

Ensuring Fine-Grained Authorized Access Control for Healthcare Applications on Cloud Provisioned Platform

Phyu Hnin Thike¹, and Nyein Nyein Oo²

Department of Information Technology Engineering, Yangon Technological University, Myanmar
¹msphyuhninthike@gmail.com, ²nno2005@gmail.com

Abstract: *In a cloud provisioned medical healthcare system, it is very important to ensure that resources and sensitive information (such as prescriptions, medical records and lab test data, etc) are accessed by and provided to authorized users and tenants whenever needed to be accessed. By using Attribute-Based Access Control (ABAC), not only role of the user but also other attributes, such as which resources he is allowed to access, what actions he is allowed to perform on those resources, what time of the date, whether he is at office or not and so on, are used for the fine-grained access control decision. Global policies are also needed to access and protect patient information from unauthorized access and to provide privacy. This ensures who can access what, under what conditions, and for what purpose. In the proposed system, XACML 3.0 standard is implemented to develop a prototype application. Through cloud computing platform (WSO2 Servers), the system can support the patients and healthcare workers to access the digital health records in hospitals easily by ensuring the patient's privacy and access to information.*

Keywords: XACML, fine-grained authorization, Policy-Based Access Control, Attribute-Based Access Control, WSO2

1. Introduction

Access control is concerned with the protection of system resources against unauthorized access. In particular, it defines a process by which the use of system resources (1) is regulated according to an access control policy; and (2) is exclusively permitted by authorized entities (users, programs, processes, or other systems) according to that policy. Role-Based Access Control (RBAC) models are receiving increasing attention as a recent generalized approach to access control. It differs from traditional identity based access control in that it takes advantage of the concept of role relations. The advantages of the concept of roles are several. It simplifies authorization administration because a security administrator needs only to revoke and assign the new appropriate role memberships if a user changes its job function [1]. Traditional RBAC is not able to specify a sufficiently fine-grained authorization policy or constraints that should be applied to an access control policy. It's not flexible with complicated access control requirements inherent to current healthcare systems.

In modern healthcare domain, a patient's EHRs can be found scattered throughout the entire healthcare sector. From the clinical perspective, in order to deliver quality patient care, it is critical to access the integrated patient care information that is often collected at the point of care to ensure the freshness of time-sensitive data. This further requires an efficient, secure and low-cost mechanism for sharing EHRs among multiple healthcare providers [2]. Health care applications requiring access control and the environments in which they operate in become more complex, an acute need for better ways to manage access control rules has arisen.

The new way for doing access control is to go for Attribute-Based Access Control (ABAC). In its most basic form, ABAC relies upon the evaluation of attributes of the subject, attributes of the object, environment conditions, and a formal relationship or access control rule defining the allowable operations for subject-object

attribute and environment condition combinations. As new subjects join the organization, no modifications to existing rules or object attributes are required as long as the subject is assigned the attributes necessary for access to the required objects. This benefit is often referred to as accommodating the external user and is one of the primary benefits of employing ABAC.

Attribute-Based Access Control (ABAC) employs multiple attributes for authorization decision, which enables the security system to be flexible, interoperable, and multifunctional. ABAC recommended access control model for promoting information sharing between diverse and disparate organizations. In 2014, Gartner Identity & Access Management Summit predicts that by 2020, 70% of all businesses will use ABAC as the dominant mechanism to protect critical assets, up from 5% today.

The proposed system is intended to implement Policy-Based and Attribute-Based access control for healthcare data in cloud computing platform using XACML 3.0 standard. As a result of storing medical record electronically, healthcare providers can eventually shift their electronic medical record (EMR) systems into clouds instead of building and maintaining dedicated data centers. By using cloud computing concept, it can increase both the efficiency of medical data management and sharing process since patients' healthcare-related data should always be accessible from anywhere at any time. In Section 2 of this paper, related work of the system is described and brief description of XACML is presented in Section 3. WSO2 servers are explained in Section 4. The architecture and implementation of the proposed prototype system is expressed in Section 5. In Section 6 and 7, conclusion and acknowledgements are described. Finally, references for this work will be listed in Section 8.

