Multi-Owner Multi-Stakeholder Access Control Model for a Healthcare Environment

Conference Paper · October 2017

DOI: 10.1109/CIC.2017.00053

2 authors:

Leila Karimi
University of Pittsburgh
4 PUBLICATIONS **3** CITATIONS

SEE PROFILE

James B. D. Joshi
University of Pittsburgh
187 PUBLICATIONS **4,252** CITATIONS

SEE PROFILE

2017 IEEE 3rd International Conference on Collaboration and Internet Computing

Multi-Owner Multi-Stakeholder Access Control Model for a Healthcare Environment

Leila Karimi, James Joshi
School of Computing and
Information
University of Pittsburgh
Pittsburgh, PA, USA
{lek86, jjoshi}@pitt.edu

***Abstract*— Pervasive usage and wide-spread sharing of Electronic Health Records (EHRs) in modern healthcare environments has resulted in high availability of patients' medical history from any location and at any time, which has potential to make health care services both cheaper and of higher quality. However, EHRs contain huge amounts of sensitive information which should be protected from unauthorized accesses, otherwise allowing these records to be accessed by multiple parties may put patient privacy at high risk. Access control solutions must assure to reflect access control policies of all healthcare providers who are involved in generating such critical records as well as authorization policies of the patient as the primary stakeholder.**

In this paper, we propose a fine-grained semantic-based access control model that supports multi-owner multi-stakeholder policy specification and enforcement. In the proposed scheme, a trusted Policy Server is responsible for evaluating access requests to patients' health information. We also handle the policy conflicts that might arise at the time of access control policy enforcement. A proof-of-concept prototype is also implemented to demonstrate the feasibility of our model.

I. Introduction

In the last two decades, the Internet has enabled an opportunity for many organizations to transfer their paper-based workflows to electronic-based ones and many of them have become connected via the Internet. The healthcare domain is not an exception. Healthcare providers use Electronic Medical Records (EMRs) for maintaining patient health information. Healthcare treatments and researches often require the integration and sharing of medical data coming from multiple sources. Such integration can lead to greater efficiency, improved patient care, and cost savings. It reduces unnecessary duplication of lab tests and clinical visits and improves the quality of emergency services, but at the same time, it introduces new security and privacy challenges.

Nowadays, it is very common for patients to have multiple healthcare providers who may be scattered throughout a country or even the world. These healthcare providers may use EHRs for integrating and sharing patient information with each other. EHRs may contain sensitive patient data such as laboratory test results, diagnoses, and medication. Integration and sharing of these records must be done by assuring patients privacy through proper access control (AC) policies. Healthcare providers have their own AC policies and mechanisms for protecting patients' medical information from unauthorized accesses. AC solutions for shared EHRs must guarantee that AC policies of all owners of the shared medical records are accurately enforced.

In addition, electronic medical devices (e.g. wearable trackers) along with medical applications on mobile devices enable patients to generate health information which can be delivered to the healthcare providers for better serving them. Patients may have different privacy concerns for sharing their sensitive health information which can restrict access to their information, so healthcare systems should provide a mechanism for their patients to define their own access control policies.

Considering aforementioned scenarios, patients' personal and medical information can be generated by multiple parties and it may move from one stakeholder to another. Assuring patients' privacy becomes a very complex and critical issue in such a heterogeneous environment.

As the medical information flows between different parties, the original AC policies of the owner of the data need also to be shared to assure authorized accesses to the sensitive data at the point of care. Different stakeholders in the healthcare domain may not be willing to share their access control policies with each other, so a trusted third party can help in integrating and enforcing AC policies of patients and various healthcare providers.

There are various forms of EHR sharing in a healthcare environment. One approach is federation of institutions, healthcare providers share their EHR records on the basis of needs. In this approach, each healthcare provider has an individual EHR system and it is responsible for controlling access to sensitive medical information. If another stakeholder needs to access some of those EHRs, it has to negotiate with the data owner to get the required authorization by creating interoperation policies or establishing role to role mappings [1].

In another approach, patients share their personal health records (PHRs) through PHR repositories such as Microsoft Health Vault [2]. In this approach, patients generate their own health information through selected medical devices or they collect their health records from their healthcare providers and insert it to the PHR repository. In such repositories, patients are responsible for controlling with whom their medical information can be shared.

0-7695-6303-1/17/31.00 ©2017 IEEE
DOI 10.1109/CIC.2017.00053

In another approach, a patient's health information can be aggregated from multiple sources as a community service, the goal of which is to "deliver the right health information to the right place at the right time" [3]. These Community Health Records (CHRs) are accessible through Health Information Exchange (HIE) which facilitates the flow of clinical information among several health information systems.

Considering all aforementioned scenarios for sharing medical information among multiple stakeholders in a healthcare environment, the need for having a comprehensive and fine-grained AC model becomes obvious. Such a model should respect authorization policies of all involved parties.

Several approaches have been proposed to solve this problem such as in [12], [25], [26], [28]. However these approaches have three key limitations. First of all, many of these solutions (e.g. [25], [26]) do not respect the patients' right in defining their AC policies, but they're focused on enterprise oriented approaches for authorization policy specification. Second, most of these approaches (e.g. [28]) do not consider the structural composition of EHRs, so the access is either granted to the entire EHR or not at all. Finally, a number of existing models (e.g. [12]) are completely patient centric and they do not include healthcare providers' rights in defining their AC policies. Patients may not have enough knowledge in defining proper authorization policies or they may intentionally try to hide some parts of their health records, so solely relying on patient-centric AC model may not be a desirable approach.

In this paper, we propose a fine-grained semantic-based AC model that assures to reflect AC policies of all healthcare providers who are involved in generating EHRs as well as authorization policies of the patient as the primary stakeholder. In the proposed scheme, authorization policies are based on attributes of different entities in a healthcare domain. A trusted third party is responsible for combining and evaluating authorization policies of multiple parties for a given access request.

The primary contributions of this paper are as follows:

1) A comprehensive and fine-grained access control model is proposed for a multi-owner multi-stakeholder healthcare environment which respects authorization policies of all owners of an EHR record.
2) The proposed model resolves policy conflicts by considering conflict resolution strategy of the owner of the information.
3) The logical representation of the proposed access control model as well as the conflict resolution algorithm is formulated through SWRL rules and SPARQL queries, respectively.
4) A proof-of-concept prototype system has been implemented to show the efficiency and effectiveness of the model.

