In proceedings of 8th IEEE International Conference on Big Data Security on Cloud (BigDataSecurity 2022)

Semantically Rich Access Control in Cloud EHR Systems Based on MA-ABE

Sharad Dixit, Karuna Pande Joshi, Seung Goel Choi, Lavanya Elluri
University of Maryland, Baltimore County, Baltimore, MD 21250 USA
{sdixit1@umbc.edu, karuna.joshi@umbc.edu, choi@usna.edu, lelluri1@umbc.edu}

***Abstract*—With the rapid implementation of Cloud-based Electronic Health Record (EHR) systems, health providers are specifically concerned about handling data privacy on the cloud. Existing methods have either scalability issues by requiring that patients grant access to their medical data or a trust issue by having a single authority, thereby creating the problem of a single point of attack. Hence there is a need to develop an EHR system that addresses these bottlenecks for safe, secure, and easy cloud-based EHR management. To address these bottlenecks, we have developed a novel framework that allows policy-based multi-authority access permission to Electronic Health Record systems used by multiple care providers from various places or organizations. This framework, residing on the Edge, has been built using the Multi-Authority Attribute Based Encryption (MA-ABE) and Semantic Web technologies to provide a safe, semantically rich approach to facilitate secure data sharing among organizations who manage different attributes of end-users using a shared dataset. This paper describes our novel approach and the proof of concept prototype that we created to evaluate our framework.**

***Index Terms*—Multi-Authority Attribute Based Encryption (MA-ABE), Attribute Based Access Control (ABAC), Knowledge Graph (Ontologies), Cloud Computing, Access Handler, Document Processor & Crypto Module.**

I. Introduction

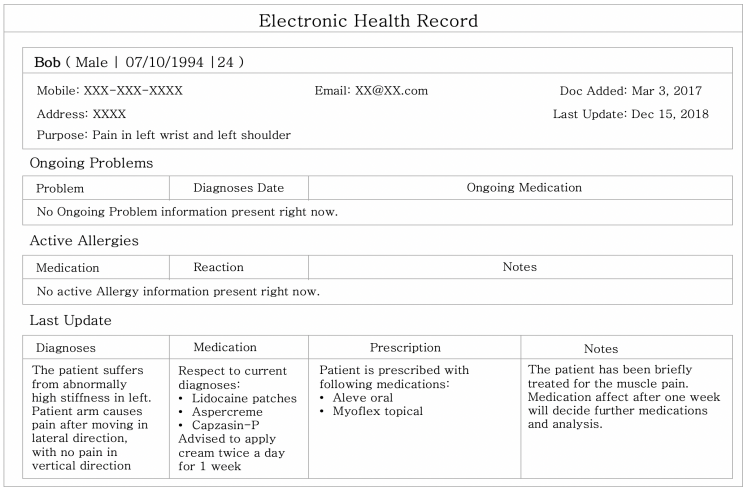
A. Motivation

Cloud-based Electronic Health Record (EHR) applications have been increasingly adopted by medical organization to store the relevant medical information of their patients. These cloud-based EHR systems permit an organization to maintain, create, and control all the electronic versions of patient data in a single location by taking advantage of cloud storage's efficiency and scalability features that enable fast retrieval and sharing of medical data. Therefore, significant increase in medical caregivers moving to cloud-based EHR systems [4], [17], [18], [20] to avail the significant cost reduction as well as the flexibility and high availability provided by the Cloud EHR systems. However, storage of electronic copies remotely with third-party cloud servers increases the possibility of attacks and data breaches leading to privacy concerns impeding the wide adoption of such services. An EHR record collects patients' health-related information to allow efficient, consistent, and universal sharing of medical data. Figure 1 shows a sample EHR record, which contains diagnoses, medication, prescription and doctor notes, and patient medical history and personal details. Because of the sensitivity of medical data, a significant challenge lies in providing secure and delegated access to EHR data. Privacy of patient's EHR record is also recognized by health care regulations such as Health Information Technology for Economic and Clinical Health (HITECH) [6], and Health Insurance Portability and Accountability Act (HIPAA) [11], [28] which require cloud service providers to promote and regulate the management and distribution of medical data; they have established rules and regulations for protecting confidentiality and privacy of medical data stored in cloud storage and are aimed at ensuring sufficient care is provided to it. In order to enforce the privacy of medical data, researchers have considered an approach of patient-centric privacy [5], [21], [31] where the patient is accountable for granting access. Although ideal from a privacy perspective, involving a patient with every access decision creates a significant system overhead, causing considerable damage to the system's scalability.

Moreover, the patient will be easily overwhelmed by the large scale of requests and the technical complexity of the system, potentially leading to incorrect access decisions that, at times, the patient may not be in a state to give this approval. Due to the above-mentioned issues, most of the research works consider a central authority (CA) to manage an access control mechanism with encryption [2], [14], [24], [31]. Existing cloud-based EHR services such as CureMD [8], and Athenahealth [3] also follow the framework of the access control management by a CA. However, this framework creates a load bottleneck on CA. CA will stop the entire system from working if it becomes inaccessible due to either software/hardware error or a denial of service attack. More significantly, the system puts too much trust in CA. Since CA has access control over all the data, if CA stops being trustworthy due to internal or external corruption, data privacy of the entire system would be endangered.

B. Our Work

We present a novel EHR access framework that overcomes the shortcomings of previous works and extends our previous work in progress [9]. In particular, our framework guarantees a secure encrypted access control mechanism in a multi-authority environment. When caregivers belong to different organizations and work in different contexts (e.g., location and/or time), data access policies are often dictated by multiple authorities. Our EHR access framework enables these authorities to specify a different set of attributes for a caregiver, and access to the patient EHR is granted to the caregiver only if



Electronic Health Record

Bob (Male | 07/10/1994 |24)

Mobile: XXX-XXX-XXXX | Email: XX@XX.com | Doc Added: Mar 3, 2017

Address: XXXX | Last Update: Dec 15, 2018

Purpose: Pain in left wrist and left shoulder

Ongoing Problems

Problem	Diagnoses Date	Ongoing Medication
No Ongoing Problem information present right now.		

Active Allergies

Medication	Reaction	Notes
No active Allergy information present right now.		

Last Update

Diagnoses	Medication	Prescription	Notes
The patient suffers from abnormally high stiffness in left. Patient arm causes pain after moving in lateral direction, with no pain in vertical direction	Respect to current diagnoses: • Lidocaine patches • Aspercreme • Capzasin-P Advised to apply cream twice a day for 1 week	Patient is prescribed with following medications: • Aleve oral • Myoflex topical	The patient has been briefly treated for the muscle pain. Medication affect after one week will decide further medications and analysis.

