An Attribute Based Access Control Framework for Healthcare System

Majid Afshar¹, Saeed Samet², Ting Hu³

¹Memorial University of Newfoundland, Department of Computer Science, St. John’s, NL, Canada.
²University of Windsor, School of Computer Science, Windsor, ON, Canada (Adjunct Professor at Memorial University of Newfoundland, Faculty of Medicine, St. John’s, NL, Canada)
³Memorial University of Newfoundland, Department of Computer Science, St. John’s, NL, Canada.

Corresponding author’s email: mman23@mun.ca

Abstract. Nowadays, access control is an indispensable part of the Personal Health Record and supplies for its confidentiality by enforcing policies and rules to ensure that only authorized users gain access to requested resources in the system. In other words, the access control means protecting patient privacy in healthcare systems. Attribute-Based Access Control (ABAC) is a new access control model that can be used instead of other traditional types of access control such as Discretionary Access Control, Mandatory Access Control, and Role-Based Access Control. During last five years ABAC has shown some applications in both recent academic fields and industry purposes. ABAC by using user’s attributes and resources, makes a decision according to an access request. In this paper, we propose an ABAC framework for healthcare system. We use the engine of ABAC for rendering and enforcing healthcare policies. Moreover, we handle emergency situations in this framework.

1. Introduction
The principal concern of Personal Health Records (PHR) is confidentiality [1], especially when patients use commercial healthcare distributed based systems, such as Microsoft HealthVault [2] or WebMD, to maintain their health records. Access Control Mechanism (ACM) is useful for providing access policy for different types of users. However, traditional access control mechanisms, such as Role-Based Access Control (RBAC) and Identity Based Access Control (IBAC), have several restrictions in distributed based systems [3], [4]. In particular, the data owner is unable to control the information when they upload their data to the server.

Access control means the process of defining and managing which users are permitted to perform what operations on which objects in a system. Availability of resources is one of the aspects of access control that conflicts with restricting access to resources that is another perspective of access control [5]. Access control creators need to know how the organization functions work and the access requirements of the system users. In case of the high-level user restrictions, the result has at least two kinds of disadvantages. First, if the user with a valid request cannot access to seeking information, then she tries to find other ways to accessing the required recourses. Second, and more important in healthcare systems is the lack of exception case handling.
Several of the access control mechanisms have been proposed in healthcare information systems [6], [7], [8], [9] without addressing exception or emergency handling. However, some of those have had to pass exceptions to the typical access control mechanisms to be able to satisfy the needs of their users in emergency cases. It is evident that handling of exception cases is not easy due to its effect on control over information flow in a healthcare system. The purpose of this study is to apply the concept of ABAC instead of existing current access control mechanisms, and also to address the problem of emergency cases in healthcare applications.

The rest of this paper is organized as follows. Section 2 includes two subsections on background and related work. In section 3, we show an ABAC framework for the healthcare system. Section 4 describes how to manage emergency situations. In section 5, contains conclusions and future work.

2. Background and Related Works

2.1. Background

ABAC describes an access control model where access rights are admitted to users by the use of policies [4]. Attributes are properties that describe specific features of the subject, object, environment conditions, and requested actions that are predefined and preassigned by an owner or administrator or authority. Attributes are formed of an arbitrary section that indicates the class of information supplied by the attribute, a name, and value.

A subject is an actual entity such as a person, process, or device that by sending requests makes changes to the system’s status. It could be a user, requestor, or mechanism acting on behalf of a user or requestor. In this work, we assume that subject and user have the same meaning. An object is a passive system-related entity that contains or receives information such as files, records in a table, databases, applications, networks, and domains. For the goal of this work, we assume that object and resource are synonymous.

By processing of a subject's request upon an object, the system runs a particular function, which is called an operation. Operations could be one or more of the following functions: read, write, edit, delete, author, copy, execute, and modify. The policy is the description of rules or connections that determine the set of permissible operations a subject should run upon an object in authorized environment conditions. The policies could contain the combination of the following kinds of attributes [4], [10]:

- User (subject) attributes,
- Resource (object) Attributes,
- Environmental attributes (conditions), and
- Administrative attributes.