II. MoMsAC Requirements

We show the requirements for a multi-owner multi-stakeholder access control (MoMsAC) model for a healthcare environment through the following example.

Example: *Alice is a patient who suffers from Spina Bifida (the most common disabling birth defect in the US). She uses a mobile health system, iMHere [7] which helps her promote self-management skills and improve her health status. Through iMHere, Alice generates lots of information about her condition by uploading pictures of her wounds, answering questionnaires about her mental status, and so on. Alice can share this information with clinicians at UPMC hospital where she is under treatment but she may have some AC policies related to her information. For example, she wants her pictures to be accessible only to her family doctor. In addition, UPMC clinicians generate other information related to Alice through clinical appointments, home visits, and her hospitalizations. This information is stored at UPMC database and only authorized users can access them based on AC policies of UPMC. One policy can be:"Only users who are Wellness Coordinators can access data which is related to Spina Bifida".*

Assume that Alice wants to move to Maryland and continue her treatments at John Hopkins Hospital. Transferring information related to her disease generated by her through iMHere or by UPMC clinicians can help in time and cost savings and may result in better health services from JH hospital. However, any access to the information should meet both Alice's and UPMC's AC policies, otherwise it may put Alice's privacy at high risk.

Later on, clinicians at JH may need to gather more information about Alice's status through lab tests and clinical meetings. This information can be merged with previous information. JH hospital may have its own policies for controlling access to this information. However, the original AC policies of Alice and UPMC hospital should also be enforced.

In more complex situations, Alice's information can be shared with other stakeholders such as insurance companies, government agencies, medical researchers, etc . Sharing of this information should meet authorization policies of all parties involved in generating the information (i.e. Alice, UPMC, and John Hopkins Hospital).

So as the medical data of patients flow between different stakeholders in a healthcare environment, the original access control policies of the owner of the data should also transfer besides the data. But in practice, different stakeholders may not be willing to share their policies directly with other stakeholders. In addition, managing pairwise sharing of authorization policies may incur huge overheads to each party (e.g. when a policy needs to be updated, a healthcare provider should inform all other parties about such an update). Making use of a trusted third party (Policy Server) is our proposed solution for such a scenario. Considering the heterogeneous healthcare environment, an MoMsAC model should fulfill the following requirements:

1) It should enable different stakeholders including patients in a healthcare environment to specify their AC policies.
2) Individual stakeholders should not be required to disclose their AC policies with other stakeholders.

3) If a requester requests access to more medical information than what he is authorized to access, the whole request should not be denied, but the allowed subset of EHR elements should be returned instead.
4) It should consider the policy conflicts that may arise when combining policies of different stakeholders and have a mechanism for resolving them.

III. Overall Architecture of the Proposed MoMsAC Framework

Figure 1 represents the overall architecture of the proposed EHR sharing system as well as the proposed AC framework. The medical records are stored in several shared repositories such as PHR, CHR, and private repositories. These records are generated by various healthcare providers or by the patients' personal medical devices. The information may be outsourced to the shared repositories (dotted blue lines) to be accessible by other stakeholders. The authorization policies related to these records are located in Policy Server. When a healthcare provider wishes to share his medical records with other stakeholders, he needs to outsource his access control policies to the Policy Server (dashed red lines).

Later on, a user may want to access medical records of a specific patient. The user (who can be the patient, himself, his family member, or his physician) sends his access request to the Policy Server. The Policy Server first locates the appropriate repository which contains the required records through the Repository Lookup Service (RLS). The information about records and their corresponding repositories are stored in RLS when owners of those records outsource the information to the shared repositories and their corresponding authorization policies to the Policy Server.

After locating corresponding repositories, the Policy Server informs the user about the required attributes for completing his request. The user needs to fetch all required attributes from a group of Attribute Providers (APs) and forward related certificates to the Policy Server. All the processes of sending access request and fetching required attributes are done by users' clients on behalf of the users.

A. Policy Server

Policy Server is the primary entity in the proposed AC model. It mediates accesses to the shared repositories and enforces authorization policies of the owners of the medical records. We assume that Policy Server is completely trusted, it follows the authorization evaluation algorithm correctly and it does not compromise the privacy of patients and healthcare providers. The Policy Server receives Access Requests, evaluates them and sends corresponding responses. Figure 1 shows the components of the Policy Server which are related to the proposed access control model.

The Knowledge Base (KB) stores the information related to the entities of the system through a set of ontologies. These pieces of information contain properties related to the entities and the relationships between them which are used in establishing AC policies and at the time of authorization evaluation. The Policy Repository (PR) stores authorization policies.

The other two components are Policy Decision Point (PDP) and Policy Enforcement point (PEP). PDP evaluates the access requests based on the properties of the involved entities (which is stored in KB) and the authorization policies (which is stored in PR). Based on this evaluation the PDP decides to either grant the access or deny it.

After evaluating the access request, the PEP is responsible for responding with the proper answer. If the access was rejected, the proper message should be returned back to the requester, otherwise, the corresponding Access Token should be sent.

IV. The Proposed MoMsAC Model

Below is the description of components in our proposed model:

- PA, the set of patients. Each $pa_i \in PA$ has a unique identifier ID_{pa_i} in the whole system
- HP, the set of healthcare providers. Each $hp_i \in HP$ has a unique identifier ID_{hp_i} in the whole system.
- OW, the set of data owners in the system.
- $OW \subseteq PA \cup HP$
- CE, the set of composite EHRs in the system. Each $C_i \in CE$ contains the health information of one patient in the system.
- V, the set of EHR elements in the system. Each $C_i \in CE$ contains a set of EHR elements $V_{C_i} \subseteq V$
- R, the set of repositories in the system.
- AP, the set of attribute providers in the system.
- A, the set of attributes in the system.
- U, the set of all users in the system. Users of the system may request to access a Composite EHR or some part of it.
- P, the set of authorization policies in the system that are shared with Policy Server.
- ON, the set of ontologies in the system that authorization policies are defined based on. Details of these ontologies are discussed in IV-A.
- $expectedAttributes : P \to 2^A$, a function returning the set of attributes that are related to a given authorization policy.
- $getOwners : PA \to 2^{OW}$, a function returning the set of owners of all EHR elements related to a given patient.
- $getRepositories : PA \to 2^R$, a function returning the set of all repositories which stores EHR elements related to a given patient.
- $getPolicies : OW \to 2^P$, a function returning the set of authorization policies related to a given owner.