Fig. 1: Sample EHR Record

the caregiver satisfies all attribute conditions. In particular, we use Multi-Authority Attribute-Based Encryption (MA-ABE) scheme [26] for data encryption. In attribute-based encryption (ABE) scheme [13], the encryption algorithm takes as input the public parameters as issued by an authority as well as a Boolean formula over a set of attributes. Each of the decryption parties will be issued private keys by the authority associated with a set of attributes. A party can decrypt a cipher-text if the attributes of his private key satisfy the Boolean formula associated with the cipher-text. A single-authority ABE works well in the setting where data is managed within one organization or trust domain. However, there are many scenarios when one wishes to describe a policy that spans multiple trust domains.

To address this issue, MA-ABE systems were introduced where multiple parties could play the role of authority. A medical record is encrypted with an MA-ABE scheme and stored in our system in cloud storage. Each of the multiple authorities controls a disjoint subset of attributes needed for decryption, and a party can decrypt an encrypted record only if the party has been granted all the necessary attributes from these multiple authorities. Since decryption is performed based only on the attributes, the authorities do not need to be available once the party has obtained the necessary attributes, relieving our system of the load bottleneck issue. We highlight key aspects of our system below.

• **Collusion resistance**. In our environment, EHR data can be accessed by any authorized user, such as doctors, nurses, and health insurers. Hence, we can not neglect the possibility that these users may intentionally or unintentionally collude together to gain access to part of EHR data they do not have the right to access separately. However, our setting and design provide complete confidentiality and data security in such circumstances.

• **Semantically rich policy specification based on a knowledge graph**. We designed and developed a comprehensive ontology, a knowledge graph, to define security and privacy measure specific to the healthcare domain. Our ontology defines in detail the concepts of a medical domain by comprehensively describing the roles and attributes of different medical organization entities and their relationships. In particular, to realize the multi-authority environment, our ontology describes different medical authorities with distinct medical attributes that they could control and manage. We also developed a fine-grained access control mechanism to extract attributes and their relations from the ontology and match them against the access policy rules to provide a comprehensive access decision.

• **Edge computing**. The architecture of our framework is based on the principles of edge computing, which refers to the notion of performing all the necessary required computations inside the organization before moving the data to any third-party server. In our framework, we have established a strong boundary for communication outside the organization at the edge. We have used a secure access control mechanism and a robust crypto module for data encryption before transferring the data out of the organization [9]. We have also implemented a Multi-Authority EHR Application to prototype this research using open-source development tools. The application provides an easy interface for all the users of the medical domain to view and/or edit the EHR record guaranteeing a semantically rich and cryptographically secure environment.

C. Organization

This paper consist of 8 sections. Section II, consist of related work, then section III presents brief overview of framework and design is discussed, followed by comprehensively explaining each module of the framework in section IV (Access Handler) and section V (Document Processor Crypto Module). Section VI demonstrates the feasibility of the proposed framework by discussing a prototype EHR application and section VII evaluates the scalalibity and performance of the framework by performing an in-depth performance analysis.

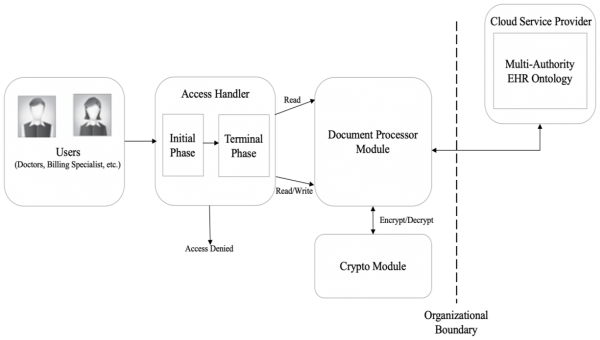


Fig. 2: System Architecture Overview

Then the paper is concluded in section VIII with the scope of future work.

II. RELATED WORK

Immense research focus has been directed towards creating secure systems for storing and sharing of EHRs, even importance of preserving data privacy and security of Electronic Health Records (EHR) in a cloud environment has been recognized by the health regulations acts like Health Information Technology for Economic and Clinical Health (HITECH) Act and Health Insurance Portability and Accountability Act (HIPAA). Hence, developing systems complying with all ethical standards and legalities is a complex research challenge. Currently, there exist several cloud-based EHR services such as CureMD [8], and Practice Fusion [25]. Organizations such as GE Healthcare [12], and Epic Health Services [10] are also investing in cloud-based EHR services.

A. Authentication-Based EHR System

Authentication-based EHR System incorporates an access control mechanism such as role-based access control (RBAC) or Access Control list (ACL) scheme to manage user's access right and places complete trust on the cloud server where the EHR system resides. Multiple access control mechanisms have been proposed to ensure authorized access to the system. Models such as the fixed access control list (ACL), Mandatory Access Control (MAC), Role-Based Access Control (RBAC) [27] have been used for securing documents. However, these models are not adequate for an organization with a complex organizational structure.

In our scheme, we have utilized Attribute-Based Access Control (ABAC) [15], [29] which is an enhancement over all the models where many users' multi-valued attributes are evaluated against the access policy before providing an access decision.

B. Cryptography-Based EHR system

An increasing importance is seen in applying Attribute Based Encryption (ABE) to develop secure systems for EHR. This is because of the high key management overhead suffered in traditional public-key encryption schemes, whereas ABE schemes are much more scalable. In particular, Narayan et al. [24] proposed a fine-grained encrypted approach where a patient's EHR file was encrypted by using the broadcast variant of ciphertext policy ABE that allow users and attributes revocation. Ibraimi et al. [14] proposed infrastructure by introducing a concept of social/professional domain and implemented Ciphertext Policy Attribute Based Encryption (CP-ABE) to encrypt the patient health records to ensure data privacy and security. Akinyele et al. [2] integrated ABE in his proposed infrastructure to secure electronic medical records on mobile devices or cloud servers. Recently, Joshi et al. [16] proposed a semantically rich and secure application for storing electronic health records with Amazon cloud service provider [aws.amazon.com]. The EHR manager application is a web-based application that enables an organization to define and enforce its access policy with ABE to tighten the security further. The application uses a knowledge graph which is stored with the cloud service provider containing all the necessary information of all the different stakeholders of a medical organization. User and Document attributes are extracted from the knowledge graph and applied against the access policy to provide a categorized access decision. The application uses CP-ABE [13] for encryption purposes to provide guaranteed delegated secure access to electronic health records.

