Delegated Authorization Framework for EHR Services Using Attribute-Based Encryption

Maithilee Joshi, Karuna P. Joshi, and Tim Finin

Abstract—Medical organizations find it challenging to adopt cloud-based Electronic Health Records (EHR) services due to the risk of data breaches and the resulting compromise of patient data. Existing authorization models follow a patient-centric approach for EHR management, where the responsibility of authorizing data access is handled at the patients end. This creates a significant overhead for the patient who must authorize every access of their health record. This is not practical given that multiple personnel are typically involved in providing care and that the patient may not always be in a state to provide this authorization. Hence there is a need to develop a proper authorization delegation mechanism for safe, secure and easy to use cloud-based EHR Service management. We present a novel, centralized, attribute-based authorization mechanism that uses Attribute Based Encryption (ABE) and allows for delegated secure access of patient records. This mechanism transfers the service management overhead from the patient to the medical organization and allows easy delegation of cloud-based EHRs access authority to medical providers.

Index Terms—Attribute based encryption (ABE), attribute based access control (ABAC), electronic health record (EHR), cloud storage, Semantic Web, access broker, knowledge graph (ontology), cloud computing

1 Introduction

An Electronic Health Record (EHR) is an electronic version of a patients health history that documents all the relevant clinical details over a period of time [1] and is maintained by healthcare providers. EHRs help organizations provide improved healthcare services by automating patient information access and management. In 2003 the U.S. Institute of Medicine published a consensus study report, Key Capabilities of an Electronic Health Record System [2], that defined EHR systems as including:

- longitudinal collection of electronic health information for and about persons, where health information is defined as information pertaining to the health of an individual or health care provided to an individual
- immediate electronic access to person—and population-level information by authorized, and only authorized, users
- provision of knowledge and decision-support that enhance the quality, safety, and efficiency of patient care; and
- support of efficient processes for health care delivery

With the broader adoption of Cloud computing, healthcare service providers are increasingly moving to Cloud based EHR services to manage their patient health records. These services are platform independent and provide aggregated patient information with robust data searching, retrieval, access and management functionality, and can also be accessed from any location in a cost effective manner. These EHR services are developed internally or purchased from vendors like CureMD,[1] Practice Fusion,[2] and Athenahealth.[3] However, maintaining electronic copies of patient health and history increases the possibility of attacks on patient data and information privacy [3]. Patient medical records contain highly sensitive personally identifiable information (PII) and so require very high level of security and privacy controls. EHR security requirements include managing the sets of access control permissions granted within an EHR and preventing unauthorized use of data, data loss, tampering and destruction [2].

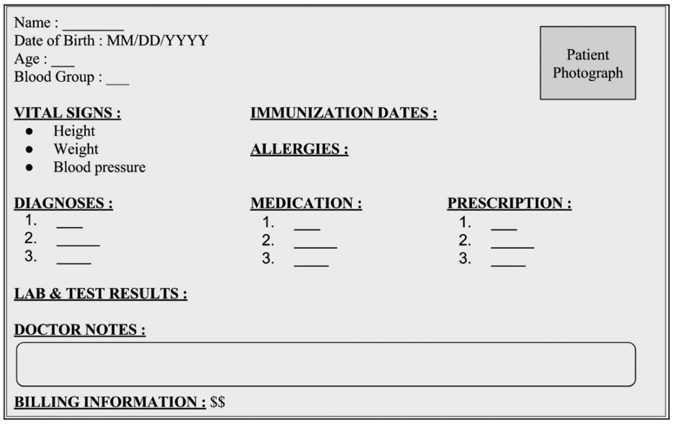
1.1 Motivation

EHR records patient's vital stats, diagnoses, medications, immunization history, laboratory and radiology reports, doctor notes and other medical facts along with patient's personal details. Based on the HL7 EHR Functional Model [2], we identified the key information fields in a typical EHR system which is illustrated in Fig. 1 and referenced in our system design. The Health Information Technology for Economic and Clinical Health (HITECH) Act [4] sets privacy standards that every medical provider should comply with while providing quality health services. The Health Insurance Portability and Accountability Act of 1996 (HIPAA) [5], [6] regulates the management and distribution of medical records by establishing standards for preserving the security and privacy of medical health data. Cloud based EHR services in the United States are required to comply with these regulatory standards

- *M. Joshi and T. Finin are with the Computer Science and Electrical Engineering Department, University of Maryland, Baltimore County, Baltimore, MD 21250 USA. E-mail: {maithi1, finin}@umbc.edu.*
- *K.P. Joshi with the Department of Information Systems, University of Maryland, Baltimore County, Baltimore, MD 21250 USA. E-mail: karuna.joshi@umbc.edu.*

Manuscript received 5 Aug. 2018; revised 15 Feb. 2019; accepted 10 May 2019. Date of publication 20 May 2019; date of current version 9 Dec. 2021. (Corresponding author: Maithilee Joshi.)
Digital Object Identifier no. 10.1109/TSC.2019.2917438

[1] www.curemd.com
[2] www.practicefusion.com
[3] www.athenahealth.com



Name : ________
Date of Birth : MM/DD/YYYY
Age : ___
Blood Group : ___

Patient Photograph

VITAL SIGNS :

- Height
- Weight
- Blood pressure

IMMUNIZATION DATES :

ALLERGIES :

DIAGNOSES :
1. ___
2. ____
3. ____

MEDICATION :
1. ___
2. ____
3. ____

PRESCRIPTION :
1. ___
2. ____
3. ____

LAB & TEST RESULTS :

DOCTOR NOTES :

BILLING INFORMATION : $$

Fig. 1. An example of the simple Electronic Healthcare Record system interface that contains electronically-stored patient information used in developing our system.

and so must ensure enhanced data protection combined with a seamless user experience that cloud services offer. This also requires that they implement strict access control mechanisms to ensure unauthorized access by any user is prohibited by their EHR service. Hence EHR systems often encrypt their dataset and have access restricted to only the caregivers directly treating the patient.

There are often scenarios, as when the patient's health suddenly deteriorates, that require records be made available to specialists (who could be remote) or other care givers who might not have initial access to the patient's health records. Existing authorization models follow a patient-centric approach where the EHR data authorization must be approved by the patient. This is not practical in every scenario and moreover the patient may not be in a state to provide this authorization when required. Hence there is a need to develop a authorization delegation mechanism where by the patient authorizes the provider, access to his/her EHR and the provider in turn delegates this authorization to appropriate employees or collaborators to access the data.

Traditional role based access models will not work as the cloud based EHR systems can be accessed from any location and from any device. So in addition to the care giver's roles, their other attributes, like location, time, duty period, etc., can also influence the delegated authorization. We have developed a novel, centralized, attribute based authorization mechanism for EHR Services that uses Attribute Based Encryption (ABE) and allows for delegated secure access of patient records. This mechanism transfers the service management overhead from the patient to the medical organization and allows easy delegation of cloud-based EHRs access authority to medical providers.

We present a comprehensive study of currently available EHR management systems and ongoing research on enhancing their information security and privacy. Our own research on this involves the combination of using Semantic Web technologies with attribute based schemes. We designed and developed a comprehensive knowledge graph ontology that can represent the entities or stakeholders of a medical organization and its patients. The ontology also represents the different EHR fields and their respective attributes as well as the various relationships between different entities in the organization. Using the attributes in this ontology, we developed a strong attribute based access control mechanism that extracts attributes from the EHR Ontology and applies policy rules to determine access permissions. Our access policy rules are based on the HIPAA policy for medical information storage and management. To further guarantee strong levels of data security, we implemented an attribute based encryption mechanism using the attributes represented in the *EHR Ontology*. We developed our prototype as an open-source, web-based application called, *EHR Manager*, that is designed for medical organizations desiring a cloud-based EHR that can guarantee strong data protection at a reasonable cost. This research also contributes towards open-source development of service-oriented cloud-based EHR, where each module independently performs its operation and supports the re-use of sub-modules.

The rest of this paper is organized as follows. Section 2 describes the related work in this area. Section 3 provides the system overview. Section 4 describes the architecture design. Section 5 describes the Access Broker and Section 6 describes the Encryption Unit in further details, followed by Section 7 which explains all the details about EHR Manager. Section 8 concludes by describing the future scope of this project and the overall conclusions of this research effort.

2 Related Work

There has been an increased adoption of cloud-based EHR services for efficient health data management and control [7], [8]. This can be attributed to the elasticity, high level of availability, and reduced cost of cloud services. Currently, there are a number of cloud-based EHR services, including CureMD.[4] Practice Fusion[5] and Athenahealth.[6] Organizations like GE Healthcare[7] and Epic Health Services[8] are also investing in cloud-based EHR services. Various research efforts have been proposed with major focus on secure, cloud-based EHR systems [7], [8]. Other researchers have also proposed trusted computing using SGX processors for Cloud security [9], [10].

However, the majority of the proposed approaches are deficient in guaranteeing a comprehensive access control and encryption mechanism. Along with this, most available applications are licensed and thus expensive to adopt. In this scenario, an open-source, low cost EHR managing application needs to be developed that can guarantee sophisticated levels of data privacy and protection. Through the *EHR Manager* application, this research effort tries to build such a solution by using all open-source development tools apart from the third party cloud services. The *EHR Manager* is an open-source tool which provides an easy interface for medical staff as well as patients to view and/or edit the EHR. Very intuitive, this application guarantees strong access control and data protection mechanism.