A. Healthcare Domain Ontologies

In order for different stakeholders in a healthcare environment to be able to share patients' information with each other, there should be a unified medical record structure. There are various generic reference models such as HL7 [9], openEHR [10], and UDS [11] that try to provide schemes that show logical relationships between different elements of

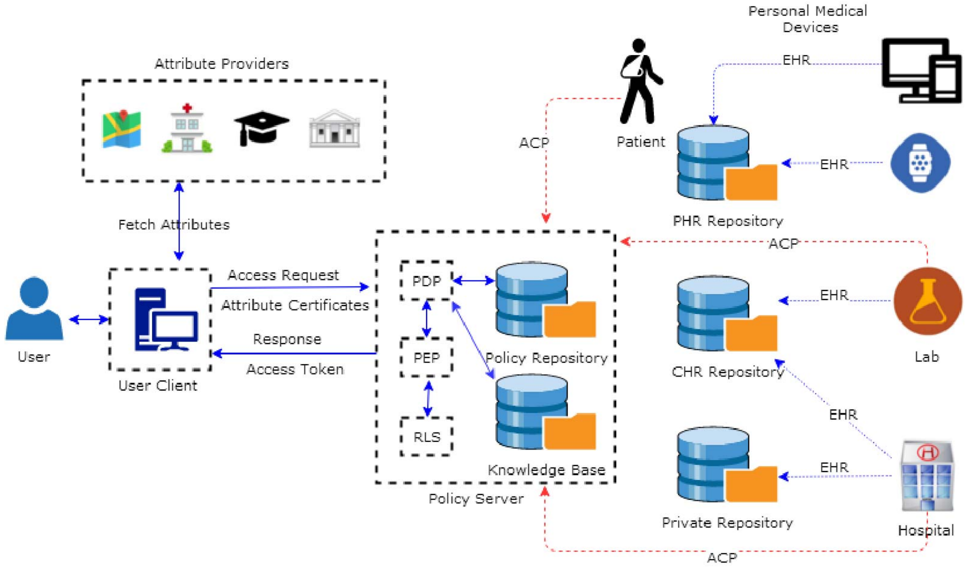


Fig. 1. Overall Architecture of the Proposed MoMsAC Framework

a patient's health record. Based on these reference models, patient's EHR records from different healthcare providers can be integrated into a composite EHR record. Figure 2 depicts the composite EHR ontology which is based on a simple Unified Data Schema (UDS) [11]. In this ontology, the Composite EHR simply includes three categories of patients information: Demographics, History and Labs. Each descendant of the Composite EHR is a medical data element that needs to be protected from an unauthorized access. The composite EHR ontology will be used in policy specification to point to the EHR records to which the access control policy refers to.

In addition to a unified composite EHR model, we need to model other entities in a healthcare environment and the relationships between them to be able to specify fine-grained authorization policies. Figure 3 represents the key entities in a healthcare environment and their relationships. There are three main classes: Healthcare Provider, User, and Composite EHR. Every entity in a healthcare domain including users, patients, composite EHRs, and actions may have various attributes such as *Sensitivity*, *Role*, *Location*, and *Time*. We can also assume different relations between users of the system such as *isFamilyDr*, *isParent*, and *isReferredPhysician*. These attributes and relations play important roles in policy specification and at the time of policy enforcement. More details are discussed in Section IV-B and V.

As suggested in [13], attributes in a healthcare domain can be categorized to two subcategories: semi-static attributes and dynamic attributes. Semi-static attributes such as user's role, EHR record's sensitivity, etc. do not change frequently. They are defined by healthcare providers and stored in KB of Policy Server through healthcare ontology. On the other hand, dynamic attributes such as patient's location or time of the action are not static and will change frequently. These attributes should be collected from Attribute Providers at the time of the access request and inserted to the KB for authorization evaluation.

B. Access Control Policy Specification

As mentioned before, an EHR element can be generated either by a patient or by a staff of a healthcare provider. Later on, this EHR may be shared with other stakeholders in a healthcare domain. We define *owner* and *stakeholder* as follows:

The *owner* of an EHR element is the one who generates that element and uploads it to the shared repository.

Back to our example, when Alice generates some information about her status through the iMHere application, she is the owner of the information. On the other hand, during her visit to UPMC, when a physician generates some records about her, the owner of the generated data is the organization that the physician is affiliated with in that session (which is UPMC here).

Each element of a composite EHR may have a set of owners which is determined at the time of integrating multiple EHR elements into that composite EHR element. The owners of such element are derived as follows:

$$owner(c_1 \cup c_2) = owner(c_1) \cup owner(c_2)$$

For example, demographic information of a patient can be generated by multiple healthcare providers each of whom is the owner of that information.

A *stakeholder* in our model is any organization/user in a healthcare environment who has some benefit in accessing

