# User Centric Access Control Policy Management Framework for Cloud Applications


**By**

**Misbah Irum**

**2011-NUST-MS-CCS-021**

**Supervisor**

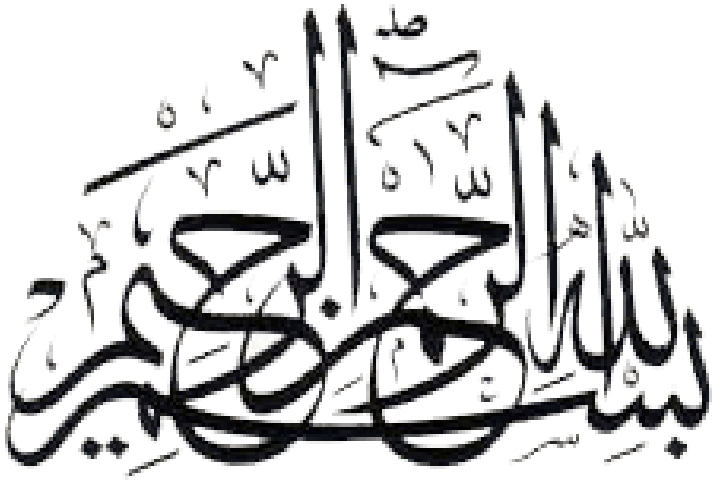
**Dr. Abdul Ghafoor**

**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.

(December, 2014)


بسم الله الرحمن الرحيم

# Approval

It is certified that the contents and form of the thesis entitled "User Centric Access Control Policy Management Framework for Cloud Applications", submitted by Misbah Irum have been found satisfactory for the requirement of the degree.

Advisor: **Dr. Abdul Ghafoor**

Signature: _____

Date: _____

Committee Member 1: **Dr. Awais Shibli**

Signature: _____

Date: _____

Committee Member 2: **Dr. Usman Younis**

Signature: _____

Date: _____

Committee Member 3: **Mr. Qaisar Choudhary**

Signature: _____

Date: _____

*Dedicated*

*to*

*my everloving Parents*

*and beloved sisters!*

# Certificate of Originality

I hereby declare that this submission titled User Centric Access Control Policy Management Framework for Cloud 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. I also verified the originality of contents through plagiarism software.

Author Name: **Misbah Irum**

Signature: _____

# Acknowledgment

First of all, I would THANKS to ALLAH ALMIGHTY for helping me in completion of my Master's Degree. By the grace of ALLAH, I am able to accomplish this goal in my life with full dedication and motivation. After that, Special Thanks to my precious Ammi and Abu for their everlasting love and too much support throughout my thesis phase.

I would like to thanks my supervisor Dr. Abdul Ghafoor who has always inspired me with his dedication and enthusiasm to work. His guidance, motivation and mentorship by far had been the most encouraging factors to complete my research work. I would like to appreciate and thank my committee members, Dr. Awais Shibli, Dr. Usman Younis, and Mr. Qaiser Chaudhary who had always given me their precious time and guided me through my thesis work. Finally thanks to my beloved husband, my sisters and my friends for supporting me throughout my thesis phase.

Misbah Irum

# Table of Contents

# List of Figures

# List of Tables

# Abstract

Cloud computing environment is a collection of various Cloud applications deployed by different Cloud service vendors for their customers. The online availability, variety and easy access of Cloud applications allow users to create, upload and store numerous resources across the Cloud. However, protection of these resources from different security threats in Cloud environment is still a serious concern for the Cloud users. Cloud applications provide diverse and complex authorization and access control mechanisms to different Cloud users. In addition to that access control is also limited and tightly coupled with the functionality of the applications and does not cater the access control requirements of individual users. Securing every resource with different, complex and customized access control solutions is a tedious task and results in poorly protected resources susceptible to unauthorized access which further leads to data theft, identity theft, fraud and different other security threats.

In this regard, a new approach to access control in Cloud environment is presented in this research work. It externalizes access control from Cloud applications and enables users to create and manage access control policies on their resources according to their access control requirements. The framework also provides users with a central control point and a standard policy definition language to specify and manage access control on all their resources scattered across the Cloud. We presented the framework and described the protocol which defines the interaction between different components of the system to specify and enforce User-Centric Access Control policies using XACML standards. To show the applicability of the designed framework, we developed a prototype using Google spreadsheet as the Cloud application. The prototype is then validated and verified from security and functional perspectives. To verify the security features of the designed system, a threat model is formulated which identify different security and access control threats and explain the protection mechanisms incorporated within the designed system to eliminate these threats. To check the correctness of the system, various categories of test cases are formulated and performed through JUnit testing. Successful execution of test cases verifies the claims of the user centric access control framework in providing users with the control to define access control policies according to their requirements.

# Chapter 1

# Introduction

Cloud computing is an evolutionary technology which provides on demand, robust and scalable computing services to the IT industry and business world. Cloud computing is defined by the National Institute of Standards and Technology (NIST) as: "*A model for enabling on-demand, ubiquitous and convenient network access to a shared pool of configurable computing resources (e.g., networks, applications, storage, servers, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This Cloud model promotes availability and contains five essential characteristics, three delivery models and four deployment models*" [1]. The important characteristics include dynamic provisioning, ubiquitous network access, rapid elasticity, shared infrastructure and managed metering. These characteristics provide benefits to organizations and individuals in the form of cost effectiveness, scalability, optimization and operating efficiencies. Furthermore, Cloud computing provides services to customers by using three delivery models which include Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). These Cloud delivery models can be deployed as Public Cloud (open to all), Private Cloud (own by organizations for their private use), Community Cloud (combination of Public and Private Clouds) and Hybrid Cloud consisting of multiple Clouds. [2].

The gradual evolvement and rapid development of Cloud computing environment provides users with endless services in the form of various Cloud applications. The online availability, low cost, ease of access and the variety of these applications has facilitated the migration of traditional desktop applications to distributed Cloud applications. Through the use of different mobile computing devices such as smart phones, laptops, tablet PCs, users can access these applications at anytime from anywhere therefore, becoming a constant source of creation, storage and distribution of numerous content online. Particularly, users are storing huge amounts of contents in the form of personal information on social networks, blogs and reviews or by hosting resources such as documents, pictures etc. on different applications. However, still a large number of individuals and organizations hesitate to migrate to Cloud environment because of its potential risks [3]. Different surveys on Cloud computing indicates that security and privacy issues are the major reasons creating hindrance in its adoption at a wider scale [3] [4] [5]. Inadequate security controls implemented by Cloud applications give rise to various security and privacy threats including unauthorized access, theft, fraud etc. [6] [7]. The development of new and effective security mechanisms to cater privacy and access control issues in cloud

environment is necessary for the adoption of this new computing paradigm by the IT industry and business world at a large scale.

The focus of this research is to provide a solution for User-Centric access control in Cloud environment. As mentioned in the above paragraphs that Cloud computing is a collection of various Cloud applications deployed by different Cloud vendors to provide a variety of services to numerous users. A Cloud user can use one or multiple Cloud applications for various purposes. However, Cloud computing does not facilitate users with a single access control and authorization mechanism using a standard policy definition language or a sole management tool for different Cloud applications. On the contrary, every Cloud application has its own authorization mechanism, access control solution and policy definition language which Cloud users employ for the protection of their resources from unauthorized access. This creates a lot of hassle for the user in terms of managing access control of different resources hosted on various Cloud applications. Furthermore, access control provided by Cloud applications is limited and tightly coupled with the functionality of the application and is not flexible to cater individual user's access control requirements [8]. It can be a good approach that the definition of access control policy should be in the control of user or owner of the resource which may increase the trust of users in the Cloud environment. Currently, the Cloud applications are not only dictating how access control policies have to be specified but also specified using their own policy formats and standards [6]. Use of such complex access control mechanism, diverse and incompatible policy languages to protect numerous resources scattered across the Cloud results in a tedious and error-prone task for the users. Thus results in poorly protected resources susceptible to unauthorized access which further leads to data theft, identity theft and various other security threats.

After analyzing the existing solution and their problems as specified in the above paragraph, we concluded that there is an imminent need of a mechanism which should enable users to define access control policies according to their own requirements. The mechanism should be in the form of an integrated system that can be used irrespective of different Cloud applications. It should allow users to create and manage access control policies for all their resources spread across the Cloud by employing a standard policy definition language from one central location. The designed User-Centric Access Control framework empowers users with full control to protect their resources from unauthorized access hosted by various Cloud applications. Users are provided with an interface which allows them to set policies for their resources which are dynamically formed at the back end. Moreover, the designed system provides users with a central control point that enables them to manage access control on all their resources irrespective of their location on the Cloud.

## 1.1 Motivation

With the rapid development of Cloud environment, users are storing numerous resources on the Cloud in the form of documents, pictures, videos, etc. They share these resources with other users and applications for various professional and personal purposes. The user (owner) of the resources specifies access control on these resources to define the sharing process. However, the extent to which the owner can define access control on its resources depends upon the access control mechanisms adopted by the Cloud applications hosting those resources. Hence, users are bounded to specify access control options provided by the Cloud applications, which often does not meet user's access control requirements. This results in unprotected resources and decrease level of trust on the Cloud application. In this regard, the main motivation of this research is to provide a User Centric solution to define access control so that users can share and protect their resources according to their own security requirements.

## 1.2 Aims and Scope

In this thesis, we aim to identify and address the most prevalent issues in the domain of Access Control in Cloud environment. To achieve this goal, first we will explore different access control frameworks, highlight the security issues and then finally aim to design and implement a secure User centric access control policy management framework for Cloud applications that meets the pressing needs of different Cloud users. Our scope, therefore, is limited to three research objectives:

- To design an authorization framework which externalizes access control from cloud applications and enable users to protect their resources according to their access control requirements.
- To provide users with an integrated central control point to manage access control on all its resources irrespective of their location on the Cloud.
- To enable users to specify access control on all it resources through a standard policy language i.e. XACML.

## 1.3 Thesis Organization

The presented thesis has been organized into different chapters in which each chapter gives certain aspects of our research. Following is the brief description of all the chapters.

- Chapter 1: entitled "Introduction" briefly describes the research area, problems and the abstraction of the designed solution. Furthermore, the chapter gives details about aims, scope and major contributions of the thesis.

- Chapter 2: entitled “Background and Related Work" gives details of literature survey which has been conducted throughout the research phase. It elaborates the traditional access control models and User centric access control models. It also compares and provides a critically analyzes of existing solutions and approaches.

- Chapter 3: entitled “Research Methodology" explains the research approach and methodology followed in this thesis. A hybrid approach has been used where different suitable research methodologies are used to achieve our objectives.

- Chapter 4: entitled “User Centric Policy Management for Cloud Applications" describes the designed system for access control in Cloud environment. The main modules of the system have been discussed along with designed protocol.

- Chapter 5: entitled “Design and Implementation of the Prototype system” describes the implementation of the prototype of the designed solution. It explains different prototype components and their working and also describes the tools used to build the prototype.

- Chapter 6: entitled “Evaluation of Research Work" explains the details of research work validation using the different Test cases and Threat Model.

- Chapter 7: entitled “Conclusion and Future Work" concludes the thesis and highlights the potential future prospects of our research work.

## *Summary*

*This chapter describes the research area in detail and gives an overview of existing access control mechanisms in cloud environment. It also highlights various problems face by different users while limiting access of their resources using existing access control mechanisms. Furthermore, it also gives an overview of the purposed access control framework and describes the motivation to carry out the research work.*

# Chapter 2

# Background and Related Work

## 2.1 Access Control in Traditional Cloud Applications

In traditional Cloud and web based applications access control is developed within the applications and is modeled according to the functionality of the application. The main focus of Cloud service providers is on the functional and operational requirements of the application and authorization mostly remains as a side issue. Furthermore, they do not take user's security and privacy requirements under consideration while designing the authorization frameworks. To analyze and explain the problems faced by users while employing such access control mechanism, we describe a simple scenario as follows:

### 2.1.1 Scenario

To find the shortcomings and problems of existing access control mechanism adopted by Cloud and Web applications, we examine a scenario depicting a Cloud user who creates, stores and shares various resources with other users and applications through different Cloud services.

In this scenario, we have assumed a Cloud user Chris who is a frequent traveler and a hobbyist photographer. While traveling, he takes various pictures, makes videos and documents his entire trips. To store and share these resources, he uses three famous Cloud applications which are called as CPics (a service to host pictures), CVideos (a video hosting service) and CDocs (a Cloud based document management system to create and manage different types of documents).

After uploading his resources, Chris wanted to share some of his videos with his friends Bob and Alice. To do so, he logs into CVideos and specify access control by choosing certain options provides by CVideos to reflect his sharing requirements. Chris also decided to share some photos and trip reports with Alice and Bob and logs into CPics and CDocs and specify access control on pictures and documents separately as shown in figure 2.1

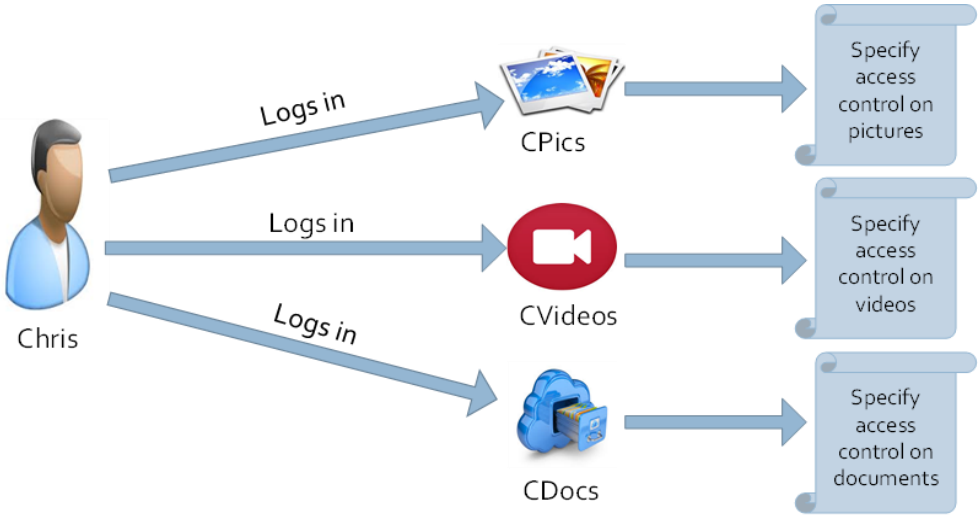



**Figure 2.1: Access control specification in different Cloud applications**

After sometime other friends of Chris wishes to see his photos and other details about his trips. Therefore, Chris logs into every application and modify access control polices to share the same resources with other friends. Furthermore, when he needs to share different other resources with the same friends, he then creates new access control policies on each resource hosted on a different Cloud application.

