evaluation criteria.

National institute of standards and technology (NIST) provides the secu-

curity guidelines and procedures for enterprises to securely execute their processes and operations. Since the access control mechanism is considered as an important element for determining the legal use of operations and resources, NIST has formulated the quality metrics for the critical analysis of access control systems (Hu et al., 2006). Access control systems must be evaluated on the basis of these metrics before making it functional and operational in practical scenarios (Jansen, 2010). It provides the better view of security features for the specific access control model incorporated in system that helps to find out whether the model is suitable or not for particular situation. As access control systems for Cloud must have to cater the variety of customers in dynamic environment (Kumaraswamy et al., 2010), their critical analysis is somewhat more challenging and demanding. We have selected the following NIST access control quality metrics for the performance evaluation of Cloud access control systems:

4.1.1 Least Privilege

Principle of least privilege indicates the granular level to which the user can access the required resource. It defines the level at which access mediations are assigned to user for resources such as defining access rights for users attributes. Cloud resources are accessed by large number of users having different access requirements, so access policies must be formulated to incorporate the least privilege principle. Being a reliable platform of services, Cloud environment must follow this principle in order to provide improved protection of resources.

4.1.2 Separation of duty

Separation of duty requires limiting object access to subjects in order to reduce the security breach in case of any inconsistency. According to it, rights are assigned to users corresponding to their duties. If it is not the case data is revealed to multiple users which cause leakage of information in certain scenarios. This feature helps in segregation of multiple users data residing on Cloud.

4.1.3 Management Complexity

Creation of access control policies needs to assign access rights which vary in different access control models. This parameter is considered important in evaluating the performance of access control system in order to determine the number of steps required to assign the user capabilities. Since the Cloud

platform needs to be highly flexible and reliable to handle the customer demands, this feature has a great impact to analyze the usability factor of access control systems.

4.1.4 Performance of access control enforcement mechanism

Enforcement mechanism involves the operations required for granting permission as well as the verification of user access request. Performance of access control enforcement mechanism is measured in terms of computational complexities of system. Computational overhead is the critical factor for Cloud access control as Cloud acts as a logical service model to entertain users achieving better performance.

4.1.5 Policy Conflicts

There are scenarios when two or more rules cause conflict in access decision, so there must be specified procedure to manage this situation. Cloud environment being a service delivery model must have to cater these conflicts while making access decision in order to improve the reliability.

4.1.6 Horizontal Scope

It defines the ability of access control system to be easily incorporated in different environments and platforms. Horizontal scope must be wide to cater all types of operations and processes and provide secure access to resources through access control system. Cloud Computing offer services for wide variety of customers so access control system must be compatible and provide interoperability for all scenarios.

4.1.7 Configuration Flexibility

Access control systems must be flexible and reliable to provide the clear separation of policies and mechanisms for better performance. This feature will allow handling the wider set of access control policies and providing resistance against attacks. Cloud platform must provide the robust access control mechanism that requires the little configuration management to be adopted for all applications and environments.

4.2 Comparative Analysis of Access Control Systems on Cloud

Cloud Computing paradigm possesses security concerns both at service provider side and client side. While sharing resources or utilizing services, the service provider must ensure that no illegal or malicious users exhaust these Cloud resources. On the other side, customers must also ensure that data privacy is maintained and Cloud servers are not being compromised. Cloud based authorization mechanism restricts unauthorized access to resources and these mechanism must include the required features to deliver services efficiently. In this section we discuss and analyze the access control systems that are suggested for Cloud environment according to NIST defined evaluation parameters. Some of them are encryption based in which access is restricted in such a way that the data owner reveals decryption key only to those users having the required attributes for the file being accessed. This analysis is shown in Table 3.1 by assigning three levels low, medium and high according to satisfaction level of the specified features.

Table 4.1: Comparison of Access Control Systems on Cloud

Characteristics	RBAC	TRBAC	ARBAC	ABE FGAC	HABE	CBAC
Least Privilege	medium	high	high	high	high	high
Separation of Duty	medium	high	high	high	high	high
Complexity	low	low	high	high	high	low
Performance of Enforcement Mechanism	medium	medium	high	low	low	medium
Policy Conflicts	high	high	high	high	high	high
Horizontal Scope	high	low	high	low	medium	low
Configuration Flexibility	high	low	low	low	low	low

4.2.1 Role Based Access Control

Proposed RBAC system(Sirisha and Kumari, 2010) follows the *least privilege* principle by assigning rights according to role specification and user attributes. In addition to this, roles are defined in a static manner and cannot be modified dynamically according to change in organization security requirements; therefore *separation of duty* is partially followed. It is relatively less *complex* because access control policies are defined based on roles and few of the user attributes. *Enforcement mechanism* involves two steps of attribute validation and role verification; hence this system performs well in distributed nature of Cloud. On the negative side, there is no defined procedure in the system to handle the *policy conflicts* that may occur due to the imprecise specification of policy. The proposed RBAC system provides access mediations at API level which increases its *scope* across different Cloud applications. Simple configuration of APIs for diverse range of applications eventually results in high *configuration flexibility* for this system.

4.2.2 Task-Role Based Access Control

Least privilege is supported in TRBAC system(Narayanan and Gunes, 2011; Thomas and Sandhu, 1998) by making an instance of the task which exists till the task is being performed and as soon as it is complete, access rights are revoked. The proposed system follows both the static and dynamic *separation of duty*. Static separation of duty is achieved at task assignment level i.e. no role can be assigned to two or more tasks at the same time. Dynamic separation of duty is performed through task instance which are created dynamically when task is initiated and prevents execution of two or more tasks by the same role. Policy specification is done on the basis of defined workflow and non-workflow tasks which in turn are associated to roles and roles are assigned to users. According to these defined tasks and processes, users are assigned capabilities which reduce the *complexity* of policy specification. *Enforcement mechanism* of the system consists of task validation and their corresponding role validation, making its performance better in case of health care systems. This is not the case for environments in which tasks and role parameters are not sufficient for accurate access decision. There is no proper mechanism defined to avoid the *policy conflicts* that arise between access decisions of different policies. Although the task and role parameters are suitable in multi-tenant health care systems; they cannot be considered appropriate in all Cloud based applications where there are consumers having different access requirements hence limiting the *scope*. The proposed system parameters need to be modified for environments that

demand multiple factors for access decision, reducing systems *configuration flexibility*.

4.2.3 Attribute and Role Based Access Control

Since ARBAC (Mon and Naing, 2011) is the composition of RBAC and ABAC, *least privilege* is supported by granting permissions according to specified attributes and role parameters in policy. *Separation of duty* is achieved in a way that each subject and resource is associated with particular attributes based on which job functions and access rights are defined. User and data classification levels are defined according to which privacy preferences and access policies are formulated. Hence *complexity* of defining policies becomes high with the increase in user classification levels. Policies are enforced by validating the defined attributes for subject, resource, environment and user roles, which improves the performance of *enforcement mechanism* and reliability. *Policy conflicts* avoiding procedure is not mentioned in proposed ARBAC system which may occur due to the difference in access decision of multiple policies. Incorporation of additional parameter like environment attributes (that can manage the system related properties and characteristics) helps in increasing *horizontal scope* of the system across different platforms and applications. The main decision factor is attributes according to which access rights and permissions are assigned in ARBAC. Management of attributes (subject, resource, environment) in different scenarios require detailed configuration modifications which results in low *configuration flexibility*.

4.2.4 Attribute Based Encryption Fine Grained Access Control

In the proposed system (Li et al., 2010), *least privilege* principle is followed by defining access structure for each user. If user access structure matches with the requested file attributes then access is granted to data hosted on Cloud. Access is defined at fine grained level which means that privileges are associated for the basic unit of data file. *Separation of duty* is followed in a way that jobs are defined for all the system entities; data owner, Cloud provider, consumer and third party auditor. Cloud provider can keep the encrypted data files, user can access them if their access structure is matched with the file attributes specified by the data owner. Third party performs auditing of access requests with the help of tracing algorithm in order to detect the malicious entity. Policy specification of this proposed system requires to define access structure for each user which may become *complex* because of their

varying access requirements. Data owner defines the threshold value in policy specification that represents the number of attributes to be matched for each user request. This threshold value is proposed for specific scenario and cannot cope with varying nature of Cloud applications. System introduces large overhead in terms of mathematical operations and algorithms which affects performance of enforcement mechanism. *Policy conflicts* are not managed by this system, which may occur due to difference between decisions of two or more access control policies. *Scope* for the proposed attribute based encryption system is limited because it is not preferable to use complex algorithms with lightweight applications as it reduces the system efficiency. Also such systems require great amount of time to execute the mathematical operations and algorithms thus introducing the delay in access response. The system is not flexible enough because it requires the management of complex operations which decrease its applicability in different environments. There is also an issue of key management and distribution to authorized users for decrypting required data files that result in low *configuration flexibility*.

4.2.5 Hierarchical Attribute Based Encryption Access Control

The proposed hierarchical technique (Wang et al., 2010) follows the *least privilege* by assigning the specific IDs and attributes to the respective domain at fine grained level. Tasks are divided for domain masters to handle the operations of attribute management and administration thus following the *separation of duty* principle. *Complexity* to specify access control policies is higher because tasks and processes are distributed at individual level and domains. *Enforcement mechanism* is complicated in terms of mathematical operations and functions that badly affect the system performance and efficiency. Also there is no defined method available to handle the *policy conflicts* in this proposed hierarchical system. *Scope* of the technique is limited to organizations having hierarchical structure of system entities. System tasks and processes are distributed at each node, but the addition of security feature for each single entity decreases the *configuration flexibility* and does not remain impressive for Cloud environment.

4.2.6 Capability Based Access Control

Least privilege is followed for this technique (Sanka et al., 2010) by assigning access rights for the basic unit of data file. Duties are clearly defined for Cloud customers with the specification of access rights in capability access list. Users can only perform the functions specified by data owners in

their corresponding list following the *separation of duty*. The system specifies policy by defining permissions in the capability list with user id and file id that will somehow simplify the policy creation process and reduces *complexity*. Performance of the *enforcement mechanism* depends on the key generation and hashing algorithms, which might be the bottleneck for this system. *Policy conflicts* are not managed in this proposed system. Capability list contains the static entities of users and their corresponding allowable objects which are not well suited for dynamic environment like Cloud. It does not consider the multiple factors for access decision which is the major requirement for distributed environments results in limiting its *scope*. Double encryption is used in the proposed technique to provide strong cryptographic strength through which keys management, configuration and their distribution to large number of customers become overhead. It will make the system inflexible to be adopted in different computing platforms decreasing its *configuration flexibility*.

4.3 Conclusion

After the detailed analysis of Cloud based access control systems, we have concluded that the above specified NIST defined access control features must be present in Cloud based access control systems. These features will help to achieve the better performance in dynamic and distributed environment of Cloud. We have analyzed that the access control mechanism for Cloud must be flexible and reliable to provide features such as delegation of rights and user revocation dynamically. Moreover, Cloud based access control system must monitor the changing attributes values of access control policies in order to perform the accurate access decision throughout the access phase.

Chapter 5

A Brief Overview of UCON and XACML

This chapter provides the basic understanding of UCON model core features and its processes. The three main decision factors of UCON are authorization, obligation and condition on the basis of which UCON processes have been defined. These processes have been explained in the chapter along with their examples. Further, XACML basic components and their entities and XACML flow of access control policies have been described in this chapter.