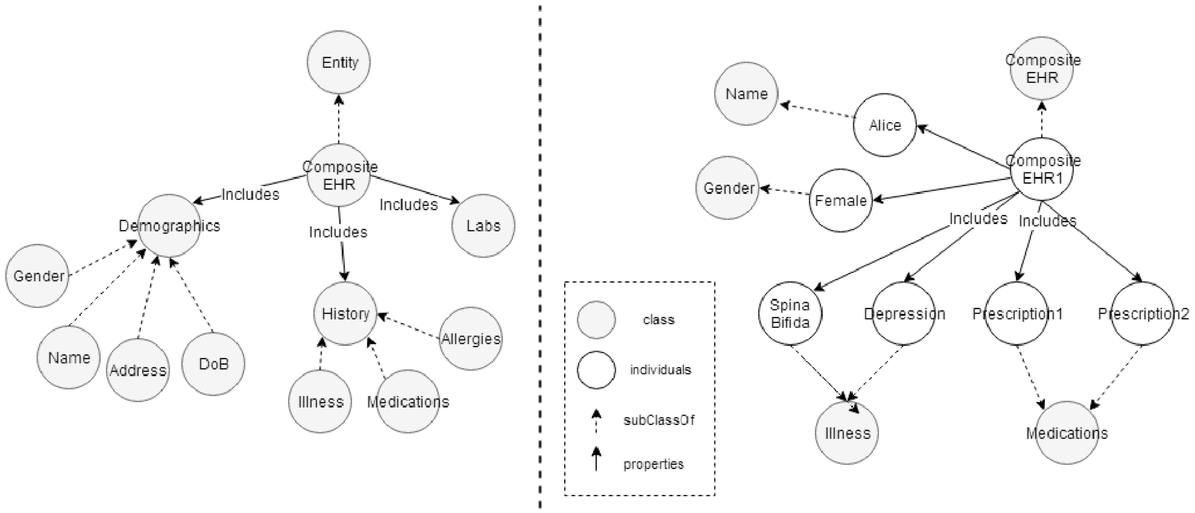


Fig. 2. Composite EHR Ontology

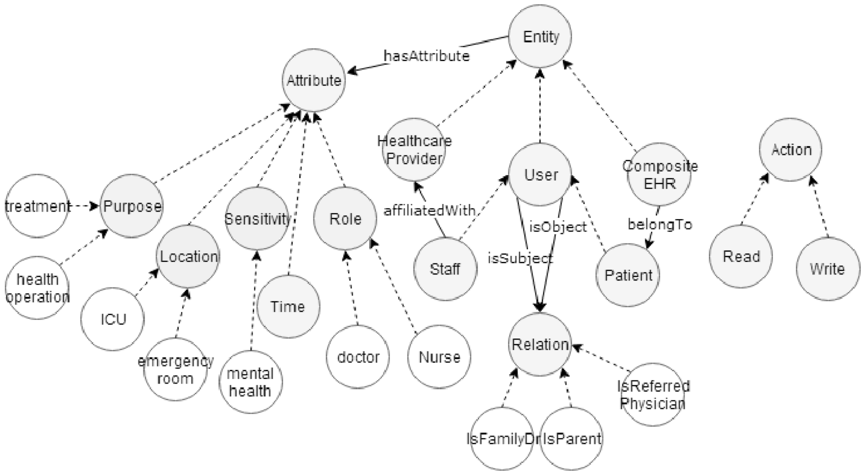


Fig. 3. Healthcare Ontology

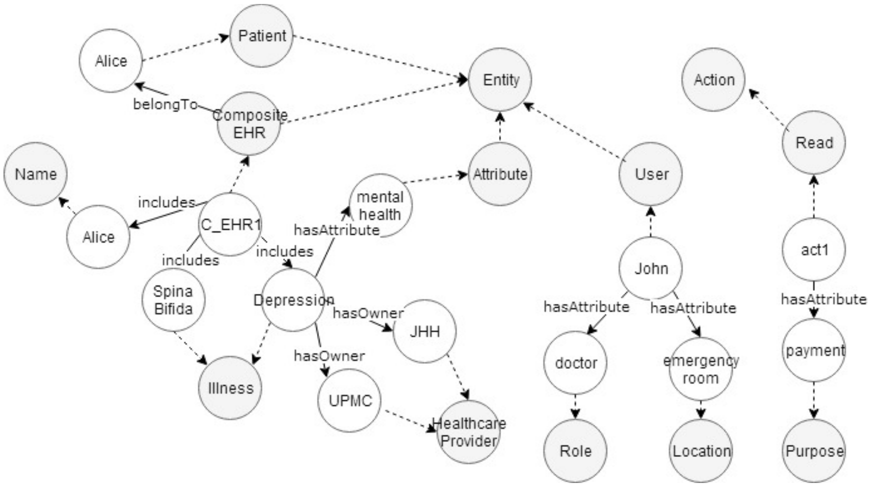


Fig. 4. Instantiated Healthcare Ontology

patients' information. The owner of an EHR element determines the stakeholders of such element and the related information is stored through healthcare ontology in Policy Server.

Back to our example, assume that an insurance company needs to get access to some part of Alice's information for investigating her claim. In such scenario, the insurance company is one of the stakeholders of Alice's EHR.

More formally, for each composite EHR element $c_i \in CE$, its stakeholders include:

- Patient pa_i who this EHR element belongs to.
- Any organization/user in the healthcare environment who is interested in accessing that EHR. Such stakeholder may later want to share some part of the information with other stakeholder.

The owner of an EHR element is responsible for establishing access control policies related to that element and outsourcing those policies to the Policy Server. A stakeholder of an EHR element can also define his own authorization policy for that element. When there is a conflict between different authorization policies for an EHR element the owners' conflict resolution strategy will help in determining the final access decision.

There should be a unified policy specification scheme for all EHR owners to define their authorization policies based on that. Our proposed policy specification scheme is grounded upon the composite EHR ontology and the healthcare ontology. In addition, we adapt the idea of filtration properties from [12] and define the Filtration Attribute Set concept as follows:

Definition 1: [**Filtration Attribute Set**]. Let $\mathcal{A}$ be the set of defined attributes/relations in the system, and for each $\mathcal{A}_i \in \mathcal{A}$, let $Range(\mathcal{A}_i)$ be the set of valid values for that attribute/relation. A filtration attribute set is specified as

$$FA = \{< \mathcal{A}_i, \mathcal{V}_i >| \ \mathcal{A}_i \in \mathcal{A} \ \& \ \mathcal{V}_i \subseteq Range(\mathcal{A}_i)\}$$

The defined attribute/relation set and the possible valid values for each attribute/relation in the above definition can be extracted from the EHR and Healthcare Ontology. We can also use patterns in specifying filtration attribute sets. Pattern "_" means any value within the attribute/relation range, and pattern "{_}" means any set within the attribute/relation dimension.

The filtration attribute sets are used in policy specification for determining specific subjects, objects, and actions based on their attributes or their relations to other entities.