2. Related Work

It is thus essential that new access control policies and mechanisms are devised for federated Electronic Health Record systems, to ensure not only that sensitive patient data is accessible by authorized personnel only, but also that it is available when needed in life-critical situations. [3] showed how the required level of data security can be achieved through a judicious combination of three mechanisms, namely Discretionary Access Control(DAC), Mandatory Access Control (MAC) and Role-Based Access Control(RBAC).

In access control, the main concern is privacy, where access should only be granted to the information required by an actor in any situation. Clinicians may well disagree with this from the viewpoint that it is better to have broad access. In [4], an approach was suggested to access control that combines guidelines and learning from observations and logs.

Security is currently one of the main concerns and several initiatives are currently ongoing aimed at achieving a standardized way for supporting integrity, confidentiality, and access control for XML web services. [5] defined requirements for access control of designing medical Web services, and provide security for collaborative business process using web services in an open environment using WS-Policy measure for transport layer security.

In [6], Policy-Based Access Control model (PBAC) is discussed in details. In PBAC model, a resource is governed by a document that exactly specifies what subject credentials and requirements must be fulfilled in order to obtain access. PBAC is by now the de-facto standard model for enforcing access control policies in service-oriented architectures. A widely used implementation of PBAC and ABAC is given by the eXtensible Access Control Markup Language (XACML). It defines a language for the definition of policies and access requests, and a workflow to achieve policy enforcement. XACML is currently used as a basis for enforcing access control in many large scale projects.

With the development of large distributed systems, Attribute-Based access control (ABAC) has become increasingly important. XACML is a defacto standard for Policy-Based and Attribute-Based access control. XACML is "cloud ready", and solves significant problems in the cloud and its architecture supports externalization of Authorization. This work provides effective fine-grained authorization for Policy-Based and Attribute-Based access control in cloud provisioned healthcare systems by using XACML 3.0 standard. Since it

can provide flexible access decisions for healthcare data storage, healthcare workers and patients are enable to create, manage and access patients' healthcare information from anywhere at any time.

3. XACML

Early experiences using XACML in distributed systems have proven positive. The language is indeed useful for specifying arbitrarily complex policies in a wide variety of (distributed) applications and environments. While targeted at traditional access control systems, XACML also proves practical for expressing privilege management policies and defining privilege statements. The standard format works well in tying together heterogeneous systems, and already fosters development of common tools. Its open standard status, definition in XML, and availability of open source projects has already drawn support from diverse applications. XACML's ability to tie into other authorization systems makes it a natural inter-operability point, even for legacy systems. Its expressive semantics and extensible nature also make it useful as an intermediary language.

For the proposed system, XACML 3.0 standard is used to ensure privacy and access to healthcare information. Many applications, especially web applications, need access control policies to authorize for various resources. The eXtensible Access Control Markup Language (XACML) is the widely used standard for specifying access control policies. This is a set of standards maintained by the Organization for the Advancement of Structured Information Standards (OASIS). It is not only a declarative access control policy language implemented in XML but also a processing model describing how to evaluate authorization requests according to the rules defined in policies. In many large scale projects, it is used as a methodology for enforcing access control.

XACML version 3.0 was recently introduced. XACML 3.0 also allows a requester to ask several queries at once to which the PDP responds with a single answer with multiple decisions. This reduces the amount of bandwidth required in web-based scenarios and can serve to reduce the communication overhead. Finally, XACML 3.0 also introduces new attribute functions and data types, including XPath, as well as new policy combination algorithms. These are designed to improve the entire policy by allowing for smoother, faster, and more reliable XACML processing. It provides a flexible and mechanism independent representation of access rules that vary in granularity; It allows the combination of different authoritative domains' policies into one policy set for making access control decisions in a widely distributed system environment.

XACML is primarily an Attribute Based Access Control System (ABAC), where attributes associated with a user or action or resource are inputs into the decision of whether a given user may access a given resource in a particular way. Role Based Access Control (RBAC) can also be implemented in XACML as a specialization of ABAC. This language is very expressive and can be used to define a lot of different kind of policies. The choice of this language was due to the completeness of its expressivity for access control rules. Many AC models (like RBAC, ABAC, etc.) can be defined with this language. It has a good flexibility for defining rule conditions. Due to the diversity of AC models used in different cloud platforms, XACML is opted as a suitable standard to express AC rules.