[4] http://www.curemd.com
[5] http://www.practicefusion.com
[6] http://www.athenahealth.com
[7] http://www.gehealthcare.com
[8] http://www.epichealthservices.com

2.1 Automating Electronic Health Records

Automating medical health record management systems has been the focus of much past research [11], [12], [13], [14]. The privacy and security of the patient health record being of utmost importance, this field of research has seen various approaches being suggested [12], [13], [15]. The Health Insurance Portability and Accountability Act of 1996del (HIPAA) provides data privacy, security and safeguarding acts for protecting electronic medical information of individuals [5], [6]. HIPAA provides guidelines for electronic medical record management for balancing individual privacy with respect to medical records along with the need to protect health of the masses. The Health Information Technology for Economic and Clinical Health Act aims towards maintaining electronic medical records by ensuring quality, safety, efficiency, privacy and security [4]. Complying with the legalities of managing medical records and at the same time developing an easy-to-use electronic health record system becomes a major research and development challenge. There exist many EHR management tools like GE Healthcare, Epic, CureMD etc. which provide EHR services using cloud computing.

2.2 Previous Attribute Based Access Control Work

Previously, we developed a semantically rich access control model based on Attribute Based Access Control (ABAC) [16]. This model evaluated an access decision based on the attributes of the user requesting a document and those of the requested document. We designed and implemented an ontology to demonstrate the use of ABAC in an organizational setting. Access control decisions were evaluated against an organizational confidentiality policy. This work demonstrated the use of policy-based, Semantic Web approach of implementing ABAC at a document level. The system has been improved to evaluate an access decision on the fields of a document rather than the entire document. The improved system can now categorize the permitted access instead of just a binary decision. Apart from this, the previously developed system demonstrated the concept of *edge computing* [17] where the organizational boundary was considered to be the edge of the system. The cloud service provider was considered as an untrusted entity and thus lied beneath the organizational edge. All data transactions between the organizational edge and the cloud were encrypted using an Oblivious Storage and the Oblivious RAM (ORAM) [18], [19]], obscuring the access patterns between the organization and the cloud service provider.

2.3 Access Control Mechanisms

Various access control models have been proposed, including Mandatory Access Control (MAC), Role Based Access Control (RBAC) [20], and many others. Jin et al. defined the Attribute Based Access Control model, which supports features of the pre-existing access control models [21]. Modeling access control policies has been a topic of interest. XACML is a policy model, based on the XML specification language [22] which uses attributes to impose access control. The Rei policy language [23] is based on deontic concepts and uses N3 rules and CWM for reasoning. ROWLBAC [24] and KAoS are also based on OWL [25]. For representing policies and rules formally, The Web Ontology Language (OWL) [23], [26], [27] serves to be very efficient while representing security policies. Complex ontologies can be effectively represented by using OWL. OWL representation of ABAC policies have been presented in [27]. In this work, basic constructs like User, Subject, Object, Permission, are defined as OWL classes. The User Attribute, Subject Attribute and Object Attribute are defined using OWL properties.

2.4 Attribute Based Encryption

To protect data privacy and threats, various encryption models have been proposed. Attribute Based Encryption is one approach where a user's ciphertext, secret key and private key are associated with her attributes [28], [29], [30]. Goyal et al. proposed an attribute based system called the Key-Policy Attribute Based Encryption (KP-ABE) [28] in which ciphertexts are tagged with attributes corresponding to access control structures. Their model supports Hierarchical Identity-Based Encryption (HIBE). Bethencourt et al. have developed a system called the Ciphertext-Policy Attribute Based Encryption (CPABE) for implementing ABE using the attributes of the user encrypting the document [29]. The *EHR Manager* uses the CPABE toolkit to prototype the research effort.

ABE has been one of chosen technologies for electronic health record management systems too [31], [32], [33]. Akinyele et al. have presented a design and implementation of Electronic Medical Records (EMRs) using attribute based encryption on mobile devices [31]. In their system, they provide off-line support for updating the medical records with support for eventual consistency. However, their model does not support a field-level encryption of the EHR. Researchers at Microsoft developed a patient controlled electronic medical record system with attribute based encryption [32]. As the name suggests, this system put all the access control in the patients hands. The control and distribution of access keys was the patients responsibility. However, this approach requires a high level of control overhead on the patients end. The *EHR Manager* however, does not impose any overhead on the patient. The central system handles all the secure access and distribution of the EHR.

2.5 Semantic Web Technologies

We have used Semantic Web technologies to develop the EHR ontology, the reasoning component of our system and for prototype development. These enable us to build the schema using W3C standardized languages that support our design requirements, which include interoperability, sound semantics, Web integration, and availability of tools and system components. Semantic Web tools enable data to be annotated with machine understandable meta-data, allowing the automation of their retrieval and their usage in correct contexts. Semantic Web technologies include languages such as Resource Description Framework (RDF) [34] and Web Ontology Language [26] for defining ontologies and describing meta-data using these ontologies as well as tools for reasoning over these descriptions. OWL Semantic Web knowledge can also be encoded in rule format using several approaches, including N3-logic rules [35], SWRL rules [36] and RIF, the new W3C standard for Rule Inter-change Formalism. These technologies can be used to provide common semantics of service information and policies enabling all agents who understand basic Semantic Web technologies to communicate and use each others data and Services effectively.

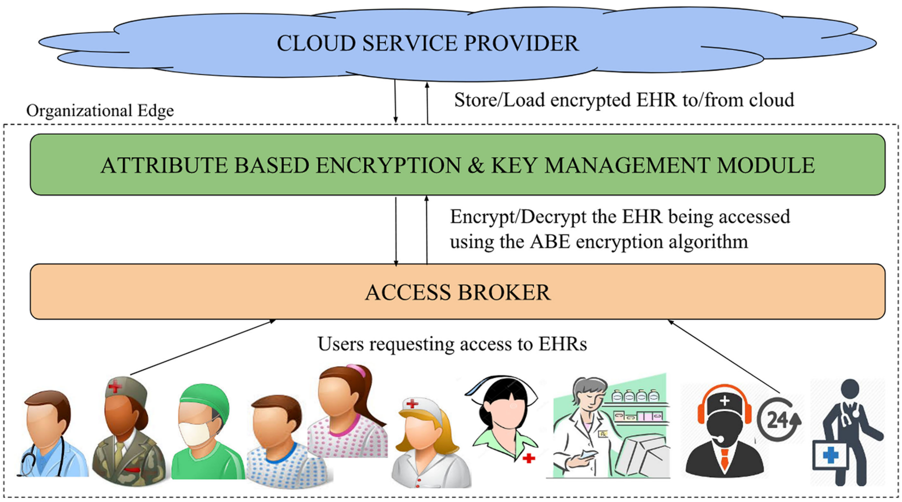

CLOUD SERVICE PROVIDER
Store/Load encrypted EHR to/from cloud
Organizational Edge
ATTRIBUTE BASED ENCRYPTION & KEY MANAGEMENT MODULE
Encrypt/Decrypt the EHR being accessed using the ABE encryption algorithm
ACCESS BROKER
Users requesting access to EHRs
24

Fig. 2. Our System is composed of four levels.

Our most fundamental requirement is for a representation that supports interoperability at both the syntactic and semantic levels. OWL has a well-defined semantics grounded in first order logic and model theory, allowing programs to draw inferences with the assurance that the subsequent interpretation is sound. An important advantage for OWL over many other knowledge-representation systems is that it has well defined subset profiles guaranteeing sound and complete reasoning with various levels of reasoning complexity and designed to work with popular implementation technologies, such as OWL QL for databases and OWL RL for rule-based systems.

A second design requirement is for a language that is designed to integrate well with the Web, which has become the dominant technology for today's distributed information systems. OWL is built on basic Web standards and protocols and is evolving to remain compatible with them. It is possible to embed RDF and OWL knowledge in HTML pages and several search engines (including Google) will find and process some embedded RDF. RDF is also compatible with Microdata, a Web Hypertext Application Technology Working Group HTML specification that is used to nest semantic statements within existing content on web pages. Microdata has been adopted by Schema.org, a collaboration of the major Web search companies and has been used to define a number of basic ontologies that are being supported by search engines.

3 System Overview and Overall Design

One of our primary objectives is to develop a highly secure, attribute based access mechanism for a Cloud based EHR service that will provide flexibility of data access to end users along with a sophisticated data encryption scheme. Using Semantic Web technologies, like OWL and SWRL, along with Attribute Based Encryption techniques, we were able to build an EHR service that allows easy data sharing and distribution in a highly secure fashion. We currently host this service using the Amazon AWS instance and are developing a version on OpenStack to allow us to compare the performance on the two platforms.

The HL7 EHR functional model [2] specifies that applications must adhere to the rules or policies established to control access and protect the privacy of EHR information. Security measures assist in preventing unauthorized use of data and protect against loss, tampering and destruction. The main security functions include user or entity (such as another application) authentication, authorization, access control, patient access management, non-repudiation, secure data exchange, secure data routing, information attestation, and patient privacy and confidentiality. We have referenced this functional model in our design and collaborated with our colleague, Dr. Eliot Siegel, who is Professor and Vice Chair at the University of Maryland School of Medicine, Department of Diagnostic Radiology, as well as Chief of Radiology and Nuclear Medicine for the Veterans Affairs Maryland Healthcare System, to understand how EHR systems are used by caregivers in an hospital. His insight helped us in designing the process flow of our system.

