8th International Conference Conference on Collaborative Computing: Networking, Applications and Worksharing , Collaboratecom 2012 Pittsburgh, PA, United States, October 14-17, 2012

Secure Sharing of Electronic Health Records in Clouds

Ruoyu Wu[1], Gail-Joon Ahn[1], Hongxin Hu[2]
[1]Arizona State University, [2]Delaware State University
{ruoyu.wu, gahn}@asu.edu; hhu@desu.edu

***Abstract*—In modern healthcare environments, healthcare providers are more willing to shift their electronic medical record systems to clouds. Instead of building and maintaining dedicated data centers, this paradigm enables to achieve lower operational cost and better interoperability with other healthcare providers. However, the adoption of cloud computing in healthcare systems may also raise many security challenges associated with authentication, identity management, access control, trust management, and so on. In this paper, we focus on access control issues in electronic medical record systems in clouds. We propose a systematic access control mechanism to support selective sharing of composite electronic health records (EHRs) aggregated from various healthcare providers in clouds. Our approach ensures that privacy concerns are accommodated for processing access requests to patients' healthcare information. We also demonstrate the feasibility and efficiency of our approach by implementing a proof-of-concept prototype along with evaluation results.**

***Index Terms*—Cloud Computing; Electronic Health Record; Access Control; Security**

I. Introduction

In modern healthcare domain, electronic health records (EHRs) [5] have been widely adopted to enable healthcare providers, insurance companies and patients to create, manage and access patients' healthcare information from anywhere and at any time. Typically, a patient may have many different healthcare providers including primary care physicians, specialists, therapists, and miscellaneous medical practitioners. Besides, a patient may have different types of insurances, such as medical insurance, dental insurance and vision insurance, from different healthcare insurance companies. As a result, 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. Particularly, in some emergency healthcare situations, immediate exchange of patient's EHRs is crucial to save lives. However, in current healthcare settings, healthcare providers mostly establish and maintain their own electronic medical record (EMR) systems for storing and managing EHRs. This kind of self-managed data centers are very expensive for healthcare providers. Besides, the sharing and integration of EHRs among EMR systems managed by different healthcare providers are extremely slow and costly. Such an inefficient usability and low cost-effective fashion become the biggest obstacles for moving healthcare IT industry forward [22]. A common and open infrastructure platform can play a vital role in addressing and changing such a situation.

Cloud computing has become a promising computing paradigm drawing extensive attention from both academia and industry [15]. This paradigm shifts the location of computing infrastructure to the third-party service providers who handle the management of hardware and software resources. It has shown tremendous potential to enhance collaboration, scale, agility, cost efficiency and availability. As such, healthcare providers along with many other software vendors are more and more willing to shift their EMR systems into clouds instead of building and maintaining dedicated data centers. Cloud computing, as cornerstone, can not only increase the efficiency of medical data management and sharing process, but also enable us to access healthcare services ubiquitously since patients' healthcare-related data ought to be always accessible from anywhere at any time. It is noted that managing healthcare applications in clouds would make revolutionary changes in the way we currently deal with healthcare information.

It is tremendously beneficial for both healthcare providers and patients to have EHR applications and services in clouds. However, such an adoption may also cause various security challenges associated with identity management, access control, policy integration, compliance management and so on [2], [19], [20], [21]. If those challenges cannot be properly resolved, it hinders the successful deployment of EMR systems in clouds. Among those challenges, this paper mainly focuses on access control issues when EHRs are shared with various healthcare providers in cloud computing environments. The sharing process is complex and involves multiple entities. Due to the potential disclosure of medical records, patients' privacy concerns need to be considered in security and privacy mechanisms that should be well-integrated into healthcare systems and enforceable across a variety of heterogeneous systems in clouds, where patients fully lose control over their EHRs. In particular, a shared EHR instance may consist of sensitive healthcare information such as demographic details, allergy information, medical histories, laboratory test results, and so on. Access control solutions must be in place to guarantee that access to such sensitive information is limited only to those entities who have a legitimate need-to-know

978-1-936968-36-7 © 2012 ICST
DOI 10.4108/icst.collaboratecom.2012.250497

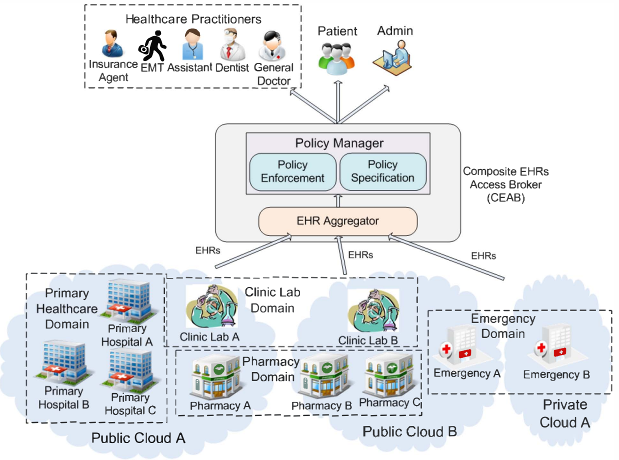

Fig. 1: Approach Overview

privilege authorized by patients. For example, a patient may not be willing to share his medical information on a particular disease with a dentist unless a specific treatment is required. Therefore, a systematic and flexible security mechanism is desirable to selectively share EHRs based on access control requirements.