Although, the above ABE-based systems have a weakness that assumes a central authority in the system. All the workload and trust of issuing keys and attribute-related tasks like validating originality of user attributes is managed by this central authority, which creates a load bottleneck and creates security and privacy issues as it gives the central authority the control to access encrypted files. Moreover, there is no related work to semantically rich access control mechanism for fine-grained access in these system. The key difference of our system from the above works is that our system realizes a multi-authority system which provides a better reality; different organizations usually outsource some of their tasks or form their subdomains which tend to become authorities controlling and certifying their own set of attributes.

In our scheme, we have utilized enhanced MA-ABE [26] where multiple authorities control a disjoint subset of attributes, and authorities cannot pool data to get access to the encrypted files. Recently, Li et al. [19] proposed an attribute-based infrastructure and introduced a concept of a personal/public domain for secure management of patient health records in a multi-authority scenario. They proposed to apply Key Policy ABE (KP-ABE) in the personal domain and Multi-Authority ABE (MA-ABE) in the public domain to have scalable key management. However, the system puts a lot of control overhead on the patient. It does not provide a semantically rich and robust access control mechanism for fine-grained access to the system, whereas our system transfers all service management overhead from the patient to the authorities and medical organizations with a fine-grained access control mechanism.

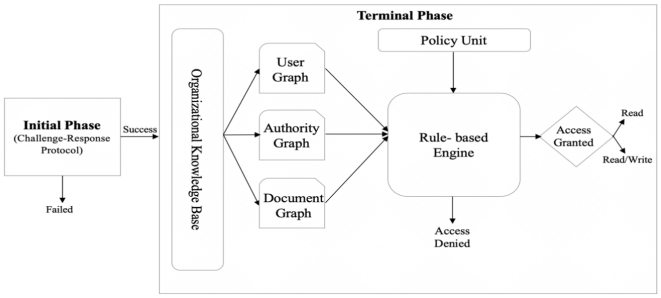


Fig. 3: Access Handler Architecture

III. FRAMEWORK OVERVIEW & DESIGN

The primary goal is to develop a framework that provides easy access and management of EHR data without compromising on data confidentiality and privacy of the records. We show how the framework transfers the service management overhead from the patients or a centralized authority to multiple authorities and provides secure delegated access to EHR documents.

A. Threat Model

In our system, there are data owners, multiple users and authorities, but a single cloud service provider stores all the EHR documents in the cloud storage. Users come from a broad medical universe; he or she can be a doctor, billing specialists, pharmacists, etc. Each user receives access rights based on his attributes matched against the confidential policies defined by the organizations. Users may be corrupted and try to gain unauthorized access to a medical record. For example, an insured person may try to obtain information beyond their allocated access to develop marketing strategies. Data owners (patients) have read access to their respective documents. The cloud provider may be compromised by an adversary as well. We consider that a corrupted cloud service provider will behave in an honest-but-curious manner [22]; that is, although the provider follows the complete regulations and arrangements, it will try to obtain as much information as possible passed between the complete end-to-end parties.

B. Overview of our Framework

The primary objective of our framework is to provide secure delegated EHR access in a multi-authority environment. The key idea is to realize an environment of multiple authorities, each governing a disjoint subset of attributes. Users obtain their obligated multi-valued attributes and attributes-based secret keys from respective authorities without interacting with the data owner to reduce overhead from the owner or any need of central authority and place the burden on different multiple authorities. A single-authority ABE works well in the setting where data is managed within one organization or trust domain. However, there are many scenarios when one wishes to describe a policy that spans multiple trust domains. To address this issue, MA-ABE systems were introduced where multiple parties could play the role of authority.

Our framework is split into two major parts, based on the principles of Edge computing [30] where we have defined our secure ”edge” as the organizational boundary (see Figure 2 for the overall system architecture).

- Entities inside the organizational boundary are controlled and managed by the organization and hence are considered trusted units. The first part of our framework is considered inside an organizational boundary consisting of the Access Handler module and Document Processor & Crypto module. Different consumers request a login to the system to which the initial authentication is performed by traditional challenge-response protocols in the Access Handler module. If the access handler allows the request, then the request is forwarded to the Document Processor & Crypto Module for document encryption and decryption. We have implemented Multi-Authority Attribute Based Encryption (MA-ABE) [26] with Symmetric Encryption for encryption purposes in this module.
- On the other hand, the second part lies outside the organizational boundary, consisting of the untrusted server provided and managed by cloud service providers.

IV. ACCESS HANDLER

Different consumers request a login to the system to which the initial authentication is performed by traditional challenge-response protocols in the Access Handler module. Suppose the user qualifies the initial authentication phase. A complete access control decision is evaluated in the terminal authentication phase where read, write or no access is provided to users using Attribute-Based Access Control (ABAC) to carry out a strong access control mechanism semantically. To validate the extensive access decision, respective semantics of the user and document (attributes instances) are extracted from the authority, user and document graphs. Conclusively, an access decision is then carried out by matching the extracted attribute instances against the privileged access policies defined by an organization. Each authority defined in the authority graph allocates the user with their respective attributes, which the authority controls and binds them to their claimed attributes.

Knowledge graph. To implement an access control mechanism, a Multi-authority EHR Ontology (Knowledge Graph) was developed using Web Ontology Language (OWL) [23] to define security and privacy measures specific to the healthcare domain. Developed ontology defines in detail the concepts of a medical domain by comprehensively describing the roles and attributes of different medical organization entities and their relationships. In particular, to realize a multi-authority environment, developed ontology describes different medical authorities with distinct medical attributes that could be controlled and managed by them. Developed access control mechanism refers to this knowledge graph (representing the access policies for organizations) and provides an access decision for each access request. Semantic Web Rule Language (SWRL) is used to reason over the ontology based on the access policies. One of the captivating advantages of the developed access

mechanism is that any number of authorities or the attributes controlled by them can be added within the extensible medical domain, allowing health organizations to have more flexibility in their access policies. This key idea behind Access Handler is to implement an access control mechanism of the system using Attribute Based Access Control (ABAC). This module guarantees a precise access to the requesting person according to his attributes. Unlike other access control mechanisms, this scheme regulates access at a more acceptable level by evaluating user's multi-value attributes against the access policies.

Architecture. The module consists of two phases (Figure 3) [9], i.e., initial and terminal. The initial phase of the module implements traditional challenge-response protocols to provide initial access to the system where if the user qualifies it, the request is further handled by the terminal phase where comprehensive access control authentication is performed to provide a read-write or no access decision. The terminal phase of the module contains three critical submodules Organizational Knowledge Base, Rule-Based Engine, and Policy Unit. Further, this section explains each submodule of the Access handler in detail.