## 2.1.2 Problems in Traditional Access Control Solutions

To identify the limitations of the existing access control mechanisms in the Cloud environment, we critically analyze the scenario described in the previous section. Following are the details of different weaknesses that we observed.

Firstly, Cloud applications have mostly implemented isolated authorization mechanisms in which access control is tightly incorporated with the functionality of the application. Such authorization mechanisms provide limited access control options which are unable to configure particular user's access control requirements. These access control solutions often address simple circumstances where resources are made public or private to certain number of users. Chris, for example, must employ the authorization mechanism implemented by CPics, CVideos and CDocs which may not fulfill his all the access control requirements. They may not have the functionality to enable Chris to form different user groups and specify access control on them. Astonishingly, many popular applications do not have this simple functionality. Furthermore these access control solutions mostly lack fine grained access control functionality. For example if Chris

wants to share some sections of his particular trip document to Alice and some other sections with Bob, but CDocs's authorization mechanism does not provide such fine grained access control functionality and forced Chris to just share the whole document with his friends.

Secondly, another important weakness in the existing solutions is the absence of a standard access control policy language in different Cloud applications. Every Cloud application uses different access control solutions based on unique policy languages more suitable to their functional requirements. For example, in our scenario, CPics may have adopted a simple access control list or matrix mechanism while CVideos and CDocs applications may have implemented a more flexible mechanism. This does not enable Chris to define access control policies on his pictures, videos and trip documents only once because they are hosted by different Cloud applications. Furthermore, if Chris transfers his resources to another Cloud application, he then has to redefine the same policies using a different policy language on the other Cloud application.

Another weakness we found is the management of access control policies spread across the Cloud. User various resources and their access control policies are hosted by different Cloud applications User have a limited view of all the applied access controls and if he want to add new policies, update the existing ones or to audit them, he then have to log in to every Cloud applications and configure access control appropriately. This results in a very challenging and error prone task for the user. In the context of the above mentioned scenario, Chris does not have a complete view of the specified access control polices related to his documents, pictures and clips at CDocs, CPics and CVideos respectively. With the continuous increase of resources, Chris host on the Cloud, organizing and handling relations among access control polices and resources results in a very complex and tedious task.

Based on our analysis of the scenario and the weakness we observed and found out the following limitations of existing access control solutions:

- Existing Access control solutions are inadequate and do not cater individual user's security and access control requirements.
- Users have to employ diverse and complex access control policy languages and tools.
- Users do not have a unified control point to manage access control policies scattered across numerous Cloud applications.

## 2.2 User Centric Access Control Models

User centric authentication for Cloud and traditional web environment is widely adopted through the use of OpenID [34], Shibboleth [35] and many other frameworks mentioned in [9] which allow users to control their identity information according to their own security requirements. However, access control in traditional web applications and Cloud services is not user-centric. Little research is done in this domain which is as follows:

### 2.2.1 Related Work

A User centric authorization framework provided by Moorsel et.al [10] externalizes access control form web applications and provides it in the form of an external authorization server. Users are redirected to the Authorization server where they can create and store access control policies regarding their resources. Request to access a protected resource is also redirected to the authorization server which after evaluating policies provides requestor with an authorization token. The web application verifies the authorization token and provides requestor with the access to the requested resource. The following figure shows the steps involved in the whole process.

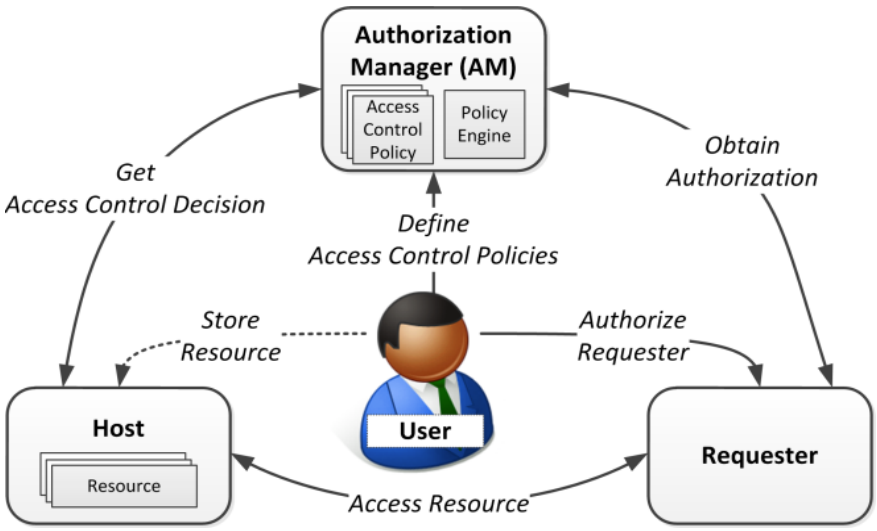



**Figure 2.2: User managed Access control for Web**

The proposed framework allows user to choose Authorization server based upon their preferences. A trust relationship has to be established first between the Authorization server and the web application. The framework allows a user centric approach to access control in Web environment but the designed protocol is complex and comprises of many steps. The back and forth redirection from host to authorization server is a time consuming process. Furthermore the framework does not facilitate the offline creation of policies and user has to be online to access the authorization server and specify access control. Moreover, the authorization token in possession of the requestor is susceptible to different attacks and can lead to unauthorized access of resources.

A unified user centric approach is adopted by Kapil Singh in [11]. The paper describes a framework which allows users to set access control on their content before uploading it on the web applications. The framework consist of two major components 1) xAccess extension which resides in the user's computer and is used to set access control on contents 2) xAccess server component resides on the application's server which parse the contents according to the access control specified by the user. xAccess extension provides user with an interface and tools to specify access control polices which are then translated into categorizes of xAccess base model. The base model comprises of Role Base Access Control (RBAC) scheme which convert the categories into particular roles and define access control on these roles. Figure shows the designed framework.

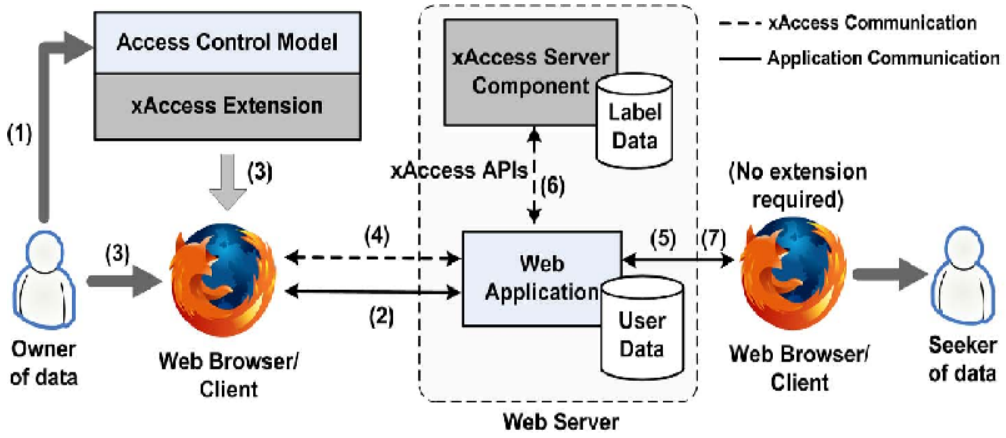



**Figure 2.3: xAccess Architecture**

The web application has delegated its authorization functionality to the xAccess server component. All access requests are forwarded to the xAccess server component which filters the requested resources according to the categories defined by the owner. The framework provides a user centric access control model; However, It is not a generic system and operates in close environment. The author mentioned that different access control models have to be implemented

in xAccess extension in order to translate them into base model. Furthermore the xAccess server component must be installed in every Web application and different API's have to be implemented for the communication between different applications and xAccess server component.

Security as a service in Cloud environment is gaining a lot of popularity now a days. Different Cloud service providers are providing various security solutions like identity management system (IDMS), intrusion detection and prevention systems etc. as Cloud services. Authors of [12] presents a Policy management system as a service in Cloud environment which provides a unified control point to users to specify and manage access control on their resources hosted on the Cloud. The main component of the system is a Policy Management Service Provider (PMSP) server that provides users with an interface and tools to generate and manage access control policies on their resources. PMSP facilitates users by eliminating the need to understand and employ different access control mechanisms of different Cloud application. Instead, users can create access control policies by using a unified policy language and can manage all the access control policies from a single control point.

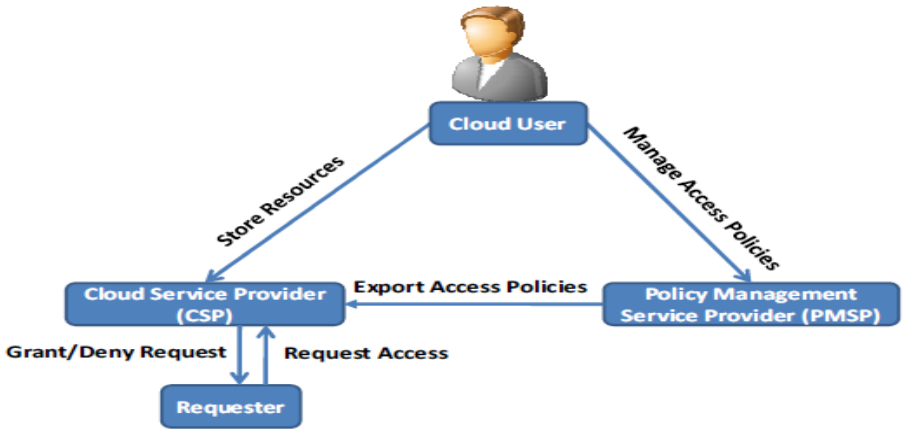



**Figure 2.4: Policy Management framework for Cloud Environment**

In the designed system, PMSP only provides policy creation and management features. Once the policies are created they are transferred to the Cloud service provider (CSP) hosting the resources. In order to transfer these policies to the hosting CSP's repository, the underlying policies first have to be translated into the CSP's specific policy language. This can cause a lot of compatibility issues and API's for every CSP has to be implemented for the translation process. Furthermore, there is no relationship and coordination between the policy creation and policy enforcement modules. User creates policies at the PMSP according to their requirements while CSP enforce these policies. Different access control mechanisms are adopted by different CSPs

and they may not have the built in functionality to enforce policies according to the user specifications.

OAuth [13] is an industrial standard authorization delegation protocol which enables content owners to delegate limited access to their content to other clients or third party applications. It allows the owner to give access to his resources to other users and applications without sharing any credentials such as usernames and passwords. To gain access to a user's resource, the client or application first sends a request to the owner. The owner provides the requestor with an authorization grant which is then sent to the authorization server by the requestor. The authorization server generates and provides the requestor with an authorization token. Requestor sends this token to the hosting application which provides him access to the resource.

OAuth is widely deployed with in the Web environment and is used by very popular social networking web applications. However, there are certain security threats associated with such an open standard, which are that it does not allow individual users to specify access control according to their requirements. Thus, the protocol can also enable third party applications to get unauthorized access to other contents without the owner's consent.

In another study [14] a user centric approach is employed by the authors to protect user's privacy information. They describe a model in which user is given control to protects its private information i.e. username, password and other identity information. User's privacy policies are separated from user's access control policies and are stored at user's side. All requests regarding the private information is sent to the Server which in turns gets the information from the owner. The owner also specifies the privacy policies regarding the private information to the Sever. The server enforces these policies onto the private information and provides the requestor with the access to this information according to the owner's requirements. This allows user to have full control over the privacy policies enabling him to protect its private information according to its own security requirements. Figure shows the architecture of the designed system.

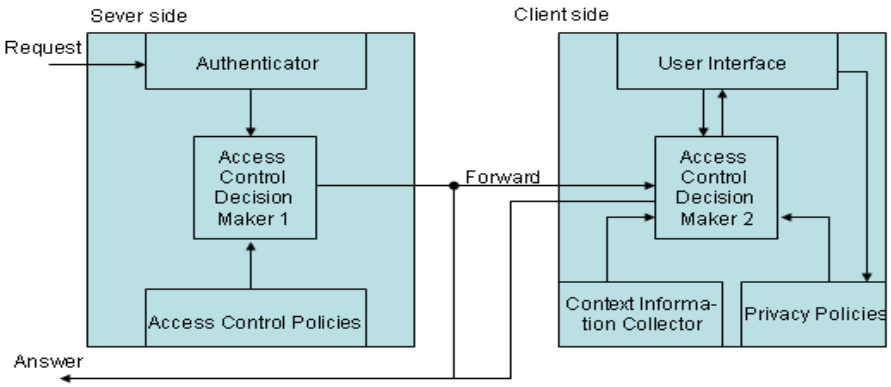



**Figure 2.5: User-centric privacy access control model**

The model protects user's privacy by protecting its private information; however, user can not define access control policies regarding their content and resources. Access control policies are specified and stored by the server side thus not achieving full protection on its resources.

### 2.2.2 Limitations

Considerable literature exists on access control models from different perspectives [15] [16] [17] [18] [19] [20]. However, little research work has been done on user centric access control models in Cloud environment and different models that have been purposed are explained above. Analyses of the existing work highlights the key weaknesses of the underlying models and are beneficial in providing a foundation for our protocol. Different frameworks uses different techniques to provide user centricity however some of the models are too complex to employ and some of them operates in close environment and cannot be deployed as a generalized framework in Cloud environment.