2.2. Related Work

The Role-Based Access Control (RBAC) model was formally presented by David F. Ferraiolo and Richard Kuhn [11]. By using RBAC, organizations can assign users to defined role(s). Then users can obtain all the permissions of the roles of which they are the member. Roles can be generated for the various tasks in an organization and based on ability, and eligibility, users are assigned to the roles. Users are able to select other roles based on the system policy management. Roles can be given new permissions from resources, and permissions can be revoked from roles as required.

The above description has turned RBAC as a primary model of option for access control applications. However, RBAC suffers from some major problems, especially in healthcare system. First and the major weakness of RBAC is role explosion [12]. When an organization that using RBAC faces a rising number of various real-world roles it needs an increasing quantity of RBAC roles to define permissions well. In this case, handling all those roles becomes a complicated circumstance [13]. Next, RBAC gives an almost same level of access to users who have the same roles. Although RBAC is scalable and adaptable and has been implemented a lot in different applications [14], [15], [16], it is not proper for the healthcare systems [17].

3. Access Control Architecture
As shown in Fig 1, this example contains the following primary functional nodes: The Policy Enforcement Point (PEP), the Policy Decision Point (PDP), the Policy Administration Point (PAP) and the Policy Information Point (PIP). Those parts also called ABAC engine. Once aforementioned segments are in a schema together, they have to perform in compatibility to present proper access control decisions.

As part of authorization process, PDP is the heart of the system and responsible for assessment on policies and attributes that come from PAP and PIP, respectively, to return an access control decision. In other words, it makes the access decisions by assessing the appropriate policies and attributes. The PDP should return the decision procedures based on the ACM's computational languages like XACML [18]. Of course, one of the principal duties of the PDP is to reduce the risk of collision between system polices and corresponding Meta-Policies (MP).

Then, the output of PDP must be enforced by the next component. This function refers to the PEP. In response to a user’s request, the PEP performs the appropriate access decisions that is prepared by the PDP. In this system, user could be a doctor, researcher or nurse. The result or output of PEP could either allow or deny user access to the requested protected patients’ records.

Packages of PDP and PEP work as authorization services and they need some information regarding user and patient’s attributes as part of the input to properly perform their duties. PIP addresses this issue, and it works as the retrieval source of attributes as well as healthcare environmental data, and other data needed for policy assessment. For storing and retrieving polices in an appropriate repository, we need another function, called PAP. PAP should also be able for creating, editing, running and debugging policies and MPs by a user interface. To explain how this proposed framework can handle a user request and protect patient’s privacy, we use a simple scenario as follows.

First of all, we should have the primary policy for the whole system. Consider in a hospital, doctors could have access to all patient’s individual records, and nurses just have access to identity patients’ information, and researchers can see statistical information only if the number of patients is greater than 50. Besides, each set of policy must have a supplementary segment that specifies requester, type of resource and action. In this case, data requester must be a doctor, nurse or researcher. Resource demonstrates clinical health data like patient’s individual records that contain high-level sensitive data, and the action is the set of proper functions that data requester is authorized and permitted to do on the resource, such as read, write and execute. If all conditions mentioned in supplementary segment of the policy are satisfied by a data requester, like a doctor, then the policy set might be considered as an applicable policy.

Consider the following table that shows our example and some additional conditions in supplementary segment. Each employee in hospital has a role attribute. If the value of this role equals to doctor, nurse or researcher, then they might be able to see some results from sensitive health information.

Assume a doctor as a data requester needs to read a particular patient’s record and this record has been registered two years ago (Request 1). Based on the supplementary segment in Table 1 that request satisfies all the conditions. Thus, the policy can consider as an applicable policy. On the other hand, patients can totally determine who can access to their health data and for what part of data by adding some policies. For instance, a patient just allows her doctor to have access only to her last year treatments. Also, the patient is not willing to share his health data with any researcher.