A. Organizational Knowledge

This sub-module is the primary knowledge store, which captures all the credentials of every entity in the EHR domain in the knowledge graph. We developed the Multi-Authority EHR Ontology (Figure 4) using semantically rich Web Ontology Language (OWL) to store roles and multi-valued attributes of different stakeholders as an instance of classes developed in the ontology related to the EHR domain. This ontology contains three sub-parts; Authority graph, User graph, and Document graph. Authorities are the attribute authorities of the medical domain, each governing disjoint subset of attributes. These authorities control and allocate attributes to the users and provide users' respective secret keys. For authorities, we have created an Authority graph that describes the multi-valued attributes controlled by each authority with their relation to the user graph. For example, the Doctor of Medicine (M.D.) attribute is authorized and controlled by the American Medical Association authority (AMA) and Allergy&Immunization (A.I.) attribute is authorized by the American Board of Medical Specialties (ABMS). Users are the authenticated employees of the organization with validated credentials to access the EHR system. We have built a user graph to store the necessary attributes and credentials related to users and their relationship with the attributes controlled by authorities. For example, a Doctor will have attributes like name, duties, and Doctor of Medicine (M.D.). Documents are the Electronic Health Record (EHR) of patients, which details all the clinical reports of an individual affiliated with the medical organization. All the EHR documents are assumed to be available within the cloud server in an encrypted format. Finally, we have also created a knowledge graph for the documents, which entails all the required attributes related to EHR documents such as belongsTo, createdBy, and accessLevel.

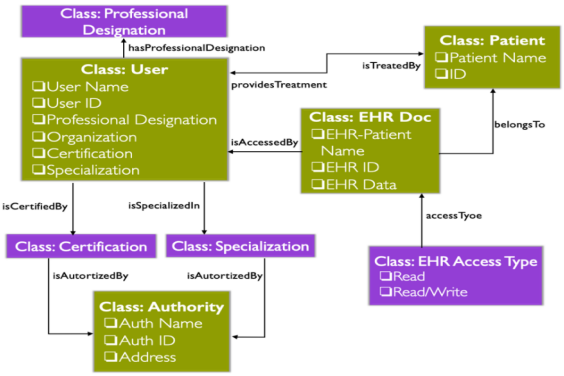


Fig. 4: Snapshot of EHR Ontology

B. Policy Unit

This unit is used for storing all the privileged access policies of an organization to store rules in Semantic Web Rule Language (SWRL) rules. An organization defines these policies to regulate and manage access decisions to confidential and valuable information. Our system works in a multi-authority system that allows an organization to declare and realize their confidential policies depending on the multiple authorities within the organization. This helps an organization to realize the reality more freely than the previous work [16] where only a single authority exists. Multiple confidential access policies can be stated in our system where if a user has to validate against multiple policies, then our system correctly executes the rules by validating the user to all the required policies. For implementation and prototyping purposes, HIPAA policies were used as access policies to determine access control over patient EHRs.

C. Rule Based Engine

Policy Unit works coherently with Rule Based Engine to infer an access decision. Rule-Based Engine is responsible for carrying out the access decisions based on access policies provided by the Policy Unit matching against document and related authority-user attributes extracted from the organization's knowledge graph. The Rule-Based Engine implements SWRL against attributes instances for providing access decisions read, write, and if the attributes of the user do not comply with the document policies, the user is denied access to the document.

Figure 5 depicts an example SWRL rule depicting how a user's access request to specific fields of the EHR is evaluated. Described rule states that if a user has credentials as a senior doctor with Doctor of Medicine (M.D.) certification authorized from American Medical Association (AMA) and working in Hospital ward "A" by authorization from Hospital "Alpha" has to write access to patient's EHR.

V. Document Processor & Crypto Module

This module, as shown in Figure 6 [9] contains two submodules.

- *Document Processor Module:* It carries out the duties of document fetching and communication outside the

```
SeniorDoctor (?sd) ^
EHRdoc (?EHRDoc) ^
Certification (?c, ?M.D.) ^
HospitalWard(?hw, ?hwA) ^
Authority (?Auth1, ?AMA) ^
Authority(?Auth2, ?HospitalAlpha) ^
isCertifiedBy (?sd, ?c) ^
isAuthorizedBy (?c, ?Auth1) ^
worksIn(?w, ?hw) ^
isAuthorizedBy(?w, ?Auth2)
-> hasWriteAccessMedication (?sd, true)
```

Fig. 5: Example SWRL 1 rule

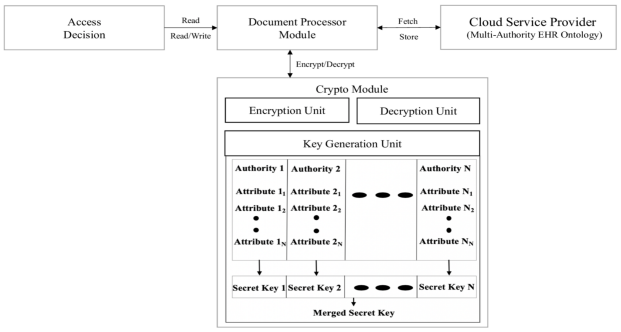


Fig. 6: Document Processor & Crypto Module

Let N be the number of authorities and M be the number of users. Let $A_1, \ldots, A_N$ denote the authorities and $U_1, \ldots, U_M$ the users. Let $\mathsf{AS}_1, \ldots, \mathsf{AS}_N$ be the disjoint sets where each AS_i denotes the set of attributes that authority A_i controls. Let US_j denote the set of attributes that U_j receives. Each attribute att of user U_j (i.e., $\mathsf{att} \in \mathsf{US}_j$) is from some authority, i.e., $\mathsf{att} \in \bigcup_{i=1}^{N} \mathsf{AS}_i$.
Access decision evaluation. Let $\mathsf{AP}_1, \ldots, \mathsf{AP}_X$ be the access policies that the system describes.
For a user U_j with US_j:
$\exists k : \mathsf{AP}_k(\mathsf{US}_j) = 1 \implies$ Access EHR Doc according to AP_k (read and/or write).
Encryption with MA-ABE. Let $E_1, \ldots, E_Y$ be the EHR documents. To encrypt an EHR doc E_ℓ, create a ciphertext policy CP_ℓ for E_ℓ and then encrypt E_ℓ along with CP_ℓ using MA-ABE. Here CP_ℓ is decribed as a monotone boolean formula based on some attributes in $\bigcup_{i=1}^{N} \mathsf{AS}_i$.
Decryption using MA-ABE. Decryption is performed as follows.
For user U_j with US_j:
if $\mathsf{CP}_\ell(\mathsf{US}_j) = 1$ then U_j can decrypt EHR Doc E_ℓ.