In [11], [12], an access control mechanism was proposed to support patient-centric selective sharing of composite EHRs. However, this approach assumes that all healthcare providers adopt a unified EHR schema. Since different healthcare providers in clouds may utilize various EHR schemas to represent their healthcare data, such an assumption cannot be applicable in cloud environments. In this work, we attempt to overcome such a limitation by proposing a comprehensive access control mechanism to facilitate the selective sharing of composite EHRs from multiple healthcare providers in cloud computing environments. We present algorithms for EHRs data schema composition and cross-domain EHR aggregation. A proof-of-concept prototype system deployed in a cloud environment demonstrates the effectiveness and efficiency of our approach.

The rest of this paper is organized as follows. In Section II, we present our broker-based authorization approach which supports the selective sharing of composite EHRs in cloud computing environments. Section III discusses the system design of our prototype system with a case study. Section IV describes implementation details and system evaluation followed by the related work in Section V. We conclude the paper and discuss the future research directions in Section VI.

II. Broker-based Composite EHRs Authorization

In this section, we present a broker-based authorization approach to support selective sharing of EHRs, which manages each access to composite EHRs that are integrated from various healthcare providers in cloud computing. Fig. 1 shows an overview of our approach. Healthcare providers from various domains such as primary care, pharmacy, clinic lab, emergency care and so on host their EMR systems in clouds to achieve the features such as lower operation cost, higher interoperability, and ubiquitous service delivery. They can reside in a single cloud or multiple clouds (public cloud, private cloud, or hybrid cloud) depending on their deployment needs. The *Composite EHRs Access Broker (CEAB)* module consists of two sub-modules: the *EHR Aggregator* sub-module retrieves and aggregates distributed EHRs among clouds to construct virtual composite EHRs; and the *Policy Manager* sub-module supports the specification and enforcement of access control policies to regulate sharing of composite EHRs. Three types of stakeholders are involved: patients are the owners of EHRs who specify access control policies to control who can access which portions of EHRs. Healthcare practitioners access EHRs and are usually associated with specific healthcare providers. In addition, administrators perform administrative functions.

A. Logical EHR Model

A patient's EHRs are typically dispersed over a wide range of distributed EMR systems in clouds. Different EMR systems have different data schemas to manage logical and semantic relationships between data elements drawn from various medical domains. Such medical domains include patient demographics, labs, medications, encounters, imaging and pathology reports, and a variety of other medical domains from primary, specialty and acute care settings. To support the selective sharing of EHRs in clouds, we leverage a hierarchical structure proposed in our previous work [11], [12] to represent EHRs from various healthcare domains such as pharmacies, primary care, clinic labs, healthcare insurance and so on. Each node in the hierarchical structure is labeled and the root of the hierarchical structure represents a particular EHR instance. There are two types of nodes: *field node* and *group node*. Field nodes are leaves of the hierarchical structure which represent elementary information regarding the EHR. Related field nodes are placed to each other to form an information group node. For example, field node 'name', 'address', 'birthday' of a patient are very often grouped together to construct an information group node 'demographics'. Moreover, several related group nodes can form a super-group node (Note that a super-group node is still a group node). As an example, the group node 'demographics' of a patient is likely to be grouped together with other group nodes such as 'allergies' and 'drugs' to form a super-group node to represent an EHR object in pharmacy healthcare domain. Thus, this bottom-up characterization reflects the hierarchical nature of the logical EHR model. Formally, we give the definition of the logical EHR model as follows:

Definition 1: [**Logical EHR Model**] An EHR object is represented as a 3-tuple $T = (r, V, E)$, where

- r is the root of the whole EHR object;
- V is a set of nodes within the hierarchical structure of EHR object such that $V = V_f \cup V_g$ where V_f is a set of field nodes which are leaves in the hierarchical structure and V_g is a set of group nodes which are formed by a set of leaves or a set of other group nodes in the hierarchical structure.
- $E \subseteq V \times V$ is a set of links between nodes. $e_{ij} \in E$ represents the link between node $i \in V$ and node $j \in V$.

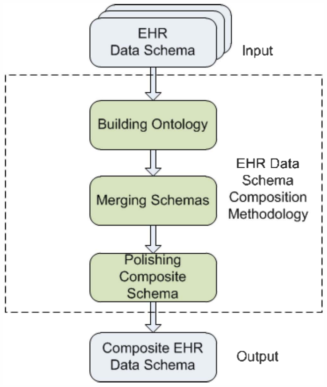

Fig. 2: EHR Data Schema Composition Approach

As an example, the EHR data schema represented in the logical EHR model for the pharmacy healthcare domain is shown in Fig. 3. The root node 'EHR instance' consists of three group nodes: 'Demographics', 'Allergies' and 'Drugs'. Group node 'Demographics' contains five field nodes including node 'Name', 'DoB', 'Age', 'Addr' and 'Gender' to describe demographic information in this EHR instance. Both group node 'Allergies' and 'Drugs' contain other group nodes as well as field nodes to describe medical information regarding allergies and drugs.

B. EHR Data Schema Composition

In this section, we discuss our approach for EHR data schema composition. We assume all source EHR data schemas to be integrated have already been represented in our defined logical EHR model. As shown in Fig. 2, the input of our approach includes multiple EHR data schemas from different healthcare domains such as pharmacy, primary care, clinic lab and so on. The output is the composite EHR data schema. There are three major steps such as building ontology, merging schemas, and polishing composite schema in our schema composition approach.