## *Summary*

*This chapter presents the existing work in the underlying research domain. It has elaborated the privacy issues in existing access control frameworks and elaborate them through the analysis of a scenario. The chapter further gives the overview of proposed user centric access control models in Cloud along with the analysis of these models to highlight their limitations and weaknesses. Analysis of existing framework also helped us in designing our own user centric access control framework and protocol from different perspectives.*

# Chapter 3

# Research Methodology

## 3.1 Introduction

The word research is generally used to describe a systematic study of phenomenon or the process of searching the knowledge and gather information specific to some domain under consideration. It can also be defined as a scientific investigation and exploration to discover the new facts and gain information [21]. People perform research whenever there is need to answer the highlighted questions and the appropriate solutions for identified problems [22]. The existing literature highlights that scientists and scholars have given different meanings to this concept, thus elaborating its effectiveness and usage with widespread perspectives. Research is the keen desire for knowledge that encourages us to inquire and discover the unknowns. The two main types of scientific research are deductive research and inductive research. Deductive research follows the top-to-bottom approach and works from the more general to the more specific. On contrary inductive research approach also called as bottom-up approach works from specific observation to general principles and theories[23] [24].

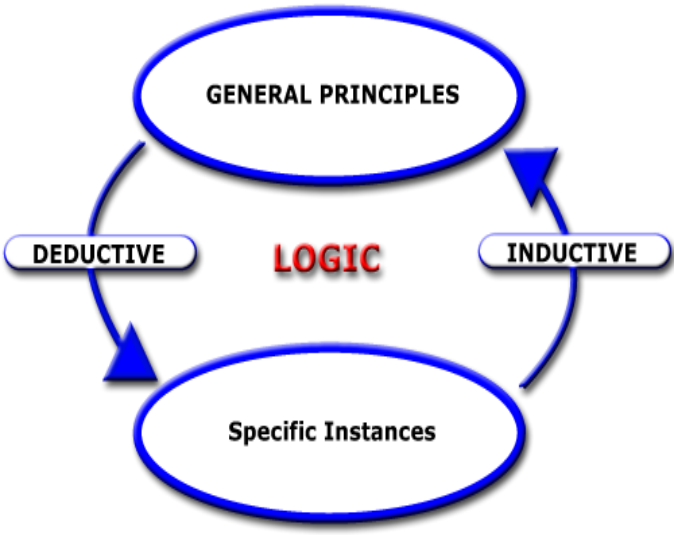



**Figure 3.1: Deductive and Inductive Research Approach**

## 3.2 Thesis Research Methodology

The aim of the current research activity is to describe the problem and then provide a solution and draws conclusion by narrowing down the focus. So, we have followed a deductive approach to solve the problem. Figure 1 shows the steps follows in a deductive approach:

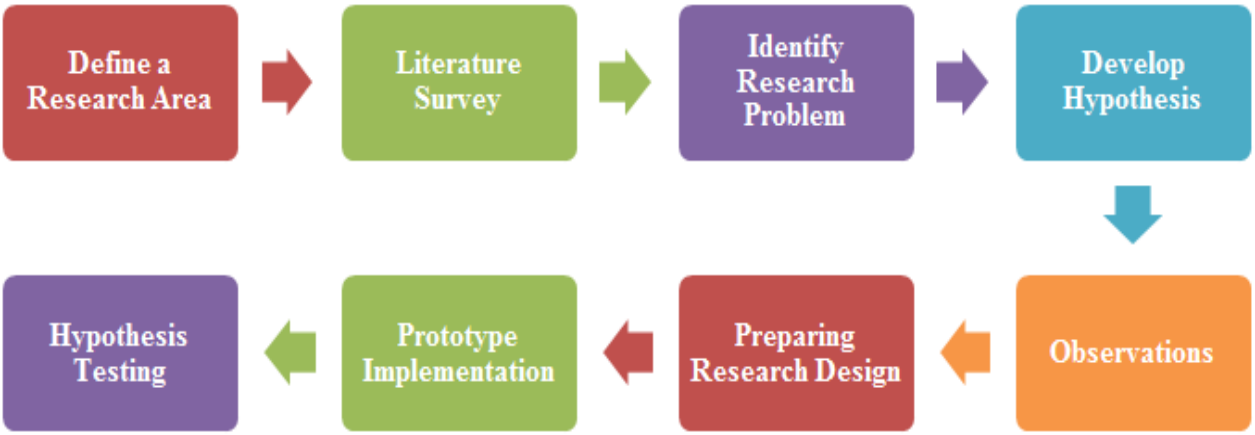



**Figure 3.2: Steps in Deductive Research Approach**

### ➢ Define a Research Area

First the problem domain has been identified by conducting extensive literature survey on state of the art technologies and related work. Cloud computing security issues have been identified as the main research area out of which we focus on access control which is one of the critical security and privacy issues in cloud computing environment.

### ➢ Literature Review

After defining the research area, the next step is to conduct the detailed literature survey which helps in identification of research problem. The survey was carried in two directions. First we study the traditional access control models and find their shortcomings by analyzing them through different scenarios. Secondly, a survey on different user centric access control models was done and is critically analyzed to find the advantages and limitations of each model. The holistic literature survey of existing access control models in Cloud domain provides us a strong foundation to formulate our thesis research problem.

## ➢ Identify Research Problem

Our extensive literature survey conducted in previous step of research process helps us to formulate three significant research problem statements related to access control in Cloud paradigm.

(a) There is a need to propose a user centric access control framework for Cloud environment which enables users to define access control on resources according to their requirements.

(b) There is a need to provide a standard authorization mechanism and access control policy language in Cloud environment.

(c) There is a need to provide users with an integrated central control point to manage resources and related access control policies scattered across the Cloud.

## ➢ Develop Hypothesis

Deductive research methods help us to develop the hypothesis for our research on basis of state-of-the-art literature survey and the problem statements formulized in the previous step. Our hypothesis is divided into three main questions in line with the identified problem statements.

(a) Is it possible to develop a framework which will put users in full control to define access control policies according to their requirements in Cloud environment?

(b) Is it possible to use a standardized authorization mechanism and an access control policy language in Cloud environment?

(c) Is it possible to provide users with a central control point to specify and manage access control policies on all their resources hosted on various Cloud applications?

## ➢ Observations

Using deductive research approach the following observations have been made to support the above developed hypothesis for thesis research.

- Cloud computing is still facing various security challenges despite of its various benefits and significant importance.

- Cloud applications provide users with limited access control options which do not fulfill user's access control requirements and results in poorly protected resources susceptible to different security threats.

- Different User centric access control models have been proposed for web and Cloud applications but are too complex and have different limitations.

- Different authorization mechanism and policy languages like XACML can be used as a standard in Cloud environment.

- User centric access control framework provides users with a unified control point to manage access control on all resources.

## ➢ Preparing Research Design

The research design has been formulated by doing the analysis of existing traditional access control models and proposed user centric access control models for Cloud environment. The research proposes the design for the identified problems by keeping in view the essential features of a user centric access control framework. The designed framework consists of different components and underlying protocols necessary for the communication between these components.

## ➢ Prototype Implementation

We have proved the identified problem statements and the devised hypothesis by implementing a prototype of the designed solution. The prototype implements the components and protocols of the designed framework and provide user with an interface to specify access control on its resources and XACML policies are dynamically formed at the back end. Google Spreadsheet application is used to show the applicability and integration of the designed framework with real world Cloud applications.

## ➢ Hypothesis Testing

The developed hypothesis is verified through validation of implemented User centric access control framework. In this regard, we have used different test cases and scenarios to verify and validate our proposed designed. The results of these test cases confirm that our hypothesis is true and the implemented system can effectively provide user with the control to define access control on all its resources according to its requirements.

## 3.3 Research Contributions

As now a days users are storing more and more resources on the cloud applications to share with other users or applications for various purposes. Users employ access control mechanism provided by cloud applications to share these resources with other users. However, the application centric access control mechanism does not cater individual user's requirements and left users worried about the protection of their resources from unauthorized access and different other security threats.

This research solve the above mentioned problems from a very different approach and provides a solution by externalizes access control from Cloud applications. Externalization of access control from cloud applications allow to develop a standard integrated authorization and access control mechanism that can be used with different application in the Cloud environment. This research provides such a user centric access control framework for cloud applications. The framework is designed by keeping in view user's access control requirements and the limitations of different proposed user centric access control models. User requirements are gathered by finding the shortcomings of the current application centric access control models. Critical analyses of different user centric access control models yields their limitations in providing a generalized user centric access control model for cloud environment.

The designed framework allows users to protect their resources by enabling them to define access control according to their requirements. The framework also provides users with a platform to control and manage resources and access control policies from a single point in Cloud environment. Every component of the designed framework has been chosen with respect to some responsibility. The main component is the policy specification module which provides user with the interface and tools to specify access control on its resources and XACML policies are dynamically formed at the back end. These policies are then stored onto the authorization server which generates decision by evaluating these policies. XACML is used as a standard and all access request and access decisions are formed using XACML policy language. Prototype of the designed framework is implemented by using real world cloud application to show its applicability and integration with existing cloud applications.

By putting users in control for specifying access control on their resources, the framework minimizes different security threats like unauthorized access and data leakage and increase user's trust on the Cloud application. At the end verification and validation of the designed framework is done by evaluating the implemented prototype through different test cases and scenarios. The result of the evaluation verifies our claims and shows the applicability and effectiveness of the designed framework.

## *Summary*

*Different research methodologies have been proposed for the research process which can be followed by the researchers according to their requirements. Each of these research methods has its different steps and impacts, which normally depend upon the research domain under consideration. The researcher can select any particular research methodology depending upon the targeted domain of research and scientific phenomenon. For the presented thesis, we have selected a Deductive research methodology which is used at various stages of research process to formulate the research problems, develop hypothesis and verification of implemented system. The chapter also describes the contributions made by this research work and features of the designed framework and its advantages over the existing models are explained in detail.*

# Chapter 4

# Construction of User Centric Access Control Policy Management Framework for Cloud Applications

After a detailed review and analysis of existing access control models and protocols, we have designed a framework which consists of the essential components for providing User-Centric access control policy management in Cloud environment. The detailed architecture and its underlying components are described in detail as follows:

## 4.1 Designed Framework

The designed framework provides users with the control to protect their resources according to their authorization and access control requirements. The designed framework consists of different components which provide different services to users to achieve user centricity in cloud environment. Each and every component of the designed framework is responsible to perform some particular tasks. The main component is the policy specification module which provides users with the interface and tools to define access control on their resources using XACML policy format. The component authorization server stores these policies and generates access control decisions based on the XACML policy file. The framework also incorporates a Strong Authentication server (SA) to verify the identity of different users. In the designed framework Authorization server, SA server and Cloud application components are deployed on a Cloud infrastructure. Whereas Policy specification module are resides on the user side and all communication between the user and cloud services is done through Gateway which is the entry point for the Cloud. Figure 4.1 shows the architecture of the designed user centric access control framework.

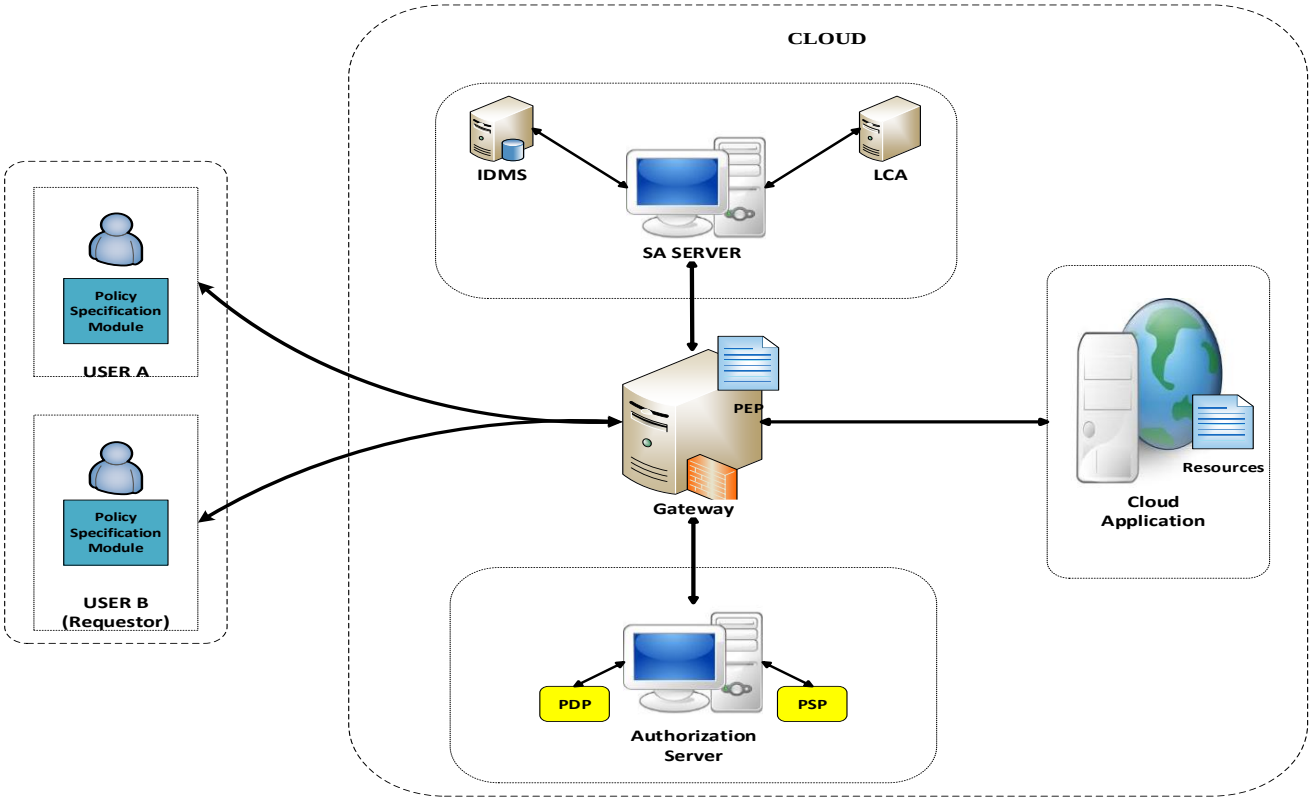



**Figure 4.1: Architecture of User Centric Access Control Policy Management Framework for Cloud Applications**

## 4.2 Components

Figure 4.1 illustrates the main architecture of the designed framework, whereas, the different components of the framework have been discussed further in detail below:

- **Strong Authentication Server (SA Server)**

  Authentication server holds the responsibility for managing user authentication. It validates the users as authentic or unauthentic by making decisions on the received authentication requests. This authentication server is based on FIPS 196 unilateral strong authentication protocol [25] using certificates. Users authenticate themselves by providing their digital certificates to the authentication server which provides SAML tickets after verification to the authenticated users for single sign on purposes.