Fig. 7: Mathematical representation

organizational boundary. The EHR document is retrieved from the cloud server by the Document Processor Module after a qualified (read or write) access decision provided by the Access Handler.

- *Crypto Module:* It performs necessary encrpytion and decryption tasks.

If the access handler approves the request, then the request is forwarded to the *Document Processor & Crypto Module*. If the access decision granted by the *Access Handler* was read, then the *Document Processor & Crypto Module* gets the document from the server and expects for the decryption keys to be given by the user, which the the user attains from his corresponding multiple authorities. If the decryption keys given by the user are valid then the system can decrypt the documents.

For write access, all modules function similar to read access. Although, after the user modifies the document, the *Crypto Module* encrypts the document using the encryption keys in agreement with the access policy stated by the organization. Encrypted document is transferred and stored on the server by making a new node recording all the details of the modifications in the patient's EHR.

A. Details of Crypto Module

Crypto Module performs necessary encryption and decryption tasks. This module implements Multi-Authority Attribute Based Encryption (MA-ABE) [26]. The framework provides enhanced user privacy by employing MA-ABE in encrypting medical records. In particular, one fascinating feature of MA-ABE [7] is that aggregation of decryption keys from multiple parties doesn't increase the decryption power. In other words, suppose neither Alice nor Bob can decrypt ciphertext C individually. Then, they will not be able to decrypt the ciphertext C even if they collaborate and merge their decryption keys. A similar level of security is guaranteed even against the coalition of multiple authorities. Since our system is realized in a multi-authority setting (instead of a single central authority), along with preventing authorities from combining their data to obtain access to documents, our system proves to be highly secure for data privacy. As another benefit, the MA-ABE scheme [26] that we use handles a large universe of attributes at meager operational cost, which leads to a highly scalable key management. We show in Figure 7 the mathematical representation by extending the case of a single authority [16] to that of multiple authorities.

Crypto Module contains three critical sub-units: *Key Generation Unit, Encryption Unit and Decryption Unit*.

Key Generation Unit. Key Generation Unit implements two functions: one for merging secret keys provided by the user to produce a final merged secret key during decryption of the EHR document, the second for providing encryption keys (public keys) of relevant authorities during encryption decryption. Users obtain there secrets from their respective authorities, which control and manage their attributes. For this, the Key Generation unit exposes an endpoint for the registered authorities by which they can generate user keys by providing their respective secret keys.

Encryption Unit. After the EHR document is modified or generated by the user, the respective encryption the policy is fetched from the cloud server, which is used to encrypt the EHR a document by simultaneously communicating with the key generation unit to obtain the public keys required for encryption. For efficiency, each EHR document is first encrypted by symmetric encryption by randomly generating an encryption key, which is then encrypted with MA-ABE and stored together with the EHR document.

The ciphertext of an EHR document is represented as:

$$\mathsf{Enc}(doc) := (\mathsf{Enc}_{PK}^{\text{ma-abe}}(k), \mathsf{Enc}_{k}^{\text{aes}}(doc))$$

where an EHR document (doc) is initially encrypted with symmetric key encryption ($\mathsf{Enc}^{\text{aes}}$) and then the symmetric encryption key k is encrypted with MA-ABE ($\mathsf{Enc}^{\text{ma-abe}}$) where PK denotes the public key.

Decryption Unit. Encrypted EHR document is decrypted by using secret keys provided by the user with the use of Key Generation Unit to merge them and produce a final secret key, to provide to Decryption Unit. If the user is provided with a "write" decision, the modified EHR document is encrypted by calling the encryption unit again. Finally, the ontology is updated by creating a new node for storing the encrypted EHR document on the cloud server.

VI. Multi-Authority EHR Application

To validate the feasibility of our framework, we first implemented our framework. In particular, we implemented attribute-based access control to provide users with the precise access they are qualified based on their attributes. We have also implemented multi-authority attribute-based encryption to prevent data leaks and threats. By employing MA-ABE in encrypting medical records, we ensure that aggregation of decryption keys from multiple users does not lead to a security flaw that increases the decryption power, all at a meager operational cost.

Then, we developed a prototype EHR application on top of our framework. This application is an open-source web application. The application has been developed based on the principles of micro-service architecture; each sub-module is an independent service with its functionality and together serves as a suite of services. This design pattern is highly reusable in developing other applications that require similar functionalities.

Our framework and Multi-Authority EHR Application have been developed by using complete open source tools, leading to meager cost.

A. Web-Based Application

The Multi-Authority EHR Application is a web-based application built on the principles of micro-services based architecture with Model-View-Controller (MVC) architecture. The application can be divided into two independent services.

- *Front-End service.* It is the user interface for the user to interact with the application developed using HTML, CSS, Javascript and jQuery. Whenever the user interacts with the Front-End service of the application, REST with JSON call occurs, which communicates with the Backend service and retrieves a response displayed to the user.
- *Back-End service.* It is the core of our application framework, which implements attribute-based access controllers and multi-authority attribute-based encryption. The complete backend of the application is developed in Java using Spring Framework.

We now describe the independent services which constitute the backend of our application.

Ontology Query Service. In order to extract data from the ontology (i.e., fetch user, EHR field level and authority related attributes), this was service was built to act as a the bridge between the backend of the application and RDF/OWL ontology.

Open-source OWL API was used to build this service. Written in the Java programming language, OWL API provides a reference implementation for creating, manipulating and serializing OWL Ontologies. It also provides easy implementations for querying the ontology for extracting attributes and properties of entities.

Rule Based Engine Service. To help attribute-based access controller, this service takes the extracted attributes and matches them against the policies defined in SWRL rules. The service implements a reasoner and provides access decisions based on attributes matched against policies. This service can also be used to add, delete or modify exiting SWRL rules.

We implemented this service using SWRL API, which is a JAVA based open-source library working with the OWL-based SWRL rule and SQWRL query languages.

MA-ABE with Symmetric Encryption Toolkit. In order to prevent any data leaks and threats, this service was built. This service is the most crucial part of the Multi-Authority EHR Application, whose function is to provide an encryption mechanism using MA-ABE. This service implements Multi-Authority Attribute Based Encryption (MA-ABE) [26] with Symmetric Encryption for encryption purpose in this module by using Charm-Framework [1] to create a complete cryptographically secure encryption toolkit for our framework.