In our system, we began by concentrating on implementing a policy defined attribute-based access control component of the EHR system and designed a simple user-id/password based authentication scheme. Our system provides access to all stakeholders including different caregivers and patients. The system does not currently support EHR data exchange and routing, which is part of our planned future work. Fig. 2 shows an overview of our EHR system, which is divided into four levels. Level 1 is where users request access to an EHR of interest. At level 2, users are authenticated and requested actions evaluated with respect to access rules, policies, user attributes and EHR attributes. If the action is permitted, any required updates to the EHR are made at Level 3, and these updates are encrypted using the attributes of the user and the concerned EHR. Finally, at Level 4, is the cloud service provider where the data is to be sent and stored. Levels 1 to 3 lie inside the organizational edge and Level 4 lies outside. All entities outside the edge are considered to be untrusted.

As shown in the figure, there are multiple stakeholders of this system, including doctors with different specializations, nurses, emergency service personnel, pharmacists and patients. Each entity of this system has to go through a screening process through the *Access Broker*, an access control module that uses Attribute Based Access Control to control the type and amount of access to patient EHRs. On receiving

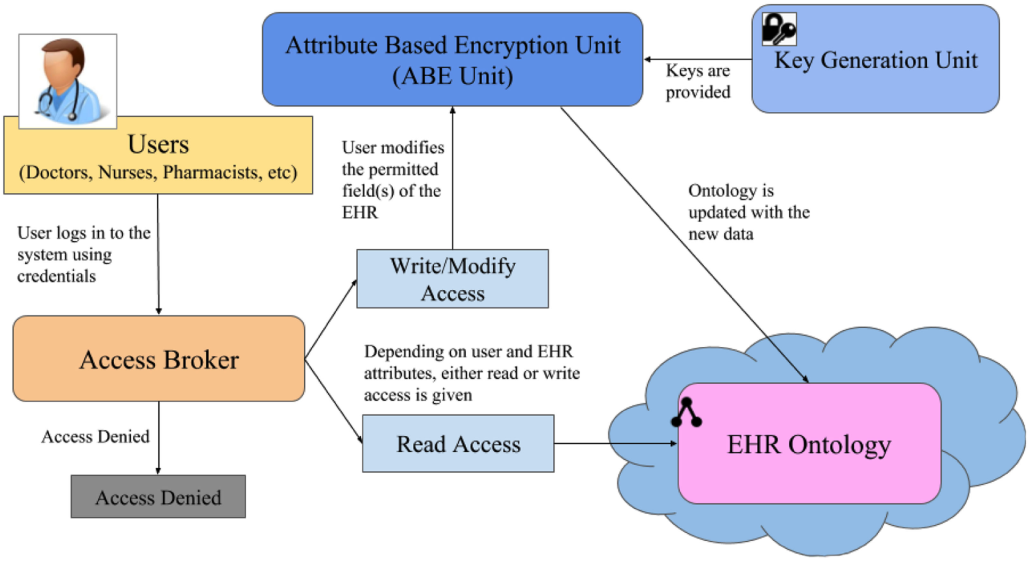

Attribute Based Encryption Unit
(ABE Unit)
Keys are provided
Key Generation Unit
Users
(Doctors, Nurses, Pharmacists, etc)
User modifies the permitted field(s) of the EHR
Ontology is updated with the new data
User logs in to the system using credentials
Write/Modify Access
Access Broker
Depending on user and EHR attributes, either read or write access is given
Access Denied
Read Access
EHR Ontology
Access Denied

Fig. 3. System architecture.

an affirmative response from the *Access Broker*, the user request to access an EHR field is passed to the *Encryption Unit* where the modified EHR field data (if any) is re-encrypted and stored in the system securely. The cloud service provider acts like a data storage center for storing the *Organizational Knowledge Base*, which details the relationships between different entities in the medical organization ecosystem. This knowledge base is represented as a knowledge graph, supported by a semantically-rich ontology represented in OWL.

4 Architecture Design

The system architecture shown in Fig. 3 consists of four main modules: Access Broker, Encryption Unit, Key Generation Unit and EHR Ontology. The data flow in the system is as follows. Medical organization users first login to the system using their credentials and the system carries out a comprehensive access control check to authenticate the user via the *Access Broker*. Our earlier design [16] used Attribute Based Access Control to carry out a strong access control mechanism with an organization-specific confidentiality policy and render a boolean decision. We have enhanced it to further categorize the access decision (as described in Section 5) to also determine the type of access permissions, e.g., read, write or modify.

The system next waits for the user to access the EHR. Once done, it then needs to encrypt the updated details of the accessed EHR fields, which is done by the *Encryption Unit*. This unit uses Attribute Based Encryption for encrypting the EHR field. It extracts the users attributes from the main ontology which is stored with a public cloud service provider, in our case Amazon Web Services. Using these attributes, the EHR field is encrypted where user attributes serve as their private key for the EHR field. This key generation is done by the *Key Generation Unit*, which uses the keys provided by the *Encryption Unit* to encrypt the EHR. Section 6 will describe further details. The encrypted text then needs to be updated in the *EHR Ontology*. To do this, a new node is created which records all the details of a patients visit to the medical organization. Maintaining visits as a node in the ontology enables easy querying and data recording. Finally, this ontology is saved with a cloud service provider. Following is a mathematical representation of the system implementation.

$\mathit{User\ set}\ U = \{U_1, U_2, ...U_n\}$
$\mathit{User\ Attribute\ Set}\ US = \{UA_1, UA_2, UA_3, ...UA_n\}$
$\mathit{EHR\ set}\ E = \{E_1, E_2, ...E_n\}$
$\mathit{EHR\ attribute\ set}\ ES = \{EA_1, EA_2, EA_3, ...EA_n\}$
$\mathit{EHR\ Fields\ Set}\ EF = \{EF_1, EF_2, ...EF_n\}$
$\mathit{EHR\ Fields\ Subset}\ EFS \subset EF$
$\mathit{Policy\ set}\ PS = \{PS_1, PS_2, ...PS_n\}$
$\mathit{Decryption\ Policy\ set}\ DS = \{DS_1, DS_2, ...DS_n\}$
$\forall\ \mathit{User}\ U, \exists\ \mathit{User\ Attribute\ Set}\ US$
For evaluating access decision
$\mathit{For\ each\ User}\ X \wedge \mathit{EHR}\ Y \wedge \mathit{EHR\ Fields\ Set}\ Z,$
$\mathit{If}\ US\ \mathit{satisfies\ any\ one\ from\ policy\ from}\ PS \rightarrow$
$\mathit{Read\_and\_or\_Write}\ (\mathit{User}\ X, \mathit{EHR}\ Y, EFS)$
For encryption using ABE
$\mathit{For\ each\ User}\ X \wedge \mathit{EHR}\ Y, \exists\ \mathit{Fields\ Subset}\ Z,$
$X \wedge Y \wedge \mathit{User\ Attribute\ Set}\ US \wedge Z \rightarrow \mathit{Encrypted\ EHR\ field}$
$\mathit{where}\ US \subset DS$
For decryption using ABE
$\mathit{If\ User\ Attribute\ Set}\ US \subset DS$
$US \wedge EF \rightarrow \mathit{Decrypted}\ EFS$

5 Access Broker

The *Access Broker* uses concepts from Attribute Based Access Control to manage and enforce access control, guaranteeing the right authorization access to only the authenticated users. Using Semantic Web techniques, this module extracts the user and EHR field attributes from the knowledge base, feeds them to the reasoner and thus regulates the access permissions. Unlike traditional RBAC mechanisms, the *Access Broker*, can regulate access down to the field level of the EHR as it references users' attributes and not just their role.

Fig. 4 shows the architectural view of the *Access Broker* which consists of three main sub-modules: the *Organizational Knowledge Base*, the *Rule Based Engine* and the *Policy*

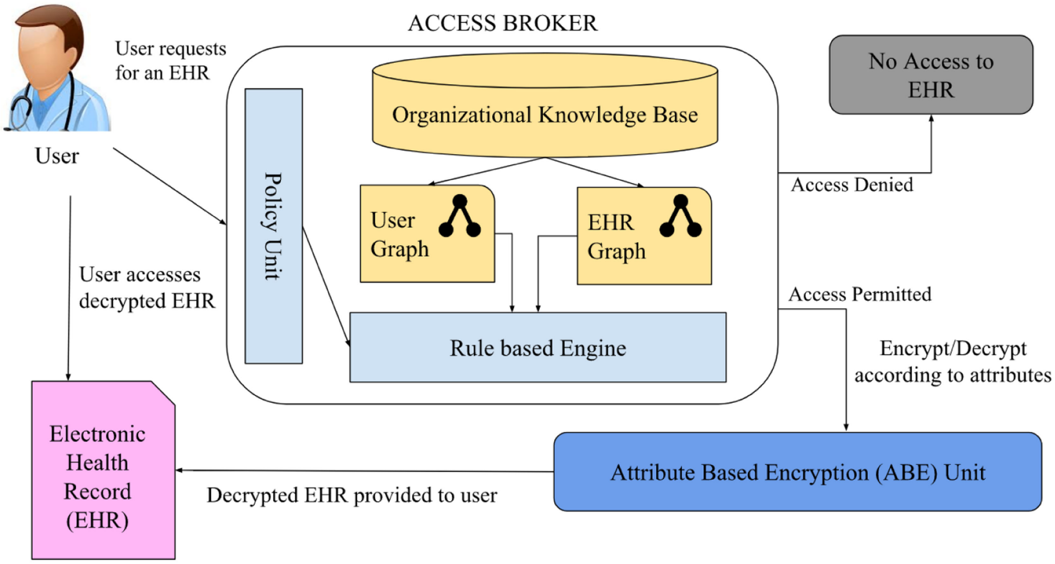

ACCESS BROKER
User requests for an EHR
User
User accesses decrypted EHR
Policy Unit
Organizational Knowledge Base
User Graph
EHR Graph
Rule based Engine
No Access to EHR
Access Denied
Access Permitted
Encrypt/Decrypt according to attributes
Attribute Based Encryption (ABE) Unit
Electronic Health Record (EHR)
Decrypted EHR provided to user

Fig. 4. Access broker architecture.