- **Cloud User (CU)**

  CU is a client which uses various Cloud applications for different purposes. Cloud User can create stores and shares resources with other users or applications. A policy specification

module which acts as Policy Administration Point (PAP) resides on the user's computer to provide an interface for the user to create XACML policies necessary for the authorization decision. PAP manages the generation of XACML policies with a particular target. This target contains three fields termed subject, resource and action. These three fields need to be set in order for any particular policy to be generated or evaluated in the future. This module facilitates user to create customize XACML policies on its resources according to its own requirements.

- **Gateway**

  Gateway is the entry point for the Cloud. It acts as proxy server between Cloud users and other components of the designed framework. It is responsible for handling the requests initiated at the user end and the responses from the servers and passes the messages in either direction. It handles Cloud user's access requests and gets access control decision from the authorization server against those requests. It then fetches the resources from the Cloud application and parses it according to the user's specifications and then provides requestor with the requested resource. Therefore, it also acts as a Policy enforcement Point (PEP) in the designed system.

- **Authorization Server**

  Authorization server consists of three modules, Policy Decision Point (PDP) ,Policy Storage Point (PSP) and a Policy Information Point(PIP). PSP act as a policy repository and store access control policies in the form of XACML policy files. PDP receives access requests from the gateway, fetches the appropriate policies from the policy repository and evaluates it accordingly to generate an access decision.PIP is used by the authorization server to retrieve attribute values. These attributed values are context and environment based and needed to correctly evaluate a policy.

- **Cloud Application**

  A Cloud application acts as an on demand application granting access to a particular service. It enables Cloud users to create, store and share resources (documents, files, pictures etc.) with other users or applications. In the designed framework the Cloud application has delegated its authorization and access control functionality to the authorization server and to the user respectively. Access control decisions are generated by the authorization server on the behalf of Cloud application. It contains a resource repository and is only responsible for storing resources created or uploaded by users.

- **Requestor**

  Requestor is an authenticated entity that interacts with the cloud application to access protected resources of another user in the designed framework. The entity can be a user trying to access a protected resource e.g. a document of another user or it can be a photo

editing application trying to access photos stored by user's photo hosting application. The requestor sends an access request to the gateway which provides requestor with the access the requested resource according to the decision generated by authorization server.

## 4.3 Protocol

The designed protocol consists of three sub protocols which are Authentication and Authorization, Access control policy specification and Accessing protected resource. Before explaining these steps in detail some assumptions that must be considered as pre-requisite are as follows:

- All Cloud users are registered to the authentication server.
- Cloud service providers have already defined and stored access control policies in the authorization server about their registered Cloud users.
- Trust relationship is already established between Cloud application and authorization server

### 4.3.1 Authentication and Authorization Process

The first step for the Cloud user is to verify its identity to the authentication server. As shown in figure 4.2 the Cloud user sends an authentication request along with its certificate to the authentication server. The authentication server queries the IDMS and verifies user's certificate. After verification authentication server generates an authentication ticket for the Cloud user for single sign on. The ticket is encrypted with the gateway's public key and contains the following information.

**Ticket = EG (IDuser || Time Stamp || lifetime || Digital Signature)**

*Where, EG = Public key of gateway*

The Cloud user sends this ticket along with the application request it wants to access to the gateway. The application request contains the URL of the requested Cloud application. Gateway upon receiving the request decrypts the authentication ticket by using the key $E_M$ (shared by SA server) and verifies it as legitimate request by comparing the signed and unsigned parts of the message. It then generates an XACML authorization request for the Authorization server with the received User ID as a target subject and Cloud application URL/descriptor as the target resource. Authorization server fetches the policy created against the target subject from the policy repository and evaluates the received request based on the policy description. The

evaluation response is sent to the gateway for further action. If the response is equal to "permit" gateway sends the user ID along with login request to the Cloud application which in return provides user with the access to its account on the Cloud application.

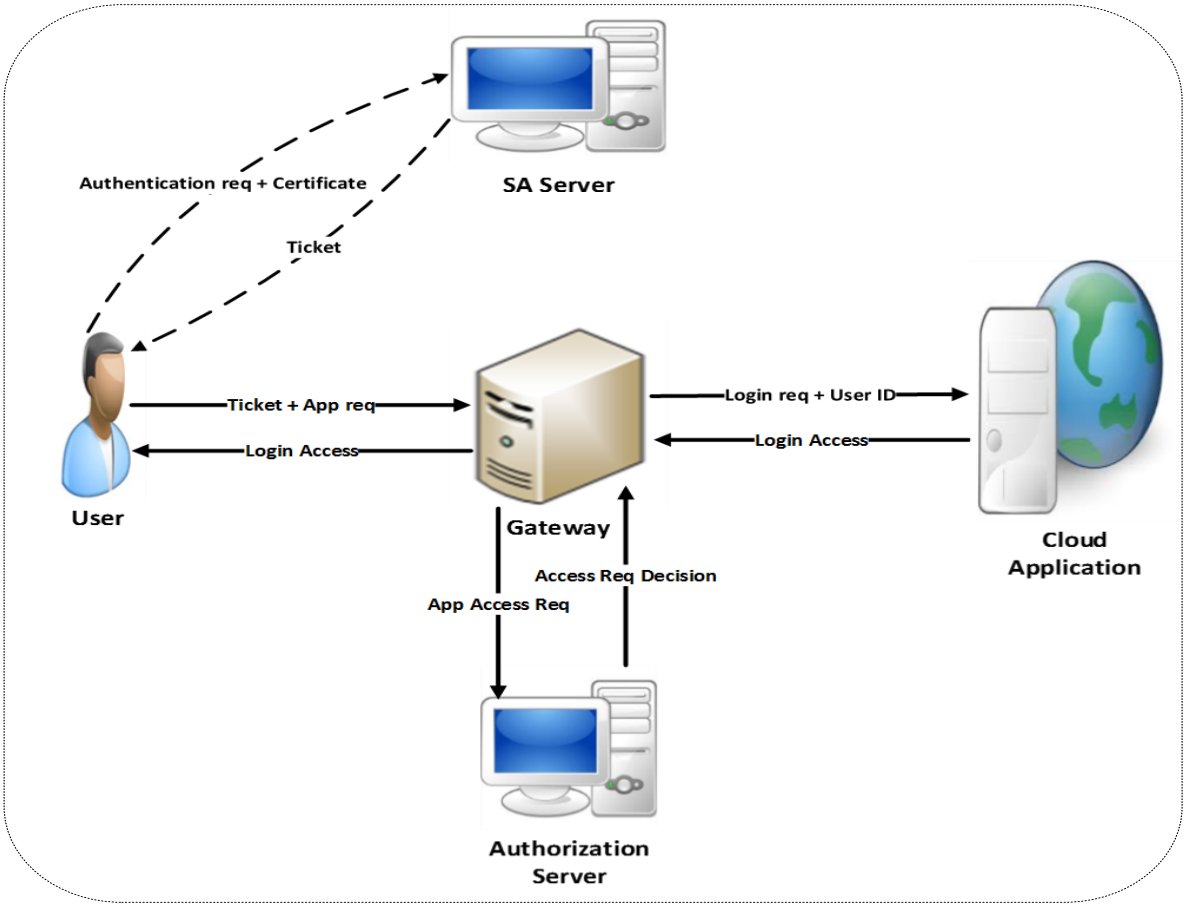



**Figure 4.2: Authentication and Authorization Process**

## 4.3.2 Access Control Policy Specification Process

After getting access to a particular Cloud application the user can now create, update or share resources with other users or applications. To share a resource, Cloud user must define access control policies to govern the sharing process. Cloud User can define these policies by using the policy specification module. From within this module Cloud User is free to customize, create and assign access rights to other users or applications according to his own security and access control requirements. The policy specification module provides the user with an interface and tools to define access control on its resource and access control policies are dynamically formed at the back end using XACML policy format. After defining policies, user uploads these policies onto the authorization server. The authorization server stores these policies into the policy storage module. Similarly the user can edit policies for previously created resources and updates them on the authorization server. Figure 4.3 illustrate the steps of this process.

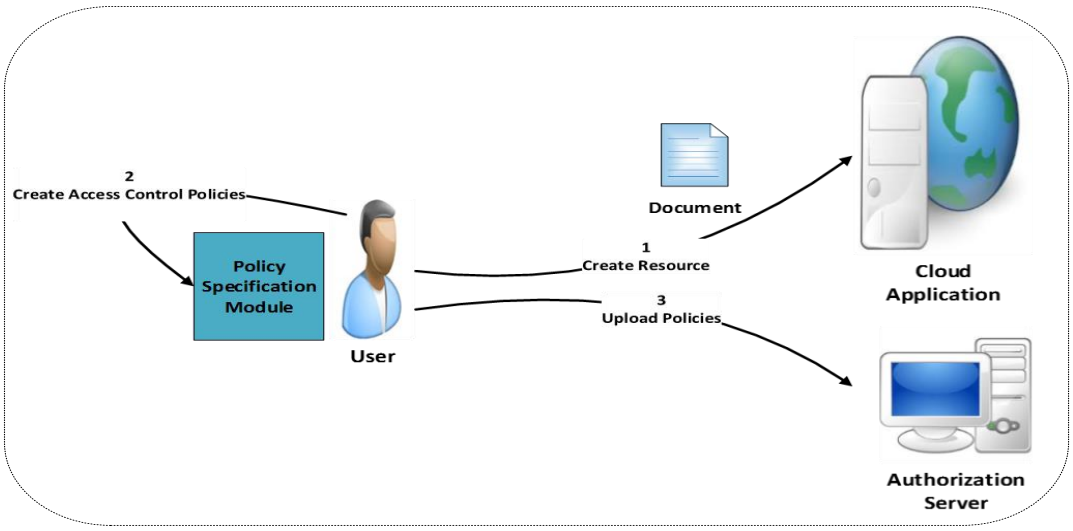



**Figure 4.3: Access Control Policy Specification Process**

### 4.3.3 Accessing Protected Resource

In this step a Requestor sends an access request of a protected resource of another Cloud user as shown in figure 4.4. The access request contains the URL of the protected resource. Gateway send User B's access request (for the protected resource of Cloud User) to the authorization server in XACML request format. Authorization server fetches policies regarding the particular resource and generates an access decision after evaluating these policies. The decision is then send to the gateway which after interpreting it fetches the resource from the Cloud application and parse it according to the received decision. It then provides User B access to only those parts of the resource which he has been assigned rights on. In case of deny or unknown decision gateway denies access to the requestor.

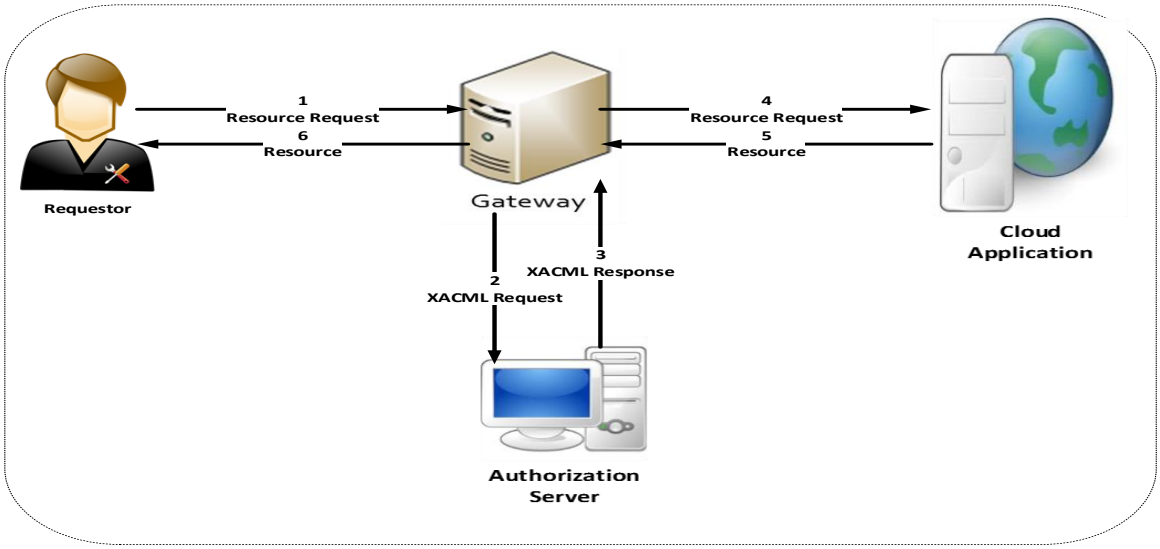



**Figure 4.4: Accessing Protected Resource**

A flow diagram depicting the complete steps of the above describe protocol is shown in figure 4.5.