4. WSO2

WSO2 is an open source application development software company focused on providing service-oriented architecture (SOA) solutions for professional developers. WSO2 is a key contributor to Apache web services projects including Apache Axis2, Apache Rampart, Apache Synapse, Apache Axiom and more. WSO2 projects are free and open source released under Apache License Version 2.

WSO2's platform is the only comprehensive middleware platform that is 100% open source with no gimmicks. This gives unlimited opportunity to explore and experiment with our platform with no licensing costs. Most importantly, it can be downloaded and played with the same version that runs in production at no cost. WSO2 not only uses an open-source license, but also follows an open development process, which customers can observe and provide input into. The WSO2 platform is made up of over 25 products covering all major

categories from integration and API management to identity and mobile. All products are built using a single code base; therefore, they work seamlessly with each other so less engineering resources are needed when integrating them. It can support for private, public, and hybrid clouds that the same application can run on-premise and on the cloud. As a result, the WSO2 platform is future proof as our solution don't need to be re-architected when it is moved to the cloud. WSO2's business provides 24*7 support with a guaranteed response time of 1 hr by the engineers that built the product and know the code inside out. WSO2 software runs billions of enterprise_critical transactions per year. It is used by hundreds of major companies throughout the world, including Boeing, eBay and Cisco.

4.1. WSO2 Identity Server

The **WSO2 Identity Server** is an open source identity and entitlement management server having support for Information Cards, OpenID and XACML. In XACML, core logic related to policy evaluation resides in a software component called "XACML Engine". WSO2 Identity Server (WSO2 IS) has a XACML engine embedded and supports Policy-Based access control with XACML 2.0 and 3.0. The XACML engine of the Identity Server acts as a PAP, PDP and a PIP. The entitlement service of Identity Server can be enforced via PEP. Using the WSO2 IS as the PDP, the authorization of requests that are coming into a particular web application can be checked.

4.2. WSO2 Application Server

WSO2 Application Server (WSO2 AS) is an enterprise ready Web services engine powered by Apache Axis2. It is a lightweight, high performing platform for Service Oriented Architectures, enabling business logic and applications. Bringing together a number of Apache Web services projects, WSO2 AS provides a secure, transactional and reliable runtime for deploying and managing Web services.

It is Cloud Native, providing a firm foundation for hosting shared, multi-tenant, elastically scaling SaaS applications. It brings together best of breed open source technologies for Web Applications (i.e. Apache Tomcat), Web Services (i.e. Apache Axis2), RESTful services (ie: JAX-RS) with WSO2's open source management, monitoring, clustering, and logging extensions.

When entitlement is provided, there has to be a point where the requests are intercepted and checked for authorization. That particular point is named as a **PEP**. To provide entitlement for proposed healthcare web application requests, the WSO2 enterprise middleware platform uses a **Servlet Filter** as the PEP. This **Entitlement Servlet Filter** feature is available in WSO2 Application Server 5.0.1 onwards.

In this work, WSO2 Identity Server is used as as the Policy Decision Point (PDP) and also as XACML engine to check the authorization of requests for healthcare data that are coming into the proposed healthcare web application and WSO2 Application Server is used to host the proposed healthcare web application.

5. Proposed System Architecture and Implementation

The architecture of the proposed system is shown in Fig: 1 and Policy-Based and Attribute-Based access control has been implemented in the proposed system. WSO2 Application Server is used as a web application container to host the proposed web application. WSO2 Identity Server is used as the XACML Policy Decision Point (PDP) to store Attribute-Based policies for ensuring fine-grained authorization. In the proposed system, four types of users such as physician, administrator, patient and staff are defined with their respective access permissions. The proposed web application provides functions for users to explore the system based on their particular roles. Also, the system is built upon ABAC model (as a specialization of RBAC) by exposing different views of the healthcare database corresponding to the authenticated user roles. System administrators for each tenant manages authentication and authorization for their own domain by defining users, roles, permissions, resources, constraints and policies for the authorization. The provider creates system administrator role for each tenant and assigns access rights to them. Access control policies, as shown in TABLE I, that are based on subject, resource, action and environment attributes are created and stored in WSO2 Identity Server. The access control

decision is designed to depend on the attributes of the requestor, such as subject, resource, action and environment for ensuring the right access without permission misuse.