TABLE I: Node Ontology

Class Label	Class Nodes
Demographic	Demographic, Demo, Profile
Gender	Gender, Sex
DoB	DoB, Birthday, Birth Date
...	...

In the first step, we build a node ontology based on ISO EHR Standards [1] shown in Table I. More specifically, we identify all semantically equivalent nodes from various EHR data schemas using the approach introduced in [6], and then construct classes with ontology labels defined in the ISO EHR Standard. For example, 'Demographic', 'Demo' and 'Profile' represent three different nodes from schemas to be integrated but they are semantically equivalent. They are categorized into a class with a label of 'Demographic' since 'Demographic' is referred in the ISO EHR Standard. Some ontology tools such as Knoodl [13] and NeOn [16] can be utilized in this step to build the node ontology.

In the second step, we merge multiple EHR data schemas into a composite EHR data schema. The general merging process is pair-based: for a set of source EHR data schemas to be integrated, the first two EHR data schemas are merged first. Then, the intermediary composite EHR data schema generated by the first two schemas is further merged with the third EHR data schema. We continue this process until all EHR data schemas are processed.

Algorithm 1: *MergeTwo*(T_i, T_j) $\rightarrow T_c$

```
Input: Two EHR data schemas $T_i$, $T_j$
Output: A composition EHR data schema $T_c$
if $T_i$ and $T_j$ are empty schemas then
    return empty schema
else
    if $T_i$ is empty schema then
        return $T_j$;
    else
        if $T_j$ is empty schema then
            return $T_i$;
        else
            if depth($T_i$) != depth($T_j$) then
                $T_d \leftarrow$ largerDepth($T_i$, $T_j$);
                $T_s \leftarrow$ the rest schema;
            else
                if numberOfFieldNodes($T_i$) != numberOfFieldNodes($T_j$)
                then
                    $T_d \leftarrow$ moreFieldNodes($T_i$, $T_j$);
                    $T_s \leftarrow$ the rest schema schema;
                else
                    $T_d \leftarrow$ randomChoose($T_i$, $T_j$);
                    $T_s \leftarrow$ the rest schema;
                end
            end
            insertSubSchema($T_d$, $r_d$, $T_s$, $r_s$);
            return $T_d$;
        end
    end
end
/* Definition of insertSubSchema function:/
insertSubSchema(Schema $T_1$, Node $v_1$, Schema $T_2$, Node $v_2$)
begin
    if Scan $T_1$ from $v_1$ heading to bottom level,
    there exist node m such that m = $v_2$ then
        foreach $n \in getImmediateChild(v_2)$ do
            insertSubSchema($T_1, m, T_2, n$);
        end
    else
        Insert sub-schema rooted at $v_2$ in schema $T_2$ into $T_1$ rooted at $v_1$;
        return;
    end
end
```

The details of merging two EHR data schemas are shown in Algorithm 1. Two EHR data schemas are fed as an input and the output is a composite EHR data schema. The general idea is to insert sub-schemas of one schema into proper locations of the other schema. The sub-schema may consist of one or more nodes. If both schemas are empty, an empty schema is returned. If one of these two schemas is empty, the other schema is returned. The main body of the algorithm is executed when both schemas are not empty. In this case, we first need to choose one of the two schemas as a destination schema. This process is based on following three rules: (1) if the two schemas are of different depths, the schema with more levels is chosen as the destination schema; (2) for two schemas of the same depth, the one with more field nodes is chosen as the destination schema. (3) If the two schemas have the same numbers of depths and field nodes, we randomly pick one as the destination schema. The destination schema is denoted by T_d, and the other schema is the source schema denoted by