Unit. We discuss each sub-module of the *Access Broker* in detail below.

5.1 Organizational Knowledge Base

The *Organizational Knowledge Base* stores information about every entity belonging to the medical organization in a knowledge graph including both the ontology schema and rules and data encoded using them. The graph captures the roles and attributes of the different stakeholders of the medical organization along with the various relationships between them. We have designed and created the ontology by referencing our earlier HIPAA ontology [37], the medical standards specified by National Healthcareer Association, HealthIT.gov and National Institutes of Health. The *Organizational Knowledge Base* is critical in delivering correct attributes for the *Access Broker* and the *ABE Unit* to run. The role of this unit inside the *Access Broker* is to successfully deliver correct attributes of the entities (users and EHR fields) and accurately reflect the changes made by the medical staff in the patient's EHR field. Fig. 5 shows a snapshot of the ontology with its core classes and their properties.

We host the EHR Ontology with a third-party cloud service provider, in this case the Amazon Web Services (AWS) cloud platform. Our statistical analysis results proved that hosting the ontology on cloud platform reduced the performance times by a considerable amount. Section 8 shows the results of the test performed.

5.2 Policy Unit

Every organization has its own set of rules for document access. These set of rules comprise the confidentiality policy of the organization. In medical scenarios, the common policies, apart from the organization specific ones are the rules and standards set by the HIPAA Act and the HiTech Act.

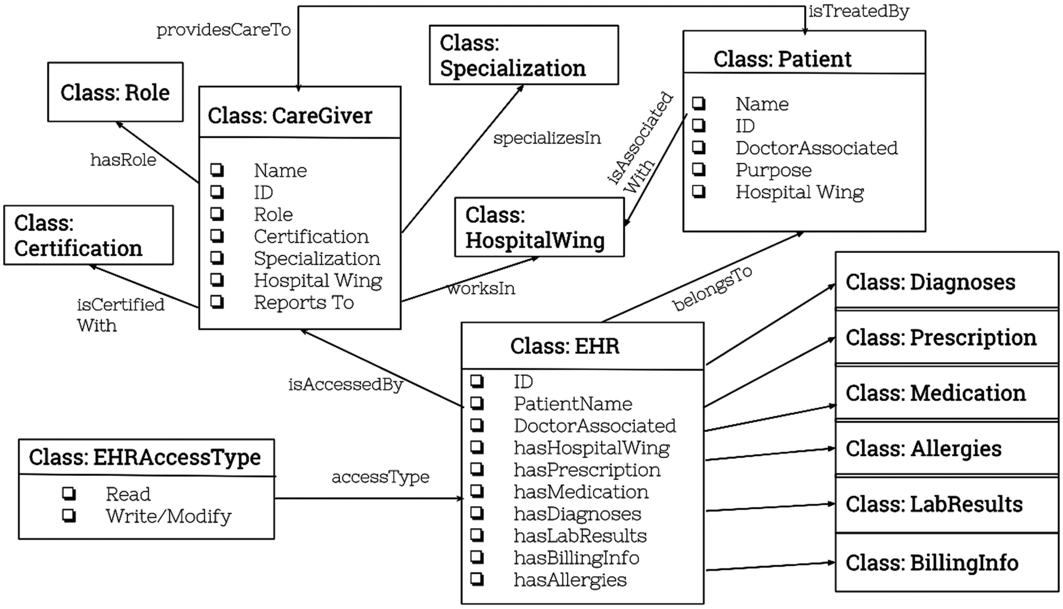


Fig. 5. Snapshot of EHR ontology.

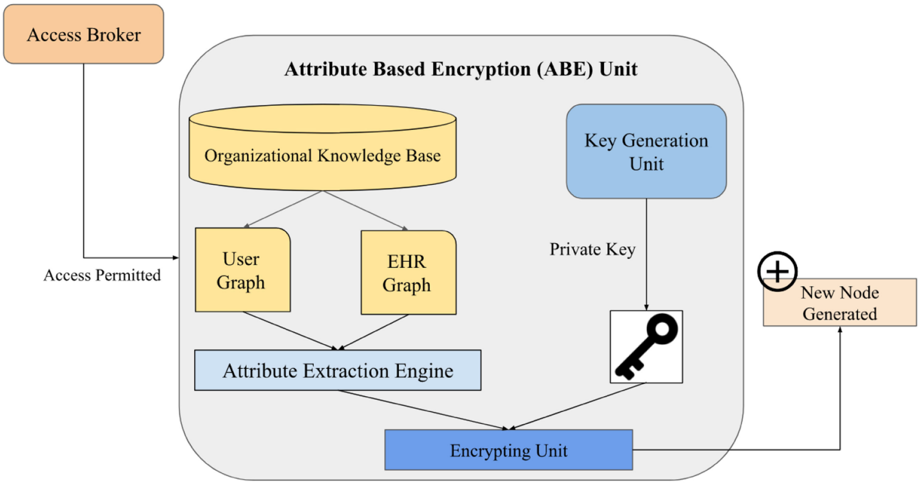


Fig. 6. Encryption unit architecture.

The *Policy Unit* stores all these access policies which are crucial in determining the access permissions. In terms of where the *Policy Unit* fits inside the *Access Broker*—this module provides content to the SWRL rules. Meaning, the *Rule Based Engine* takes in a policy from the *Policy Unit*, converts it to a SWRL rule and then further determines access permissions. For implementation and prototyping purposes, we have used the HIPAA policies, as the policies that determine access control over patient EHRs.

5.3 Rule Based Engine

The *Rule Based Engine* uses the Semantic Web Rule Language (SWRL) to use the confidentiality policies for implementing access control decisions. The *Rule Based Engine* requires user and document attributes from the ontology for carrying out access control decisions. Running these rules results in an access decision. Here, the *Access Broker* has been modified to categorize the access decisions as either read or write. Also, instead of evaluating the access decision for an entire document, the modified *Access Broker* evaluates access decisions at a field-level. This means, a user may or may not be granted access to the entire EHR but may be granted access to specific fields depending on attributes only.

```
SeniorDoctor(?se) ^ worksIn(?se,Maternity)^
   specializesIn(?se,Gynaecologist)^
   isCertifiedBy(?se,GYN)^Patient(?p)^
   providesTreatment(?se,?p)^EHR(Medication)^
   EHR(Prescription)^EHR(VitalStats)^
   EHR(Diagnoses)^EHR(Allergies)^
   EHR(DoctorNotes)^EHR(LabResults)->
   canModifyLabResults(?se, true)^
   canModifyPrescription(?se,true)^
   canReadVitalStats(?se, true)^
   canModifyMedication(?se, true)^
   canModifyAllergies(?se,true)^
   canModifyDiagnoses(?se, true)
```

The SWRL rule below shows how a users access request to certain fields of the EHR is evaluated using conditions on a *Senior Doctor* with attributes like specialization, the hospital wing where the *Senior Doctor* works, and the certification with which the *Senior Doctor* is decorated. With these attributes and the hospital policy, the *Senior Doctor* is permitted to access only a subset of the EHR fields like *Lab Results, Prescription, Vital Stats, Medication, Allergies* and *Diagnoses*.

The following rule shows an example rule where a *Junior Doctor* with certain attributes can access only those fields to which a *Senior Doctor* to whom this *Junior Doctor* reports has access to.

```
JuniorDoctor(?jd)^HospitalWard(?hw)^
   SeniorDoctor(?sd)^Certification(?c)^
   EHR(Medication)^EHR(Diagnoses)^
   EHR(Allergies)^worksIn(?sd, ?hw)^
   worksIn(?jd, ?hw) ^isCertifiedBy(?jd, ?c)^
   reportsTo(?jd,?sd)^canModifyMedication
   (?sd, true) ^canModifyAllergies(?sd, true)^
   canModifyDiagnoses(?sd, true) ->
   canReadMedication(?jd, true)^
   canReadDiagnoses(?jd, true)^
   canModifyAllergies(?jd, true)
```

The rule engine extracts user and EHR field attributes from the EHR Ontology by querying it. Next, it feeds these extracted attributes to the SWRL rules and eventually delivers an access decision.