5.1 Usage Control Model

UCON being an attribute based access control model accommodates the security requirements by the addition of more than one decision factors which makes it more reliable and flexible (Park and Sandhu, 2002). This model primarily restricts the usage of digital objects and provides the efficient mechanism to include the traditional access control models. Previous access control models only encompass authorization rules in making access decision; rather UCON model also consider the obligations and environmental conditions. Moreover collaborative environments demand the need of enhanced provisioning and controlled access to digital resources. In addition to the immutable attributes that are explicitly modified by the administrator, ucon model also manages the system controlled mutable attributes by constant monitoring throughout the stages of usage session.

5.1.1 Model Features

UCON model identifies three types of subjects; consumer, provider and identifyee. Consumers are the subjects who make request to perform certain action on object. Providers are the individuals who own services and issue the rights to the requesting party. Identifyee is the entity whose confidential information is incorporated within digital object. It is an optional group of subjects which may or may not be present depending on system requirements however it is always present in case of systems having users confidential information. Depending on the job functions of subjects, three types of rights (actions) are specified namely consumer, provider and identifyee rights which indicates the set of actions or privileges on digital objects (Park and Sandhu, 2002). Apart from these, there are other actions as well that fall in the category to perform updates in attributes values during the phases of usage session which are termed as usage control actions (Zhang, 2006).

UCON model also classify the objects as privacy sensitive and privacy non sensitive objects that determines whether the object contains critical information of identifyee subject or not. Improper management of privacy sensitive objects cause security breaches which results in data disclosure to unauthorized users and compromising data integrity. There is another phenomenon of UCON model called as reverse UCON in which the position of consumers and providers are inverted depending on the scenario. Complete classification of UCON subjects, objects and rights is shown in Figure 5.1.

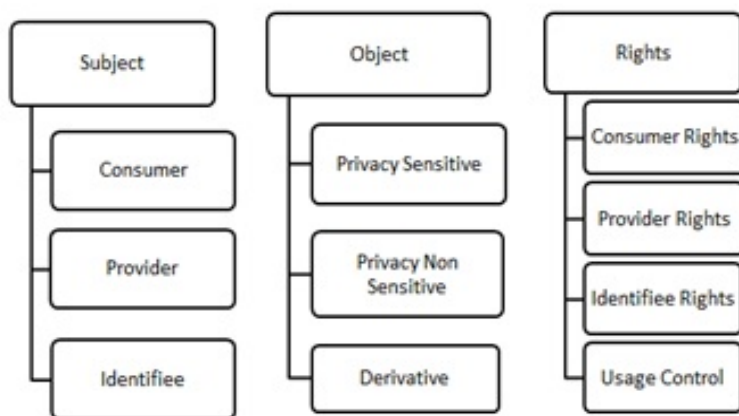


Figure 5.1: UCON Classification of Subject, Object and Rights

All the traditional models include authorization rules that need to be satisfied before the using the resources called as rights related authorization rules. UCON identify additional obligations related authorization rules

which are the set of actions related to access request to be completed before granting permissions of resources. Obligations are first time addressed by UCON model to improve the accuracy of access decision by enforcing users to perform certain actions before access. Obligations also act as functions that check whether the obligatory actions are fulfilled by the requesting entity or not. In order to further increase the accuracy of authorization decision, another factor to be considered important in UCON model are the environmental conditions such as ip addresses, current date or time. Conditions can be of two types; dynamic (stateful) and static (stateless). Dynamic conditions have constantly changing information so they needs to be evaluated for every update and static conditions do not have to be checked for each update during usage session (Park and Sandhu, 2002).

5.1.2 Model Processes

UCON identifies the processes of access decision as ABC (authorization, obligation, condition) models (Zhang, 2006; Park and Sandhu, 2004; Zhang et al., 2004). These models are specified on the basis of

- pre and ongoing phase of authorization, obligation and conditions
- pre, ongoing and post updates of attribute values

Below is the list of UCON processes that are specified depending on the combination of above mentioned factors:

- Pre authorization with immutable attribute
- Pre authorization with pre update
- Pre authorization with ongoing update
- Pre authorization with post update
- Ongoing authorization with immutable attribute
- Ongoing authorization with pre update
- Ongoing authorization with ongoing update
- Ongoing authorization with post update
- Pre obligation with immutable attribute
- Pre obligation with pre update

- Pre obligation with ongoing update
- Pre obligation with post update
- Ongoing obligation with immutable attribute
- Ongoing obligation with pre update
- Ongoing obligation with ongoing update
- Ongoing obligation with post update
- Pre condition with immutable attribute
- Ongoing condition with immutable attribute

Some of the models mentioned above are not useful in some scenarios i-e when decision is to be made before the access; ongoing updates are not useful like pre authorization with ongoing updates and pre obligations with ongoing update. These models do not have any impact on current usage decision rather it affects the upcoming access events. UCON model does not handle the changing environmental constraints for pre and ongoing models of conditions, so we will not consider the condition models with updates (Park and Sandhu, 2004).

UCON Authorization Processes

Authorization are functional rules defined over subject and object attributes that have to be evaluated before access is granted to determine whether the subject is allowed to perform a specific action or not.

Pre Authorization with immutable Attributes: Authorization rules are evaluated before access rights are exercised and there is no any update in attribute values during, before or after the usage. Predicates are checked when a request is generated or before an access is granted.

Definition: This process has following components:

$$\begin{aligned}
 & S, O, R, ATT(S), ATT(O) \text{ and } preA \\
 & (\text{subject, object, rights, subject attributes, object attributes and pre} \\
 & \quad \text{authorizations respectively}) \\
 & Allowed(s, o, r) \rightarrow preA(ATT(s), ATT(o), r)
 \end{aligned}$$

Example: In mandatory access control, subject can read object if subject clearance is greater than object classification or in case of write if object classification is greater than subject clearance.

Pre Authorization with pre update: Authorization rules are checked or update occurs in either subject or object attributes before access is granted. These rules must be evaluated to true before the pre update occurs.

Definition: This process has same components as pre authorization with immutable attributes, adds pre update procedure for subject and object attributes.

$$\text{Allowed}(s, o, r) \rightarrow \text{preA}(\text{ATT}(S), \text{ATT}(O), r) \\ \text{preupdate}(\text{ATT}(s)), \text{preupdate}(\text{ATT}(o))$$

Example: In DRM pre-paid credit, subject can access object if subjects credit is greater than objects value and as a result of it, objects value will have been deducted from subjects credit. An update to credit has been performed by the system before granting access to subject.

Pre Authorization with ongoing update: Authorization rules are evaluated before access is granted and update occurs during the usage session. This update cannot change access decision during current usage session but can affect future requests. Since the authorization rules are evaluated before the access and ongoing updating of attributes does not have impact on current session, it is not practically useful for real time scenarios.

Pre Authorization with post update: Authorization rules are checked before decision and update occurs after usage session.

Definition:

$$\text{postupdate}(\text{ATT}(s)), \text{postupdate}(\text{ATT}(o))$$

Example: DRM member-based metered payment allow subjects to read objects if both belong to same reading group and subjects expense is updated by deducting objects cost after the session.

Ongoing Authorization with immutable attributes: In ongoing authorization models, rules are evaluated continuously during the usage session and if the rules are not satisfied at any stage rights are revoked.

Definition: Ongoing authorization models have following components:

$$\begin{aligned}
&S, O, R, ATT(S), ATT(O), onA \text{ (ongoing authorizations)} \\
&\quad Allowed(s, o, r) \rightarrow true \\
&\quad Stopped(s, o, r) \leftarrow \sim onA(ATT(S), ATT(O), R)
\end{aligned}$$

Example: Employee has been assigned some temporary role with certificate for specific project in which he has to access sensitive information. His certificate is being continuously checked, if its number is present in certificate revocation list that temporary membership is revoked.

Ongoing Authorization with pre update: Authorization rules are evaluated during the usage process and update occurs before subject access the object. There are no checks involved before the subject starts access as in pre authorization models.

Definition: Ongoing authorization models have following components:

$$\begin{aligned}
&S, O, R, ATT(S), ATT(O), onA \text{ (ongoing authorizations)} \\
&\quad Allowed(s, o, r) \rightarrow true \\
&\quad Stopped(s, o, r) \leftarrow \sim onA(ATT(S), ATT(O), R) \\
&\quad preupdate(ATT(s)), preupdate(ATT(o))
\end{aligned}$$

Example: Object attribute is a list of subjects accessing this object and system attribute is system clock called as start time. When subject starts to access object, two updates have been performed; object list will be updated by adding requesting subject and current time of system clock will be assigned to start time attribute.

Ongoing Authorization with ongoing update: Authorization rules are checked during the usage process and also one or more update occurs during this process.

Definition: Ongoing authorization models have following components:

$$\begin{aligned}
&S, O, R, ATT(S), ATT(O), onA \text{ (ongoing authorizations)} \\
&\quad Allowed(s, o, r) \rightarrow true \\
&\quad Stopped(s, o, r) \leftarrow \sim onA(ATT(S), ATT(O), R) \\
&\quad onupdate(ATT(s)), onupdate(ATT(o))
\end{aligned}$$

Example: Two subject attributes are: status (busy, idle), idle time. In this case revocation is performed for the subject having longest idle time. Idle time will be calculated by checking the status of subject which is the ongoing update of attributes.

Ongoing Authorization with post update: Authorization rules are evaluated during the whole process and update occurs after the usage session.

Definition: Ongoing authorization models have following components:

$$\begin{aligned}
& S, O, R, ATT(S), ATT(O), onA \text{ (ongoing authorizations)} \\
& \quad Allowed(s, o, r) \rightarrow true \\
& \quad Stopped(s, o, r) \leftarrow \sim onA(ATT(S), ATT(O), R) \\
& \quad \quad postupdate(ATT(s)), postupdate(ATT(o))
\end{aligned}$$

Example: There is a policy that more than ten subjects or subjects having earliest start time are not allowed to access object, if this is the case access rights are revoked during the usage session. Post update occurs in object list of accessing subjects by removing the subject and assigning start time to null value showing that subject is no longer involved in access.

UCON Obligation Processes

Obligations are the core feature of UCON defined as actions related to access request that have to be fulfilled by the subject before giving access or during usage of objects.

Pre Obligation with immutable attributes: Pre obligations are kind of history functions that evaluate whether certain action has been performed or not. Attributes are not changed before, during or after the usage process.

Definition: This process has following components:

$$\begin{aligned}
& S, O, R, ATT(S), ATT(O) \\
& \text{obligation subjects, obligation objects, obligations (OBS, OBO, OB)} \\
& \quad \text{pre obligation predicates, pre obligation elements (preB, preOBL)} \\
& \quad \quad \text{preOBL is the subset of } OBS \times OBO \times OB \\
& \quad \quad \text{preFulfilled: } OBS \times OBO \times OB \rightarrow \{true, false\} \\
& \text{getPreOBL: } S \times O \times R \rightarrow 2^{\text{preOBL}} \text{ (function to select pre obligation} \\
& \quad \quad \text{elements)} \\
& \quad \quad \text{preB= preFulfilled (obsi, oboi, obi)} \\
& \quad \quad \text{if getPreOBL}(s, o, r) = \phi, \text{preB}(s, o, r) = true \\
& \quad \quad \text{allowed}(s, o, r) \rightarrow \text{preB}(s, o, r)
\end{aligned}$$

Example: Each time user requests for certain object usage, he has to click on license agreement.

$$\begin{aligned}
& OBS = S, OBO = \text{license agreement}, OB = \text{agree} \\
& \text{getPreOBL}(s, o, r) = \{(s, \text{license agreement}, \text{agree})\}
\end{aligned}$$

Pre Obligation with pre update: Obligations have to be performed before granting access and update occurs before the access is given.