MA-ABE associates a document to be encrypted with a particular, unique decryption policy and user's secret key with their respective attributes. This decryption policy is a logical expression of attributes of the entities involved in the document usage. The users whose set of attribute values satisfies the decryption policy are allowed to decrypt and use the document.

This toolkit was developed in python and was wrapped around Java's Spring Boot service for fast and easy deployment of the toolkit as a service. Developed MA-ABE with symmetric encryption toolkit provides us with five command-line tools *MaAbeGlobalSetup, MaAbeAuthSetup, MaAbeKeygen, MaAbeEncrypt, and MaAbeDecrypt*. Initially, in the framework, global public parameters are generated by *MaAbeGlobalSetup*. Based on global public parameters, each authority in the system generates their respective public and master secret key pairs using *MaAbeAuthSetup*. The *MaAbeEncrypt* is used to encrypt the file first with symmetric encryption using a randomly generated encryption key, which is then encrypted with MA-ABE. *MaAbeDecrypt* is used to decrypt the encrypted file if the user provides valid secret keys. These commands are automated in our system; that is being called by backend service using REST with JSON. Then, by using *MaAbeKeygen*, each authority provides users with their secret keys associated with their respective attributes.

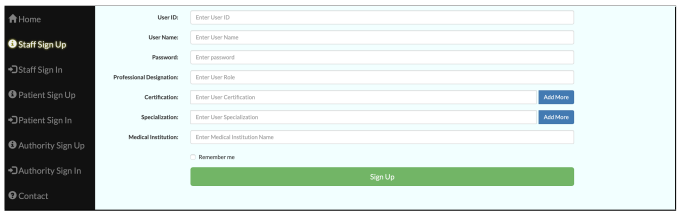


Fig. 8: Staff Sign Up

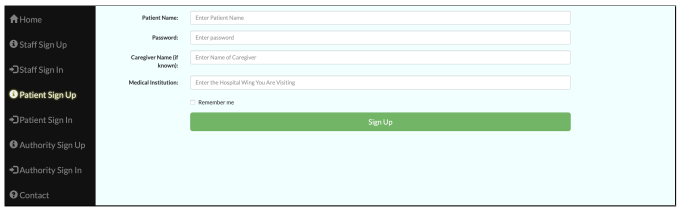


Fig. 9: Patient Sign Up

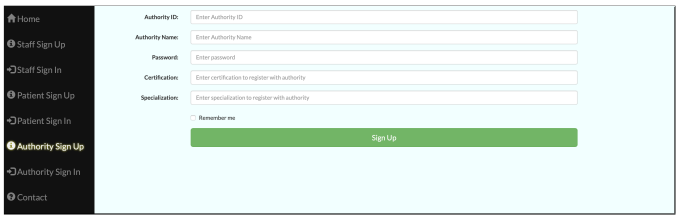


Fig. 10: Authority Sign Up

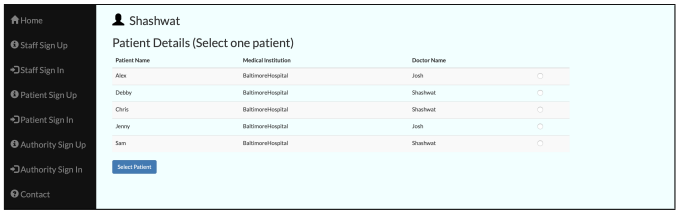


Fig. 11: Patient Selection

B. Knowledge Representation & Management

To develop and design the *Multi-Authority EHR Ontology*, Protege[1] tool was used. Developed by the Stanford Center for Biomedical Informatics Research, Protege is an open-source, free, knowledge graph editor and management system. Protege supports easy ontology development and maintenance. It supports a huge pallet of features, with support for visual representation in terms of a graph. The ontology explanation feature supports debugging and helps reduce or completely remove any inconsistencies in the ontology design and development. The support for refactoring ontologies helps in merging ontology, entity renaming, etc. Multi-Authority EHR Application uses ontology to carry out access decisions based on attribute based access control and fetches attributes for encryption toolkit.

C. Application Flow and Prototype

The application provides interface for users, patients and authorities to register themselves as well as to sign up in the system to access the EHR documents.

As shown in Figure 8, medical users can register themselves by providing their unique id, name, medical certifications, specializations, and the associated Medical Institution. They can add additional medical certifications or specializations if they have more than one. Once registration has been completed, an automated message will be sent to the authorities controlling those respective certifications, specialization, and medical institutions to verify them and create their secret keys. If the authorities authorize the user, the user successfully gets registered with their keys stored in a standard location on the cloud server, and a new entry is created for the user in the Multi-Authority EHR Ontology.

Figure 9 shows the registration interface for patients. The application requires their key attributes like name, caregiver name and medical institution they are associated with, and other key attributes. After successful registration, a new entry is created in the ontology.

If an authority wants to register themselves, an endpoint is exposed only to them, which provides the authority with an interface, as shown in Figure 10. The authority registers itself by providing the authority's unique id, name, certification, and specializations they control and monitor. After registering successfully, a new entry is created in the ontology.

If a registered caregiver tries to sign in to our system, the application runs Access Handler to determine the patients to which the user is qualified. This is done by the backend service, where the ontology query service extracts user attributes and provides them to rule-based engine service. Figure 11 shows the list of patients to which the user has access to after evaluating the user attributes against the policies.

After selecting a patient, the access handler identifies the type of access either 'read' or 'read/write'. They can view all the fields to which the access is permitted. The writing action would be absent for the fields where write access is not permitted but only read is permitted. When a patient is selected, the Document Processor & Crypto Module is invoked to fetch the EHR to which the user has access and decrypt the EHR documents. This is done by MA-ABE with Symmetric Encryption Toolkit wrapped around Java service.

Figure 12 shows an example of a registered Senior Doctor accessing the system. If the doctor wishes to change any of the fields he has access to, he can do it by clicking the edit button. Figure 13 shows the action of changing the EHR record. Also, after changing the record, Document Processor & Crypto Module encrypts the document according to respective policy and stores it on the cloud server. When a registered patient tries to access the application, the patient can only view the EHR records as the access handler provided only read access. Also, when an authority sign in our application through the endpoint exposed to them, they get access to the medical certifications and specialization they control and monitor to edit.