Based on the aforementioned patient policy (conditions), the Request 1 should be rejected due to the value of attributes of data requester, that cannot satisfy the patient’s policy. Of course, the system must have a strategy in especial occasions to avoid conflicts. As an example, if the system receives two decisions that made by PDP at the same time on a same resource, then the system should call a particular algorithm to handle the conflict.
Figure 1. Using ABAC in healthcare system.

Table 1. The Main Policy Set For Hospital

| Data Requester | Type of Resource                      | Action                      |
|----------------|--------------------------------------|-----------------------------|
| Doctor         | Patient’s history records             | Read, Write                 |
| Nurse          | Identity information of patients      | Read                        |
| Researcher     | Statistical Information               | Execute if Patient count> 50|

Supplementary Segment

| Data Requester | Type of Resource                      | Action                      |
|----------------|--------------------------------------|-----------------------------|
| Doctor         | Patient’s history records (only last 5 years) | Read, Write                 |
| Nurse          | Identity information of patients      | Read                        |
| Researcher     | Statistical Information (all years except two last years) | Execute if Patient count> 50|

4. Manage Emergency Cases

Handling emergency situations in a healthcare system is so complicated. Consider a particular patient comes with or without an ambulance to the hospital, and there is no time for patient registering process that is essential for access control. We propose a mechanism to manage that situation called emergency access. In this case, the healthcare system provides some access to the patient's data. This grant is not only directly related to a job position or special role, but is also intended as a special permission associated directly with the particular data requester to be able to satisfy all the following conditions with her attributes.

- Data requester must be a doctor.
- The doctor must work in an emergency section.
- The doctor must be at work at the time of the request.

The reasoning for this policy is that only a limited number of data requesters might be able to assign this permission, and they must be decided on their attributes not just based on the job position. Besides, there is a possibility of tracing the usage of emergency access by an administrator. Normally only a few users at emergency sections might be able to gain this permission. Although emergency access is only supposed to be applied in emergency cases, to prevent any abuse of this grant, we appended some additional conditions after assigning the permission. The conditions may modify lightly from one system to another, but the most common ones can define as follows:

- If the patient sets some policies after a while, then emergency access is not valid any more.
- By using the emergency access just permit user to have access only to a subset of the available patient information, not the complete patient record. To gain access to the other parts of the record, the user has to pass some other system policies.
• Information obtained using emergency access will only remain accessible for the data requester for a short period of time that can be defined based on average time needed.
• Regaining access to the patient's information after a timeout is needed to satisfy all the conditions again.

5. Conclusion
In this paper, we have illustrated the application of ABAC as a framework in the healthcare systems. Besides, we have discussed on how this framework can manage and enforce policies to grant or deny data requesters based on their attributes. ABAC supports accommodating the unexpected users. By using this feature and applying some conditions as an emergency access policy, we have demonstrated how an emergency situation could be handled with patient’s privacy protection. Although ABAC has several advantages in the healthcare systems, it is still largely in its infancy, and the list of future work and open problems are extensive. For instance, we could add some other features to address other real scenarios. Implementing the new framework with new features and check the corresponding performance are among other future directions of this work.