6 Encryption Unit

The *Encryption Unit* is the most crucial elements of the entire system architecture. This module is responsible for protecting the EHR field data against any data leaks and threats. This module uses Attribute Based Encryption to perform the data protection activity. Using the attributes from the EHR Ontology, this module applies the necessary attributes that would satisfy the decryption policy with which the document has been encrypted. In simple words, any document that is to be encrypted using ABE, is associated with a particular, unique decryption policy which is a logical expression of different attributes involved in the organizational setting. In other words, the user attributes serve as encryption/decryption keys for document protection. The *EHR Manager* uses ABE at a field level instead of the traditional approach of using ABE at a document level.

Fig. 6 shows the architecture of the *ABE Unit*. The *ABE unit* consists of four sub-modules namely the *Organizational Knowledge Base*, the *Attribute Extraction Unit*, the *Key Generation*

Unit and lastly the *Encrypting Unit*. The sections of the rest of the section describe each module in detail.

6.1 Organizational Knowledge Base

The *Organizational Knowledge Base*, as mentioned before, stores all the attributes of every stakeholder of the medical organization in the HIPAA compliant *EHR Ontology*. This ontology details the roles and attributes of the different stakeholders of the medical organization along with the various relationships between them. The *Organizational Knowledge Base* is critical in delivering correct attributes for the *Access Broker* and the *ABE Unit* to run. The role of this unit inside the *Encryption Unit* is to successfully deliver correct attributes of the entities (users and EHR fields) and correctly reflect the changes made by the medical staff in the patient's EHR field. The Key Generation Unit also requires attributes from the Knowledge Base so as to generate the decryption policy attributes.

6.2 Attribute Extraction Unit and Key Generation Unit

The Attribute Extraction Unit queries the EHR Ontology to retrieve the user and the EHR field attributes. As each users EHR is stored in the form of a node in the graph, querying this becomes a trivial task. Attribute extraction is carried out using SWRL rules.

The Key Generation Unit generates the keys required for ABE and provides it to the ABE unit for it to then encrypt/decrypt as explained in the section above. To generate the keys, it has to access the Organizational Knowledge Base as shown in Fig. 6. It needs this to create a combination of the user and EHR field attributes for the ABE unit to proceed with the encryption. For decryption too this unit provides the proper keys to the ABE unit which then decrypts the requested EHR field and provides it to the user.

6.3 Encrypting Unit

The Encrypting Unit acts like a co-coordinator for the different sub-modules of the ABE Unit. This is because, it continuously communicates with the Key Generation Unit, the Organizational Knowledge Base and the Attribute Extraction Unit. It requires the Key Generation Unit to extract the secret keys required for encryption/decryption. To perform this encryption, attribute and attribute values are required which are extracted by the Attribute Extraction Unit from the EHR Ontology. For implementation purposes, an open-source library called as the Ciphertext-Policy Attribute Based Encryption [30] is used. Further details can be found in Section 7.4.

7 *EHR Manager* Application

The *EHR Manager* Application is an open-source, service based, web application developed in Python to manage the field-level, attribute based encryption and access control of patient EHRs. This application uses attribute based access control to ensure that only the right users can access the right amount of data. Next, to guarantee a strong data encryption mechanism, this application uses attribute based encryption to protect the data based on the attributes of the user trying to encrypt/decrypt the concerned document. In other words, the secret key for encryption, is the combination of user attributes. We have developed this application in such a way that each sub-module performs its own functionality independently and together all the sub-modules serve as a suite of services. This design supports the re-use of sub-modules in developing other applications that require similar functionalities.

We have built the *EHR Manager* Application using open-source tools, Python language, libraries and APIs that are listed below. As all the development tools are open-source, the cost of the application is only that incurred for hosting the data on the Amazon cloud.

7.1 Web Development Framework

The *EHR Manager* Application is a web-based application built on the principles of the Model-View-Controller (MVC) architecture using used the Python Django framework. Using the *views, models, templates* and *URLs* of the framework, we designed this application to enable medical staff as well as patients easy and secure access to their concerned EHRs.

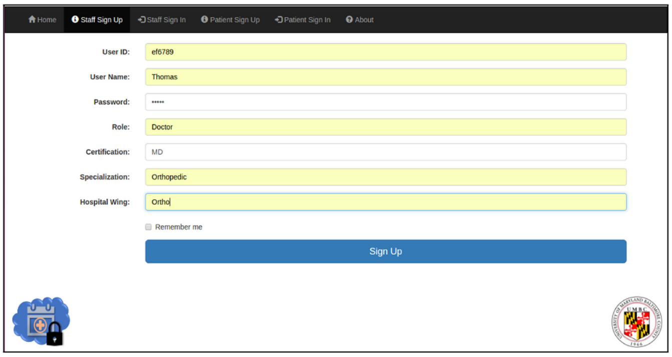
The *views.py* is a Python file that lists all the functions defined for the application. The views file works alongside the *templates, URLs and models* files respectively. The templates folder, as the name suggests, stores all the HTML templates for the application's front end. The *urls.py* is a Python file that lists all the regular expressions to be used for calling the appropriate functions written in the *views.py* file. The *models.py* file, again is a Python file which stores all the database tables and their respective schema. The framework flow is as follows. The user screen displays one of the templates from the templates folder and waits for the user to respond to the requested actions. On getting the user's input, the resulting action is associated with one of the URLs in the *urls.py* file. When this selected URL gets triggered, it calls its associated function from the *views.py* file which then performs the back-end operations for the current actions and then displays the next user page from the templates folder. The views file is responsible for making changes in the back-end database whose schema is defined in the *models.py* file. In this way, the data exchange continues back and forth using the principles of the MVC architecture.

7.2 Knowledge Management and Representation

As mentioned in the previous sections, the *EHR Manager* Application uses Semantic Web technologies to automate the attribute based access control and encryption. To design and implement the *EHR Ontology* we used the Protege [38] application. Protege supports the SWRL rule language and multiple reasoners that can support both description logic and SWRL reasoning.

7.3 Ontology Querying Library: rdflib

To extract data, i.e., the user and EHR field attributes out of the ontology, a bridge is required that can connect the Python application and the RDF/OWL ontology. An open-source library called *rdflib* is this bridge. *rdflib* is a toolkit that provides various functions to deal with ontologies and knowledge graphs. *rdflib* provides effective utilities to query the ontology and extract the necessary user and EHR field attributes.



User ID: ef6789

User Name: Thomas

Password: •••••

Role: Doctor

Certification: MD

Specialization: Orthopedic

Hospital Wing: Ortho

□ Remember me

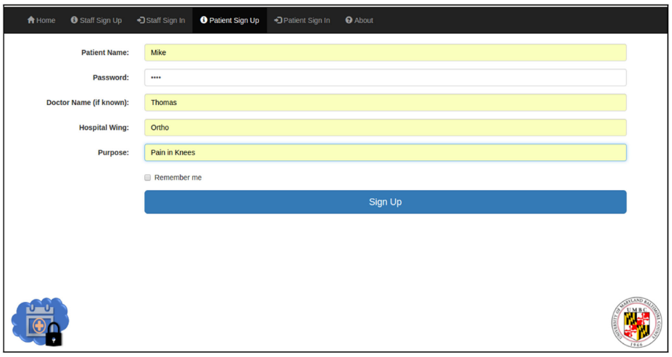
Fig. 7. Prototype: Staff registration view.

7.4 Field-Level Attribute Based Encryption: CPABE

The most crucial module of the *EHR Manager* is its encrypting unit which is carried out using ABE. The CipherText-Policy Attribute Based Encryption library is used for carrying out the encryption [30]. Researchers at the University of Texas at Austin have developed this open-source Python library that supports all the operations required to carry out ABE. CPABE associates a document to be encrypted with a particular, unique decryption policy. This decryption policy is a logical expression of attributes of the entities involved in the document usage. The users whose set of attribute values satisfy the decryption policy are allowed to decrypt and use the document.

To implement these features, the CPABE library provides four command-line tools—*cpabe-setup*, *cpabe-keygen*, *cpabe-enc* and *cpabe-dec*. *cpabe-setup* creates the public key and a master secret key which are required for the further operations. The *cpabe-keygen* utility generates a private key with a given set of attributes. Along with the attributes, *cpabe-keygen* uses the public key and the secret master key created by *cpabe-setup*. The list of attributes is specified as a space separated string. The output of *cpabe-keygen* is a private key for the user whose attributes are used for the document field encryption.

Next, *cpabe-enc* encrypts a file according to the decryption policy, which is a logical expression of attributes. This command encrypts the required file, in our case the document field content, by taking in the decryption policy by using the public key generated previously by cpabe-keygen. The encrypted file is written to a file with .cpabe extension. The cpabe-dec decrypts a file using a private key that is generated by cpabe-enc. The output of cpabe-dec is a the original file that got encrypted to a .cpabe file.