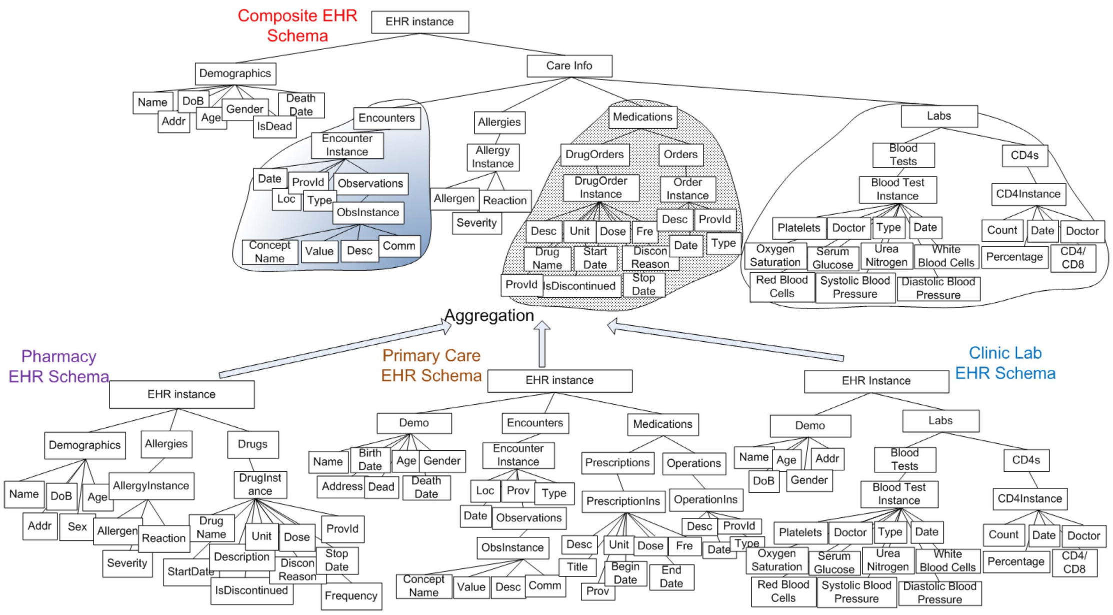

Fig. 3: EHR Data Schema Composition Example

T_s. Our algorithm works in a up-to-bottom fashion, starting from root to the bottom level of the schema. Sub-schemas in source schema T_s are recursively inserted to destination schema T_d rooted at a node $v \in V_{T_d}$, if the parent node of the sub-schema is equal to a node v and the node v does not have any immediate child node, which is equal to the root node of the sub-schema. Given two EHR data schemas to be merged, n times of insertion functions are invoked recursively for the worst case (n is the number of nodes in the source schema). For each insertion function, m times of node matching operations are conducted for the worst case (m is the number of nodes in the destination schema). Hence, the time complexity for Algorithm 1 is $O(n^2)$ for the worst case.

Consider Pharmacy EHR data schema and Primary Care EHR data schema are merged: Primary Care EHR data schema is chosen as the destination schema T_d since it has more depth and Pharmacy EHR data schema is the source schema T_s. Function *insertSubSchema* is invoked and those two EHR data schemas T_d and T_s as well as their root nodes r_d and r_s are passed as arguments. The core idea of function *insertSubSchema* is to recursively insert sub-schemas of T_s into proper locations of T_d. In the top level of recursion, it scans T_d from its root node to bottom level to check whether there exists a node m such that $m = v_2$ where v_2 is the root node of T_s. The root node of T_d is found as the node m (m can be considered as the upper boundary in T_d for the scanning step in each recursion) since both root nodes are represented using the same label 'EHR instance' and they are semantically equivalent to each other. Since the node v_2 which is the root node of T_s now has three immediate child nodes such as 'Demographics', 'Allergies' and 'Drugs' nodes, three *insertSubSchema* functions are invoked for each of those immediate child nodes and current argument m is still an 'EHR instance' node, which is the root node of T_d. In the recursion of 'Demographics' node in T_s, T_d is scanned from m to bottom. The 'Demo' node in T_d is found as m since 'Demo' and 'Demographics' nodes are semantically equivalent to each other based on the ontology shown in Table I. Then m in this recursion becomes a 'Demo' node in T_d. Since 'Demographics' node has five immediate child nodes, *insertSubSchema* function is invoked for each immediate child node. In the recursion of 'DoB' node in T_s, T_d is scanned from m which is 'Demo' node in T_d to bottom. The 'Birth Date' node in T_d is found as m since 'Birth Date' and 'DoB' nodes are semantically equivalent to each other based on the ontology shown in Table I as well. Similarly, other recursions are conducted. As shown in Fig. 3, based on above EHR data schema composition approach, three different EHR data schemas for pharmacy, primary care and clinic lab in clouds are integrated into a composite EHR data schema. After identifying all semantically equivalent nodes and building an ontology, pharmacy EHR data schema and primary care EHR data schema is first merged. The result EHR data schema is further merged with clinic lab EHR data schema. Primary care EHR data schema is chosen as the destination schema when merging the first two schemas. Sub-schemas of pharmacy EHR data schema are inserted into primary care EHR data schema.

C. Cross-domain EHR Instance Aggregation

Patients' EHR instances that carry actual medical information are organized and stored in distributed EMR systems based on their EHR data schemas. As shown in Figure 3, EMR systems from different healthcare sub-domains such as primary care, pharmacy and clinic lab adopt different EHR data schemas. To support selective EHR sharing for a patient, all related EHR instances residing in various EMR systems need to be aggregated into a composite EHR instance. Some of those EHR instances are based on the same EHR data schema if they come from the same healthcare sub-domain. Some of