In the following, we discuss some examples of common filtration attribute sets that can be expressed in a healthcare domain:

- *Location of the requester*: The location of the requester (subject) can play an important role in authorization decision, for example, a hospital may want to grant access to the specific patient's records to doctors and nurses who are in ICU room:

$$sfa1 =$$
$$\{< Role, \{doctor, nurse\} >, < Location, \{ICU\} >\}$$

- *Purpose of the request*: According to [16], the purpose of information exchange in a healthcare domain can be categorized into 11 classes including Payment, Treatment and Research. The Purpose-Based Access Control (PBAC) is based on assigning data objects with purposes. These purposes can indicate the reasons why data is collected and what they can be accessed for. Several studies show that this purpose assignment will result in greater privacy preservation [17]–[19]. Below is an example of an action filtration attribute set which restricts authorization outcome to requests with treatment as the action purpose:

$$afa1 = \{< Purpose, \{treatment\} >\}$$

- *Sensitivity level of the EHR element*: Medical data can be categorized into different classifications based on their sensitivity. These classifications include *general*, *mental health*, *drug and alcohol*, *communicable disease*, *decedent*, and so on [12]. Sensitivity classifications can be considered as another dimension in the authorization policies. For example, the following object filtration attribute set points to all EHR elements whose class is *illness* and their sensitivity is *mental health*:

$$ofa1 =$$
$$\{< Class, \{illness\} >, < Sensitivity, \{mentalhealth\} >\}$$

- *Relation between involved entities*: In Relationship-Based Access Control Models (ReBAC) [20], [21], the relationship between two involved entities is also important in the authorization decisions. For example, a patient may want to limit access to his health records only to his family doctor, which can be stated in the following subject filtration attribute set:

$$sfa2 = \{< Relation, \{isFamilyDr\} >\}$$

Definition 2: [**Access Control Policy**]. An access control policy $\mathcal{P}$ is a 5-tuple $\mathcal{P} = (\mathcal{C}, OFA, SFA, AFA, o)$, where

- $\mathcal{C}$ is the composite EHR object the policy refers to;
- OFA is the object filtration attribute set which specifies a set of EHR elements $V_P \subseteq V_C$ this policy especially points;
- SFA is the subject filtration attribute set which determines the set of subjects who are the target of this policy;
- AFA is the action filtration attribute set which defines the set of actions this policy is for; and
- $o \in \{permit, deny\}$ is the decision outcome of the policy.

C. Logical Representation of Access Control Policies

We translate access control policies into logical rules using SWRL. Given a policy $\mathcal{P} = (\mathcal{C}, OFA, SFA, AFA, o)$, the corresponding logical rule is established as follows:

- For each $< \mathcal{OA}_i, \mathcal{OV}_i > \in OFA$ the following logic literal is added to the body of SWRL rule:

$$\mathcal{OA}_i(?Obj, ?X_i) \wedge swrlb : member(?X_i, \mathcal{OV}_i)$$

- For each $< \mathcal{SA}_j, \mathcal{SV}_j > \in SFA$ the following logic literal is added to the body of SWRL rule:

$$\mathcal{SA}_j(?Sbj, ?Y_j) \wedge swrlb : member(?Y_j, \mathcal{SV}_j)$$

- For each $< \mathcal{AA}_k, \mathcal{AV}_k > \in AFA$ the following logic literal is added to the body of SWRL rule:

$$\mathcal{AA}_k(?Act, ?Z_k) \wedge swrlb : member(?Z_k, \mathcal{AV}_i)$$

The overall logical representation for a given access control policy in SWRL language is as follows, here $swrlb : member$ is an SWRL built-in atom:

$$\begin{gathered} request(?req) \wedge hasObj(?req, ?Obj) \wedge \\ hasSbj(?req, ?Sbj) \wedge hasAct(?req, ?Act) \wedge \\ CompositeEHR(\mathcal{C}) \wedge includes(C, ?Obj) \wedge \\ \bigwedge_{i=1}^{n} \mathcal{OA}_i(?Obj, ?X_i) \wedge swrlb : member(?X_i, \mathcal{OV}_i) \wedge \\ \bigwedge_{j=1}^{m} \mathcal{SA}_j(?Sbj, ?Y_j) \wedge swrlb : member(?Y_j, \mathcal{SV}_j) \wedge \\ \bigwedge_{k=1}^{o} \mathcal{AA}_k(?Act, ?Z_k) \wedge swrlb : member(?Z_k, \mathcal{AV}_k) \\ \Rightarrow o(?req) \end{gathered}$$

V. Access Control Policy Evaluation

A. Conflict Resolution

By introducing both positive and negative authorization (permit and deny outcomes), possible conflicts may arise at policy evaluation time. Different conflict resolution strategies have been proposed in literature which include *Deny_Overrides*, *Full_Consensus_Permit*, *Majority_Permit*, etc [23], [24]. Each owner of an EHR element should specify the desired conflict resolution strategy for that element. Such strategy will be used to resolve conflicts at the time of policy evaluation.

We consider two types of conflicts in our model, *Design Conflicts* and *Runtime Conflicts*.

Design Conflict occurs when two authorization policies are obviously in conflict with each other even before they are deployed. In this kind of conflict, all filtration attribute sets of one authorization policy are subset or equal to the filtration attribute sets of another policy but the decision outcome of these two policies are opposite. Here is an example of two conflicting policies in this category:

$$\begin{gathered} \mathcal{P}1 = (\_, \{< Sensitivity, \{mental\ health\} >\}, \\ \{< Location, \{emergency\ room\} >\}, \\ \{< Purpose, \{payment\} >\} \\ , deny) \end{gathered}$$

$$\begin{gathered} \mathcal{P}2 = (\_, \{< Class, \{illness\} >, \\ < Sensitivity, \{mental\ health\} >\}, \\ \{< Location, \{emergency\ room\} >\}, \\ \{< Purpose, \{treatment, payment, health\ operation\} >\} \\ , permit) \end{gathered}$$

The formal definition of two authorization policies in Design Conflict is as follow:

Definition 3: [**Design Conflict**].The two authorization policies $\mathcal{P}1$ and $\mathcal{P}2$ with different outcomes are in Design Conflict if and only if:

$$\forall i : i_{P1} \subseteq i_{P2},$$

where $i \in \{OFA, SFA, AFA\}$.

