Secure Cloud EHR with Semantic Access Control, Searchable Encryption and Attribute Revocation

Abstract—To ensure a secure Cloud-based Electronic Health Record (EHR) system, we need to encrypt data and impose field-level access control to prevent malicious usage. Since the attributes of the Users will change with time, the encryption policies adopted may also vary. For large EHR systems, it is often necessary to search through the encrypted data in real-time and perform client-side computations without decrypting all patient records. This paper describes our novel cloud-based EHR system that uses Attribute Based Encryption (ABE) combined with Semantic Web technologies to facilitate differential access to an EHR, thereby ensuring only Users with valid attributes can access a particular field of the EHR. The system also includes searchable encryption using keyword index and search trapdoor, which allows querying EHR fields without decrypting the entire patient record. The attribute revocation feature is efficiently managed in our EHR by delegating the revision of the secret key and ciphertext to the Cloud Service Provider (CSP). Our methodology incorporates advanced security features that eliminate malicious use of EHR data and contributes significantly towards ensuring secure digital health systems on the Cloud.

Index Terms—Attribute Revocation, Searchable Encryption, Electronic Health Record, Knowledge Graph (Ontology), Cloud Computing, Cloud Security

I. INTRODUCTION

Healthcare organizations are increasingly adopting cloud-based technologies to maintain their digital records efficiently. A cloud-based Electronic Health Record (EHR) service allows centralizing patient data and using the advantages of elasticity and scalability of a cloud infrastructure [5], [34], [35], [40]. The cloud also offers a highly supportive atmosphere for efficiently handling the load [2]. Moreover, cloud services are usually a more economical solution than others when they develop a technology infrastructure to deploy their services. It has become much more popular due to the pay-as-you-go concept, which incentivizes customers to pay only for what they want and how much they use.

Although the Cloud offers many advantages, it also continues to pose unique risks to healthcare organizations in terms of data privacy and security. With the recognition of the security risks, all healthcare organizations must comply with Health Insurance Portability and Accountability Act (HIPAA) [14], [50] and Health Information Technology for Economic and Clinical Health (HITECH) [49] privacy guidelines set by regulatory authorities. Failing to comply with the acts can be financially devastating to an organization. Noncompliance carries a range of penalties depending on the degree of misconduct which can result in hefty fines. Violations can also lead to an arrest. As a result, an EHR solution must abide with all applicable laws and regulations and also allow an easy and seamless exchange of patient information.

A. Motivation

EHR system with semantic access control. Recently, Walid et al. [55] proposed a cloud-based EHR system that offers a semantically rich, policy-driven mechanism that employs Attribute-Based Access Control (ABAC) [22] to evaluate users’ entrance to the system.

The architecture of their system was created using Semantic Web Technologies [7]. By referencing the HIPAA knowledge graph (ontology) developed in [23], they created a HIPAA-consistent knowledge graph. This way, the system systematically addressed the issues with the aforementioned compliance problem.

In particular, the system used a knowledge graph to derive user attributes and the EHR fields based on the type of request. The knowledge graph is queried using SPARQL, and it entails complete details of individuals in the organization and their associated unique attributes. The unique attribute control various access to different fields of an EHR. Thus, each individual has distinct access to a patient EHR.

The system also allowed the client to search through encrypted data based on keyword queries, without needing to download and decrypt all encrypted data from the cloud.

Disadvantages of the work [55]. Although their system has many good features, it still has a few disadvantages. In this work, we want to improve the system by resolving those disadvantages.

First of all, the system uses two different encryption schemes, one for searchability and the other for data encryption. It often becomes more cumbersome in large systems to have multiple schemes within a system, as more keys and policies may have to be maintained. A system would be better if it uses a single encryption scheme.

Second, although the system provides the search feature, the search time is quite slow. In the presence of Big data, it usually
requires a lot of time and analysis to search through the data to locate specific patients with certain diseases or conditions. We want to speed up the search time.

Most importantly, the system doesn’t provide revocation. It is essential that an attribute-based EHR system have an attribute revocation feature, since the organization policies and the user attributes keep changing with time. For example, a physician might have been promoted, so its attributes would change. Likewise, an employee might have moved to another department, or an employee leaves the organization, so its unique attributes must be revoked from the system. Often the organization policies also vary with time, which requires some attributes to be revoked.