6. References
[1] Señor, Inmaculada Carrión, José Luis Fernández-Alemán, and Ambrosio Toval. "Are personal health records safe? A review of free web-accessible personal health record privacy policies." Journal of Medical Internet Research 14.4 (2012).
[2] Sunyaev, Ali, Dmitry Chernyj, Christian Mauro, and Helmut Krcmar. "Evaluation framework for personal health records: Microsoft HealthVault vs. Google Health." In System Sciences (HICSS), 2010 43rd Hawaii International Conference on, pp. 1-10. IEEE, 2010.
[3] Kalam, Anas Abou El, R. El Baida, Philippe Balbiani, Frédéric Cuppens, Yves Deswarte, Alexandre Miege, Claire Saurel, and Gilles Trouessin. "Organization based access control." In Policies for Distributed Systems and Networks, 2003. Proceedings. POLICY 2003. IEEE 4th International Workshop on, pp. 120-131. IEEE, 2003.
[4] Hu, Vincent C., David Ferraiolo, Rick Kuhn, Arthur R. Friedman, Alan J. Lang, Margaret M. Cogdell, Adam Schnitzer, Kenneth Sandlin, Robert Miller, and Karen Scarfone. "Guide to attribute based access control (ABAC) definition and considerations (draft)." NIST special publication 800, no. 162 (2013).
[5] Fisler, Kathi, Shriram Krishnamurthi, Leo A. Meyerovich, and Michael Carl Tschantz. "Verification and change-impact analysis of access-control policies." In Software Engineering, 2005. ICSE 2005. Proceedings. 27th International Conference on, pp. 196-205. IEEE, 2005.
[6] Blobel, Bernd. "Trustworthiness in distributed electronic healthcare records-basis for shared care." In Computer Security Applications Conference, 2001. ACSAC 2001. Proceedings 17th Annual, pp. 433-441. IEEE, 2001.
[7] Blobel, Bernd. "Authorisation and access control for electronic health record systems." International journal of medical informatics 73, no. 3 (2004): 251-257.
[8] Hu, Junzhe, and Alfred C. Weaver. "A dynamic, context-aware security infrastructure for distributed healthcare applications." In Proceedings of the first workshop on pervasive privacy security, privacy, and trust, pp. 1-8. 2004.
[9] Schwartmann, Dirk. "An attributable role-based access control for healthcare." In International Conference on Computational Science, pp. 1148-1155. Springer, Berlin, Heidelberg, 2004.
[10] Servos, Daniel, and Sylvia L. Osborn. "Current Research and Open Problems in Attribute-Based Access Control." ACM Computing Surveys (CSUR) 49, no. 4 (2017): 65.
[11] David, Ferraiolo, and Kuhn Richard. "Role-based access controls." In Proceedings of 15th NIST-NCSC National Computer Security Conference, vol. 563. Baltimore, Maryland: NIST-NCSC, 1992.
[12] Elliott, Aaron, and Scott Knight. "Role Explosion: Acknowledging the Problem." In Software
Engineering Research and Practice, pp. 349-355. 2010.
[13] Sandhu, Ravi, Venkata Bhamidipati, and Qamar Munawer. "The ARBAC97 model for role-based administration of roles." ACM Transactions on Information and System Security (TISSEC) 2, no. 1 (1999): 105-135.
[14] Tang, Bo, Qi Li, and Ravi Sandhu. "A multi-tenant RBAC model for collaborative cloud services." In Privacy, Security and Trust (PST), 2013 Eleventh Annual International Conference on, pp. 229-238. IEEE, 2013.
[15] Li, Wei, Haishan Wan, Xunyi Ren, and Sheng Li. "A refined RBAC model for cloud computing." In Computer and Information Science (ICIS), 2012 IEEE/ACIS 11th International Conference on, pp. 43-48. IEEE, 2012.
[16] Takabi, Hassan, James BD Joshi, and Gail-Joon Ahn. "Security and privacy challenges in cloud computing environments." IEEE Security & Privacy 8, no. 6 (2010): 24-31.
[17] Mukherjee, Subhojeet, Indrakshi Ray, Indrajit Ray, Hossein Shirazi, Toan Ong, and Michael G. Kahn. "Attribute Based Access Control for Healthcare Resources." In Proceedings of the 2nd ACM Workshop on Attribute-Based Access Control, pp. 29-40. ACM, 2017.
[18] Rissanen, Erik. "Extensible access control markup language (xacml) version 3.0." (2013).

Acknowledgments
This work has been partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Research & Development Corporation of Newfoundland and Labrador (RDC).