Definition: It has the same definition as pre obligation with immutable attributes except it includes pre update procedure

$$\text{preupdate } (ATT(s)), \text{preupdate } (ATT(o))$$

Example: User has to click on license agreement to become registered user for future usage of service. After clicking once on agreement, update occurs in subjects attribute (registered) before granting permission.

Pre Obligation with ongoing update: Obligations have to be performed before granting access and there are one or more updates during the usage process. This model is not practically useful as obligations have to be checked before access and updates occur during the access which does not affect the current usage session.

Pre Obligation with post update: Obligation has to be performed before usage and one or more update occurs after the usage session.

Definition: It has the same definition as pre obligation with immutable attributes except adds post update procedure

$$\text{postupdate } (ATT(s)), \text{postupdate } (ATT(o))$$

Example: Customer has to click agreement as pre obligation for obtaining service and customer list will be updated by adding item after order has been placed.

Ongoing obligation with immutable attributes: Obligations have to be checked during the whole usage session either periodically or continuously. Ongoing obligation predicates must have to be true all the time while accessing objects.

Definition: This process has following components:

$$\begin{aligned} &S, O, R, ATT(S), ATT(O) \\ &\text{obligation subjects, obligation objects, obligations } (OBS, OBO, OB) \\ &\text{ongoing obligation predicates, ongoing obligation elements } (onB, onOBL) \\ &\text{onOBL is the subset of } OBS \times OBO \times OB \\ &\text{onFulfilled: } OBS \times OBO \times OB \rightarrow \{true, false\} \end{aligned}$$

getOnOBL: $S \times O \times R \rightarrow 2^{onOBL}$ (function to select ongoing obligation elements)
 $onB = onFulfilled (obsi, oboi, obi)$
 if $getOnOBL (s, o, r) = \phi$, $onB(s, o, r) = true$
 $allowed(s, o, r) \rightarrow preB(s, o, r)$
 $stopped(s, o, r) \leftarrow \sim onB(s, o, r)$

Example: Online service provider while giving service opens advertisement banner on client side, when it closes service is no longer available. So, opening advertisement banner during service usage is ongoing obligation which is continuously monitored during the service usage.

Ongoing obligation with pre update: Obligations have to be checked during the usage session, one or more update occurs before accessing the object and there is no usage control before the access.

Definition: It has the same definition as ongoing obligation with immutable attributes except includes procedure for pre update

$preupdate (ATT(s)), preupdate (ATT (o))$

Example: In online application, advertisement has to be clicked after every 30 minutes. Ongoing obligation is to continuously check the usage time attribute, as service is available only if usage time is multiple of 30. Before accessing service usage time attribute will be reset to zero which is the pre update.

Ongoing obligation with ongoing update: Obligations must have to be fulfilled during the usage session and also one or more update occurs during the usage process.

Definition: It has the same components as ongoing obligation with immutable attributes except includes procedures for ongoing update

$onupdate (ATT(s)) onupdate (ATT (o))$

Example: The same above scenario of advertisement can be considered in this model in which last click time is continuously updated through the usage.

Ongoing obligation with post update: Obligations have to be true during the process and one or more update occurs after the usage session.

Definition: It has same definition as onB0 except includes procedures for post update

$$postupdate (ATT(s)) \quad postupdate (ATT (o))$$

Example: The same scenario of advertisement will be considered for this model, when the access ends connection time will be reset to zero.

UCON Condition Processes

Conditions are environmental and system attributes like time of access, system load that have to be satisfied before or during the usage process. For condition processes, UCON does not consider the changing system attributes and only has pre and ongoing condition processes with immutable attributes.

Pre condition with immutable attributes: Conditions are checked before giving access and they are selected according to user request having subject and object attributes.

Definition: This process has following components:

$$\begin{aligned} & S, O, R, ATT(S), ATT (O) \\ & preCON \text{ (set of pre condition elements)} \\ & getPreCON: S \times O \times R \rightarrow 2^{preCON} \\ & preCONChecked : preCON \rightarrow \{true, false\} \\ preC (s, o, r) &= preCONChecked (preC i) \text{ where } preC i \in getPreCON (s, o, \\ & \quad \quad \quad r) \\ & allowed(s, o, r) \rightarrow preC(s, o, r) \end{aligned}$$

Example: Suppose user wants to use service, it can be accessed only during the day time. Current time is the system attribute representing environment status which is pre condition that has to be checked before the access.

Ongoing Condition with immutable attributes: Access is allowed without any decision process but environmental restrictions have to be checked during the complete usage session.

Definition: This process has following components:

$$\begin{aligned} & S, O, R, ATT(S), ATT (O) \\ & onCON \text{ (set of ongoing condition elements)} \\ & getOnCON: S \times O \times R \rightarrow 2^{onCON} \end{aligned}$$

$$\begin{aligned}
& onCONChecked : onCON \rightarrow \{true, false\} \\
onC(s, o, r) &= onCONChecked(onC\ i) \text{ where } onC\ i \in getOnCON(s, o, r) \\
& allowed(s, o, r) \rightarrow onC(s, o, r) \\
& stopped(s, o, r) \leftarrow \sim onC(s, o, r)
\end{aligned}$$

Example: Some particular service can be accessed at time interval between 8 am to 5 pm, current time is continuously monitored during the service usage, after 5 pm service is no longer available.

5.2 XACML

Today's company network consists of interconnected segments in order to exchange the information and offer services around the globe. Extensible Markup Language (XML) being a standard makes it easy to exchange the information in a specific format from external sources. Many standards have been defined using XML out of which the two specialized standards are Security Assertion Markup Language (SAML) and Extensible Access Control Markup Language (XACML). SAML defines how identity, access and security information can be transferred among organizations without changing their internal architectures. It does not describe the criteria of how this security information can be used which is covered by XACML policy specification language.

XACML was designed to replace the existing application specific access control mechanisms where each application had to formulate their own criteria for access control policy specification. XACML provides vocabulary to express the organizations access control rules and to make authorization decision. It acts as policy specification language that allows specifying the access control rules, who can access what and when. In addition to this, it also provides the request response mechanism that describes the request for accessing resources and response to these access requests (OASIS Website, 2003). XACML has the following main advantages that make it superior over other languages:

- Access control policies are written once in XACML that is compatible with different applications and there is no need to define policies separately for each application
- Application developers are not required to write code for their policy languages, instead can reuse the existing code
- XACML can encompass the most access control policy needs as well as support the new emerging requirements

- XACML policies can refer to one another which helps to refer the company wide policies and country specific policies in large organizations
- A single XACML policy can be used for different resources that avoid inconsistencies and reduce the effort of creating multiple access control policies

There are two versions of XACML i-e 1.0 and 2.0 (Moses et al., 2005), currently 3.0 has been developed which is backward compatible with the previous versions and still not mature. The basic XACML flow of access control policies is shown in the following figure:

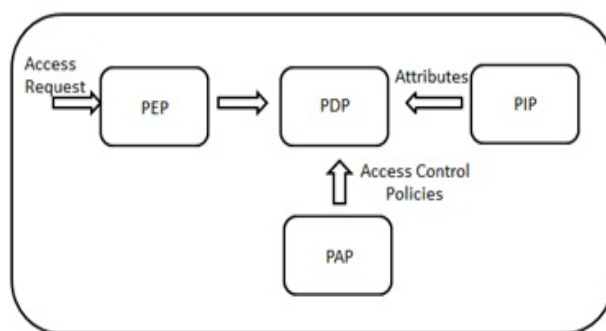


Figure 5.2: XACML Access Control Framework

Policy decision point (PDP), policy enforcement point (PEP), policy information point (PIP) and policy administration point (PAP) are different stages for the flow of access control policies in XACML. PAP is an administration point that is used to create and manage the access control policies; PIP is connected to backend database of enterprise supplying the required attributes to PDP for evaluation. After obtaining the required attributes of subject, object and environment PDP evaluates the applicable policy and returns the authorization decision to PEP which enforces that decision.

XACML policy language has three main components;

- rule
- policy
- policy set

Rule is the most basic unit which defines the specific criteria according to which access is either permitted or denied. Multiple rules combine to form

the policy in order to define the access decision of particular scenario. Each `rule` tag is evaluated on the basis of its following contents:

- `target`
- `effect`
- `condition`
- `obligation`
- `advice`

`Target` defines the set of requests that must be evaluated for the specific rule, rule contents are evaluated only for those access requests whose attributes are the same as that of rule `target`. `Condition` is represented by the boolean expression that specifies access criteria for certain situation and rule `effect` indicates the writer intention of when this rule is evaluated to be true. Rule `effect` can have two values; either true or false. In addition to the required attributes, PDP sometimes also wants to have `obligation` and `advice` for access evaluation depending on the situation. `Obligations` are the actions that must be performed by the requestor before granting access while `advice` indicates the additional information that must be provided to PEP for evaluation.

`Policy` consists of rule combining algorithms that combines access decision of all the rules to form the final access decision for particular request. Similarly, `policy set` encloses the multiple policies and their final access decision is formulated by the access decision of each individual policy with the help of policy combining algorithms. In addition to this, `policy` and `policy set` includes the same components as discussed above for `rule`.

5.3 Conclusion

After the detailed study of UCON model, we have examined that this model provides the access control at fine grained level that improves the accuracy of access decision. Different UCON processes of authorization, obligation and condition can be deployed in real world applications in order to satisfy their authorization requirements. In order to make this possible, UCON model features needs to be incorporated in a policy language to provide the proper specification for this model. We have also observed that XACML (OASIS standard of policy specification) being a generic and extensible policy

language is suitable to provide the UCON model specification. In addition to policy specification standard XACML also provides the request response phenomenon that can help to propose the complete UCON access control framework.

Chapter 6

UCON Access Control Framework

In this chapter, we discuss the complete flow of access control policies for proposed UCON access control framework. Each components operation is briefly explained in order to understand their role in this proposed framework. Another contribution that is described in this chapter is the number of extensions that have been proposed in XACML policy language according to UCON model. These extensions will allow incorporating the UCON model features in the defined policies that are specified by the UCON policy builder. In the end, detailed discussion is given for UCON policy builder module that allows formulating policies of UCON processes for authorization, obligation and condition. In addition to this, step by step procedure is provided that helps to create the policies for UCON processes according to user defined parameters and specifications.

6.1 UCON Access Control Framework

UCON is not being widely adopted as an access control model to restrict the unauthorized access. The major reason for this is that the model features are not been translated in any of the standard policy language to offer the proper formulization of model core parameters. Since UCON can facilitate the diverse range of applications like digital rights management (DRM), health care systems and social networking, it is highly encouraged to provide the formal specification of this model in generic policy language like XACML. There is a need to define the separate profile of UCON in XACML that will enable organizations to adopt this flexible model (Rissanen, 2010). Also to guarantee the accurate access decision in different deployment scenarios, it is

mandatory to propose the required alterations and additions in generic policy language of XACML which is not developed so far. We wish to propose the implementation of UCON model in XACML by incorporating additional components in order to create the XACML access control framework for UCON model.

6.1.1 Components

The proposed framework is shown in Figure 6.2, which contains the following components: Policy enforcement point (PEP), Policy decision point (PDP), Policy information point (PIP), Policy administration point (PAP) and Policy Repository which are also the basic components of XACML access control framework. Two components that have been added in this framework are highlighted in the figure that is UCON policy builder and UCON policy engine in order to support the UCON model features. Brief description of each of these components is discussed below.