Runtime Conflict occurs when two authorization policies are not in Design Conflict but after runtime configuration and based on objects, subjects, and actions attributes, the decision outcome of these policies are in conflict. Consider following policies:

$$\begin{gathered} \mathcal{P}3 = (\_, \{\_\}, \{< Role, \{Doctor\} >, \\ < Relation, isFamilyDr >\}, \{\_\}, permit) \end{gathered}$$

$$\begin{gathered} \mathcal{P}4 = (\_, \{< Sensitivity, \{mental\ health\} >\}, \\ \{\_\}, \{< Location, outOfOffice >\}, deny) \end{gathered}$$

As it can be seen policy $\mathcal{P}3$ and $\mathcal{P}4$ are not in Design Conflict but in the following scenario the outcome of theses two policies are opposite of each other:

John is the family doctor of Alice. John wants to get access to some part of Alice's medical record related to her depression from his home. Policy $\mathcal{P}3$ grants such access while $\mathcal{P}4$ denies it.

To resolve conflicts in the proposed model, we use SPARQL queries to conclude about the final decision as follows:

$$\begin{gathered} SELECT\ ?r1\ (COUNT(?r1)\ as\ m) \\ WHERE\{?req\ Permit\ ?r1\} \wedge \\ SELECT\ ?r2\ (COUNT(?r2)\ as\ n) \\ WHERE\{?req\ Deny\ ?r2\} \wedge \\ FinalDecision = m > n\ ?\ Permit\ :\ Deny \end{gathered}$$

The above procedures checks to ensure that if the number of policies granting access to a given request is higher than the number of policies denying such request then the final decision will be Permit, otherwise it will be Deny (*Majority_Permit* as resolution strategy).

B. Access Request Evaluation

In the proposed model, the access request evaluation has two phases. In the first phase, a user who wants to access health records of a patient sends his access request to the Policy Server specifying the unique identifier of that patient as well as the purpose of the access. The Policy Server looks up the repositories which contain EHR elements related to the specified patient, the owners and stakeholders of those EHR elements, and the access control policies of them referring to those elements. The Policy Server generates and returns a set of required subject and action attributes ($\mathcal{RSA}$ and $\mathcal{RAA}$) for all the related policies as follows:

- For each member of each Subject Filtration Attribute Set $< \mathcal{SA}_{ij}, \mathcal{SV}_{ij} > \in SFA_i$, the corresponding subject attribute $\mathcal{SA}_{ij}$ will be added to the Required Subject Attribute set ($\mathcal{RSA}$).
- For each member of each Action Filtration Attribute Set $< \mathcal{AA}_{ij}, \mathcal{AV}_{ij} > \in AFA_i$, the corresponding action attribute $\mathcal{AA}_{ij}$ will be added to the Required Action Attribute set ($\mathcal{RAA}$).

After receiving the required attribute sets, the requester fetches all the required attributes from a group of Attribute Providers (APs) and forward related certificates to the Policy Server. In the second phase of evaluation, for each element of requested EHR, the following steps are taken:

For each owner and stakeholder of that element, the requested access is evaluated against policy rules by considering conflict resolution strategy of the owner.

By assuming that the number of authorization rules and attribute set size are constant or they do not significantly affect evaluation time (as is shown in Section VI), the time complexity of second phase of access request evaluation is of $\mathcal{O}(mn)$ where m is the number of EHR elements in a Composite EHR and n is the average number of owners of EHR elements.

VI. Implementation

We have implemented a prototype of the proposed MoM-sAC enforcement mechanism. The prototype is written in Java on J2SE 1.8.0. We used Protégé as well as OWL-API library for developing ontologies and semantic relations between various entities of the system. The SWRL rules are used for specifying policy rules and the Pellet reasoner is used for working on ontologies and reasoning with SWRL rules. The SPARQL queries are conducted using the SPARQL-DL library. Our experiments are performed on an Intel(R) Core(TM)2 processor with 3GB of memory on a 32-bit Windows operating system.

In order to perform experiments, we developed a random policy rule generator which constructs a random rule by specifying random attributes for object, subject, and action of a request as well as a random outcome for the policy. The set of possible attributes and their valid values are shown in Table I. Figure 5 shows few examples of these random rules. In addition, a random request is also generated in which the corresponding object, subject, and action may have a set of attributes choosing from Table I.

In the first experiment, we evaluate the effect of the size of policy rules on the performance of access request evaluation. In this experiment, we test the performance of the access request evaluation for one EHR element which has one owner. Subject, object, and action of each access request have three attributes on average. Figure 6 shows the results of this experiment. It can be deducted from the figure that the access request evaluation time increases when there are more rules, but such increase is not significant, as we can see when the size of policy rules grows about 10 times (from 50 to 600), the increment of evaluation time is less than one millisecond.

In the second experiment, the effect of the entity's attribute size on the performance of access request evaluation is tested. In this experiment, 200 random rules are generated. We varied the average number of attributes for each subject, object, and action of the access request. Figure 7 shows the results of this experiment. The evaluation time has some fluctuations due to random attribute selection and random rule generation, however as we can notice from the figure, size of the attribute set does not have a significant impact on the access request evaluation time.

In the third experiment, the effect of the number of EHR elements, as well as the average number of owners of EHR elements on the access request evaluation time of a composite EHR is studied. As is discussed in Section V-B the evaluation running time has a linear relationship with both size of EHR elements and an average number of owners. Figure 8 shows such relationships. Deployment of EHR sharing systems will result in fewer repetitive tests and data acquisitions, as a result, the average number of owners will tend to one which may result in acceptable running time.

VII. Related Work

AC models and frameworks for managing EHR sharing and protecting composite EHR have been studied in recent years. Becker, et al.in [25] proposed Cassandra, an AC model for national electronic health record system of UK which is a large-scale heterogeneous distributed system. Cassandra is a role-base AC model. It does not consider spatio-temporal properties of different entities in its authorization policy. OASIS [26] is another role based AC model which was targeted for Uk national health service. Zhang et. al., introduced a role-based delegation framework to manage information sharing in a healthcare environment [27]. Hu et. al, [28] proposed a dynamic context-aware AC model for distributed healthcare applications. Their model is based on RBAC and combines roles with five primitive context types.