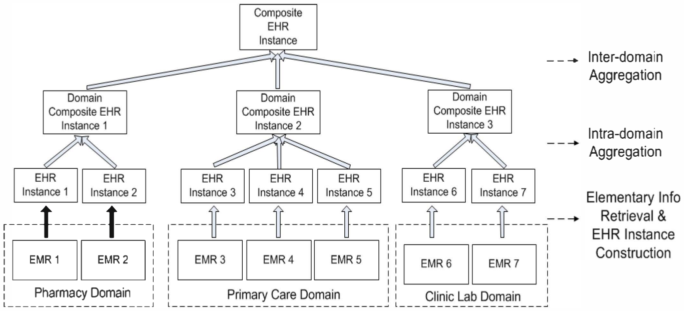

Fig. 4: EHR Instances Aggregation Procedure

them are based on different EHR data schemas if they are from different healthcare sub-domains. Hence, we propose a three-step cross-domain EHR instance aggregation approach: first, all elementary medical information from each EMR system are retrieved and corresponding EHR instances for each EMR system are constructed based on their domain EHR data schemas; second, intra-domain aggregation is performed. EHR data instances from the same healthcare domains are aggregated into domain EHR instances based on their common EHR data schemas; and lastly, inter-domain aggregation is conducted. All aggregated EHR instances across different health domains are aggregated into a composite EHR instance based on the composite EHR data schema. For example, as shown in Figure 4, a patient's EHRs are resided in seven EMR systems. EMR 1 and EMR 2 are within the same healthcare sub-domain *Pharmacy*; EMR 3, EMR 4 and EMR 5 are within the same healthcare sub-domain *Primary Care*; EMR 6 and EMR 7 are within the other healthcare sub-domain *Clinic Lab*. First, EHR Instance 1, EHR Instance 2 and so on are respectively retrieved and constructed from their corresponding EMR systems and based on their EHR data schemas. EHR Instance 1 and EHR Instance 2 are based the same EHR data schema since they are from the same healthcare sub-domain. They are integrated into the Domain Composite EHR instance 1 in the 2nd step. Similarly, Domain Composite EHR instance 2 and Domain Composite EHR instance 3 are generated. Finally, a composite EHR instance is generated from those three domain composite EHR instances by cross-domain EHR instance aggregation. The EHR data schema for the composite EHR instance is obtained by integrating EHR data schemas of those three healthcare sub-domains using the EHR data schema composition approach presented in Section II-B.

D. Access Control Policy Specification

To enable an authorized and selective sharing of patients' EHRs in clouds, it is critical for an authorization policy to determine a subject's access privileges for specific portion(s) of a composite EHR instance. Our policy specification scheme is built upon the defined logical EHR model such that access policies can be effectively defined at different granularity levels within the structure. To give a formal definition of an access control policy, we first define following concepts: *Subjects*, *Objects*, *Purposes.*

In healthcare domain, patients may give the access permission of their EHRs to identified individuals. For instance, a patient may want to indicate the following intent: "Dr. Bruce is allowed to access my medical records". In other situations, authorizations can be issued to a role such as 'dentist', 'general physician', 'pharmacist', and 'nurse'. As healthcare practitioners are usually associated with certain organizations, such a property may also be a constraint on the subject. We give the formal definition of subjects as follows:

Definition 2: [**Subject**] Let *U*, *R* and *O* be the sets of user IDs, roles and affiliated organizations. A subject *sub* is defined as a tuple *sub* = $< u, so >$ or *sub* = $< r, so >$, where $u \in U$, $r \in R$, and subjects' affiliated organization set $so \subseteq O$. Overall, the subject set *Sub* is defined as *Sub* = $(U \times 2^{O}) \bigcup (R \times 2^{O})$.

To support the selective sharing of EHRs, the definition of objects is based on the logical EHR model as follows:

Definition 3: [**Object**] Let *V* be a set of all nodes in a given EHR instance represented according to an EHR logical model denoted by *T*. An object obj_v where $v \in V$ is a set of nodes in a sub-schema of *T* rooted at node *v* such that the object set *Obj* is defined as *Obj* = 2^V.

To better protect a patient's privacy when sharing his medical information, an attribute, *purpose*, is specified in the authorization policy so that we can confine the intended purposes/reasons for data access in healthcare practice. Some examples of purpose are *payment*, *treatment*, *research*, and so on. Formally, the purpose is defined as follow:

Definition 4: [**Purpose**] Let *P* be a set of purposes for business practices in healthcare domains. The purpose *pur* is a sub set of *P*, $Pur \subseteq P$.

Definition 5: [**Access Control Policy**] An access control policy is a 4-tuple acp= (sub, obj, pur, effect), where

- $sub \in Sub$ is a subject;
- $obj \in Obj$ is an object;
- $pur \subseteq P$ is the purposes; and
- $effect \in \{permit, deny\}$ is the authorization effect of the policy.

Policies can be categorized into two types: local policy and global policy in terms of residencies of the policy. Local policies are enforced in a specific EMR system when it shares EHR instances with other systems. Global policies are enforced on the composite EHR instance in a centralized way. Both types of enforcement of polices support different granularity levels of EHRs' disclosures. Three global access control policy examples are given as follows:

- **P1**: (<GP, h1>, $obj_{Encounters}$, {treatment}, permit);
- **P2**: (<SP, h2>, $obj_{Medications}$, {treatment,research}, permit);
- **P3**: (<Dr.Lee, h2>, obj_{Labs}, {research}, deny);

In **P1**, a patient allows all general practitioners (GP) in hospital *h1* to view encounter information of his composite EHR shown in the shaded scope in Fig. 3 for treatment purpose; In **P2**, the patient allows all specialists (SP) in hospital *h2* to view medications information of his composite

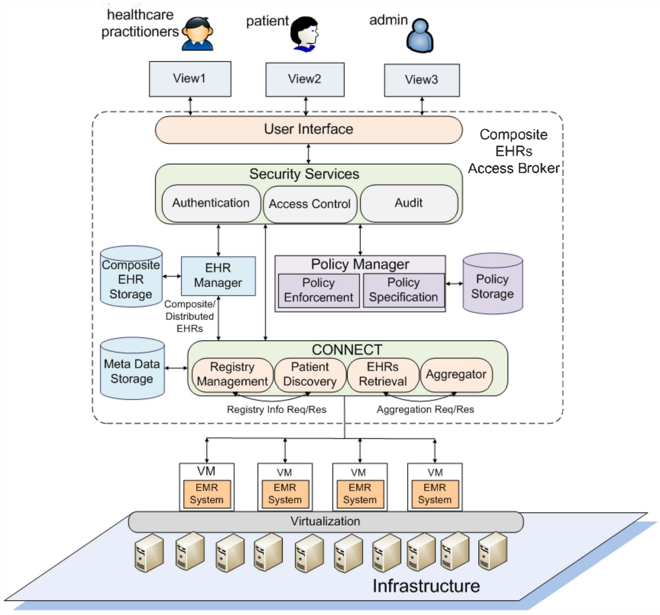

Fig. 5: System Architecture

EHR shown in the dotted scope in Fig. 3 for treatment or research purpose; and in **P3**, the patient disallows Dr. Lee from hospital *h2* to access his clinic lab information of his composite EHR shown in the non-shaded scope in Fig. 3 for research purpose.