Patient Name: Mike

Password: ••••

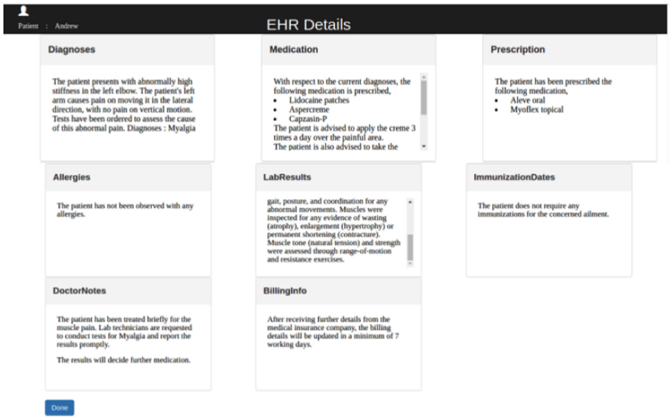
Doctor Name (if known): Thomas

Hospital Wing: Ortho

Purpose: Pain in Knees

□ Remember me

Fig. 8. Prototype: Patient registration view.



Patient : Andrew

Diagnoses

The patient presents with abnormally high stiffness in the left elbow. The patient's left arm causes pain on moving it in the lateral direction, with no pain on vertical motion. Tests have been ordered to assess the cause of this abnormal pain. Diagnoses : Myalgia

Medication

With respect to the current diagnoses, the following medication is prescribed,
- Lidocaine patches
- Aspercreme
- Capzasin-P

The patient is advised to apply the creme 3 times a day over the painful area.
The patient is also advised to take the

Prescription

The patient has been prescribed the following medication,
- Aleve oral
- Myoflex topical

Allergies

The patient has not been observed with any allergies.

LabResults

gait, posture, and coordination for any abnormal movements. Muscles were inspected for any evidence of wasting (atrophy), enlargement (hypertrophy) or permanent shortening (contracture). Muscle tone (natural tension) and strength were assessed through range-of-motion and resistance exercises.

ImmunizationDates

The patient does not require any immunizations for the concerned ailment.

DoctorNotes

The patient has been treated briefly for the muscle pain. Lab technicians are requested to conduct tests for Myalgia and report the results promptly.

The results will decide further medication.

BillingInfo

After receiving further details from the medical insurance company, the billing details will be updated in a minimum of 7 working days.

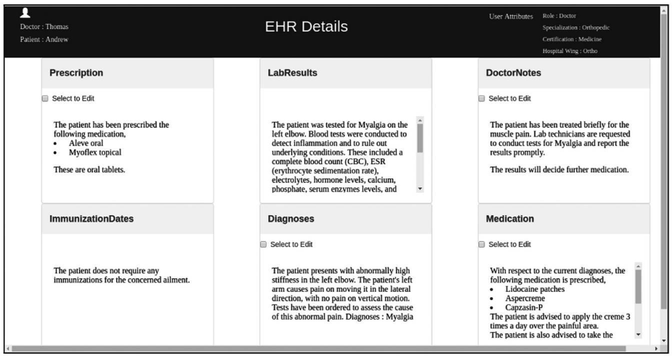
Fig. 9. Prototype: Patient's view shows the details of the health record.

7.5 Application Flow and Prototype

To use the applications, users register to the system by providing their attributes/credentials. As shown in Fig. 7, for a medical caregiver/staff, the application requests the person's unique id, name, medical certifications, specializations, the associated hospital wing, and other key attributes. For patients, the application requires their key attributes like name, the name of the medical staff the patient is primarily associated with, the hospital wing, etc., as shown in Fig. 8. On entering these details, a new entry is created for the user in the EHR knowledge graph.

Let us now consider the patient's view of the system. On registering with the system, the *EHR Manager* executes the *Access Broker* component to determine the access level. Based on the attributes of the patient, s/he is either allowed or denied access. The patient can see the entire health record for viewing/reading purposes, as shown in 9. To ensure the accuracy and integrity of the medical records, the patient is not allowed to edit any of the EHR fields.

Now let us look at the health caregiver or staff view of the *EHR Manager*. After registering the caregiver, the application again runs the *Access Broker* to determine the patient EHRs that the staff has access to. Along with this, the *Access Broker* also identifies the type of access which is either 'read only' or 'read/write' access. Depending on this access decision, the staff sees a list of the patient EHRs that they have access to. On selecting a record, they can view all the fields to which



Doctor : Thomas
Patient : Andrew

User Attributes
Role : Doctor
Specialization : Orthopedic
Certification : Medicine
Hospital Wing : Ortho

Prescription

□ Select to Edit

The patient has been prescribed the following medication,
- Aleve oral
- Myoflex topical

These are oral tablets.

LabResults

The patient was tested for Myalgia on the left elbow. Blood tests were conducted to detect inflammation and to rule out underlying conditions. These included a complete blood count (CBC), ESR (erythrocyte sedimentation rate), electrolytes, hormone levels, calcium, phosphate, serum enzymes levels, and

DoctorNotes

□ Select to Edit

The patient has been treated briefly for the muscle pain. Lab technicians are requested to conduct tests for Myalgia and report the results promptly.

The results will decide further medication.

ImmunizationDates

The patient does not require any immunizations for the concerned ailment.

Diagnoses

□ Select to Edit

The patient presents with abnormally high stiffness in the left elbow. The patient's left arm causes pain on moving it in the lateral direction, with no pain on vertical motion. Tests have been ordered to assess the cause of this abnormal pain. Diagnoses : Myalgia

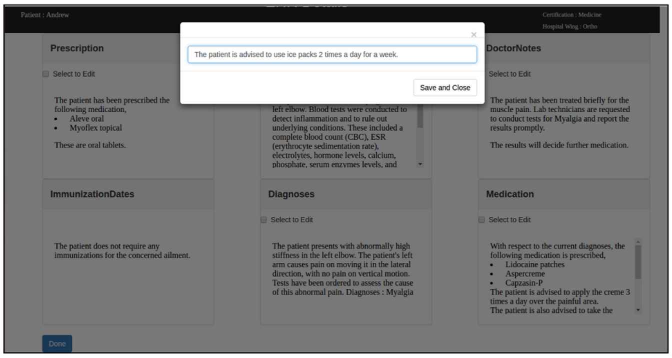
Medication

□ Select to Edit

With respect to the current diagnoses, the following medication is prescribed,
- Lidocaine patches
- Aspercreme
- Capzasin-P

The patient is advised to apply the creme 3 times a day over the painful area.
The patient is also advised to take the

Fig. 10. Prototype: Caregiver staff's view of the health record.


Patient : Andrew

Certification : Medicine
Hospital Wing : Ortho

The patient is advised to use ice packs 2 times a day for a week.

Prescription

□ Select to Edit

The patient has been prescribed the following medication,
- Aleve oral
- Myoflex topical

These are oral tablets.

left elbow. Blood tests were conducted to detect inflammation and to rule out underlying conditions. These included a complete blood count (CBC), ESR (erythrocyte sedimentation rate), electrolytes, hormone levels, calcium, phosphate, serum enzymes levels, and

DoctorNotes

□ Select to Edit

The patient has been treated briefly for the muscle pain. Lab technicians are requested to conduct tests for Myalgia and report the results promptly.

The results will decide further medication.

ImmunizationDates

The patient does not require any immunizations for the concerned ailment.

Diagnoses

□ Select to Edit

The patient presents with abnormally high stiffness in the left elbow. The patient's left arm causes pain on moving it in the lateral direction, with no pain on vertical motion. Tests have been ordered to assess the cause of this abnormal pain. Diagnoses : Myalgia

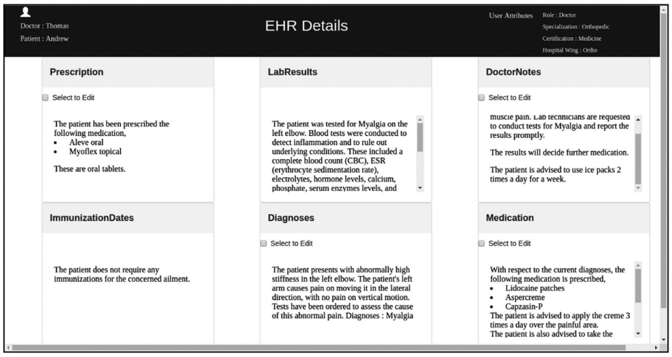
Medication

□ Select to Edit

With respect to the current diagnoses, the following medication is prescribed,
- Lidocaine patches
- Aspercreme
- Capzasin-P

The patient is advised to apply the creme 3 times a day over the painful area.
The patient is also advised to take the

Fig. 11. Prototype: Edit action.


Doctor : Thomas
Patient : Andrew

EHR Details

User Attributes
Role : Doctor
Specialization : Orthopedic
Certification : Medicine
Hospital Wing : Ortho

Prescription

□ Select to Edit

The patient has been prescribed the following medication,
- Aleve oral
- Myoflex topical

These are oral tablets.

LabResults

The patient was tested for Myalgia on the left elbow. Blood tests were conducted to detect inflammation and to rule out underlying conditions. These included a complete blood count (CBC), ESR (erythrocyte sedimentation rate), electrolytes, hormone levels, calcium, phosphate, serum enzymes levels, and

DoctorNotes

□ Select to Edit

muscle pain. Lab technicians are requested to conduct tests for Myalgia and report the results promptly.

The results will decide further medication.

The patient is advised to use ice packs 2 times a day for a week.

ImmunizationDates

The patient does not require any immunizations for the concerned ailment.