Jin et. al. [12], propose an AC model for composite EHRs which is patient-centric and fine-grained with respect to the logical structure of a composite EHR. However, this model is completely patient-centric. Patients may not have enough knowledge in defining proper authorization policies or they may intentionally try to hide some part of their health records, so solely relying on patient-centric AC model is not a wise approach. In addition, the proposed model only consider few properties of subjects (ID and role) and objects

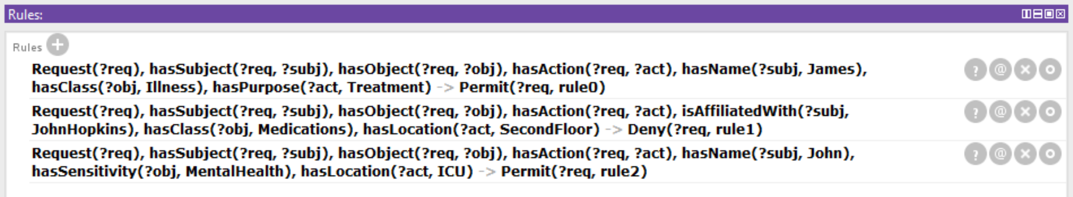
TABLE I

VARIOUS PARAMETERS USED FOR GENERATING POLICY RULES

	Attribute Set	Attribute Value Set

Object Attribute	hasSensitivity
	hasClass
	belongsTo
		{MentalHealth, Drug&Alcohol, CommunicableDisease}
		{Illness, Medications, Allergies}
		{Alice, Bob, Carol}

Subject Attribute	hasRole isAffiliatedWith hasName	{Doctor, Nurse, Admin} {UPMC, JohnHopkins, MayoClinic} {John, Emily, James}
Action Attribute	hasLocation hasPurpose hasTime	{ICU, EmergencyRoom, SecondFloor} {Treatment, Payment, HealthOperation} {Morning, WorkingHour, NonWorkingHour}
Policy Outcome		{Permit, Deny}



```
Request(?req), hasSubject(?req, ?subj), hasObject(?req, ?obj), hasAction(?req, ?act), hasName(?subj, James),
hasClass(?obj, Illness), hasPurpose(?act, Treatment) -> Permit(?req, rule0)
Request(?req), hasSubject(?req, ?subj), hasObject(?req, ?obj), hasAction(?req, ?act), isAffiliatedWith(?subj,
JohnHopkins), hasClass(?obj, Medications), hasLocation(?act, SecondFloor) -> Deny(?req, rule1)
Request(?req), hasSubject(?req, ?subj), hasObject(?req, ?obj), hasAction(?req, ?act), hasName(?subj, John),
hasSensitivity(?obj, MentalHealth), hasLocation(?act, ICU) -> Permit(?req, rule2)
```

Fig. 5. Random Policy Rules

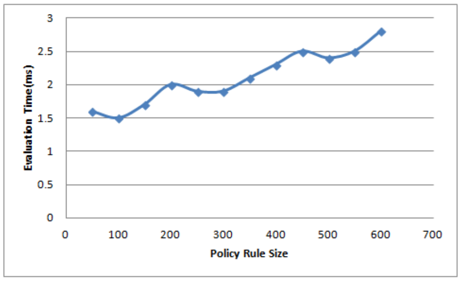


Fig. 6. Required time to evaluate an access request per different policy rule sizes

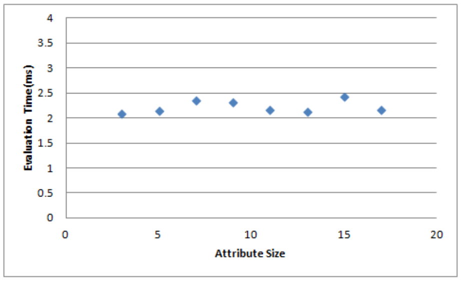


Fig. 7. Required time to evaluate an access request per different attribute set sizes

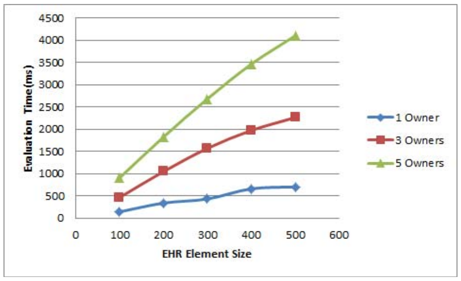


Fig. 8. Required time to evaluate an access request per different EHR element sizes and average owner sizes

(origin, sensitivity, and data type) which will greatly reduce the expressiveness of their model.

Mohan et. al. [13] propose an attribute-based, source-verifiable framework for EHR sharing. Their authorization policies are combination of system policies and patient's policies. However, they did not consider the multi-ownership of EHR elements and they did not exactly mention who is responsible for system policy specification. In addition, they did not formally specify the authorization policy structure, the conflict resolution strategy, or the request evaluation process.

VIII. DISCUSSION AND FUTURE WORK

Trust Model. As mentioned in III, in the proposed framework, all parties has a full trust in the Policy Server for conforming the protocol which may not be applicable in real practice. As a future work, we should focus on weaker assumption where Policy Server is not necessarily trusted and modify the framework to fit such assumption. We can also incorporate a trust and risk analysis model into the proposed framework to overcome such problem.

Privacy. As a follow up to the fully-trusted Policy Server, we expect the proposed framework to preserve the privacy of patients, users, and healthcare providers by protecting their attributes, access control policies and their access patterns from being compromised. However, for achieving such privacy, the channel between different parties and the Policy Server should also be secure, otherwise, they may leak sensitive information during their communication.

Response Time. In a distributed healthcare environment getting access to the complete medical history of a patient may take several hours to several days, and it usually happens through traditional channels such as paperworks, telephones, fax or their modern versions such as email. These channels are not efficient and they may not provide adequate security. In the proposed framework, such access can be provided in less than a minute and in a much more secure manner.