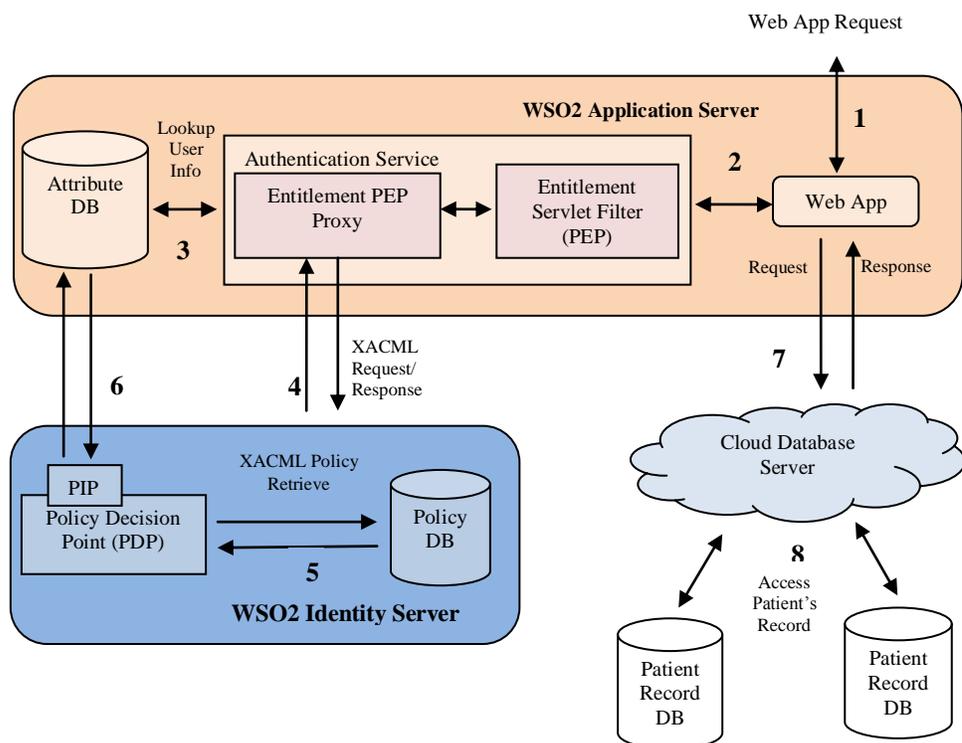


Fig. 1: Proposed System Architecture

Because RBAC can also be implemented as a specialization of ABAC in XACML standard, users are firstly classified into different roles such as administrator, physician, patient and staff. When a registered user sends a request to retrieve or insert or modify the patient’s record, it comes to a particular web application which has the engaged Entitlement Servlet Filter that is used as PEP. The Policy Enforcement Point (PEP) looks up the attribute database and validates the sender. It obtains the parameters `UserName`, `ResourceName`, `Action` and `Environment` as attributes from the request to create an XACML request. After that, it sends that request to Policy Decision Point (PDP) and waits for the Entitlement decision. The PDP finds some policies that can be applied to the request from policy DB and evaluates the decision whether the access request should be granted or denied based on Attribute-Based access control. The Policy Information Point (PIP) obtains the requested attributes from attribute database if required for evaluation.

Depending on the decision, it permits or denies the request which has come to the web application that is hosted in WSO2 AS. If the decision is Permit, then the permission is given to Cloud Database Server to access or modify the database with appropriate information. The Response is then sent with the proper response value returned from Cloud Database Server to the user through the web application. If the decision is Deny, then the appropriate error message is sent through the web application to the user. And then, the user can try again to create an access request with correct access right that was already defined for him and he will finally get the right access permission to access the patient’s record. This scenario describes how the attributes and access control policies play in vital role for achieving right access control decision and ensure access to the information.

When a physician has a log in, four types of actions are displayed and he can do any defined access to his patient’s record according to access control definitions in TABLE I. When he inserts medication information to his patient’s record, he has to authenticate himself to ensure policy-based and attribute-based fine-grained authorization. Some of sample implementation results are shown in Fig: 2, 3, 4 and 5. The corresponding authorization error page is shown if the user is not authorized person according to access control policies in policy database.