Diagnoses

□ Select to Edit

The patient presents with abnormally high stiffness in the left elbow. The patient's left arm causes pain on moving it in the lateral direction, with no pain on vertical motion. Tests have been ordered to assess the cause of this abnormal pain. Diagnoses : Myalgia

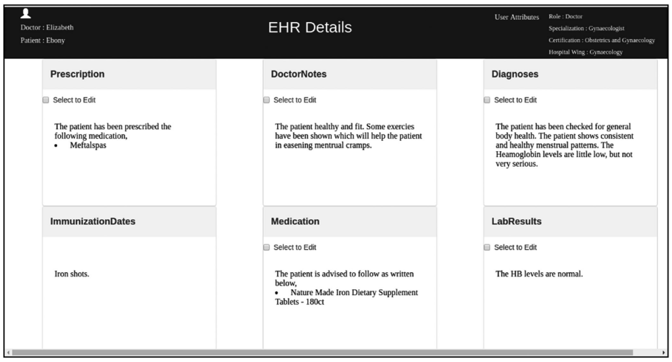
Medication

□ Select to Edit

With respect to the current diagnoses, the following medication is prescribed,
- Lidocaine patches
- Aspercreme
- Capzasin-P

The patient is advised to apply the creme 3 times a day over the painful area.
The patient is also advised to take the

Fig. 12. Prototype: Results of the edit action.


Doctor : Elizabeth
Patient : Ebony

EHR Details

User Attributes
Role : Doctor
Specialization : Gynaecologist
Certification : Obstetrics and Gynaecology
Hospital Wing : Gynaecology

Prescription

□ Select to Edit

The patient has been prescribed the following medication,
- Meftalspas

DoctorNotes

□ Select to Edit

The patient healthy and fit. Some exercies have been shown which will help the patient in easening mentrual cramps.

Diagnoses

□ Select to Edit

The patient has been checked for general body health. The patient shows consistent and healthy menstrual patterns. The Heamoglobin levels are little low, but not very serious.

ImmunizationDates

Iron shots.

Medication

□ Select to Edit

The patient is advised to follow as written below,
- Nature Made Iron Dietary Supplement Tablets - 180ct

LabResults

□ Select to Edit

The HB levels are normal.

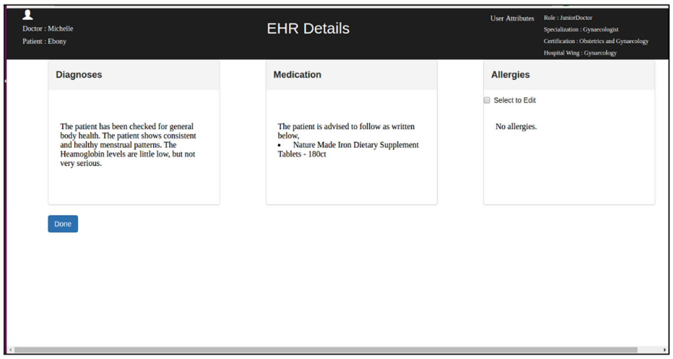
Fig. 13. Prototype: Senior caregiver staff can view more details of the health record.

the access is permitted. For the fields to which a write access is not permitted and only read is permitted, the 'Edit' action is absent.

Fig. 10 shows an example view of an Orthopedic doctor's view. Now, if the doctor wishes to edit a field to which he has access to, he can do so by clicking the 'Edit' button. Once done, the system stores the changes by encrypting it using the attributes of the doctor. Figs. 11 and 12 show this activity.

Now, to understand how the access control works according to the user's attributes, refer to Fig. 13 which shows the view of a gynecologist who is a senior doctor while Fig. 14 shows the view as can be seen by a nurse or junior doctor. As can be seen, due to the difference in the attributes, the resulting access levels are different.

In this way, the *EHR Manager* harnesses the Semantic Web and attribute based technologies to successfully guarantee a strong, robust, EHR managing application at in the field.


Doctor : Michelle
Patient : Ebony

EHR Details

User Attributes
Role : JuniorDoctor
Specialization : Gynaecologist
Certification : Obstetrics and Gynaecology
Hospital Wing : Gynaecology

Diagnoses

The patient has been checked for general body health. The patient shows consistent and healthy menstrual patterns. The Heamoglobin levels are little low, but not very serious.

Medication

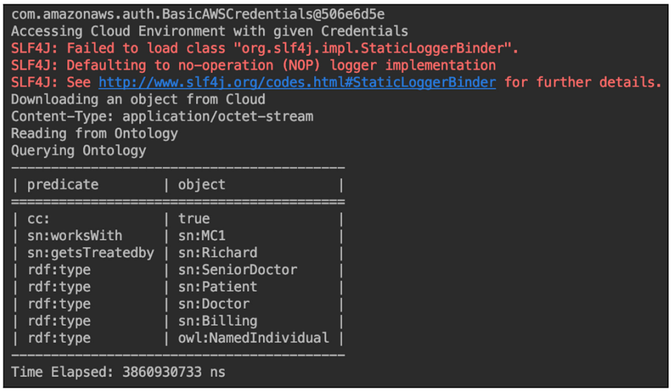
The patient is advised to follow as written below,
- Nature Made Iron Dietary Supplement Tablets - 180ct

Allergies

□ Select to Edit

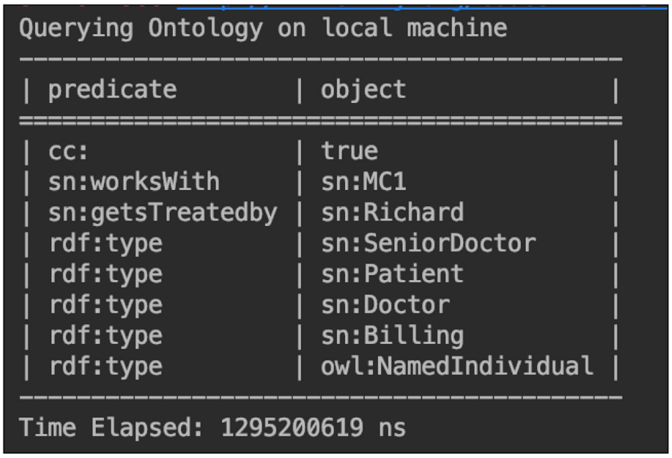
No allergies.

Fig. 14. Prototype: Junior staff has limited access and sees fewer details of the record.



```
com.amazonaws.auth.BasicAWSCredentials@506e6d5e
Accessing Cloud Environment with given Credentials
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Downloading an object from Cloud
Content-Type: application/octet-stream
Reading from Ontology
Querying Ontology
----------------------------------------
| predicate        | object              |
========================================
| cc:              | true                |
| sn:worksWith     | sn:MC1              |
| sn:getsTreatedby | sn:Richard          |
| rdf:type         | sn:SeniorDoctor     |
| rdf:type         | sn:Patient          |
| rdf:type         | sn:Doctor           |
| rdf:type         | sn:Billing          |
| rdf:type         | owl:NamedIndividual |
----------------------------------------
Time Elapsed: 3860930733 ns
```

Fig. 15. Ontology response time on cloud platform - AWS.



```
Querying Ontology on local machine
----------------------------------------
| predicate        | object              |
========================================
| cc:              | true                |
| sn:worksWith     | sn:MC1              |
| sn:getsTreatedby | sn:Richard          |
| rdf:type         | sn:SeniorDoctor     |
| rdf:type         | sn:Patient          |
| rdf:type         | sn:Doctor           |
| rdf:type         | sn:Billing          |
| rdf:type         | owl:NamedIndividual |
----------------------------------------
Time Elapsed: 1295200619 ns
```

Fig. 16. Ontology response time on local machine (edge).

8 Conclusion and Ongoing Work

EHR services are required to ensure secure and authorized access of patient data to adhere to the various regulatory acts such as HIPPA and HITECH. At the same time, they must be able to automatically delegate access of patient data to various caregivers to deliver timely treatment to patients. Security of cloud based EHR services is especially challenging since they are often accessed remotely by the end users. We have developed a novel, centralized, attribute based authorization mechanism for EHR services that uses Attribute Based Encryption to encrypt the patient records and allows for delegated secure access of patient records based on organizational policies.

This mechanism transfers the service management overhead from the patient to the medical organization and allows easy delegation of cloud-based EHR access authority to medical providers.

In our system design we have referenced the HL7 EHR functional model on information security that mandates that an EHR application must adhere to the rules or policies established to control access and protect the privacy of EHR information. We developed our EHR system by using ABE techniques (CPABE library), Semantic Web technologies, like OWL and SWRL, Python language and Amazon Cloud platform.

To automate the access policies, we have also developed a complex knowledge graph that details the roles and attributes of different stakeholders of the medical organization along with the various relationships between them. We have used a SWRL based reasoner to automate access control down to the field level. We have also developed an open-sourced web-based User Interface. To evaluate the scalability of our system, we performed performance analysis of the EHR ontology on cloud and on edge. Figs. 15 and 16 show the performance evaluation results. Note this timing data depends on many other factors: bandwidth variations, size of the S3 bucket used and the number of objects in the bucket. Apart from minor fluctuations, the average results remain constant as shown in the figures.

We currently host this service on the Amazon AWS platform. We are in process of developing this service for OpenStack Cloud and will compare the performance on the two cloud platforms. As part of our future work, we will also enhance our system to include the EHR data exchange and routing functionality that are essential for inter-organizational EHR systems. There are many additional security and privacy problems that can be addressed that we leave for future work. For example, stronger authentication mechanisms can help prevent unauthorized access by an attacker who has obtained the credentials of a physician and machine learning can be applied to recognized anomalous patterns of use.