III. System Design

A. System Architecture

Fig. 5 shows our system architecture. The bottom is an infrastructure layer which provides computing and storage capabilities to host various EMR systems. This can be achieved by several cloud computing software solutions such as XenServer [4], OpenStack [18], and Eucalyptus [7]. By leveraging the cloud infrastructure, healthcare providers can tremendously reduce their cost for building and maintaining their own data centers to host EMR systems. The middle box is the Composite EHRs Access Broker module including User Interface, Security Service module, EHR Manager module, Policy Manager module and CONNECT module. The User Interface has three different views according to users' identities: (1) healthcare practitioners are able to discover a patient with at least 3 characters of the patient's name. By selecting the desired patient, they can submit the patient's EHRs access request. Based on the authorization result, the request is either allowed or denied; (2) patients are able to view their EHRs from particular healthcare providers they are associated with or the composite EHRs aggregated from all healthcare providers they obtained services from. They can also specify policies for certain EMRs or on the composite EHRs; and (3) administrators have the capability to manage users and healthcare providers' EMR systems registered in the whole system. Security Service module consists of three sub-modules: Authentication sub-module authenticates users to make sure only legitimate users can access the system; Access Control sub-module controls users' access to EHRs from particular registered EMR systems or portions of the composite EHRs based on authorization results generated from Policy Manager; and Audit sub-module maintains all system logs. EHR Manager module retrieves distributed EHRs or the composite EHRs from CONNECT and share them with authorized users under the control of Access Control module. Policy Manager module consists of two sub-modules: Policy Specification sub-module provides capability for patients to specify their access control policies based on the scheme defined in Definition 5; and Policy Enforcement sub-module enforces corresponding policies when receiving EHRs access requests from users and generates authorization results to Access Control module. Access control policies are stored as records in a policy storage database. CONNECT module includes four sub-modules: Registry Management sub-module provides administrative functionalities such as adding, deleting, listing and updating on EMR systems hosted in cloud infrastructure; Patient Discovery sub-module enables healthcare practitioners to discover patients from all registered EMR systems and stores discovery results in a local database for caching; EHRs Retrieval sub-module retrieves all related distributed EHRs from registered EMR systems in clouds. EHR data schemas from various healthcare domains such as primary care, pharmacy and clinic lab are realized by this module. Elemental healthcare information is retrieved and constructed into EHR instances based on their EHR data schemas; and Aggregator sub-module integrates all distributed EHRs from EHRs Retrieval module to construct composite EHRs. The composite EHRs data schema and aggregation are performed as shown in Fig. 3 and Fig. 4, respectively.

This architecture is a realization of our broker-based EHRs sharing approach in cloud computing environments shown in Fig. 1. Each hosted EMR system in clouds contributes as a data source to construct composite EHRs for better healthcare service delivery. CONNECT module realizes the notion of EHR aggregator to interact with hosted EMR systems for EHRs retrieval and aggregation. Access Control module and Policy Manager ensure that only selective portions of composite EHRs are shared with healthcare practitioners who have need-to-know privileges authorized by patients.

B. Case Study

In this section, we discuss a case study to show how our approach supports the selective EHRs sharing of composite EHRs. Suppose Bob is a veteran who had a bullet wound in his abdomen during a battle before. He had a primary surgery in a VA hospital at that time. However, he did not be fully recovered due to severity of the wound. The bullet wound badly affects his pancreas system. Since then, he is suffered from diabetes and needs to periodically take prescribed medicines from a pharmacy. And he has inherited allergies to certain kinds of medicine. Hence, he has to take a special prescription from his primary doctor. Every three months, his homecare doctor needs to monitor the status of his pancreas system. One day, he had a heart attack at home and was sent to a nearby VA hospital where he usually obtains

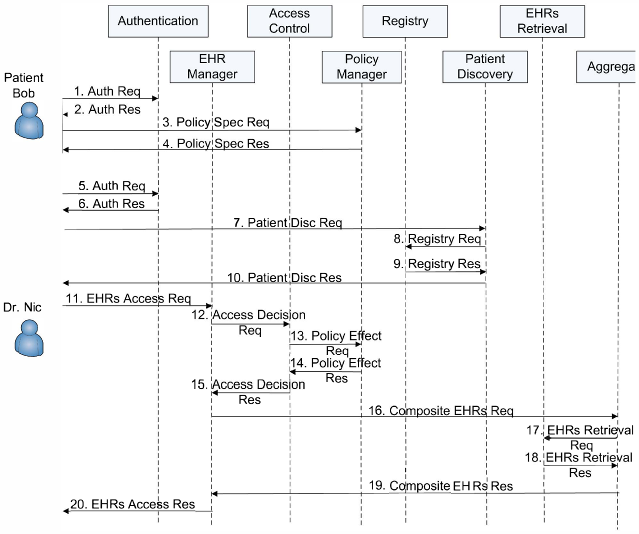

Fig. 6: System Workflow

care services by an ambulance. On the way to the VA hospital, the emergency medical technician (EMT) tried to access Bob' medical related information and carried out some emergency actions. The EMT also reported the information to the hospital. When they arrived the VA hospital, his primary doctor, Dr. Lee, had already collected all related medical information of Bob and prepared a preliminary plan.