TABLE I: Access Control Policy Definitions

User	Access Right	Type of Information To Be Accessed
Administrator	Register the users, Insert, View and Update	Basic Info
	Delete	Entire record
Staff	Insert, View and Update	Lab Test Info
Physician	View	Basic and Lab Test Info
	Insert, View and Update	Medication Info (only if he is the designated physician)
Patient	View	Basic, Lab Test and Medication Info (of his own record)



Fig. 2: Four Types of Access Action to Access Patient's Record



Fig. 3: User Authentication



Fig. 4: Required Information to Access Patient's Record



Fig. 5: Insertion of Medical Information by a Physician

Thus, the users of healthcare application can securely access their respective health records with well-managed authorization. They can easily access from various client devices, such as desktop computers, laptops, tablets and smart phones without specific software, just using a web browser to interact with the application.

6. Conclusion

The system is not completely dependent on the idea “do once- use forever approach”, meaning that it needs to be continuously observed and improved to allow dynamic access control policies to adapt to a dynamic, ever-changing environment. The system has implemented to analyze the effectiveness of XACML for fine-grained access control in cloud computing environment with healthcare application. While healthcare personnel need

access to the right information at the right time to provide the best possible care, it is also important to ensure patient privacy. The proposed system can provide healthcare workers and patients to delete, add, modify and retrieve patients' health records easily to ensure fine-grained authorization from the hospital web pages in cloud computing environment at any time. It can also be applied for other application domain according to their data and required access control policies. At the time of writing, the system is still being developed. As future work, the effectiveness of the system with experimental result and performance evaluation will be analyzed based on measurements of time per access by comparing the system with authorization and without authorization. Then, we specially intend to straight forward our proposed system to enable dynamic permission definition, assignment, and enforcement with complete administrative access control over delegation.

7. Acknowledgements

It gives me great pleasure to find an opportunity to express my deep and sincere gratitude to Dr. Myo Min Than, Professor and Head of Department of Information Technology Engineering, Yangon Technological University, for extending all the facilities of the department. I am extremely grateful to Dr. Nyein Nyein Oo from Department of Information Technology Engineering, Yangon Technological University, for their continual support and guidance to complete this paper. Last but not the least; I humbly extend my attitude to other faculty members, library staff, my friends, and administration of YTU for providing me their valuable help and time with a congenial working environment for writing this paper.

8. References

- [1] Frode Hansen, et al, "Application of role-based access control in wireless healthcare information systems", The College of Information Sciences and Technology © 2007-2013 The Pennsylvania State University.
- [2] Ruoyu Wu, Gail-Joon Ahn, Hongxin Hu. "Secure Sharing of Electronic Health Records in Clouds"; in *Proceedings of CollaborateCom 2012*, pp. 711-718.
<http://dx.doi.org/10.4108/icst.collaboratecom.2012.250497>
- [3] Alhaqbani, Bandar S. and Fidge, Colin J, "Access control requirements for processing electronic health records," in *Proceedings Business Process Management 2007 Workshops: First International Workshop on Process-Oriented Information Systems in Healthcare (ProHealth 2007)* 4928, pages pp. 371-382, Brisbane - Australia.
- [4] Lillian Rostad, "Access control in healthcare information systems", PhD. thesis, Norwegian University of Science and Technology, Trondheim ,January 2009.
- [5] Li-Qun Kuang, Yuan Zhang, Xie Han, "Access Control Policies for Web Services in Medical Aid System," in *2009 International Conference on Information Management, Innovation Management and Industrial Engineering*.
<http://dx.doi.org/10.1109/ICIM.2009.199>
- [6] Massimiliano Masi, Rosario Pugliese, Francesco Tiezzi, "Formalization and Implementation of a Standard AC Mechanism for Web Service", in *Proceeding ESSoS'12 Proceedings of the 4th international conference on engineering secure software and systems*, pp. 60-74.
- [7] <http://en.wikipedia.org/wiki/XACML>
- [8] <http://en.wikipedia.org/wiki/WSO2>
- [9] <http://wso2.com/products/identity-server/>
- [10] <http://wso2.com/products/application-server/>