ACKNOWLEDGMENTS

This research was supported by the Office of Naval Research under grants N00014-15-1-2228 and N00014-16-WX01489. We thank Dr. Seung Geol Choi (USNA), Dr. Eliot Siegel (University of Maryland Medical Center) and members of the Ebiquity Research Group for their vital input. This work was conducted using the Protege resource, which is supported by grant GM10331601 from the National Institute of General Medical Sciences of the United States National Institutes of Health.

REFERENCES

[1] K. Häyrinen, K. Saranto, and P. Nykänen, “Definition, structure, content, use and impacts of electronic health records: A review of the research literature,” *Int. J. Med. Informat.*, vol. 77, no. 5, pp. 291–304, 2008.

[2] “Electronic health record-system functional model, release 1,” in *ANSI/HL7 EHR, R1–2007*, ANSI/HL7, 2007.

[3] R. C. Barrows Jr and P. D. Clayton, “Privacy, confidentiality, and electronic medical records,” *J. Amer. Med. Informat. Assoc.*, vol. 3, no. 2, pp. 139–148, 1996.

[4] D. Blumenthal, “Launching hitech,” *New England J. Med.*, vol. 2010, no. 362, pp. 382–385, 2010.

[5] C. for Disease Control, Prevention, et al., “Hipaa privacy rule and public health. guidance from cdc and the us department of health and human services,” *MMWR: Morbidity Mortality Weekly Report*, vol. 52, no. Suppl. 1, pp. 1–17, 2003.

[6] U. D. of Health, H. Services, et al., “Summary of the hipaa privacy rule,” *Washington, DC: Author. Retrieved December*, vol. 2, 2003, Art. no. 2007.

[7] A. Bahga and V. K. Madisetti, “A cloud-based approach for interoperable electronic health records (ehrs),” *IEEE J. Biomed. Health Inf.*, vol. 17, no. 5, pp. 894–906, 2013.

[8] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, “Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 1, pp. 131–143, Jan. 2013.

[9] S. Chandra, V. Karande, Z. Lin, L. Khan, M. Kantarcioglu, and B. Thuraisingham, “Securing data analytics on sgx with randomization,” in *Proc. Eur. Symp. Res. Comput. Secur.*, 2017, pp. 352–369.

[10] F. Schuster, et al., “Vc3: Trustworthy data analytics in the cloud using sgx,” in *Proc. IEEE Symp. Secur. Privacy*, 2015, pp. 38–54.

[11] J. A. Evans, “Electronic medical records system,” U.S. Patent 5,924,074, Jul. 13, 1999.

[12] E. H. Shortliffe, et al., “The evolution of electronic medical records,” *Academic Med.*, vol. 74, pp. 414–419, 1999.

[13] M. Lavin and M. Nathan, “System and method for managing patient medical records,” U.S. Patent 5,772,585, Jun. 30, 1998.

[14] S. Narayan, M. Gagné, and R. Safavi-Naini, “Privacy preserving ehr system using attribute-based infrastructure,” in *Proc. ACM Workshop Cloud Comput. Secur. Workshop*, 2010, pp. 47–52.

[15] R. Zhang and L. Liu, “Security models and requirements for healthcare application clouds,” in *Proc. IEEE 3rd Int. Conf. Cloud Comput.*, 2010, pp. 268–275.

[16] M. Joshi, S. Mittal, K. P. Joshi, and T. Finin, “Semantically rich, oblivious access control using abac for secure cloud storage,” in *Proc. IEEE Int. Conf. Edge Comput.*, 2017, pp. 142–149.

[17] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge computing: Vision and challenges,” *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[18] D. S. Roche, A. Aviv, and S. G. Choi, “A practical oblivious map data structure with secure deletion and history independence,” in *Proc. IEEE Symp. Secur. Privacy*, 2016, pp. 178–197.

[19] E. Stefanov, M. Van Dijk, E. Shi, C. Fletcher, L. Ren, X. Yu, and S. Devadas, “Path oram: an extremely simple oblivious ram protocol,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2013, pp. 299–310.

[20] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, “Role-based access control models,” *Comput.*, vol. 29, no. 2, pp. 38–47, 1996.

[21] X. Jin, R. Krishnan, and R. Sandhu, “A unified attribute-based access control model covering dac, mac and rbac,” in *Proc. IFIP Annu. Conf. Data Appl. Secur. Privacy*, 2012, pp. 41–55.

[22] A. Anderson, A. Nadalin, B. Parducci, D. Engovatov, H. Lockhart, M. Kudo, P. Humenn, S. Godik, S. Anderson, S. Crocker, et al., “extensible access control markup language (xacml) version 1.0,” *OASIS*, 2003.

[23] L. Kagal, T. Finin, and A. Joshi, “A policy language for a pervasive computing environment,” in *Proc. IEEE 4th Int. Workshop Policies Distrib. Syst. Netw.*, 2003, pp. 63–74.

[24] L. K. J. N. R. S. W. H. W. Tim Finin, A. Joshi, and B. Thuraisingham, “Rowlbac - representing role based access control in owl,” in *Proc. 13th Symp. Access Control Models Technol.*, Jun. 2008, pp. 73–82.

[25] J. M. Bradshaw, A. Uszok, M. Breedy, L. Bunch, T. Eskridge, P. Feltovich, M. Johnson, J. Lott, and M. Vignati, “The kaos policy services framework,” in *Proc. 8th Cyber Secur. Inf. Intell. Res. Workshop*, 2013, pp. 6–16.

[26] D. L. McGuinness, F. Van Harmelen, et al., “Owl web ontology language overview,” *W3C Recommendation*, vol. 10, no. 10, 2004, Art. no. 2004.

[27] N. K. Sharma and A. Joshi, “Representing attribute based access control policies in owl,” in *Proc. IEEE 10th Int. Conf. Semantic Comput.*, 2016, pp. 333–336.

[28] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based encryption for fine-grained access control of encrypted data,” in *Proc. 13th ACM Conf. Comput. Commun. Secur.*, 2006, pp. 89–98.

[29] J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-policy attribute-based encryption,” in *Proc. IEEE Symp. Secur. Privacy*, 2007, pp. 321–334.

[30] B. Waters, “Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization,” in *Proc. Int. Workshop Public Key Cryptography*, 2011, pp. 53–70.

[31] J. A. Akinyele, M. W. Pagano, M. D. Green, C. U. Lehmann, Z. N. Peterson, and A. D. Rubin, “Securing electronic medical records using attribute-based encryption on mobile devices,” in *Proc. 1st ACM Workshop Secur. Privacy Smartphones Mobile Devices*, 2011, pp. 75–86.

[32] J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, “Patient controlled encryption: Ensuring privacy of electronic medical records,” in *Proc. ACM Workshop Cloud Comput. Secur.*, 2009, pp. 103–114.

[33] S. Narayan, M. Gagné, and R. Safavi-Naini, “Privacy preserving ehr system using attribute-based infrastructure,” in *Proc. ACM Workshop Cloud Comput. Secur. Workshop*, 2010, pp. 47–52.

[34] O. Lassila, R. Swick, et al., “Resource description framework (RDF) model and syntax specification,” WWWConsortium, 1999.

[35] T. Berners-Lee, D. Connolly, L. Kagal, Y. Scharf, and J. Hendler, “N3logic: A logical framework for the world wide web,” in *Theory and Practice of Logic Programming*. Cambridge, U.K.: Cambridge Univ. Press, 2008.

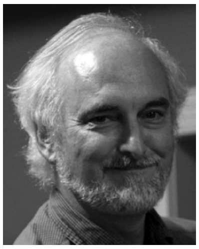
[36] I. Horrocks, P. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean, “SWRL: A Semantic Web rule language combining Owl and RuleMl,” WWWConsortium, 2004.

[37] K. P. Joshi, Y. Yesha, and T. Finin, “An ontology for a hipaa compliant cloud service,” in *Proc. 4th Int. IBM Cloud Academy Conference ICACON*, 2016, p. 55.

[38] M. Musen, “The protg project: A look back and a look forward,” *AI Matters*, vol. 1, pp. 4–12, 2015.

Maithilee Joshi received the master’s degree in computer science from the University of Maryland, Baltimore County, in May 2018. She now works as a Software Developer in Consumer Payments at Amazon, Seattle.

Karuna Joshi received the PhD degree in computer science from UMBC. He is currently an assistant professor of information systems with the University of Maryland, Baltimore County. Her research is focused on cloud automation and security, data science and health IT. She has been awarded the prestigious IBM PhD Fellowship. She also has more than 15 years of industrial experience, primarily as an IT project manager. She worked at the International Monetary Fund for nearly a decade.



Tim Finin received the degrees from MIT and the University of Illinois and has held positions with Unisys, the University of Pennsylvania, and the MIT AI Laboratory. He is the Willard and Lillian Hackerman chair in engineering and a professor of computer science and electrical engineering with UMBC. He has more than 40 years of experience in applications of artificial intelligence to problems in information systems and language understanding. His current research is focused on knowledge representation and reasoning, analyzing and extracting information from text, and enhancing information systems security and privacy.