This scenario involves four healthcare providers from different healthcare domains: primary care hospital, pharmacy, clinic lab and emergency. Each domain manages their EMR systems in clouds which store Bob's EHRs since Bob has obtained healthcare related services from them before. And their EHRs are organized and stored based on EHR data schemas. Two healthcare practitioners including Dr. Lee from the VA hospital and the EMT are also involved in the workflow. Depending on their different identities, they have different access privileges on Bob's composite EHRs. Fig. 6 illustrates how this scenario can be realized in the workflow of our system.

IV. Implementation and Evaluation

A. Implementation Details

To demonstrate the feasibility of our approach, we developed a secure selective EHRs sharing system in clouds based on our design discussed in Section III. Our cloud infrastructure environment is built using Citrix XenServer 6.0 and three Dell PowerEdge R510 rack servers with 16 cores, 30 GB RAM and 925 GB disk space for each one. We deployed OpenMRS 1.8.2 [17] as EMR systems into VMs running on the cloud infrastructure. The core EHRs aggregation and sharing logic were implemented using Java and the presentation layer was written in JavaSever Pages (JSP) technologies. We used MySQL Community Sever 5.5 for database sever. Four submodules, corresponding functionalities and related APIs of CONNECT module shown in Fig. 5 were implemented. Some of those APIs are developed with OpenMRS APIs. *Registry Management* module provides functionalities to register new EMR systems, update existing EMR systems with their IP addresses and domain types, list and delete EMR systems in clouds. *Patient Discovery* module queries each registered EMR system to discover patients with at least three characters of patients' names. *EHR Retrieval* module consists of eight sub-modules: *ConfigRetrieval* sub-module configures EHRs retrieval transactions with EMR systems. In particular, it sets up the target EMR systems and identity information including user name and password. It also manages session creation and termination with EMR systems; *RetrieveEHRInstance* sub-module constructs EHR instances based on healthcare domains they are associated with; and other six sub-modules retrieve healthcare information regarding patients' demographics, encounters, observations, allergies, medical orders and clinic lab results. *Aggregator* module conducts intra-domain EHR instance aggregation and inter-domain EHR instance aggregation. Our system also provides a web-based interface for three different kinds of users including administrators, patients and healthcare practitioners.

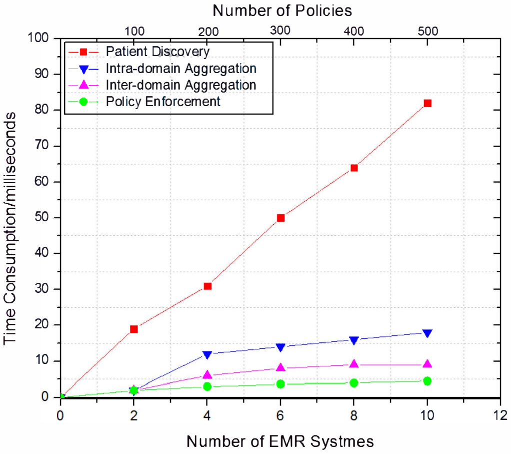

Fig. 7: System Time Overhead

B. Evaluation Results

We randomly deployed EMR systems into different VMs in our cloud environment based on the scenario mentioned in Section V. Those VMs have various configurations in terms of CPU speed, memory and disk size to simulate real-world healthcare domain. We create three types of VMs to satisfy the different resource needs of healthcare systems. The 'small', 'middle' and 'large' types of VM are respectively configured with different computing resources such as CPU, RAM and hard disk capactity. The healthcare datasets are obtained from OpenMRS software package. The management module in Fig. 5 has been deployed into a 'large' type of VM. Fig. 7 shows the time consumption for patient discovery, intra-domain EHRs aggregation, inter-domain EHRs aggregation and policy enforcement when the number of EMR systems increases. The upper line shows the time used for discovering patients is just about 78 milliseconds when the number of EMR systems is 10. The next two lower lines represent that the time consumptions of intra-domain and inter-domain aggregation respectively increase very smoothly as the number of EMR systems increases. The policy enforcement time increases even more smoothly than both intra-domain and inter-domain aggregation time increase. And when there are 500 policies in the system, the policy enforcement time takes about 5 milliseconds. Our experiments show that our aggregation process is reasonably efficient and scalable.