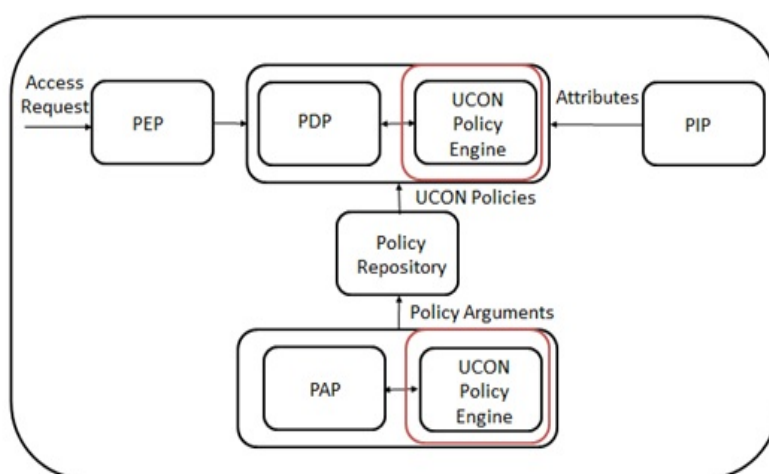


Figure 6.1: UCON Access Control Framework in XACML

Policy Enforcement Point (PEP)

PEP acts as the decision enforcement point which accepts users access request and forwards this request to PDP for evaluation. Final response of the access request is also send to this point by the PDP in order to enforce this decision. PEP also acts like front end interface presented to application users that receive the user request to access resources and generate the final response

for this request. PEP also performs this role in proposed UCON access control framework.

Policy Decision Point (PDP)

This system entity is used to perform the evaluation of user access request to generate the final access decision. When PDP receives the access request from PEP, it extracts the required policy from policy repository in order to validate the user request. If PDP does not find any policy for request, that means there is no policy defined for that specific request and this request will not be entertained. If the user request matches with any of the policy target, that policy will be evaluated by the PDP to generate the final response. There are some cases in which PDP requires the additional attributes for policy evaluation, these attributes are obtained by the PIP. In proposed access control framework, PDP is integrated with UCON policy engine that evaluates the policies according to UCON model.

Policy Information Point (PIP)

PIP is the information point that provides additional information in terms of user attributes for policy evaluation. Generally, PIP is considered as an enterprise database that includes the information of all users and this information is provided to PDP if required for policy evaluation. In UCON access control framework, PIP is also a database of user information that provides the additional user attributes required to perform the policy evaluation.

UCON Policy Engine

We have suggested to incorporate the UCON policy engine module with PDP that provides the main features of UCON model like attribute mutability and decision continuity. UCON policy engine has sub modules like attribute mutability (AM) and decision continuity (DC) .AM handles the updating of attribute values in three access phases; before, during and after access. As a result of attribute values modification, DC monitors the continuous policy evaluation of three access decision factors; authorization, obligation and condition. So the pre and ongoing models of authorization, obligation and conditions are deployed and evaluated accordingly with mutable and immutable attributes.

Policy Repository

Policy repository is a unit in XACML that resides between the PAP and PDP containing the access control policies of corresponding model. This repository contains the UCON model policies in the proposed UCON access control framework. Also this repository provides the required policy to PDP to generate the access response.

Policy Administration Point (PAP)

This is the front end interface for administrators in order to create the access control policies according to the defined specifications. Administrators formulate the access control policies through this policy administration point by giving the input for required policy parameters. These defined policies are then pushed into policy repository to be used by PDP for evaluation.

UCON Policy Builder

We have proposed additional module integrated with PAP that has the capability to interpret the newly created UCON identifiers, attributes and their values. This interpretation will help to determine whether the processing is to be performed in before, ongoing or after phase of usage session. This module is called as UCON policy builder that incorporates UCON model features in policy specification, when PAP accepts the inputs from policy administrator to formulate the generic UCON policy.

6.2 Proposed Extensions in XACML

In order to provide UCON model specification in XACML, we are going to introduce additional identifiers and attribute values to incorporate the features of UCON model.

- XACML has identifiers for subject categories like `access-subject`, `recipient-subject`, `intermediary subject`. They are used under the tag of `AttributeDesignator` that is one of the methods of attributes retrieval in XACML. `Subject-type` identifier is introduced for UCON subject categories and their values might be `consumer`, `provider` and `identiffee`.

```
urn:oasis:names:tc:xacml:1.0:subject-type:consumer
urn:oasis:names:tc:xacml:1.0:subject-type:provider
urn:oasis:names:tc:xacml:1.0:subject-type:identiffee
```

UCON subject identifiers are used along with XACML subject category `access-subject` to further specify the type of accessing subject as follows.

```
<AttributeDesignator
Category=urn:oasis:names:tc:xacml:1.0:
subject-category:access-subject
Type=urn:oasis:names:tc:xacml:3.0:subject-type:consumer>
```

- In the same way, `action-type` identifier is introduced in the attributes for the XACML category of action to reflect the UCON categories of action i-e consumer, provider, identifyee and usage control actions.

```
urn:oasis:names:tc:xacml:3.0:action-type:consumer
urn:oasis:names:tc:xacml:3.0:action-type:provider
urn:oasis:names:tc:xacml:3.0:action-type:identifyee
```

These identifiers are used under the `AttributeDesignator` tag to indicate the attributes of UCON specific action type. These attributes are then matched with the attributes extracted from the request.

```
<AttributeDesignator
Category=urn:oasis:names:tc:xacml:3.0:attribute-category:
action
Type=urn:oasis:names:tc:xacml:3.0:action-type:usage-
control>
```

- For demonstrating UCON objects, resource category identifier is created under attribute category of resource that has the value of `privacy-sensitive`, `privacy-nonsensitive` or `derivative`.

```
urn:oasis:names:tc:xacml:3.0:resource-category:privacy-
sensitive
urn:oasis:names:tc:xacml:3.0:resource-category:
privacy-nonsensitive
urn:oasis:names:tc:xacml:3.0:resource-category:derivative
```

They are used in the same way as subject or action attributes under `AttributeDesignator` tag as follows.

```
<AttributeDesignator
  Category=urn:oasis:names:tc:xacml:3.0:attribute-category:
  resource
  Type=urn:oasis:names:tc:xacml:3.0:resource-category:
  derivative>
```

- Since the UCON model consider both mutable (updating values) as well as immutable (constant values) attributes, new identifier `attribute-class` is constructed to differentiate between them. Mutable attributes are then further narrow down into pre-mutable, ongoing mutable and post mutable. This general classification of attributes is used with all of the attribute categories of subject, resource, action and environment.

```
urn:oasis:names:tc:xacml:3.0:attribute-class:immutable
urn:oasis:names:tc:xacml:3.0:attribute-class:pre-mutable
urn:oasis:names:tc:xacml:3.0:attribute-class:ongoing-
mutable
urn:oasis:names:tc:xacml:3.0:attribute-class:post-mutable
```

- Rights related authorization rules can be mapped in XACML as general rules but the time of evaluating these rules needs to be managed. So the pre and ongoing element of authorization rules is explained by the `rule-id` attribute of the rule element. Since UCON considers obligation and condition other than authorization for access decision, evaluation phase of these two factors is also indicated by this `rule-id` attribute.

```
RuleId="urn:oasis:names:tc:xacml:3.0:pre-authorization"
RuleId="urn:oasis:names:tc:xacml:3.0:ongoing-authorization"
RuleId="urn:oasis:names:tc:xacml:3.0:pre-obligation"
RuleId="urn:oasis:names:tc:xacml:3.0:ongoing-obligation"
RuleId="urn:oasis:names:tc:xacml:3.0:pre-condition"
RuleId="urn:oasis:names:tc:xacml:3.0:ongoing-condition"
```

- Obligation element is specified in XACML for mandatory actions required to be performed by the subject that can also handle the obligation related authorization rules of UCON. Obligation expression el-

ement includes arguments that are required to execute the obligation. We have proposed the new attribute namely `Fulfill-phase` for specification of pre and ongoing obligations. It indicates the access phase during which obligation must have to be satisfied by PEP, so it may have the values of `pre-access`, `ongoing-access` that reveals the pre and ongoing obligations.

```
<ObligationExpression
ObligationId="urn:oasis:names:tc:xacml:ucon-example:
obligation:license-agreement
Fulfill-phase="pre-access">
```

- Condition element in XACML contain single expression element which includes functions to be evaluated. We have introduced additional attribute `condition-type` under the condition element to present the UCON dynamic and static conditions. Furthermore pre and ongoing element of condition models are expressed by introducing new attribute called as `evaluation-phase` under the condition element. It can have the value of `pre-access` or `ongoing access`.

```
<Condition
Condition-type = "urn:oasis:names:tc:xacml:3.0:condition-
type:dynamic
Evaluation-phase= ongoing-access">
```

6.3 UCON Policy Builder

UCON policy builder module has been implemented in java that provides multiple options to policy administrator in order to specify the features for UCON policy. As described in section 5.1.2, there are three types of processes for UCON model; authorization, obligation and condition processes. This module is integrated with PAP that allows formulating the policy for all of the UCON processes as shown in Figure 6.2. We will provide the detail procedure of policy specification for all of these UCON processes in the following sections.

6.3.1 Authorization Process

We consider the scenario of financial application for policies specification that has service of creating vouchers. First we will formulate the policy for

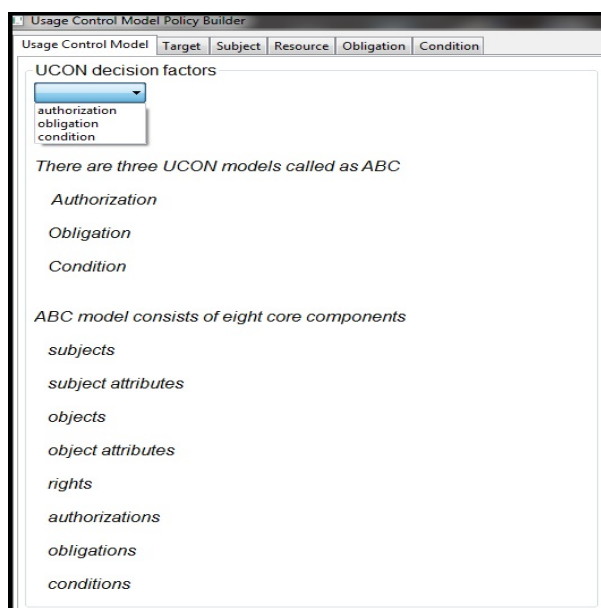


Figure 6.2: UCON Policy Builder

pre authorization process which evaluates the subject and resource attributes before giving access to voucher entry service. The policy will be such as

“Subject having id *ghazia@gmail.com*, role of *director general* and belongs to *administration* department can access the voucher entry service and create vouchers of amount *5000* if his name is included in the *accessing list*“

In this case, the subject attributes are *id*, *role* and *department* and object attributes are *accessing list* and *voucher limit* that has to be checked.

First the authorization decision factor is selected in order to create the policy for pre authorization process. Then subject and resource attributes that have been identified above for policy are selected, their match id functions and data types are also defined in subject and resource tabs respectively.

In addition to this model parameters are specified that includes attribute class (immutable, pre, ongoing, post), access phase (pre or ongoing authorization), policy id , policy description, rule id, rule description, rule effect and rule combining algorithms .