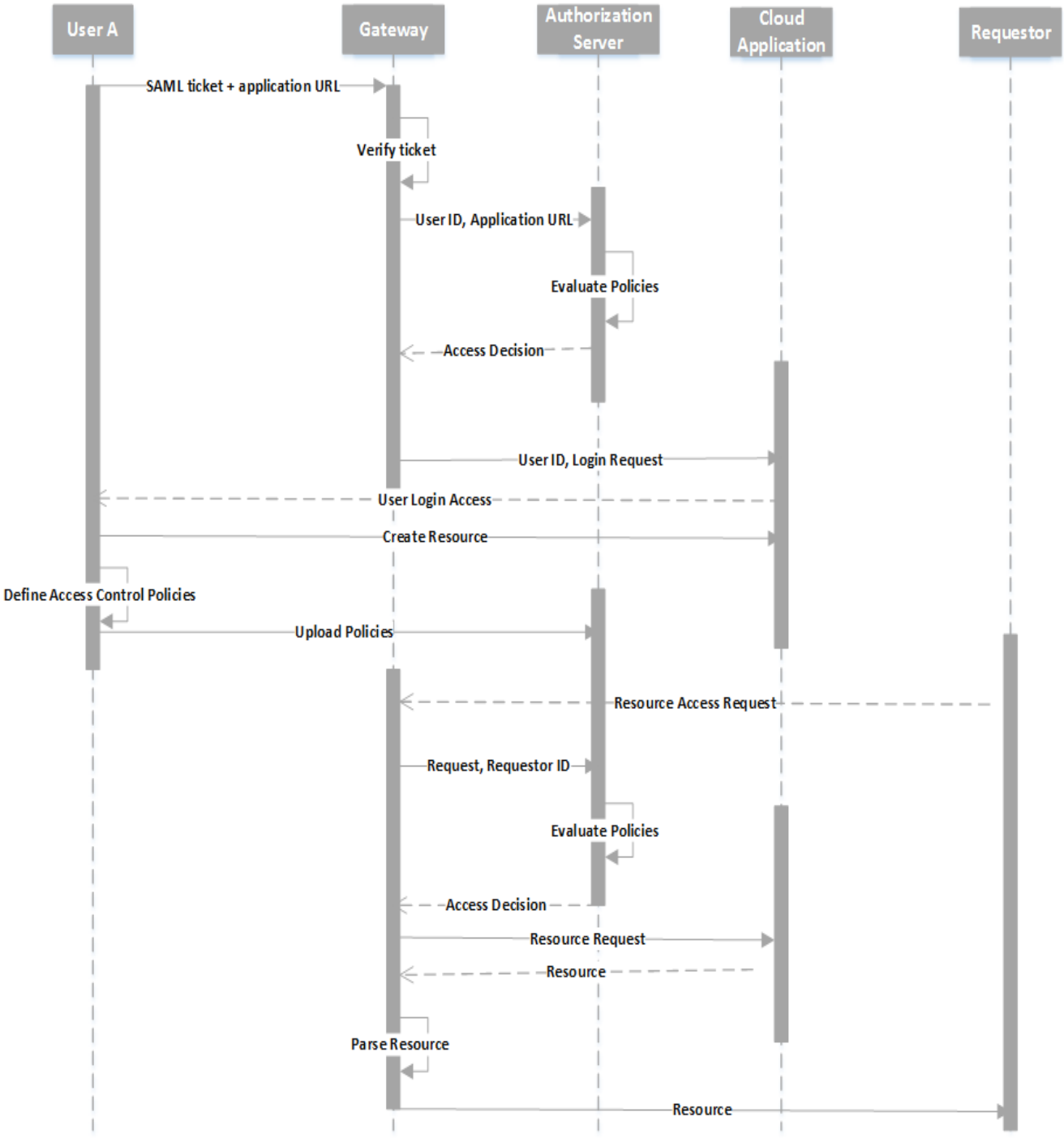



**Figure 4.5: Protocol Flow**

## *Summary*

*Architecture of the designed framework is described in detail and an architecture diagram of the designed solution is presented in this chapter. It also explains the purpose and major tasks of each component i.e. SA Server, CU, Gateway, Authorization Server, Cloud Application and Requestor. The chapter also explained the protocol which defines the interaction and communication between the components in detail.*

# Chapter 5

# Prototype Implementation

## 5.1 Related Technologies

Implementation of the user centric access control policy management framework for cloud applications involves the use of and XACML and Google Spreadsheet API. Below, we elaborate each one of them to help understand their functionality and significance.

### 5.1.1 XACML

Extensible Access Control Markup Language (XACML) [26] is an open standard access control policy language developed by OASIS. It covers all the detailed and general access control requirements and provides different points of extension to define new data types, policy/rule combining algorithms, functions, etc. XACML engine consist of a policy set which contains different policies and each policy contains a target, rule and a rule combining algorithms. The typical implementation of XACML includes a Policy Administration Point (PAP), a module responsible for the generation and management of the access control policies, Policy Enforcement Point (PEP), which prepares the request after extracting the attributes from query and collects the additional attributes related to requester and resource in question, and forwards the request to Policy Decision Point (PDP). PDP then finds appropriate policies by matching the targets and evaluate the policies to return the appropriate response (Permit, Deny, Intermediate or Not Applicable) to the PEP. We choose XACML because it is standard, generic, allows distributed policies, is really powerful and flexible enough to be used as a standard access control policy definition language in the Cloud environment and to enable users to specify access control according to their requirements.

### 5.1.2 Google Spreadsheet Application

Google spreadsheet application is an online Cloud application which provides spreadsheet processing services. We use Google spread sheet application in our implementation to show the applicability and integration of the designed framework with existing Cloud applications. User can create, update, import and share spreadsheets by using google spreadsheet application. It can be used to store and manipulate different types of data e.g. financial records, personal or other business data. Google spreadsheet service is widely use because of it's of access and sharing process. Google applies Access Control List mechanism to define the sharing process and users can make the whole spreadsheet either Public (visible to every one), Private (only visible to owner), anyone with the link and Specific people (by providing their email addresses). However, it does not provide coarse grained and fine grained access control functionality e.g. it does not allow users to share different records of the spreadsheet to different users and to group users together to specify access control policies on it.

## 5.2 Use Case Scenarios

To explain the prototype implementation we describe two use case scenarios. The first scenario describes the process of defining access control polices by using the policy specification module and the second scenario describes the process of user accessing a protecting resource.

### 5.2.1 Access Control Policy Specification (ACPS) - Use Case 01

Use-case 01 depicts an access control specification scenario, where a User A define access control on its resource (a google spreadsheet) using the policy specification module. After defining policies user upload these policies onto the policy repository of the authorization server. The detail steps are as follows:

1. To specify access control on a specific resource, user first selects the resource which is a google spreadsheet, then chooses a subject on whom this policies applies and then choose the action that the person will apply and the effect of that action as shown in figure 5.1.

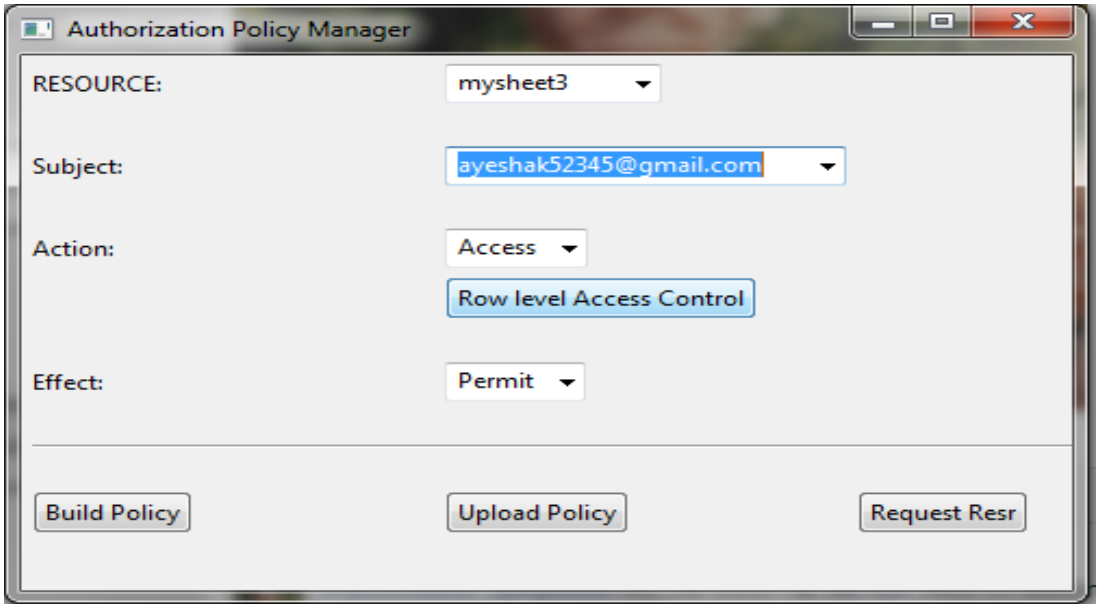



**Figure 5.1: Policy Specification Module**

2. To set access control on each record of the spreadsheet that which rows the User B (subject) can access of a particular spreadsheet, User A selects the row level access control option and enter the row numbers to set access control on each record as shown in the figure 5.2.

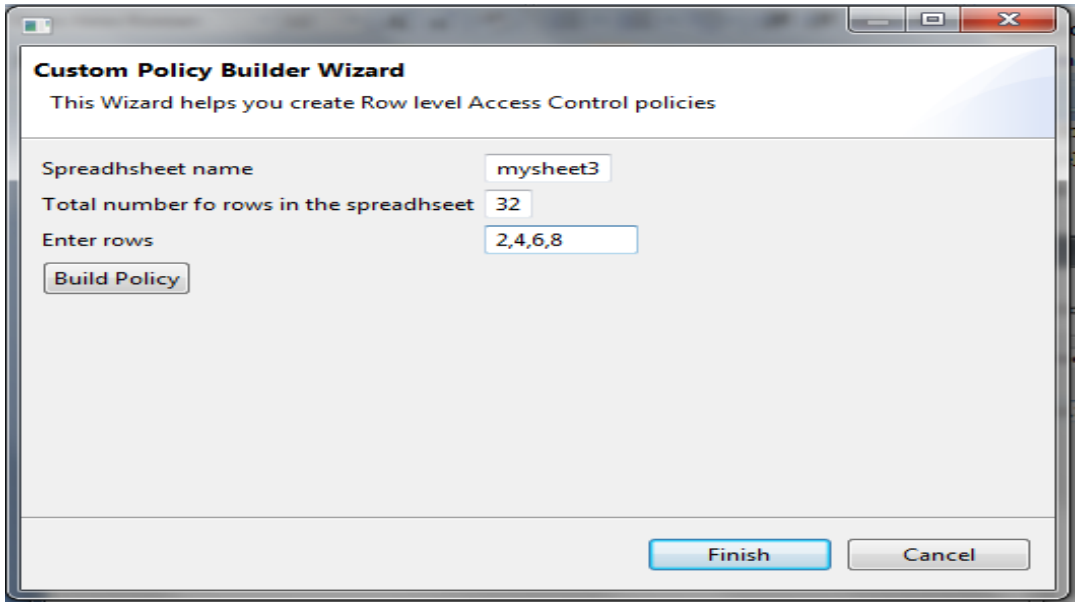
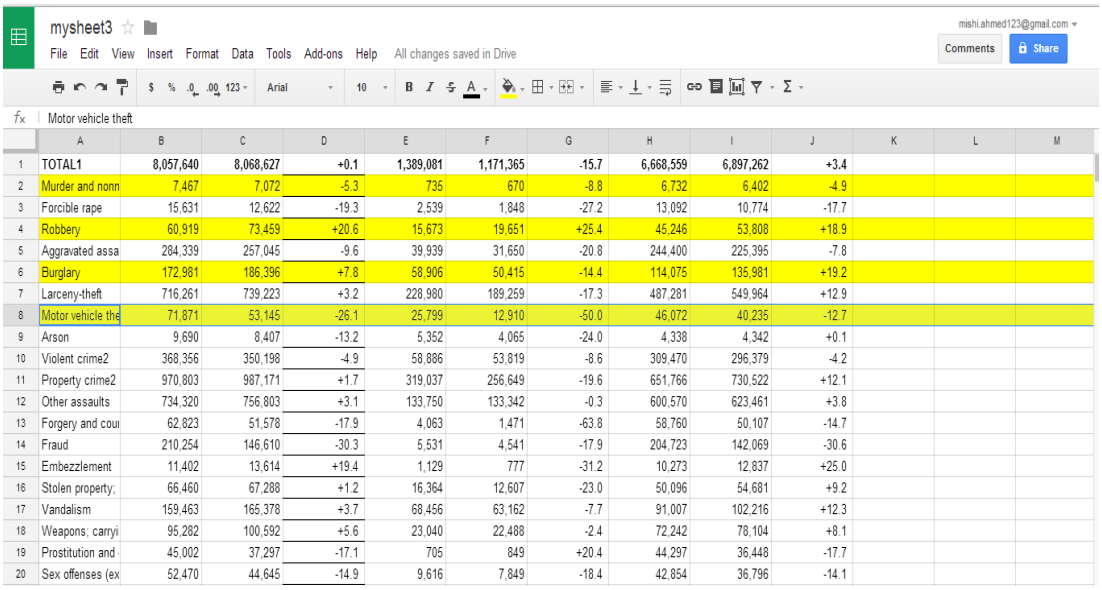