V. Related Work

In [11], [12], Jin et al. proposed a unified access control scheme which supports patient-centric selective sharing of virtual composite EHRs using different levels of granularity, accommodating data aggregation and various privacy protection requirements. However, this approach assumes that all healthcare providers adopt a unified EHR schema, which is not applicable in cloud environments. In contrast, our work supports EHRs aggregation from various healthcare providers considering different EHR data schemas in cloud environments. In [22], Zhang et al. identified a set of security requirements for eHealth application Clouds and proposed an EHR security reference model to support the sharing of EHRs. In [10], Jafari et al. proposed a patient-centric digital right management (DRM) approach to protect privacy of EHRs stored in clouds based on the patient preferences. However, those approaches are not fine-grained and cannot accommodate selective EHR sharing requirements. Al Kukhun et al. [3] examined mobile querying of distributed XML databases within a pervasive healthcare system. Whereas their approach is not cloud-based and does not consider the needs of EHR integration from different healthcare providers. In [14], Li et al. proposed a novel framework of access control to realize patient-centric privacy for personal health records in cloud computing by leveraging attribute based encryption (ABE) techniques. Their approach mainly ensures that EHRs are shared with a selective set of users. Our approach focuses on sharing selective portions of access control objects with authorized users.

VI. Conclusion and Future Work

In this paper, we have identified and articulated the selective EHRs sharing issue in healthcare cloud computing environments. To address this issue, a broker-based access control mechanism has been presented. We has also proposed an EHR data schema composition approach to generate composite EHR data schema. Based on this schema, distributed EHR instances from various healthcare domains can be aggregated into a composite EHR instance. By enforcing access control policies specified by patients, selective portions of the composite EHR instance are able to be shared with authorized healthcare practitioners. A proof-of-concept EHR sharing system has been implemented and evaluated to demonstrate the feasibility of our approach.

As part of our future work, we would conduct more comprehensive evaluations on our system with a real-world healthcare dataset. We would also investigate how to address policy composition issues [8], [9] and how to support fine-grained delegation mechanism for EHR sharing in cloud computing environments. In addition, we would apply our approach to support EHR sharing using consumer devices such as smart phone and tablet to cover the whole healthcare ecosystem.

Acknowledgments

This work was partially supported by the grants from National Science Foundation (NSF-IIS-0900970 and NSF-CNS-0831360) and Department of Energy (DE-SC0004308).

References

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

[2] G. Ahn, H. Hu, J. Lee, and Y. Meng. Representing and reasoning about web access control policies. In *Computer Software and Applications Conference (COMPSAC), 2010 IEEE 34th Annual*, pages 137–146. IEEE, 2010.

[3] D. Al Kukhun and F. Sedes. Adaptive solutions for access control within pervasive healthcare systems. *Smart Homes and Health Telematics*, pages 42–53, 2008.

[4] I. Citrix Systems. XenServer 6, 2011. http://www.citrix.com/English/ps2/products/product.asp?contentID=683148.

[5] C. DesRoches, E. Campbell, S. Rao, K. Donelan, T. Ferris, A. Jha, R. Kaushal, D. Levy, S. Rosenbaum, A. Shields, et al. Electronic health records in ambulatory carea national survey of physicians. *New England Journal of Medicine*, 359(1):50–60, 2008.

[6] E. Dragut, W. Wu, P. Sistla, C. Yu, and W. Meng. Merging source query interfaces on web databases. In *Proceedings of the 22nd International Conference on Data Engineering, 2006.*, pages 46–46. IEEE, 2006.

[7] I. Eucalyptus Systems. Eucalyptus Cloud Computing Software, 2011. http://www.eucalyptus.com/.

[8] H. Hu and G. Ahn. Enabling verification and conformance testing for access control model. In *Proceedings of the 13th ACM symposium on Access control models and technologies*, pages 195–204. ACM, 2008.

[9] H. Hu, G. Ahn, and K. Kulkarni. Anomaly discovery and resolution in web access control policies. In *Proceedings of the 16th ACM symposium on Access control models and technologies*, pages 165–174. ACM, 2011.

[10] M. Jafari, R. Safavi-Naini, and N. Sheppard. A rights management approach to protection of privacy in a cloud of electronic health records. In *Proceedings of the 11th annual ACM workshop on Digital rights management*, pages 23–30. ACM, 2011.

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

[12] J. Jin, G. Ahn, H. Hu, M. Covington, and X. Zhang. Patient-centric authorization framework for electronic healthcare services. *Computers & Security*, 30(2):116–127, 2011.

[13] Knoodl, 2012. http://www.knoodl.com.

[14] M. Li, S. Yu, K. Ren, and W. Lou. Securing personal health records in cloud computing: Patient-centric and fine-grained data access control in multi-owner settings. *Security and Privacy in Communication Networks*, pages 89–106, 2010.

[15] P. Mell and T. Grance. The nist definition of cloud computing (draft). *NIST special publication*, 800:145, 2011.

[16] NeOn, 2012. http://www.neon-toolkit.org.

[17] OpenMRS-Open source health IT for the planet, 2011. http://openmrs.org/.

[18] openstack.org. Open source software for building private and public clouds, 2011. http://openstack.org/.

[19] H. Takabi, J. Joshi, and G. Ahn. Security and privacy challenges in cloud computing environments. *Security & Privacy, IEEE*, 8(6):24–31, 2010.

[20] R. Wu, G. Ahn, H. Hu, and M. Singhal. Information flow control in cloud computing. In *Proceedings of the 6th International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pages 1–7. IEEE, 2010.

[21] R. Wu, G.-J. Ahn, and H. Hu. Towards hipaa-compliant healthcare systems. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, pages 593–602. ACM, 2012.

[22] R. Zhang and L. Liu. Security models and requirements for healthcare application clouds. In *Proceedings of 3rd IEEE International Conference on Cloud Computing*, pages 268–275. IEEE, 2010.