In the end, policy target is defined that contains the subject, resource and action attribute. Subject attribute is *user id*, resource attribute is the *URI of voucher service* and action attribute is *access*. These attributes are first matched with the access request at the time of policy evaluation.

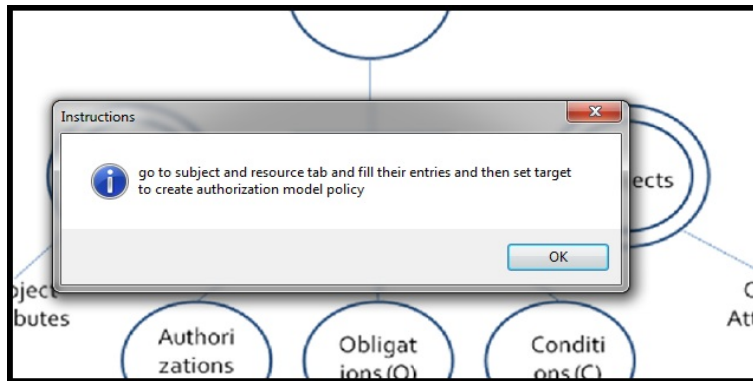


Figure 6.3: Authorization Process

Subject Attribute	Subject Match Id	Subject Designator DataType
subject-role	string-equal	string
subject-department	string-equal	string

Figure 6.4: Subject Attributes

Resource Attribute	Resource Match Id	Resource Designator DataType
voucher-limit	double-equal	double
accessing-list	string-equal	string

Figure 6.5: Resource Attributes

After the above specification, the following policy of pre authorization process has been created.

Attribute Class	Access phase
<input type="text" value="immutable"/>	<input type="text" value="pre"/>
Policy Id	Policy Description
<input type="text" value="ucon-policy"/>	<input type="text" value="access-policy"/>
Rule Id	Rule Description
<input type="text" value="pre authorization"/>	<input type="text" value="first-rule"/>
Rule effect	Rule Combining Algorithm
<input type="text" value="permit"/>	<input type="text" value="permit-overrides"/>

Figure 6.6: Model Parameters

Subject	<input type="text" value="subject-id"/>	Resource	<input type="text" value="resource-id"/>	Action	<input type="text" value="action-id"/>
Subject Match Id	<input type="text" value="rfc822Name-match"/>	Resource Match Id	<input type="text" value="anyURI-equal"/>	Action Match Id	<input type="text" value="string-equal"/>
Subject Designator Type	<input type="text" value="rfc822Name"/>	Resource Designator Type	<input type="text" value="anyURI"/>	Action Designator Type	<input type="text" value="string"/>

Figure 6.7: Policy Target

```

<Policy PolicyId="ucon-policy" RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-
combining-algorithm:permit-overrides">
<Description>access-policy</Description>
  <Rule RuleId="urn:oasis:names:tc:xacml:2.0:ucon-example:pre- authorization" Effect=
  "Permit">
    <Description>first-rule</Description>
    <Target>
      <Subjects>
        <Subject>
          <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:rfc822Name-match">
            <AttributeValue
              DataType="http://www.w3.org/2001/XMLSchema#string">ghazia@gmail.com</AttributeValue>
            <SubjectAttributeDesignator
              AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
              DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name"
              AttributeClass="urn:oasis:names:tc:xacml:1.0:subject:attribute-class:immutable"/>
          </SubjectMatch>
          <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
              DataType="http://www.w3.org/2001/XMLSchema#string">Director General</AttributeValue>

```

```

    <SubjectAttributeDesignator
      AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-role"
      DataType="urn:oasis:names:tc:xacml:1.0:data-type:string"
      AttributeClass="urn:oasis:names:tc:xacml:1.0:subject:attribute-class:immutable"/>
    </SubjectMatch>
  </Subject>
</Subjects>
<Resources>
  <Resource>
    <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
      <AttributeValue
        DataType="http://www.w3.org/2001/XMLSchema#anyURI">webservice</AttributeValue>
      <ResourceAttributeDesignator
        AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
        DataType="http://www.w3.org/2001/XMLSchema#anyURI"
        AttributeClass="urn:oasis:names:tc:xacml:1.0:resource:attribute-class:immutable"/>
      </ResourceMatch>
    <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:double-equal">
      <AttributeValue
        DataType="http://www.w3.org/2001/XMLSchema#double">50000.0</AttributeValue>
      <ResourceAttributeDesignator
        AttributeId="urn:oasis:names:tc:xacml:1.0:resource:voucher-limit"
        DataType="http://www.w3.org/2001/XMLSchema#double"
        AttributeClass="urn:oasis:names:tc:xacml:1.0:resource:attribute-class:immutable"/>
      </ResourceMatch>
    </Resource>
  </Resources>
<Actions>
  <Action>
    <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
      <AttributeValue
        DataType="http://www.w3.org/2001/XMLSchema#string">access</AttributeValue>
      <ActionAttributeDesignator
        AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
        DataType="http://www.w3.org/2001/XMLSchema#string"
        AttributeClass="urn:oasis:names:tc:xacml:1.0:action:attribute-class:immutable"/>
      </ActionMatch>
    </Action>
  </Actions>
</Target>
</Rule>
/Policy>

```

In the same way, ongoing authorization policy can be created by specifying the ongoing attribute class for subject and resource attributes according to scenario.

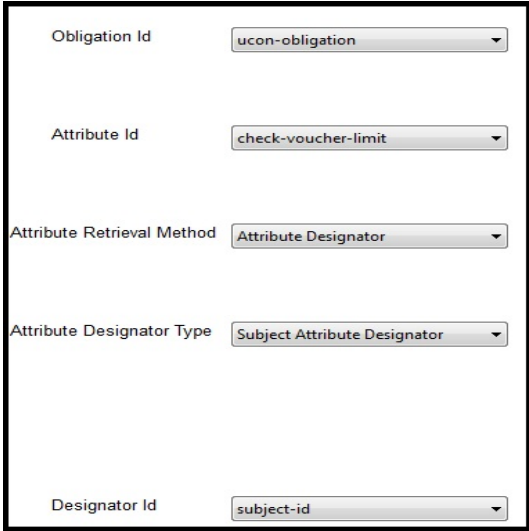
6.3.2 Obligation Process

Same financial application scenario is considered for the obligation process which includes the specification of obligation related XACML parameters along with policy target. Obligations are categorized as system related obligations and subject related obligations (Katt et al., 2008). System related obligations are those actions that are executed by the service provider in order to ensure the verification of requesting party. On the other hand subject

related obligations are performed by the requesting subject which is enforced by the service provider. Let policy for system related ongoing obligation is

“Service provider will continuously check the voucher limit of subject having id *ghazia@gmail.com*, if the voucher limit exceeds 5000 then the voucher entry service is no longer available“

After selecting the obligation decision factor, the process related parameters access phase, policy id, policy description, rule id, rule description, rule effect and rule combining algorithm are defined in obligation tab. Obligation id and attribute id are specified for which obligation will be defined. There are two different methods of attribute retrieval in XACML that can be used for attributes within obligation tag. Attribute designator is used to retrieve the attributes defined in policy while attribute selector is used to define the path of node from which attributes are extracted. Since the current scenario is to continuously monitor the voucher limit, the attribute ids will be check voucher limit and disable post voucher option. Attribute designator is selected for these two attributes, first attribute is for subject and second one corresponds to resource.



Obligation Id	ucon-obligation
Attribute Id	check-voucher-limit
Attribute Retrieval Method	Attribute Designator
Attribute Designator Type	Subject Attribute Designator
Designator Id	subject-id

Figure 6.8: Subject Attribute Designator

Same policy target is defined for the obligation process as for authorization process described above. In obligation processes, we have assumed that the target attributes are immutable and there is no need to define attribute class separately for these target attributes.

Obligation Id	ucon-obligation
Attribute Id	disable-post-voucher-option
Attribute Retrieval Method	Attribute Designator
Attribute Designator Type	Resource Attribute Designator
Designator Id	resource-id

Figure 6.9: Resource Attribute Designator

Access phase	Policy Id
ongoing	ucon-example
Policy Description	Rule Id
ucon-model-policy	ongoing obligation
Rule Description	Rule effect
ucon-rule	permit
Rule Combining Algorithm	
permit-overrides	

Figure 6.10: Model Parameters

After the above specification of parameters, the following ongoing obligation policy has been formulated.

```
<Policy PolicyId="ucon-example" RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:
  rule-combining-algorithm:permit-overrides">
  <Description>access-policy</Description>
  <PolicyDefaults><XPathVersion>2</XPathVersion></PolicyDefaults>
  <Rule RuleId="urn:oasis:names:tc:xacml:2.0:ucon-example:ongoing-obligation"
    Effect="Permit">
```

```

<Description>ucon-rule</Description>
<Target>
  <Subjects>
    <Subject>
      <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:rfc822Name-match">
        <AttributeValue
          DataType="http://www.w3.org/2001/XMLSchema#string">ghazia@gmail.com</AttributeValue>
        <SubjectAttributeDesignator
          AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
          DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name"/>
        </SubjectMatch>
      </Subject>
    </Subjects>
    <Resources>
      <Resource>
        <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
          <AttributeValue
            DataType="http://www.w3.org/2001/XMLSchema#string">webservice</AttributeValue>
          <ResourceAttributeDesignator
            AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
            DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
          </ResourceMatch>
        </Resource>
      </Resources>
    <Actions>
      <Action>
        <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue
            DataType="http://www.w3.org/2001/XMLSchema#string">access</AttributeValue>
          <ActionAttributeDesignator
            AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
            DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ActionMatch>
        </Action>
      </Actions>
    </Target>
  </Rule>
  <Obligations>
    <Obligation
      ObligationId="urn:oasis:names:tc:xacml:ucon-example:ucon-obligation" FulfillOn=
      "OngoingPermit">
      <AttributeAssignment
        AttributeId="urn:oasis:names:tc:xacml:2.0:check-voucher-limit"
        DataType="http://www.w3.org/2001/XMLSchema#string">
        SubjectAttributeDesignator
        AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id" DataType
        ="http://www.w3.org/2001/XMLSchema#string"></AttributeAssignment>
        <AttributeAssignment
          AttributeId="urn:oasis:names:tc:xacml:2.0:disable-post-voucher-option"
          DataType="http://www.w3.org/2001/XMLSchema#string">
          ResourceAttributeDesignator
          AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id" DataType
          ="http://www.w3.org/2001/XMLSchema#string"></AttributeAssignment>
        </Obligation>
      </Obligations>
    </Policy>

```

Pre obligation policy can be created in the same way by specifying the attributes for pre obligation scenario. Accepting license agreement before using

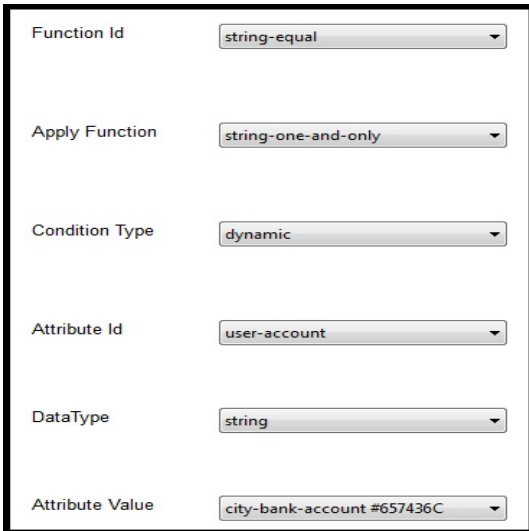
the service is one of the example of pre obligation process.