**Figure 5.2: Row Level Access Control**



mysheet3

File Edit View Insert Format Data Tools Add-ons Help All changes saved in Drive

mishi.ahmed123@gmail.com

Comments Share

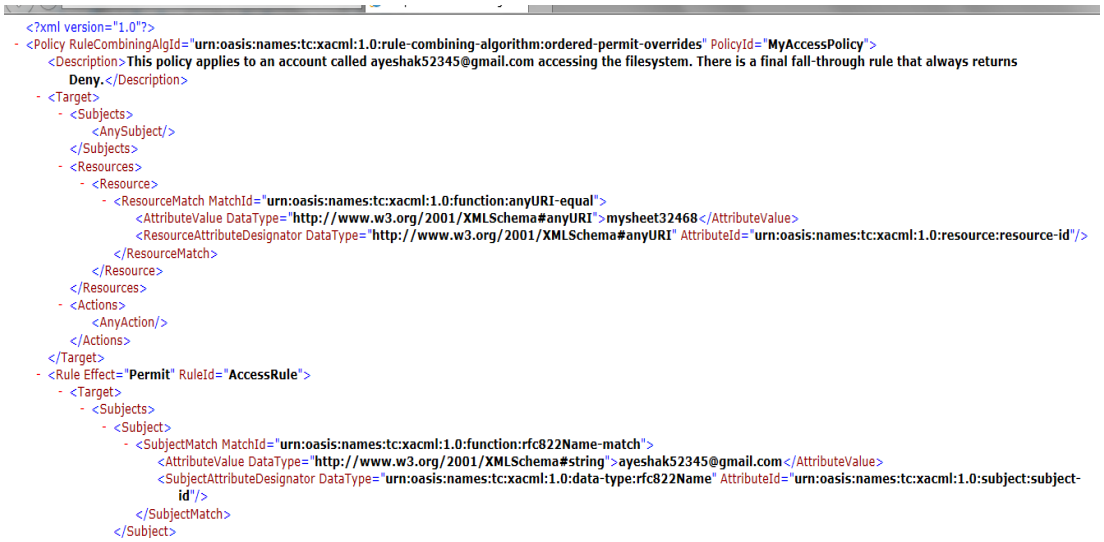
Arial 10

fx Motor vehicle theft

| | A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | TOTAL1 | 8,057,640 | 8,068,627 | +0.1 | 1,389,081 | 1,171,365 | -15.7 | 6,668,559 | 6,897,262 | +3.4 | | | |
| 2 | Murder and nonn | 7,467 | 7,072 | -5.3 | 735 | 670 | -8.8 | 6,732 | 6,402 | -4.9 | | | |
| 3 | Forcible rape | 15,631 | 12,622 | -19.3 | 2,539 | 1,848 | -27.2 | 13,092 | 10,774 | -17.7 | | | |
| 4 | Robbery | 60,919 | 73,459 | +20.6 | 15,673 | 19,651 | +25.4 | 45,246 | 53,808 | +18.9 | | | |
| 5 | Aggravated assa | 284,339 | 257,045 | -9.6 | 39,939 | 31,650 | -20.8 | 244,400 | 225,395 | -7.8 | | | |
| 6 | Burglary | 172,981 | 186,396 | +7.8 | 58,906 | 50,415 | -14.4 | 114,075 | 135,981 | +19.2 | | | |
| 7 | Larceny-theft | 716,261 | 739,223 | +3.2 | 228,980 | 189,259 | -17.3 | 487,281 | 549,964 | +12.9 | | | |
| 8 | Motor vehicle the | 71,871 | 53,145 | -26.1 | 25,799 | 12,910 | -50.0 | 46,072 | 40,235 | -12.7 | | | |
| 9 | Arson | 9,690 | 8,407 | -13.2 | 5,352 | 4,065 | -24.0 | 4,338 | 4,342 | +0.1 | | | |
| 10 | Violent crime2 | 368,356 | 350,198 | -4.9 | 58,886 | 53,819 | -8.6 | 309,470 | 296,379 | -4.2 | | | |
| 11 | Property crime2 | 970,803 | 987,171 | +1.7 | 319,037 | 256,649 | -19.6 | 651,766 | 730,522 | +12.1 | | | |
| 12 | Other assaults | 734,320 | 756,803 | +3.1 | 133,750 | 133,342 | -0.3 | 600,570 | 623,461 | +3.8 | | | |
| 13 | Forgery and cour | 62,823 | 51,578 | -17.9 | 4,063 | 1,471 | -63.8 | 58,760 | 50,107 | -14.7 | | | |
| 14 | Fraud | 210,254 | 146,610 | -30.3 | 5,531 | 4,541 | -17.9 | 204,723 | 142,069 | -30.6 | | | |
| 15 | Embezzlement | 11,402 | 13,614 | +19.4 | 1,129 | 777 | -31.2 | 10,273 | 12,837 | +25.0 | | | |
| 16 | Stolen property; | 66,460 | 67,288 | +1.2 | 16,364 | 12,607 | -23.0 | 50,096 | 54,681 | +9.2 | | | |
| 17 | Vandalism | 159,463 | 165,378 | +3.7 | 68,456 | 63,162 | -7.7 | 91,007 | 102,216 | +12.3 | | | |
| 18 | Weapons; carryi | 95,282 | 100,592 | +5.6 | 23,040 | 22,488 | -2.4 | 72,242 | 78,104 | +8.1 | | | |
| 19 | Prostitution and | 45,002 | 37,297 | -17.1 | 705 | 849 | +20.4 | 44,297 | 36,448 | -17.7 | | | |
| 20 | Sex offenses (ex | 52,470 | 44,645 | -14.9 | 9,616 | 7,849 | -18.4 | 42,854 | 36,796 | -14.1 | | | |

**Figure 5.3: Spreadsheet "mysheet3" of User A**

3. After specifying the access control user clicks the build policy button which creates an XACML policy at the back end depicting user's access control requirements. The policy file as shown in figure 5.3 is then uploaded onto the authorization server by clicking the upload policy button.



```
<?xml version="1.0"?>
- <Policy RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:ordered-permit-overrides" PolicyId="MyAccessPolicy">
    <Description>This policy applies to an account called ayeshak52345@gmail.com accessing the filesystem. There is a final fall-through rule that always returns
      Deny.</Description>
  - <Target>
    - <Subjects>
        <AnySubject/>
      </Subjects>
    - <Resources>
      - <Resource>
        - <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#anyURI">mysheet32468</AttributeValue>
            <ResourceAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#anyURI" AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"/>
          </ResourceMatch>
        </Resource>
      </Resources>
    - <Actions>
        <AnyAction/>
      </Actions>
    </Target>
  - <Rule Effect="Permit" RuleId="AccessRule">
    - <Target>
      - <Subjects>
        - <Subject>
          - <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:rfc822Name-match">
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">ayeshak52345@gmail.com</AttributeValue>
              <SubjectAttributeDesignator DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name" AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-
                id"/>
            </SubjectMatch>
          </Subject>
```

**Figure 5.4: XACML Access Control Policy**

4. The policy stores the information that user B can access the resource "mysheet3" of user A. The row information is stored in a database by the gateway which is also a policy enforcement point.

| subject | resource | rows | uniqnum |
|---|---|---|---|
| misbah@gmail.com | mysheet | 24567 | mysheet24567 |
| misbah@gmail.com | mysheet2 | 2,4,5,6,7 | mysheet224567 |
| misbah@gmail.com | mysheet | 1,2,5,7,10,12,20 | 1257101220 |
| misbah@gmail.com | mysheet | 10,12,34,30 | 10123430 |
| misbah@gmail.com | mysheet2 | 2,23,3,4 | mysheet222334 |
| misbah@gmail.com | mysheet | 2,3,5 | mysheet235 |
| misbah@gmail.com | mysheet | 2,3,5 | mysheet235 |
| ayesha@gmail.com | mysheet2 | 3,5,7 | mysheet2357 |
| ayeshak52345@gmail.com | mysheet4 | 2,4,5,7,10 | mysheet4245710 |
| ayeshak52345@gmail.com | mysheet4 | 2,4,5,7,10 | mysheet4245710 |
| ayeshak52345@gmail.com | mysheet3 | 2,4,6,8 | mysheet32468 |
| ayeshak52345@gmail.com | mysheet1 | 5,7,8,9 | mysheet15789 |

**Figure 5.5: Row Information Table**

## 5.2.2 Accessing a Protected Resource – Use case 02

In use case 02 a user B (Requestor) request to access a protected resource of user A. Gateway forwards the request to the authorization server which generates an access decision. Gateway provides user B with the access to the resource according to the decision generated by the authorization server. The steps involved are elaborated below:

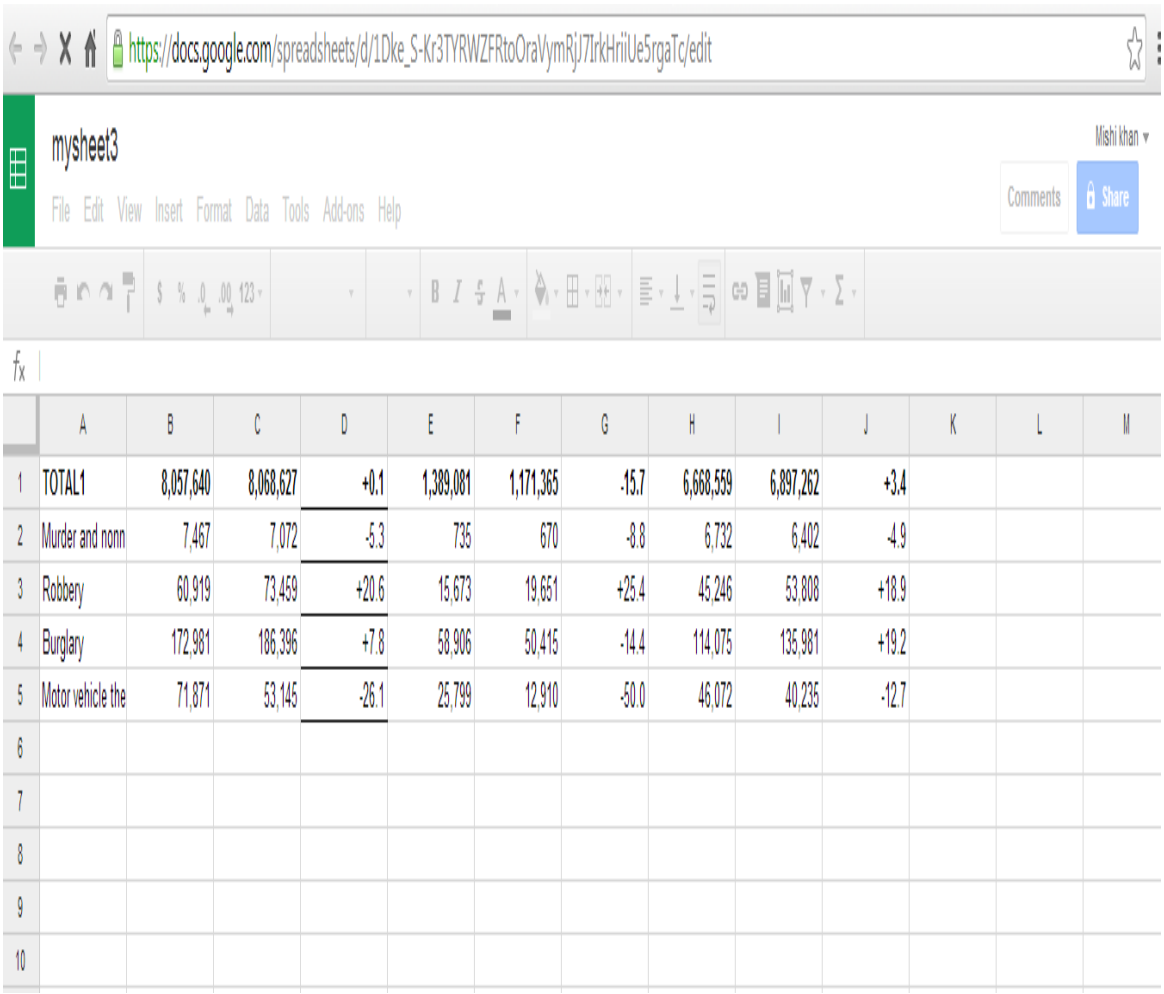
1. User B sends and access request to the gateway to access the protected Spreadsheet (mysheet3) of User A. Gateway formulates an XACML request as shown in the figure 5.6 and sends it to the authorization server.

```
<Request>
<Subject SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-
category:access-subject">
<Attribute AttributeId="group"
DataType="http://www.w3.org/2001/XMLSchema#string"
Issuer="admin@users.example.com"><AttributeValue>developers
</AttributeValue></Attribute>
<Attribute
AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<AttributeValue>ayeshak52345@gmail.com</AttributeValue>
</Attribute>
</Subject>
<Resource>
<Attribute
AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
DataType="http://www.w3.org/2001/XMLSchema#anyURI">
<AttributeValue>mysheet15789</AttributeValue></Attribute>
</Resource>
<Action>
<Attribute
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<AttributeValue>access</AttributeValue></Attribute>
</Action>
</Request>
```

**Figure 5.6: XACML Request**

2. Authorization server interprets the request and fetches the relevant policy from the policy storage module. Authorization server evaluates the policy against the request and generates and access control decision and sends it to the gateway. As indicated in the policy that user B can access spreadsheet "mysheet3" of user A so the generated decision is "Permit".

3. Gateway then retrieves the row numbers from the database and fetches the spreadsheet from Google server and parses it accordingly and provides User B with the spreadsheet that only contains the records permitted to user B by user A as shown in figure 5.7.




| | A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | TOTAL1 | 8,057,640 | 8,068,627 | +0.1 | 1,389,081 | 1,171,365 | -15.7 | 6,668,559 | 6,897,262 | +3.4 | | | |
| 2 | Murder and nonn | 7,467 | 7,072 | -5.3 | 735 | 670 | -8.8 | 6,732 | 6,402 | -4.9 | | | |
| 3 | Robbery | 60,919 | 73,459 | +20.6 | 15,673 | 19,651 | +25.4 | 45,246 | 53,808 | +18.9 | | | |
| 4 | Burglary | 172,981 | 186,396 | +7.8 | 58,906 | 50,415 | -14.4 | 114,075 | 135,981 | +19.2 | | | |
| 5 | Motor vehicle the | 71,871 | 53,145 | -26.1 | 25,799 | 12,910 | -50.0 | 46,072 | 40,235 | -12.7 | | | |
| 6 | | | | | | | | | | | | | |
| 7 | | | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | |
| 9 | | | | | | | | | | | | | |
| 10 | | | | | | | | | | | | | |

**Figure 5.7: Spreadsheet Access to User B (Requestor)**

The complete workflow of the implemented prototype is depicted in figure 5.8.