Multi ownership. The proposed framework respects both patients and healthcare providers as an owner of medical records and provides a mechanism which empowers all owners in expressing their access control policies which is not a case in most of the existing access control models in a healthcare environment. They are either focused on patient-centric approaches or they only account on healthcare providers as the main principal authority.

Policy Granularity. A composite EHR record may contain hundreds of records, each of them may have different sensitivity and require different protection level. The proposed framework considers the logical structure of a composite EHR and it provides a fine grained access control model. However, such fine grained model may result in complexity especially for patients who are not expert in access control models. As a future work, we should focus on increasing the usability of our model to overcome this challenge.

IX. CONCLUSION

In this paper, we have proposed MoMsAC, a multi-owner multi-stakeholder access control model for a healthcare environment which empowers different healthcare providers as well as patients to express their fine-grained access control policies on their EHR records. For such purpose a Policy Server based access control framework has been suggested through which owners of the information outsource their authorization policies to the Policy Server and Policy Server is the main responsible point for evaluating access requests based on those policies and returning proper responses. The proposed access control model is built on the composite EHR ontology and the healthcare ontology, which helps the owners of the medical information in defining their access control policies. Different Semantic Web based standards are used for specifying, evaluating, and enforcing authorization policies. We have also implemented a prototype of the proposed model to demonstrate the applicability of our system.

ACKNOWLEDGMENT

This work has been supported by National Science Foundation under the grant DGE-1438809.

REFERENCES

[1] Martino, L. D., Ni, Q., Lin, D., and Bertino, E. ”Multi-domain and privacy-aware role based access control in ehealth.” In Second International Conference on Pervasive Computing Technologies for Healthcare, pp. 131-134, 2008.

[2] Microsoft Health Vault. https://www.healthvault.com.

[3] Chesapeake Regional Information System for our Patients. https://www.crisphealth.org/

[4] McGuinness D. L., Van Harmelen F., and others. ”OWL web ontology language overview.”, W3C recommendation, (2004).

[5] Horrocks I., Patel-Schneider P. F., Boley H., Tabet S., Grosof B., Dean M., and others. ”SWRL: A semantic web rule language combining OWL and RuleML.”, W3C Member submission, 21:79, (2004).

[6] Prud E., Seaborne A., and others. ”Sparql query language for rdf.”, (2006).

[7] Parmanto, B., Pramana, G., Yu, D. X., Fairman, A. D., Dicianno, B. E., and McCue, M. P. ”iMHere: a novel mHealth system for supporting self-care in management of complex and chronic conditions.” JMIR mHealth and uHealth, 1(2), e10, 2013.

[8] Hardt, D. ”The OAuth 2.0 authorization framework.”, 2012.

[9] HL7. Hl7 reference information model. http://www.hl7.org/Library/data-model/RIM/modelpage-mem.htm.

[10] openEHR Community. openEHR. http://www.openehr.org.

[11] dbMotion. White paper: The critical role of integrated patient information in the delivery of high quality healthcare, January 2008.

[12] Jin, J., Ahn, G.-J., Hu, H., Covington, M. J. and Zhang, X. ”Patient-centric authorization framework for sharing electronic health records.” In Proceedings of the 14th ACM symposium on Access control models and technologies, pp. 125-134. ACM, 2009.

[13] Mohan, A., Bauer, D., Blough, D.M., Ahamad, M., Bamba, B., Krishnan, R., Liu, L., Mashima, D. and Palanisamy, B. ”A patient-centric, attribute-based, source-verifiable framework for health record sharing”, 2009.

[14] Wu, R., Ahn, G. J., and Hu, H. ”Secure sharing of electronic health records in clouds.” In Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), pp. 711-718, 2012.

[15] ISO EHR Standards, 2007. http://www.openehr.org/standards/iso.html.

[16] Dimitropoulos, L. L. ”Privacy and security solutions for interoperable health information exchange”, Assess., 2007.

[17] Byun, J.-W. and Li, N. ”Purpose based access control for privacy protection in relational database systems”, VLDB J., vol. 17, no. 4, pp. 603-619, 2008.

[18] Yang, N., Barringer, H., and Zhang, N. ”A purpose-based access control model”, in Information Assurance and Security, pp. 143-148, 2007.

[19] Ni, Q., Bertino, E., Lobo, J., Brodie, C., Karat, C.M., Karat, J., and Trombeta, A. ”Privacy-aware role-based access control”, ACM Trans. Inf. Syst. Secur., vol. 13, no. 3, p. 24, 2010.

[20] Fong, P. W. ”Relationship-based access control: protection model and policy language”, In Proceedings of the first ACM conference on Data and application security and privacy, pp. 191-202. ACM, 2011.

[21] Fong, P. W. and Siahaan, I. ”Relationship-based access control policies and their policy languages”, In Proceedings of the 16th ACM symposium on Access control models and technologies, pp. 51-60, ACM, 2011.

[22] Pritts, J. and Connor, K. ”The implementation of e-consent mechanisms in three countries: Canada, england, and the netherlands”, SAMHSA report, http://ihcrp.georgetown.edu/pdfs/prittse-consent.pdf, 2007.

[23] Li, N., Wang, Q., Qardaji, W., Bertino, E., Rao, P., Lobo, J., and Lin, D. ”Access control policy combining: theory meets practice”, in Proceedings of the 14th ACM symposium on Access control models and technologies, pp. 135-144, 2009.

[24] Hu, H., Ahn, G.-J., and Jorgensen, J. ”Multiparty access control for online social networks: model and mechanisms”, IEEE Trans. Knowl. Data Eng., vol. 25, no. 7, pp. 1614-1627, 2013.

[25] Becker, M. Y., and Sewell, P. ”Cassandra: Flexible trust management, applied to electronic health records”, In Computer Security Foundations Workshop, pp. 139-154, 2004.

[26] Eyers, D. M., Bacon, J., and Moody, K. ”OASIS role-based access control for electronic health records”, IEE Proceedings Software, 153(1), 16, 2006.

[27] Zhang, L., Ahn, G. J., and Chu, B. T. ”A role-based delegation framework for healthcare information systems”, In Proceedings of the seventh ACM symposium on Access control models and technologies, pp. 125-134, 2002.

[28] Hu, J., and Weaver, A. C. ”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.