6.3.3 Condition Process

This module also allows formulating the pre and ongoing condition policies for UCON model that includes the specification of condition related features. In the scenario of voucher entry service, pre condition is to allow the user to create vouchers in specific account numbers. Let the policy for pre condition process is

“Subject having id *ghazia@gmail.com* can only create the vouchers in *city-bank-account #657436C*, other than this account number this specific user is not allowed to create voucher“

After selecting the condition decision factor, the same model parameters are selected in the condition tab as for obligation and authorization processes. Function id that is used to evaluate the condition is selected and apply function is specified that executes the condition function in order to extract the single value. Further condition type identified in UCON model, attribute id, data type and attribute value are defined according to the condition.



Function Id	string-equal
Apply Function	string-one-and-only
Condition Type	dynamic
Attribute Id	user-account
DataType	string
Attribute Value	city-bank-account #657436C

Figure 6.11: Condition Parameters

Same policy target is defined as for obligation and authorization process in order to create the policy for pre condition process.

Access phase	Policy Id
<input type="text" value="pre"/>	<input type="text" value="ucon-policy"/>
Policy Description	Rule Id
<input type="text" value="access-policy"/>	<input type="text" value="pre condition"/>
Rule Description	Rule effect
<input type="text" value="first-rule"/>	<input type="text" value="permit"/>
Rule Combining Algorithm	
<input type="text" value="permit-overrides"/>	

Figure 6.12: Model Parameters

Following pre condition policy has been specified after defining the above parameters.

```

<Policy PolicyId="ucon-policy" RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:
rule-combining-algorithm:permit-overrides">
  <Description>access-policy</Description>
  <Rule RuleId="urn:oasis:names:tc:xacml:2.0:ucon-example:pre-condition" Effect=
  "Permit">
    <Description>first-rule</Description>
    <Target>
      <Subjects>
        <Subject>
          <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:rfc822Name-match">
            <AttributeValue
              DataType="http://www.w3.org/2001/XMLSchema#string">ghazia@gmail.com</AttributeValue>
            <SubjectAttributeDesignator
              AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
              DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name"
              AttributeClass="urn:oasis:names:tc:xacml:1.0:subject:attribute-class:immutable"/>
          </SubjectMatch>
        </Subject>
      </Subjects>
    </Target>
    <Resources>
      <Resource>
        <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
          <AttributeValue
            DataType="http://www.w3.org/2001/XMLSchema#anyURI">webservice</AttributeValue>
          <ResourceAttributeDesignator
            AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
            DataType="http://www.w3.org/2001/XMLSchema#anyURI"
            AttributeClass="urn:oasis:names:tc:xacml:1.0:resource:attribute-class:immutable"/>
        </ResourceMatch>
      </Resource>
    </Resources>
    <Actions>
      <Action>

```

```

    <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
    <AttributeValue
    DataType="http://www.w3.org/2001/XMLSchema#string">access</AttributeValue>
    <ActionAttributeDesignator
    AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
    DataType="http://www.w3.org/2001/XMLSchema#string"
    AttributeClass="urn:oasis:names:tc:xacml:1.0:action:attribute-class:immutable"/>
    </ActionMatch>
  </Action>
</Actions>
</Target>
  <Condition Type="dynamic"
  FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
  <SubjectAttributeDesignator
  AttributeId="user-account" DataType="http://www.w3.org/2001/XMLSchema#string"
  Issuer="admin@users.example.com"/>
  </Apply>
  <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">city-bank-
  account #657436C
  </AttributeValue>
  </Condition>
</Rule>
</Policy>

```

Similarly ongoing condition policy can be created by specifying the condition related functions and attributes. Ongoing condition can be to give service access by limited number of times. This condition will be monitored continuously and when the usage limit reached service is revoked from the users.

6.4 Conclusion

We have proposed the complete workflow of UCON access control framework by the addition of two modules. One of the modules is used to create the policies incorporating UCON core features that define the parameters for UCON processes of authorization, obligation and condition. It provides user friendly interface for policy specification that allows creating the policies for different types of UCON processes depending upon access decision factors and three access phases of usage session. The other module provides the continuous monitoring to handle the dynamic change throughout the access phase that is one of the distinguishing features of UCON model. This proposed framework supports the UCON of XACML that is based on the newly created attributes and identifiers in order to formulate the UCON policies. Further these policies are evaluated by the UCON policy engine module depending on multiple UCON decision factors to generate the accurate access decision.

Chapter 7

Validation of UCON Model Features

Evaluation methodology of UCON XACML policies are described in this chapter that is based on UCON policy ontology. The basic purpose of this ontology is to highlight the UCON model features and processes of authorization, obligation and condition decision factors. At the end of chapter, characteristics of UCON processes are evaluated with the help of two example scenarios for voucher entry web service.

7.1 Evaluation Methodology

There are different methods of evaluation depending upon either evaluation is performed for prototype or a complete system. Prototype evaluation can be performed by different formal methods present in literature that are used according to design requirements . On the other hand, system evaluation can be done by specifying the set of features that are analyzed for the developed system (Ross et al., 2005). There are also quantitative and qualitative evaluation methods that are mostly used to refer the generated data of the proposed system or prototype.

We have carried out our research on attribute based UCON model through highlighting the model core features in XACML policy language. This will help to formulate the UCON model policies in XACML according to user defined features. Further these policies are evaluated in such a way to satisfy the evaluation mechanism of UCON model. We have developed UCON policy ontology that describes the model entities and their relationship reflecting the model core features. These features are then analyzed for two example scenarios of voucher entry web service.

7.1.1 UCON Policy Ontology

Access control policy complete ontology has been illustrated in NIST Guidelines for Access Control Evaluation Metrics in order to convey the basic access control primitives (Vincent Hu, 2012). This ontology also helps to determine the relationship among these access control primitives. Similarly, we have designed the complete ontology for Usage Based Access Control policy to highlight the model core features and its corresponding processes. UCON model features shown in this ontology are then evaluated with the help of two simulates scenarios for voucher entry web application. The complete description of the UCON policy ontology is given below.

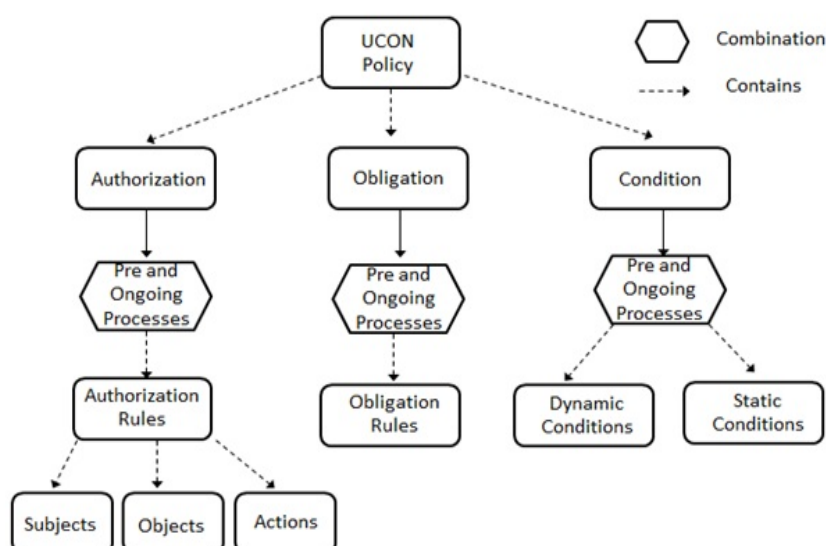


Figure 7.1: UCON Policy Ontology

UCON policy ontology comprises of three main decision factors; authorization, obligation and condition. Each of these authorization, obligation and condition has two processes that evaluate the access request before or during the access phase called as pre and ongoing respectively. Pre and ongoing authorization processes encompass authorization rules that define which specific object can perform the defined action on specific object. In order to enhance the accuracy of access decision, pre and ongoing obligations are introduced in UCON model. Obligation rules are identified for these pre and ongoing processes that describe the specific action or task a user has to be performed before accessing the resource. Condition processes of pre and ongoing phases include the static and dynamic conditions of UCON model

depending upon the policy requirements.

7.1.2 Web Service Usage (Scenario 1)

Web service of voucher entry has been developed that can be deployed for any educational institution, organization and enterprises. We have simulated the two example scenarios for this voucher entry web service in order to verify the UCON model features. This voucher entry web service accepts and evaluates the access request acting as both PEP and PDP. In first example scenario, limit has been defined to access this voucher entry web service. Policy is defined as: user having role of *director general* can create vouchers in voucher entry service for the three times during the access phase.

Following figures show the execution of voucher entry web service in JDeveloper Studio, left window shows the http request header and right window indicates the corresponding http header response for this request.

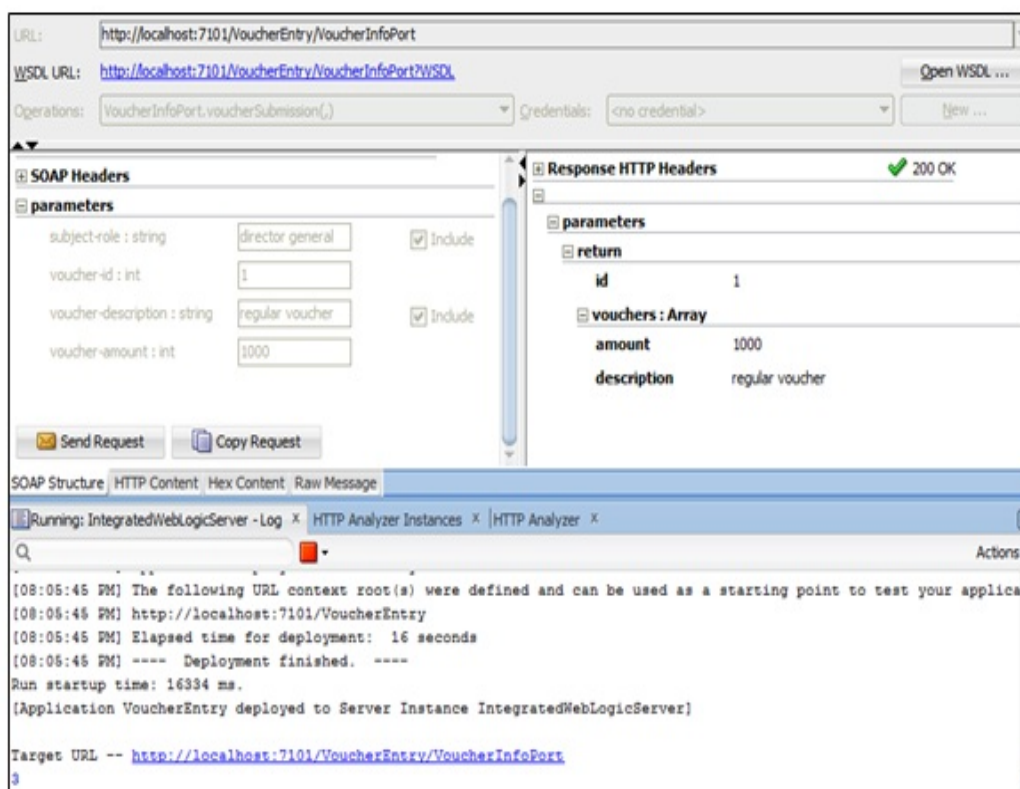


Figure 7.2: First Access Request

User enters the *role attribute*, *voucher id*, *voucher description* and *voucher*

amount that are required to generate the access request for creating voucher. The submitted values for this voucher entry are shown in the response header as the request is send. The bottom window shows the deployment and execution of web service on web logic server integrated with JDevelepor along with the usage counter for web service.