In this way, the Multi-Authority EHR Manager harnesses

[1]protege.stanford.edu

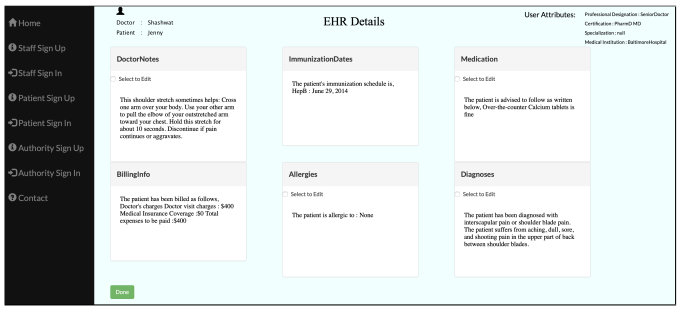


Fig. 12: Senior Doctor EHR View

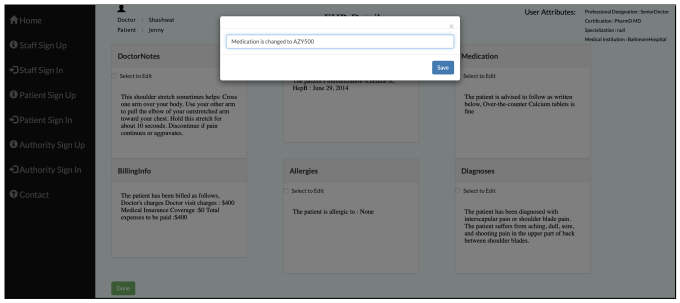


Fig. 13: Modify EHR View

the semantic web and attribute-based technologies in a multi-authority environment overcoming the bottlenecks suffered by previous works.

VII. Framework Performance Evaluation

To evaluate the scalability and performance of our system, this section provides a detailed analysis of all independent operations of our application when the ontology is stored on edge. This analysis is done to show the overhead caused by each of the actions in our application. Further, this section compares our systems' performance when all the data was stored on an OpenStack cloud platform and Edge.

Figure 14 presents the detailed time analysis of each of the independent operations that exist in our application. As shown, operation *User Registration* is measured in two parts, where the first part is the user's credentials being transferred from Front-End service to Backend service, and then the second part is the user's credentials being stored in the Ontology as a new triple by the Back-End service. Patient and authority registration works in a similar mechanism; hence apart from minor fluctuations, the timing remains constant.

Operation *Registered User Access Application*, is a measure of four actions, first user's credentials being transferred from Front-End service to Backend service and getting verified that is if the user qualifies the initial authentication phase in the Access Handler, second is the measure of the time taken by access handler to provide access decision for the list of patients the user's has access to based on the policies. The third action is when the user selects a patient; then the access handler provides access decision to which of the fields of the EHR document the user has access to and with what kind of access to those fields. Fourth action is when the fields are retrieved and decrypted by the Document Processor & Crypto Module. Operation *Storing Modified EHR* is a measure of two sub-operations, first when the user modifies the EHR field and saves it. Second, the modified field is received at the Back-End service, encrypted and stored by Document Processor & Crypto Module.

Patient's sign-in works similar to user sign; however, only three of the user's actions are credentials getting transferred from Front-End service to Backend service for verification and then access handler to provide access decision to which of the fields of the EHR document does the patient has access. Finally, fields get retrieved and decrypted. The response time for these three sub-operations is shown in *Registered Patient Access Application* operation.

Operation	Sub-Operations	Response Time (sec)
User Registration	• Credentials received • Credentials added to ontology creating new triple	0.00237
Registered User Access Application	• Credentials received • Credentials verification	0.00123
	• Running Access Handler • Provided Authorized Patient List	0.00313
	• Selected Patient Received • Running Access Handler to provide Access Decisions	0.33015
	• Fetching EHR fields, performing decryption	0.33023
Storing Modified EHR	• Modified EHR field received • Performing encryption • Storing encryption modified EHR	0.00234
Registered Patient Access Application	• Patient credentials received • Credentials verification	0.0011
	• Running Access Handler • Access Decision provided	0.3255
	• Fetching EHR fields, performing decryption	0.3256

Fig. 14: Performance Analysis of Independent Operations

Environment	Operation	Sub-Operations	Response Time (sec)
OpenStack Cloud Platform	Running Access Handler	Credentials received Querying ontology on OpenStack Running Access Handler	1.857
Organizational Edge	Running Access Handler	Credentials received Querying ontology on Edge Running Access Handler	0.572

Fig. 15: Comparison of our system's performance between OpenStack & Edge

Figure 15 presents with performance analysis of our system for the operation *Running Access Handler* when all the data with the knowledge graph is stored on the OpenStack cloud platform and compares with the data being on the Edge.

VIII. Conclusion & Future Work

Paper proposes a semantically rich and cryptographically secure framework for the EHR's realized in a multi-authority setting, addressing the challenges and bottlenecks suffered by current systems. The framework overcomes the problems suffered by previous works and transfers all the service management overhead from the patient to the different authorities in the environment. The approach implements attribute-based access control to ensure the right access privileges and further tightens the security by implementing Multi-Authority Attribute Based Encryption. Using this system, organizations can handle the electronic health records securely. Finally, Multi-Authority EHR Application is developed to test and evaluate the claims, realized in a multi-authority environment and .

As part of future work, currently, the system allows for revocation of the user's attribute only through brute force.

However, more enhanced and optimized methods are needed to tackle the revocation of attributes. Also, patient end delegation is not incorporated in the system; patients should always retain the right to revoke access privileges and their corresponding decryption key. This could be one area of expanding this project to handle temporal access. Incorporating keyword searches over the encrypted EHR data could also be an area for expanding the project. Hence, this research project has a substantial future score.

ACKNOWLEDGMENT

This research was supported by the Office of Naval Research and the National Science Foundation. We thank Adam Aviv, Travis Mayberry, and Daniel Roche, and members of the Ebiquity Research Group for their vital feedback.

REFERENCES

[1] Joseph A. Akinyele, Christina Garman, Ian Miers, MatthewW. Pagano, Michael Rushanan, Matthew Green, and Aviel D. Ru-bin. 2013. Charm: a framework for rapidly prototyping cryp-tosystems. Journal of Cryptographic Engineering 3, 2 (01 Jun 2013), 111–128.

[2] Joseph A. Akinyele, Christoph U. Lehmann, Matthew D. Green, Matthew W. Pagano, Zachary N.J. Peterson, and Aviel D. Ru-bin. 2010. Self-Protecting Electronic Medical Records Using At-tribute-BasedEncryption.

[3] Athenahealth. (Accessed April 1, 2020). https://www.athenahealth.com.