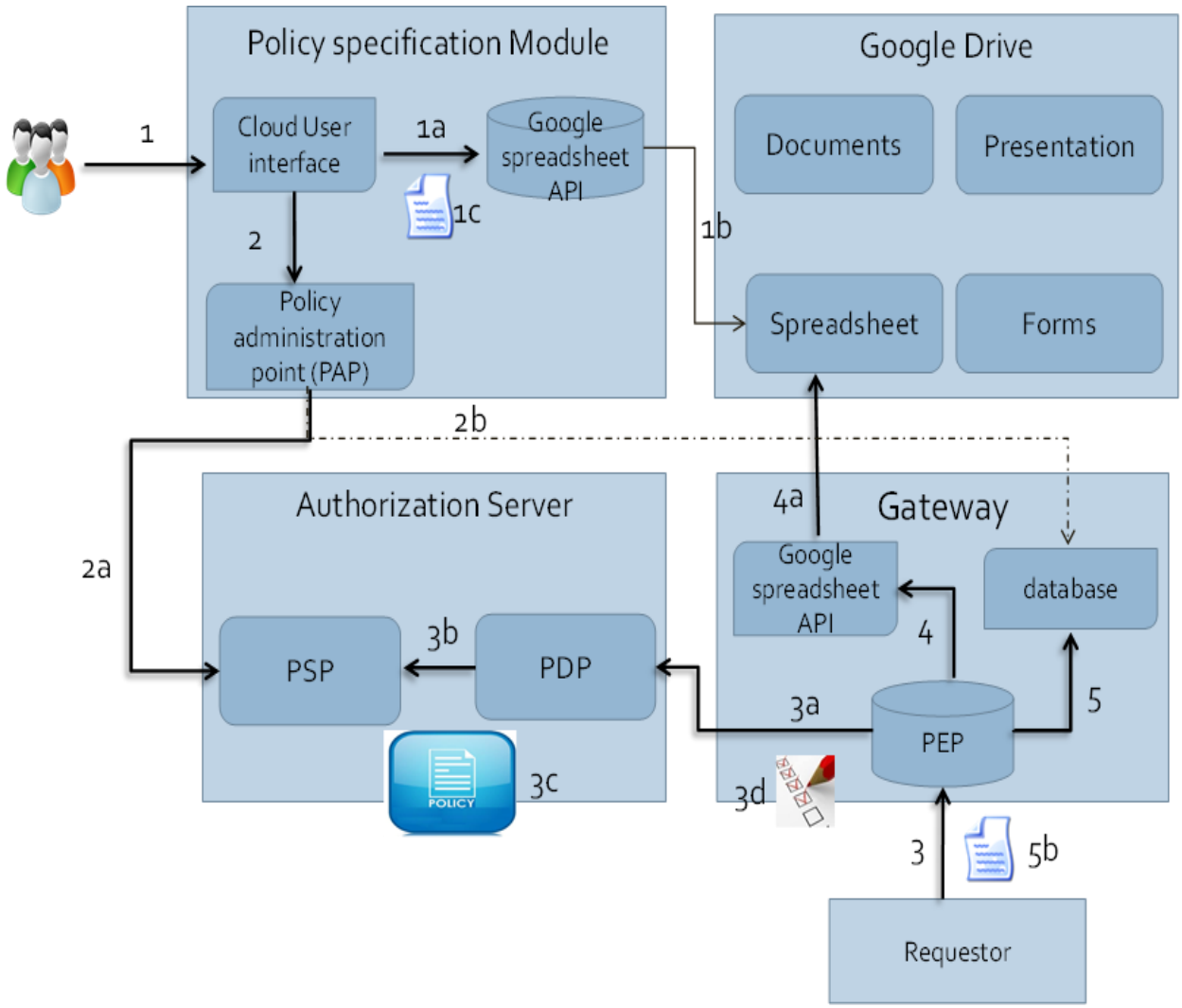



**Figure 5.8: Workflow of the implemented prototype**

## 5.3 Development Toolkit

The above discussed user centric access control system and underlying protocol has been developed using the following main tools and technologies

- Microsoft Windows Operating System
- Eclipse (Juno) J2EE

- Google spreadsheet API (Java) 3.0
- MySQL Server version 5.1.72
- JDK version 7.0
- SUN XACML API 2.0

## *Summary*

*This chapter explains the implementation of the prototype of the designed system. The explanation of the prototype is done through the use of use case scenarios. The implemented prototype provides user with the functionality to set access control on google spreadsheets according to their requirements. It enables user to specify fine grained access control e.g. access control on individual records of the spreadsheet which Google does not provide. The implemented prototype shows the reliability and integration of the designed framework with existing Cloud applications. The verification and validation of the designed framework is described in the next chapter.*

# Chapter 6

# Evaluation of Research Work

## 6.1 Evaluation Methodology

There are different types of evaluation methods both for a complete system and for a prototype of the designed system. Some of them are based on formal methods to evaluate the prototype of a system can be found in literature. On the other hand, evaluation of a complete system can be done by specifying a set of characteristic and features (according to some standard) that should be analyzed to see compliance of the developed system features with the given standards [27]. There are also other methods such as quantitative and qualitative evaluation methods which are used to measure the system from different perspectives e.g. performance, effectiveness, security, etc. We evaluated the designed system from security and functional perspective.

## 6.2 Validation based on NIST Security Criteria

NIST document on Directions in Security Metrics Research [28] provides useful practical guidelines to define security metrics to measure the security properties of a system. It defines various methods and criteria which can help designers to measure and analyze the security aspects of their system. In addition to this, several documents on evaluation metrics including [29], [30] and assessment of access control systems [31] describe recommended security control for information systems and define guidelines for evaluation. A metric is defined as a system of measurement to analyze system parameters based on quantifiable measures. For information system's security, the measures take into account the features of the system that add to its security. Security metrics in such cases consider measurement of few entities of the system which possess a security property, assessed to get a measured value. Security measures for an organization should include the fulfillment of organizational objectives and its security needs. The well-established security aspects which are considered to evaluate a system and its security properties are listed below:

- Qualitative and Quantitative Properties
- Organizational Security Objectives
- Leading versus Lagging Indicators
- Measurement of Large versus the Small
- Correctness and Effectiveness

We follow the Quantitative and Qualitative Properties aspect to evaluate the security properties of our system and Correctness and effectiveness aspects to evaluate its functional properties.

## 6.2.1 Qualitative and Quantitative Properties

Qualitative properties are based on the characteristics or measures of software including complexity, flexibility, portability, reliability, etc. However, quantitative properties are based on the measurement based on well-defined figures and numbers [32]. Security properties are complex to be evaluated based on strict qualitative or quantitative properties. NIST Direction in Security Metrics defines that qualitative assignments are sometimes used to measure the quantitative properties such as vulnerabilities found in software are measured through low, medium and high scale [29]. We use the qualitative and quantitative properties to evaluate the design of our system. In this regard, we identify the security threats related to User centric access control framework and subsequently highlight the mechanisms which have been incorporated in the framework to prevent fulfillment of malicious goals.

### 6.2.1.1 Threat versus Security Mechanism

The table 6.1 describes the threat model evaluation which we have carried out to measure the security features of the designed framework. Different threats pertinent to Access control framework have been identified and various protection mechanisms have been incorporated within the designed system to mitigate these threats.

**Table 6.1: Threat vs. Security Mechanisms for Access Control**

| Threat | Protection Granted | Protection Mechanism |
|---|---|---|
| Unauthorized access | Yes | User centric access control mechanism protects resources according to user's requirements and ensures granular level authorized access on resources. |
| Privacy leakage | Yes | Policy specification module allow users to specify access control on their personal information in the form of profiles on blog and social networking websites according to their requirements protecting personal information leaking to unauthorized personnel. |
| Denial of Service | Yes | Requests need to be validated by authorization server to get application and resource access. |
| Weak Authentication | Yes | FIPS-196 Strong Authentication Protocol is used for authentication which mitigates attacks on passwords. |
| Bypass | Yes | Designed system does not allow request to bypass PEP or PDP to access resources. |

The first threat which is identified is unauthorized access to resources hosted on different Cloud applications. Unauthorized access can then further lead to data theft identity theft, fraud and different other security threats. The designed system minimizes these threats by enabling users to protect their resources according to their requirements by using the Policy specification module. Users can specify granular level access control policies, thus mitigating the risk of unauthorized access to their resources. Similarly, Users can set access control policies on their personal information in order to minimize personal information leaking to illegitimate personals. Other security threats like Denial of service and Bypass are mitigated by providing Gateway and Authorization server components in the designed framework. These components are responsible for verifying and evaluating different requests and protect resources from different security threats.

## 6.2.2 Correctness and Effectiveness Properties

Correctness property is used to ensure the proper implementation of the designed system. On the other hand, effectiveness property is used to measures that how well the different components of the system work together [33]. We use the correctness and effectiveness properties to evaluate the working of the implemented prototype of the designed system. Correctness and effectiveness assessments are basically performed theoretically on the basis of reasoning instead of actual

evaluation of system components (both software and hardware). Similarly, we have designed and performed different test-cases to confirm the working of implemented features and highlight the mechanisms which have been incorporated in User centric access control framework to achieve those features.

### 6.2.2.1 JUnit Testing:

To verify the claims of user centric access control framework that it provides user centric access control features and protect resources according to user's requirements we have developed a prototype as explained in chapter 5. In this regard, we define our functional objectives which we validate based on different test cases through JUnit. In order to examine and test the correctness of features introduced in User centric access control framework, we formulated different categories of test cases as shown in table 6.2.

**Table 6.2: Test Cases**

| Category | Test Cases | No. of Test Cases Planed | No. of Test Cases Executed | No. of Test Cases Executed Successfully | No. of Defects Found |
|---|---|---|---|---|---|
| User-Centric Policy Creation Test Cases | Policy Creation Test | 20 | 20 | 20 | 0 |
| | Policy Upload Test | 20 | 20 | 20 | 0 |
| Authorization Test Cases | Authorization Test-Accept | 20 | 20 | 20 | 0 |
| | Authorization Test-Deny | 20 | 20 | 20 | 0 |
| Granular level Access Control Test Cases | Row Information Storage Test | 20 | 20 | 20 | 0 |
| | Row Information Retrieval Test | 20 | 20 | 20 | 0 |
| Total | | 120 | 120 | 120 | 0 |

Table 6.2 describes the different categories of test cases performed to check the correctness of features of the designed framework. Each category contains two different types of test cases, each of which is performed 20 times by using different values. Parameterized test cases were formulated in JUnit to run each test with different parameters (values) 20 times. The first category of test cases include Policy file creation and Policy file upload test cases as shown in table 6.3 and 6.4.

**Table 6.3: Policy File Creation Test**

| | |
|---|---|
| **Test Case Title** | User Centric Access Control Policy Creation Test |
| **Test Case ID** | Test-01 |
| **Test Case Objective** | To check the correctness of XACML policy file creation process |
| **Pre-Condition** | Policy specification module should be in running state |
| **Post Condition** | Policy file written in XACML should be created |
| **Procedure** | 1. Specify subject, resource, action and effect in the policy specification module.<br>2. Click the Build policy button to create the policy file. |
| **Expected Result** | XACML policy file created in a specified location |
| **Actual Result** | XACML policy file was created |
| **Status** | Pass |
| **Carried out On** | 5-3-2014 |
| **Carried out By** | Misbah Irum |

Table 6.3 depicts a Policy file creation test case. This test case is formulated and performed to check the policy file creation on the specified location. The table describes the pre and post conditions necessary for the execution of this test case. If the expected and actual results match then the status of this test case is Pass which verifies the policy file creation feature of the designed system.

**Table 6.4: Policy File Upload Test**

| | |
|---|---|
| **Test Case Title** | User Centric Access Control Policy Creation Test |
| **Test Case ID** | Test-01 |
| **Test Case Objective** | To check the correctness of XACML policy file upload process |
| **Pre-Condition** | Policy specification module should be in running state |
| **Post Condition** | Policy file should be uploaded on Authorization server |
| **Procedure** | 3. Specify subject, resource, action and effect in the policy specification module.<br>4. Click the Build policy button to create the policy file.<br>5. Click the upload policy button to upload file. |
| **Expected Result** | XACML policy file uploaded on a specified location |
| **Actual Result** | XACML policy file was uploaded |
| **Status** | Pass |
| **Carried out On** | 5-3-2014 |
| **Carried out By** | Misbah Irum |

Policy File upload test case in table 6.4 is formulated to check the correction of Policy upload function of the designed system. After creating the policy file, user needs to upload this file onto the authorization server. This test case is performed to check whether the policy file is uploaded on the specified location on the authorization server or not. The result of the test case verifies the correct functioning of the policy upload feature of the designed system.

The second category of test cases checks the correctness of features provided by the authorization server. Authorization server is responsible for generating decisions by evaluation access control policies. In this category of test cases the correct generation of access control decisions is checked.

**Table 6.5: Authorization Test-Accept**

| | |
|---|---|
| **Test Case Title** | Authorization Test-Accept |
| **Test Case ID** | Test-02 |
| **Test Case Objective** | To check the correctness of authorization process |
| **Pre-Condition** | Policy must be created to enforce authorization and grants requestor access to the resource |
| **Post Condition** | Requestor should be granted access to the requested resource |
| **Procedure** | 1. Requestor sends an access request to the gateway.<br>2. Gateway generates an XACML request and forwards it to the authorization server.<br>3. Authorization server evaluates the policy and generates a decision as "Permit" and sends it to the gateway.<br>4. Gateway provides requestor with the access to the requested resource. |
| **Expected Result** | Access Granted |
| **Actual Result** | Access Granted |
| **Status** | Pass |
| **Carried out On** | 5-3-2014 |
| **Carried out By** | Misbah Irum |

Table 6.5 depicts the authorization test case in which the decision generated by the authorization server is "Permit" if the access control policy grants access to the requestor. This test case was executed 20 times using different policies which permits particular requestors access to their resources. Authorization server correctly evaluates these policies and generates correct decisions every time.