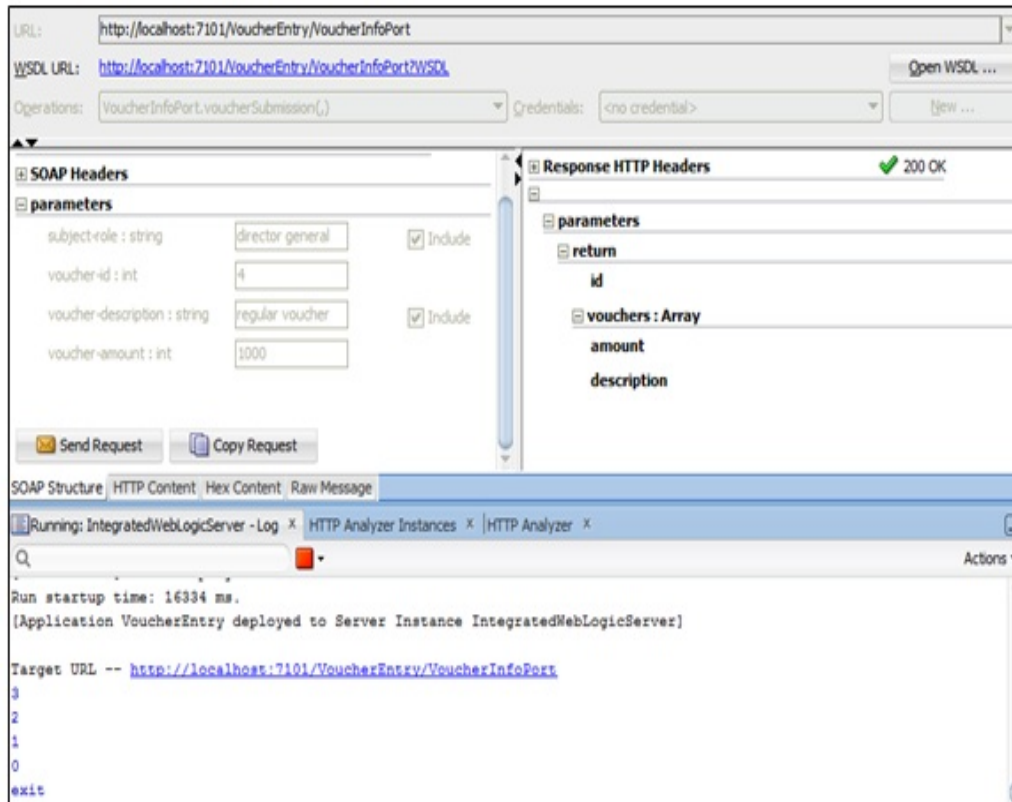


Figure 7.3: Usage Limit Reached

User generates access request three times of same amount to create voucher. When access request is generated for the fourth time web service is no longer available and voucher will not be created.

In the above scenario, ongoing condition process of UCON is being evaluated that the user cannot create the voucher more than three times during the access phase. Further, UCON pre authorization process is also being evaluated that verifies the user role attributes before granting access to voucher entry web service. If user enters the attribute value other than *director general*, he will not be allowed to access the web service in order to submit vouchers.

7.1.3 Voucher Limit (Scenario 2)

Second example scenario defines the policy: user having *role* attribute of *director general* can create vouchers in account no *5748* and *6748* of amount *5000* during the access phase.

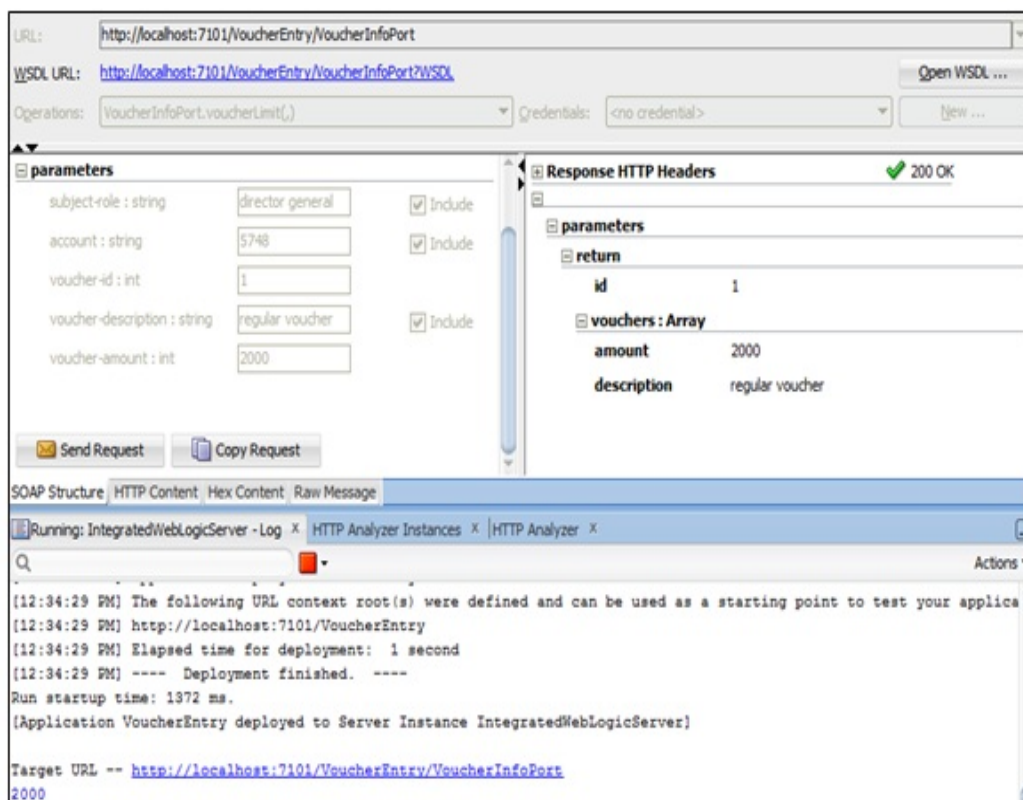


Figure 7.4: Voucher of amount 2000

User creates the voucher two times of amount *2000* in account no *5748* and the voucher has been created of total amount of *4000*. Then user creates the voucher of amount *1000* by sending request third time and the total *voucher amount* is now *5000*. If user still submits more vouchers, they will not be submitted and service is not available as the limit has been reached of *voucher amount* of *5000*.

This scenario is the combination of pre authorization, ongoing condition and system related ongoing obligation processes of UCON model. Pre authorization process is being followed in the same way as in first example scenario through the verification of role attribute. System related ongoing obligation is being evaluated in such a way that user is not allowed to create vouchers

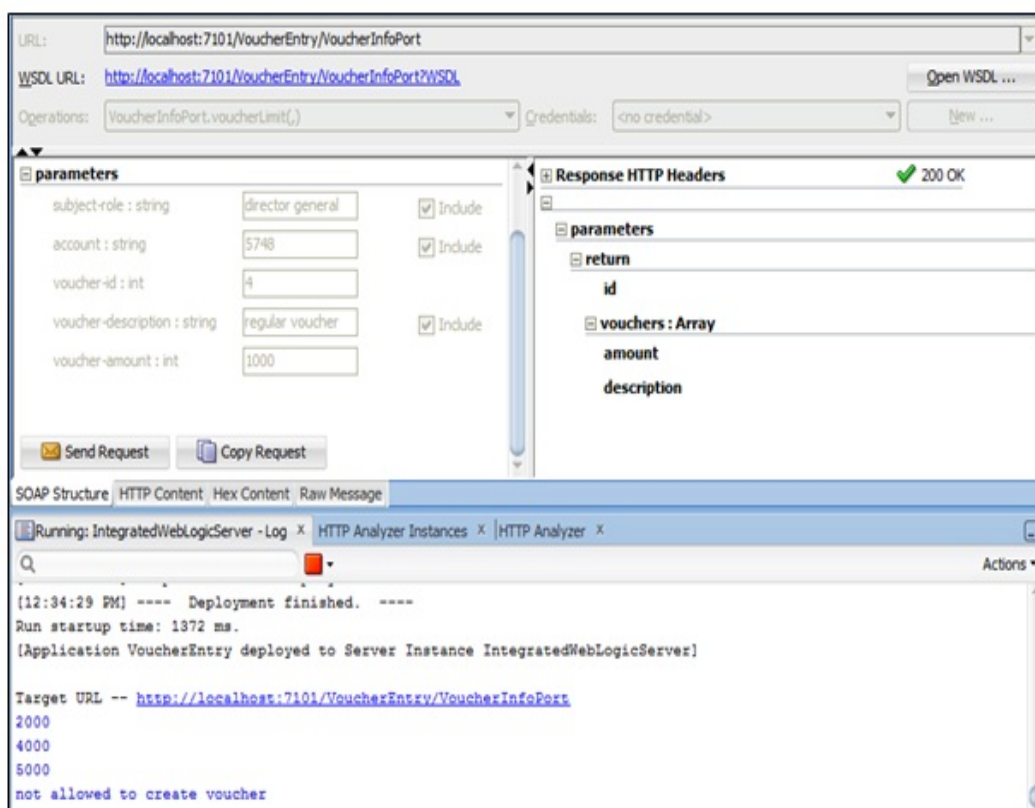


Figure 7.5: Voucher Limit Reached

other than two accounts *5748* and *6748*. This is being controlled by the web service (acting as a system) that does not allow user to create vouchers in other accounts. Ongoing condition process is being fulfilled by checking the *voucher limit* that does not allow creating vouchers of amount more than *5000* during the access phase.

7.2 Conclusion

It has been observed that access control model features can be highlighted and discussed with the help of comprehensive and detailed ontology. This ontology includes the model characteristics and demonstrates the workflow by determining the relationship among the model entities. The same concept has been used in our research for the validation of Usage Based Access Control Model (UCON) features for web based application. UCON policy ontology has been developed that indicates the core features and processes

of UCON model. As a result two example scenarios of voucher creation have been simulated that validates the UCON model core features i-e continuous monitoring of access decision and attribute modification during the access phase. These example scenarios also validate some of the UCON processes of authorization, obligation and condition. Similarly, the other combination of UCON processes can also be verified according to scenario.

Chapter 8

Conclusion and Future Research

8.1 Conclusion

The current research has been carried out in two directions; access control models for cloud environment and Usage Based Access Control for web based applications. This thesis aim to provide the analysis of existing cloud based access control models that help to highlight the authorization requirements for cloud dynamic environment. Keeping in view these authorization requirements, it has been concluded that Usage Based Access Control is the most suitable access control model for dynamic environment.

First contribution of this research is the detailed study in the domain of access control models for cloud environment. There are few of the access control models that have been proposed for cloud environment in order to address the authorization issues. These models target the specific problems and scenarios and do not cover the complete authorization requirements of cloud environment. It is therefore required first to provide the authorization features for dynamic environment that can then be incorporated in cloud based access control models. In order to address this need, we have performed the comparative analysis of existing cloud based access control models based on NIST defined access control evaluation features. These features are combined to formulate the assessment criteria for cloud based access control models in which three levels of high, medium and low have been defined. These levels have been assigned to assessment features according to their satisfaction level in the corresponding cloud based access control model. The proposed assessment criteria highlight the set of features that must be offered by cloud based access control models in order to provide the dynamic authorization.