[4] Arshdeep Bahga and Vijay Krishna Madisetti. 2013. A Cloud-based Approach for Interoperable Electronic Health Records (EHRs). IEEE Journal of Biomedical and Health Informatics 17 (2013), 894–906.

[5] Josh Benaloh, Melissa Chase, Eric Horvitz, and Kristin Lauter. 2009. Patient Controlled Encryption: Ensuring Privacy of Elec-tronic Medical Records.

[6] David Blumenthal.2010. Launching HITECH.New England Journal of Medicine362, 5 (2010)https://doi.org/10.1056/NEJMp0912825

[7] Melissa Chase and Sherman S.M. Chow. 2009. Improving Priva-cy and Security in Multi-authority Attribute-based Encryption. In Proceedings of the 16th ACM Conference on Computer and Communications Security (Chicago, Illinois,nUSA) (CCS '09). ACM, New York, NY, USA, 121–130. https://doi.org/10.1145/1653662.1653678

[8] CureMD. (Accessed April 1, 2020). https://www.curemd.com.

[9] S. Dixit, K. P. Joshi, and S. Geol Choi. 2019. Multi Authority Access Control in a Cloud EHR System with MA-ABE. In2019 IEEE International Conference on Edge Computing (EDGE). 107–109.

[10] EpicHealthServices https://www.epichealthservices.com.

[11] Centers for Disease Control and Prevention. [n.d.]. HIPAA privacy rule and public health. Guidance from CDC andthe US Department of Health and Human Services. ([n. d.]).

[12] GE Healthcare.(Accessed April 1, 2020). https://www.gehealthcare.com.

[13] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. 2006. Attribute-based Encryption for Fine-grained AccessControl of Encrypted Data. InProceedings of the 13th ACM Conference on Computer and Communications Security(Alexandria, Virginia, USA)(CCS '06). ACM, New York, NY, USA, 89–98. https://doi.org/10.1145/1180405.1180418

[14] L. Ibraimi, M. Asim, and M. Petković. 2009. Secure management of personal health records by applying attribute-basedencryption. InProceedings of the 6th International Workshop on Wearable, Micro, and Nano Technologies for PersonalizedHealth. 71–74. https://doi.org/10.1109/PHEALTH.2009.5754828

[15] Xin Jin, Ram Krishnan, and Ravi Sandhu. 2012. A Unified Attribute-Based Access Control Model Covering DAC,MAC and RBAC. InData and Applications Security and Privacy XXVI, Nora Cuppens-Boulahia, Frédéric Cuppens, andJoaquin Garcia-Alfaro (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 41–55.

[16] Maithilee Joshi, Karuna Joshi, and Tim Finin. 2018. Attribute Based Encryption for Secure Access to Cloud Based EHRSystems. 932–935. https://doi.org/10.1109/CLOUD.2018.00139

[17] M. Li, S. Yu, N. Cao, and W. Lou. 2011. Authorized Private Keyword Search over Encrypted Data in Cloud Computing. In2011 31st International Conference on Distributed Computing Systems. 383–392. https://doi.org/10.1109/ICDCS.2011.55

[18] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou. 2013. Scalable and Secure Sharing of Personal Health Records in CloudComputing Using Attribute-Based Encryption.IEEE Transactions on Parallel and Distributed Systems24, 1 (Jan 2013),131–143. https://doi.org/10.1109/TPDS.2012.97

[19] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou. 2013. Scalable and Secure Sharing of Personal Health Records in CloudComputing Using Attribute-Based Encryption.IEEE Transactions on Parallel and Distributed Systems24, 1 (Jan 2013),131–143. https://doi.org/10.1109/TPDS.2012.97

[20] Hans Löhr, Ahmad-Reza Sadeghi, and Marcel Winandy. 2010. Securing the E-health cloud.IHI'10 - Proceedings of the1st ACM International Health Informatics Symposium, 220–229. https://doi.org/10.1145/1882992.1883024

[21] K. D. Mandl, P. Szolovits, and I. S. Kohane. 2001. Public standards and patients' control: how to keep electronic medicalrecords accessible but private.BMJ322 (Feb 3 2001), 283–7.

[22] Tim Mather, Subra Kumaraswamy, and Shahed Latif. 2009. Cloud Security and Privacy: An Enterprise Perspective onRisks andCompliance. (01 2009).

[23] Deborah L McGuinness, Frank Van Harmelen, et al.2004. OWL web ontology language overview.W3C recommendation10, 10 (2004), 2004.

[24] Shivaramakrishnan Narayan, Martin Gagné, and Reihaneh Safavi-Naini. 2010. Privacy preserving EHR system usingattribute-based infrastructure.Proceedings of the ACM Conference on Computer and Communications Security, 47–52.https://doi.org/10.1145/1866835.1866845

[25] PracticeFusion.(Accessed April 1, 2020).https://www.practicefusion.com.

[26] Yannis Rouselakis and Brent Waters. 2015. Efficient Statically-Secure Large-Universe Multi-Authority Attribute-BasedEncryption. InFinancial Cryptography and Data Security, Rainer Böhme and Tatsuaki Okamoto (Eds.). Springer BerlinHeidelberg, Berlin, Heidelberg, 315–332.

[27] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman. 1996. Role-Based Access Control Models.Computer29, 2 (Feb. 1996), 38–47. https://doi.org/10.1109/2.485845

[28] Matthew A. Scholl, Kevin M. Stine, Joan Hash, Pauline Bowen, L. Arnold Johnson, Carla Dancy Smith, and Daniel I.Steinberg. 2008.SP 800-66 Rev. 1. An Introductory Resource Guide for Implementing the Health Insurance Portability andAccountability Act (HIPAA) Security Rule. Technical Report. Gaithersburg, MD, United States.

[29] N. K. Sharma and A. Joshi. 2016. Representing Attribute Based Access Control Policies in OWL. In2016 IEEE TenthInternational Conference on Semantic Computing (ICSC). 333–336. https://doi.org/10.1109/ICSC.2016.16

[30] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu. 2016. Edge Computing: Vision and Challenges.IEEE Internet of Things Journal3, 5 (Oct 2016), 637–646. https://doi.org/10.1109/JIOT.2016.2579198

[31] S. Yu, C. Wang, K. Ren, and W. Lou. 2010. Achieving Secure, Scalable, and Fine-grained Data Access Control in CloudComputing. In2010 Proceedings IEEE INFOCOM. 1–9. https://doi.org/10.1109/INFCOM.2010.5462174, Vol. 1, No. 1, Article . Publication date: January 2021.