**Table 6.6: Authorization Test-Deny**

| | |
|---|---|
| **Test Case Title** | Authorization Test-Deny |
| **Test Case ID** | Test-03 |
| **Test Case Objective** | To check the correctness of authorization process |
| **Pre-Condition** | Policy must be created to enforce authorization and denies requestor access to the resource |
| **Post Condition** | Requestor should not be given access to the requested resource |
| **Procedure** | 1. Requestor sends an access request to the gateway.<br>2. Gateway generates an XACML request and forwards it to the authorization server.<br>3. Authorization server evaluates the policy and generates a decision as "Deny" and sends it to the gateway.<br>4. Gateway denies requestor with the access to the requested resource. |
| **Expected Result** | Access Denied |
| **Actual Result** | Access Denied |
| **Status** | Pass |
| **Carried out On** | 5-3-2014 |
| **Carried out By** | Misbah Irum |

The third category of test cases checks the fine grained access control features of the designed system. The first test case as shown in table 6.7 is formulated and performed to check correct fine grained access control policy file creation at a specified location.

**Table 6.7: Fine Grained Policy Creation Test**

| | |
|---|---|
| **Test Case Title** | User Centric Access Control Policy Creation Test (Fine Grained Policy) |
| **Test Case ID** | Test-04 |
| **Test Case Objective** | To check the correctness of XACML policy creation process |
| **Pre-Condition** | Policy specification module should be in running state |
| **Post Condition** | Policy file written in XACML should be created |
| **Procedure** | 1. Specify subject, resource, action and effect in the policy specification module.<br>2. Click row level access control button and specific row numbers that the subject can access.<br>3. Click the Build policy button to create the policy file. |
| **Expected Result** | XACML policy file created in a specified location |
| **Actual Result** | XACML policy file was created |
| **Status** | Pass |
| **Carried out On** | 5-3-2014 |
| **Carried out By** | Misbah Irum |

**Table 6.8: Fine Grained Authorization Test-Accept**

| | |
|---|---|
| **Test Case Title** | Fine Grained Authorization Test-Accept |
| **Test Case ID** | Test-05 |
| **Test Case Objective** | To check the correctness of authorization process |
| **Pre-Condition** | Policy must be created to enforce authorization and grants requestor access to the resource |
| **Post Condition** | Requestor should be granted access to the requested resource |
| **Procedure** | 1. Requestor sends an access request to the gateway.<br>2. Gateway generates an XACML request and forwards it to the authorization server.<br>3. Authorization server evaluates the policy and generates a decision as "Permit" and sends it to the gateway.<br>4. Gateway fetches row number from the database and parses the resource (spreadsheet) accordingly.<br>5. Gateway provides requestor with the access to specific rows of the requested resource. |
| **Expected Result** | Access Granted only to specific rows |
| **Actual Result** | Access Granted only to specific rows |
| **Status** | Pass |
| **Carried out On** | 5-3-2014 |
| **Carried out By** | Misbah Irum |

Table 6.8 depicts the fine grained authorization test case to check the correct generation of fine grained decision by the authorization server. This test case validates the granular level access control features of the designed system. User specifies certain row numbers of the spreadsheet that are accessible to a particular user. This test case checks whether the requestor is given access to the correct specified rows or not. The test case was performed by using different policies allowing access to different row numbers to different users. The execution of the test case results in positive at each iteration thus validating the claims of the designed User Centric access control framework in providing fine grained access control to resources in Cloud environment.

## *Summary*

*We have evaluated the User centric access control policy management framework from both security and functional perspective. We identified the research problems, and followed a comprehensive research methodology to fulfill the objectives. The design of the functionalities is then evaluated based on the security metrics defined by NIST. In this regard, we identify one main security metric and validate on the basis of identified threats on Access control models and the mechanisms which prevent the threats. Finally, a prototype implementation is developed to test the proposed functionalities and test cases are used to prove that desired goals are achieved.*

# Chapter 7

# Conclusion and Future Directions

## 7.1 Conclusion

This research focuses on access control from a different perspective and provides a User centric access control policy management framework as a solution for various traditional access control problems faced by numerous users in the Cloud environment. The designed framework provides users with full control over their resources in order to protect them against different security threats. The framework enables users to specify access control regarding their resources according to their access control and security requirements. Moreover, the designed framework also eliminates the need of employing different access control mechanisms and diverse and complex access control policy definition languages offered by various Cloud applications. Instead of this, it facilitates users by providing them with a single authorization and access control mechanism and a standard policy definition language to specify access control on various resources scattered across the Cloud. Users can define, edit and manage all the applied access control from a unified control point on all its resources independent of their location on the Cloud. A prototype of the designed framework was developed which enable users to specify access control on their Google spreadsheet according to their requirements. It facilitates them to specify record level access control to share only specific rows of a spreadsheet with other users.

To validate the features of the designed system, we evaluated the prototype from security and functional perspectives. To verify the security features of the designed system, a threat model is formulated which identify different security and access control threats and explain the protection mechanisms incorporated within the designed system to eliminate these threats. To check the correctness of the features provided by the system, various categories of test cases are formulated and performed through JUnit. Successful execution of test cases verifies the claims of the user centric access control framework in providing users with the control to define access control policies according to their requirements.

## 7.2 Future Research Directions

User centric access control comes along with a number of potential initiatives which require substantial research work and development by the research community.

- In the designed framework we use Attribute Based Access Control (ABAC) model to protect resources. Incorporation of different access control models like Content Based Access Control (CBAC) and Role Base Access Control (RBAC) models should be done to provide users with more fine grained options to protect resources according to their requirements.

- Another future direction for User centric access control is to involve user in decision making process in real time. There are multiple scenarios where data or resources are of critical importance and real time user consent is necessary for the protection of resources from unauthorized access. This can be achieved by incorporating user within the decision making process and the request to be sent to the user via email or message before any decision generated by authorization server.

- Protection of the communication messages between different components of the designed framework can be another future direction in this domain. Confidentiality of XACML request and response will secure the framework and protect it against interception and alteration threats.

- Access control policies are stored on the authorization server. Protection of authorization server and access control policies is necessary for the security of the designed framework. Encryption of access control polices can be done and different security mechanism like Intruder detection system (IDPS) and firewalls should be deployed onto the authorization server.

# Bibliography

1. P. Mell, T. Grance, "*The NIST Definition of Cloud Computing. NIST Special Publication 800- 145 (Draft)*", Retrieved February 12, 2014, from http://csrc.nist.gov/publications/drafts /800-145/Draft-SP- 800-145 cloud-definition.pdf

2. Wayne Jansen, Timothy Grance, "*Guidelines on Security and Privacy in Public Cloud Computing*" NIST, NIST Special Publication 800-144, December 2011.

3. D. Catteddu and G. Hogben, "*Cloud Computing: Benefits, risks and recommendations for information security*", ENISA2009, http://www.enisa.europa.eu/activities/riskmanagement /files/deliverables/Cloud-computing-risk-assessment/at download/fullReport.

4. Todd Steiner, Hamed Khiabani, "*An IntroductionTo Securing a Cloud Environment*", SANS institute 2012.

5. B. R. Kandukuri, R. Paturi, A. Rakshit "*Cloud Security Issues*", published in the proceeding of 6th IEEE International Conference on Services Computing, pp. 517-520, Bangalore, India, 2009.

6. H. Takabi, J. B. D. Joshi, and G. J. Ahn, "*Security and Privacy Challenges in Cloud Computing Environments*", published in the proceeding of IEEE International Conference on Security and Privacy, Vol. 8, No. 6, pp. 25-31, 2010.

7. F. Xu, J. He, X. Wu and J. Xu, "*Privacy-Enhanced Access Control Model*", published in the proceeding of International Conference on Networks Security, Wireless Communications and Trusted Computing (NSWCTC 2009), Wuhan, China, April 25-26, 2009.

8. H. Takabi, J. B. D. Joshi, and G. J. Ahn, "*SecureCloud: Towards a Comprehensive Security Framework for Cloud Computing Environments*", published in the proceeding of 1st IEEE International Workshop Emerging Applications for Cloud Computing (CloudApp 2010), pp. 393-398, Seoul, South Korea, 2010.

9. El Maliki, T; Seigneur, J, "*A Survey of User-centric Identity Management Technologies*" published in the proceeding of The International Conference on Emerging Security Information, Systems, and Technologies, pp.12,17, 14-20 Oct. 2007

10. Machulak, M.P., van Moorsel, A., "*Architecture and Protocol for User-Controlled Access Management in Web 2.0 Applications*", published in the proceeding of 30th International Conference on Distributed Computing Systems Workshops (ICDCSW), pp.62-71, 21-25 June 2010.

11. Singh, K. "*xAccess: A unified user-centric access control framework for web applications*", published in the proceeding of Network Operations and Management Symposium (NOMS), pp.530-533, 16-20 April 2012.

12. Takabi, H., Joshi, J.B.D., "*Policy Management as a Service: An Approach to Manage Policy Heterogeneity in Cloud Computing Environment*". published in the proceeding of 45th Hawaii International Conference on System Science (HICSS), pp.5500-5508, 4-7 Jan, 2012.

13. E. Hammer-Lahav, D. Recordon, D. Hardt; "*The OAuth 2.0 Authorization Protocol draft-ietf-oauth-v2-12*"; January 21, 2011; http://tools.ietf.org/pdf/draft-ietf-oauth-v2-12.pdf

14. Fei Xu; Jingsha He; Xu Wu; Jing Xu, "*A User-Centric Privacy Access Control Model*", published in the proceeding of 2nd International Symposium on Information Engineering and Electronic Commerce (IEEC), pp.1,4, 23-25 July 2010

15. Ei Ei Mon; Thinn Thu Naing, "*The privacy-aware access control system using attribute-and role-based access control in private Cloud*", published in the proceeding of 4th IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT),pp.447,451, 28-30 Oct. 2011

16. Zhu Tianyi; Liu Weidong; Song Jiaxing, "*An Efficient Role Based Access Control System for Cloud Computing*" , published in the proceeding of IEEE 11th International Conference on Computer and Information Technology (CIT), pp.97,102, Aug. 31 2011-Sept. 2 2011.

17. Wenhui Wang; Jing Han; Meina Song; Xiaohui Wang, "*The design of a trust and role based access control model in Cloud computing*", published in the proceeding of 6th International Conference on Pervasive Computing and Applications (ICPCA), pp.330,334, 26-28 Oct. 2011

18. Ting-Kuang Wu; Yung-Wang Lin; Iuon-Chang Lin, "*A Cloud-User Access Control Mechanism Based on Data Masking*", published in the proceeding of Sixth International Conference on Genetic and Evolutionary Computing (ICGEC), pp.165,168, 25-28 Aug. 2012

19. Miao Zhou; Yi Mu; Susilo, W.; Man Ho Au; Jun Yan, "*Privacy-Preserved Access Control for Cloud Computing*", published in the proceeding of IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp.83,90, 16-18 Nov. 2011

20. Li, J., Zhao, G., Chen, X., Xie, D., Rong, C., Li, W., Tang, L., and Tang, Y, "*Fine-grained data access control systems with user accountability in cloud computing*", presented in Second International Conference on Cloud Computing Technology and Science (CloudCom), 2010 IEEE, pages 8996.

21. Anamika, "*Research methodology: An introduction*", www.newagepublisher.com /samplechapter/000896.pd, 2010, accessed: 2013-10-15.

22. W. C. Booth, G. G. Colomb, and J. M. Williams, "*The craft of research*", University of Chicago Press, 2003.

23. C. Kothari, "*Research methodology: methods and techniques*", New Age International, 2004.

24. A. Hevner and S. Chatterjee, "*Design Research in Information Systems, Integrated Series in Information Systems*", 2010, ch. Design Science Research in Information Systems.

25. Zhang, N., Q. Shi, and M. Merabti., "*Anonymous public-key certificates for anonymous and fair document exchange*", published in IEE Proceedings-Communications 147, vol.6 pp.345-350.

26. S. Proctor, "*Sun's xacml implementation-programmer's guide for version 1.2*," 2004.

27. Ross, R., Katzke, S., Johnson, A., Swanson, M., Stoneburner, G., Rogers, G., and Lee, A, "*Recommended security controls for federal information systems*", NIST Special Publication.

28. W. Jansen, "*Directions in security metrics research*", DIANE Publishing, 2010.

29. K. S. Vincent Hu, "*Guidelines for access control system evaluation metrics*", National Institute of Standards and Technology, Tech. Rep., 2012, http://csrc.nist.gov/publications /nistir/ir7874/nistir7874.pdf.

30. R. Ross, S. Katzke, A. Johnson, M. Swanson, G. Stoneburner, G. Rogers, and A. Lee, "*Recommended security controls for federal information systems*," NIST Special Publication, vol. 800, p. 53, 2005.

31. V. C. Hu, D. Ferraiolo, and D. R. Kuhn, "*Assessment of access control systems*", US Department of Commerce, National Institute of Standards and Technology, 2006.

32. B. Johnson and L. Christensen, "*Educational research: Quantitative, qualitative, and mixed approaches*", SAGE Publications, Incorporated, 2007.

33. W. Jansen, "*Directions in security metrics research*", http://csrc.nist.gov/publications /nistir/ir7564/nistir-7564 metrics-research.pdf, April 2009, [Online; accessed Jan-2014].

34. http://openid.net/get-an-openid/what-is-openid/ [Online; accessed Oct-2013].

35. http://www.jisc.ac.uk/whatwedo/themes/accessmanagement/federation/shibboleth.aspx [Online; accessed Oct-2013].