It also helps the research community to develop the comprehensive access control model to satisfy the dynamic nature of cloud environment.

Second research contribution is related to an attribute based access control model known as Usage Control Model (UCON). Despite of the significant features of UCON model, it is not being widely adopted by large organizations as an access control model. Major reason for this is that the model specification has not been provided in any of the policy specification language. In order to overcome this issue, we have proposed the UCON profile in OASIS standard of XACML policy specification language that supports the policy specification for UCON model. UCON profile has been formulated through XACML extensions either in the form of newly created identifiers or adding the new attribute values. The proposed UCON profile is used by the UCON policy builder module of access control framework. Similarly, another module of UCON policy engine has been added in this framework to perform the evaluation of UCON model access control policies. This UCON access control framework perform different functions such as obligation monitoring, attributes management, maintaining history of updated events of certain access request, notifies user about the current policy status for request. UCON profile in XACML enables organizations to adopt this flexible UCON model in order to provide the enhanced protection from unauthorized users. This profile enhance the existing functionality of XACML policy flow modules by considering the multiple factors in making access decision to offer detailed features of UCON model.

8.2 Future Research

To date, the already existing access control models for cloud environment are not thoroughly evaluated to highlight the dynamic authorization features. Our research analysis concludes that Usage Access Control Model (UCON) is suitable enough to fulfill the dynamic authorization requirements for cloud environment. Despite of substantial features being offered, UCON model still needs to incorporate the additional security features for the better protection from unauthorized usage of resources. Therefore, number of challenges in this domain needs meticulous research and must be effectively cater to provide the reliable security in Cloud environment. We identify the following future directions in which research can be carried out to address some of the challenges.

We have proposed the assessment criteria for cloud based access control models that specify the essential features for dynamic authorization of cloud environment. The assessment criteria can further be improved by the specifi-

cation of functional and non-functional requirements in order to provide the detailed taxonomy for the authorization requirements of cloud environment.

We have proposed UCON model for web based applications to provide the better resource protection. Still there is a need of detailed performance and security analysis of UCON model that helps to overcome the security loopholes and incorporate the additional security features such as revocation and rights delegation in model.

Policy conflict is one of the major issues that have not been handled properly in UCON model. There must be a proper mechanism for this issue that investigate the policies incompleteness, inconsistencies that arise due to grant and permit access for same rules and access requests for which policies have not been defined.

UCON profile must be developed in access control policy languages other than XACML to investigate the compatibility of model features in different languages. Formal verification and comparison must then be performed for these profiles in order to provide the most appropriate and comprehensive profile for UCON model.

Bibliography

- Abi Haidar, D., Cuppens-Boulahia, N., Cuppens, F., and Debar, H. (2006). An extended rbac profile of xacml. In *Proceedings of the 3rd ACM workshop on Secure web services*, pages 13–22. ACM.
- access-matrix (2012). Access control matrix. http://en.wikipedia.org/wiki/Access_Control_Matrix. (Last visited: October 2012).
- Ali, T., Nauman, M., Hadi, F., and bin Muhaya, F. (2010). On usage control of multimedia content in and through cloud computing paradigm. In *Future Information Technology (FutureTech), 2010 5th International Conference on*, pages 1–5. IEEE.
- Anoop Singhal, T. W. (2007). Guide to secure web services. *NIST Special Publication*.
- Bayse, G. Z. (2004). A security checklist for web application design. Technical report, SANS Institute InfoSec Reading Room. http://www.sans.org/reading_room/whitepapers/securecode/security-checklist-web-application-design_1389.
- Brunette, G. and Mogull, R. (2009). Security guidance for critical areas of focus in cloud computing v2. 1. *Cloud Security Alliance*, pages 1–76.
- Calero, J., Edwards, N., Kirschnick, J., Wilcock, L., and Wray, M. (2010). Toward a multi-tenancy authorization system for cloud services. *Security & Privacy, IEEE*, 8(6):48–55.
- Danwei, C., Xiuli, H., and Xunyi, R. (2009). Access control of cloud service based on ucon. *Cloud Computing*, pages 559–564.
- Ferraiolo, D. and Kuhn, D. (2009). Role-based access controls. *arXiv preprint arXiv:0903.2171*.
- Gouglidis, A. and Mavridis, I. (2010). On the definition of access control requirements for grid and cloud computing systems. *Networks for Grid Applications*, pages 19–26.
- Han, W. and Lei, C. (2011). A survey on policy languages in network and security management. *Computer Networks*.
- Hu, V., Ferraiolo, D., and Kuhn, D. (2006). *Assessment of access control*

- systems*. US Department of Commerce, National Institute of Standards and Technology.
- Jansen, W. (2010). *Directions in security metrics research*. DIANE Publishing.
- Jansen, W. and Grance, T. (2011). Guidelines on security and privacy in public cloud computing. *NIST Special Publication*, pages 800–144.
- Kaefer, G. (2010). Cloud computing architecture. *Siemens*, <http://www.scribd.com/doc/56483000/Cloud-Computing-Architecture-Gerald-Kaefer>.
- Katt, B., Zhang, X., Breu, R., Hafner, M., and Seifert, J. (2008). A general obligation model and continuity: enhanced policy enforcement engine for usage control. In *Proceedings of the 13th ACM symposium on Access control models and technologies*, pages 123–132. ACM.
- Kumaraguru, P., Cranor, L., Lobo, J., and Calo, S. (2007). A survey of privacy policy languages. In *Workshop on Usable IT Security Management (USM 07): Proceedings of the 3rd Symposium on Usable Privacy and Security, ACM*.
- Kumaraswamy, S., Lakshminarayanan, S., Stein, M., and Wilson, Y. (2010). Domain 12: Guidance for identity & access management v2. 1. *Cloud Security Alliance*. Online: <http://www.cloudsecurityalliance.org/guidance/csaguide-dom12-v2>, 10.
- Lazouski, A., Martinelli, F., and Mori, P. (2010). Usage control in computer security: A survey. *Computer Science Review*, 4(2):81–99.
- Li, J., Zhao, G., Chen, X., Xie, D., Rong, C., Li, W., Tang, L., and Tang, Y. (2010). Fine-grained data access control systems with user accountability in cloud computing. In *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*, pages 89–96. IEEE.
- Lu, J., Li, R., Varadharajan, V., Lu, Z., and Ma, X. (2009). Secure interoperation in multidomain environments employing ucon policies. *Information Security*, pages 395–402.
- Masood, R., Shibli, M., and Bilal, M. (2012a). Usage control model specification in xacml policy language. *Computer Information Systems and Industrial Management*, pages 68–79.
- Masood, R., Shibli, M., et al. (2012b). Comparative analysis of access control systems on cloud. In *Software Engineering, Artificial Intelligence, Networking and Parallel & Distributed Computing (SNPD), 2012 13th ACIS International Conference on*, pages 41–46. IEEE.
- Mon, E. and Naing, T. (2011). The privacy-aware access control system using attribute-and role-based access control in private cloud. In *Broadband Network and Multimedia Technology (IC-BNMT), 2011 4th IEEE*

- International Conference on*, pages 447–451. IEEE.
- Moses, T. et al. (2005). Extensible access control markup language (xacml) version 2.0. *Oasis Standard*, 200502.
- Nadalin, A., Kaler, C., Hallam-Baker, P., and Monzillo, R. (2004). Web services security: Soap message security 1.0 (ws-security 2004), oasis standard 200401, march 2004. See: <http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf>.
- Narayanan, H. and Gunes, M. (2011). Ensuring access control in cloud provisioned healthcare systems. In *Consumer Communications and Networking Conference (CCNC), 2011 IEEE*, pages 247–251. IEEE.
- OASIS Website (2003). brief introduction to xacml. https://www.oasis-open.org/committees/download.php/2713/Brief_Introduction_to_XACML.html. (Last visited: September 2011).
- OASIS Website (2006). Oasis web services security (wss) tc. https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss. (Last visited: July 2012).
- Olden, E. (2011). Architecting a cloud-scale identity fabric. *Computer*, 44(3):52–59.
- Park, J. and Sandhu, R. (2002). Towards usage control models: beyond traditional access control. In *Proceedings of the seventh ACM symposium on Access control models and technologies*, pages 57–64. ACM.
- Park, J. and Sandhu, R. (2004). The ucon abc usage control model. *ACM Transactions on Information and System Security (TISSEC)*, 7(1):128–174.
- Rimal, B., Choi, E., and Lumb, I. (2009). A taxonomy and survey of cloud computing systems. In *INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on*, pages 44–51. IEEE.
- Rissanen, E. (2010). Xacml v3.0 core and hierarchical role based access control (rbac) profile version 1.0 (committe specification 01). Technical report, Technical report, OASIS, <http://docs.oasisopen.org/xacml/3.0/xacml-3.0-rbac-v1-spec-cs-01-en.pdf>.
- role-based-access-control (2012). Nist rbac model. http://en.wikipedia.org/wiki/NIST_RBAC_model. (Last visited: October 2012).
- Ross, R., Katzke, S., Johnson, A., Swanson, M., Stoneburner, G., Rogers, G., and Lee, A. (2005). Recommended security controls for federal information systems. *NIST Special Publication*, 800:53.
- Sanka, S., Hota, C., and Rajarajan, M. (2010). Secure data access in cloud computing. In *Internet Multimedia Services Architecture and Application (IMSAA), 2010 IEEE 4th International Conference on*, pages 1–6. IEEE.
- Sengupta, S., Kaulgud, V., and Sharma, V. (2011). Cloud computing

- security-trends and research directions. In *Services (SERVICES), 2011 IEEE World Congress on*, pages 524–531. IEEE.
- Shehab, M., Bertino, E., and Ghafoor, A. (2005). Secure collaboration in mediator-free environments. In *Proceedings of the 12th ACM conference on Computer and communications security*, pages 58–67. ACM.
- Sirisha, A. and Kumari, G. (2010). Api access control in cloud using the role based access control model. In *Trendz in Information Sciences & Computing (TISC), 2010*, pages 135–137. IEEE.
- Subashini, S. and Kavitha, V. (2011). A survey on security issues in service delivery models of cloud computing. *Journal of Network and Computer Applications*, 34(1):1–11.
- Thomas, R. and Sandhu, R. (1998). Task-based authorization controls (tbac): A family of models for active and enterprise-oriented authorization management. *Database Security*, 11:166–181.
- Vincent Hu, K. S. (2012). Guidelines for access control system evaluation metrics. Technical report, National Institute of Standards and Technology. <http://csrc.nist.gov/publications/nistir/ir7874/nistir7874.pdf>.
- Wang, G., Liu, Q., and Wu, J. (2010). Hierarchical attribute-based encryption for fine-grained access control in cloud storage services. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 735–737. ACM.
- Yu, S., Wang, C., Ren, K., and Lou, W. (2010). Achieving secure, scalable, and fine-grained data access control in cloud computing. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE.
- Zhang, X. (2006). *Formal model and analysis of usage control*. PhD thesis, Citeseer.
- Zhang, X., Nakae, M., Covington, M., and Sandhu, R. (2008). Toward a usage-based security framework for collaborative computing systems. *ACM Transactions on Information and System Security (TISSEC)*, 11(1):3.
- Zhang, X., Park, J., Parisi-Presicce, F., and Sandhu, R. (2004). A logical specification for usage control. In *Proceedings of the ninth ACM symposium on Access control models and technologies*, pages 1–10. ACM.