B. Our Work

In this paper, we improve the EHR system in [55] by resolving the issues mentioned above. Below, we overview our system.

Underlying ABE encryption. We use the revocable, searchable ABE scheme introduced by Wang et al. [59]. The scheme provides both searchability and data encryption, which fits our purpose. The scheme also allows us to outsource the computation to the cloud, which we describe next.

Outsourcing the computation to cloud. In general, our system requires fewer computations as most functions are safely delegated to the cloud service provider (CSP). This is achieved by splitting each user’s secret into two keys and having one key uploaded to the CSP while the other key is kept secret to the user.

For example, since each user has a dedicated private key stored within the CSP, partial decryption is delegated to the CSP. The output from the partial decryption still hides the plaintext from the CSP and outsiders. However, given the partial decryption from the CSP, the user can recover the plaintext record with much less effort.

Faster search time. Owing to the presence of Big Data, searching through encrypted data requires utmost attention. The search time can also be improved, thanks to the outsourcing framework.

In particular, to search for a keyword in the encrypted index for the EHR database, the user first creates a token connected with a keyword query. For privacy, the token hides the keyword from the CSP.

Once the CSP gets the search token, the CSP uses the token to run the search algorithm over the ciphertexts to see which ones have the privately linked keyword (s). When the indexes, keywords, and the user’s attributes are set to meet the ciphertext access control policies, the cloud service retrieves the search results. The encoded version of the message containing the keyword is sent back to the recipient.

Our technique enables keyword search with substantially reduced network latency and client-side computing costs compared to prior work [55].

Attribute revocation. The knowledge graph records all users and patients in our framework, considering all attribute changes in the system along with the associated EHR fields. So, the knowledge graph functions to revoke unwanted user attributes and help to protect patient privacy.

The outsourcing framework also benefits when the system performs revocation. In particular, when a user attribute is revoked, the revocation takes place only in the CSP level. This works because a user secret is split into two keys, and revoking only one key would still disable the function of a user secret. The secret key that lies within the user remains firm during the entire revocation process, which greatly simplifies the revocation process.

Edge computing. The term edge computing [51] refers to the need to analyze data locally before sending it to the cloud, and we have followed this principle in our system. Inside the organizational periphery, which we refer to as the edge in our system, we enforce an access control mechanism on the records. All users are checked only within the organization’s borders, preserving their anonymity. Within the organization limit, we have implemented a robust encryption approach that protects data integrity from privacy risks until moving it to the cloud. As a result, the frontier continues to be a formidable barrier to data protection.

Threat model. Cloud users, while storing their data on the cloud, usually categorize CSP on one of these threat models: the Honest-But-Curious (HBC) adversary model, where CSP run the programs and algorithms correctly, but might look at the information passed between entities; or the malicious adversary model, where providers behave in an arbitrary manner that may be hostile to the cloud customer [42]. We have considered the HBC threat model for our approach since cloud users trust the functionality of their applications running on the cloud, but they may not fully trust the CSP whose dataset is stored in distant Cloud data centers. First of all, Clouds may be exposed to tainted workers who fail to adhere to data protection standards. Secondly, cloud applications may be subject to external cyberattacks, and cloud users may not be aware of the possible repercussions on their data security when such invasions occur. The users worry that the CSP might attempt to decrypt the data by analyzing it thoroughly or monitoring data traffic between users. We presume our framework to withstand a compromised user attempting to decode ciphertext with her decryption key and gain knowledge. We also assume our system to be resilient in the face of a user alliance trying to crack the ciphertext with decryption keys that no single member of the coalition can decode with her decryption key.

C. Organization

The remainder of the paper is structured as follows – We discuss related work in Section II, preliminaries in section III, system architecture in Section IV, system implementation in Section V, and conclusions in Section VI.
II. RELATED WORK

A. Electronic Health Record System

Digital health record systems are commonly employed to enhance hospital services, improve treatment efficacy, and reduce premiums [16], [25]. EHR records a patient’s vital stats, diagnoses, medications, immunization history, laboratory and radiology reports, doctor notes, and other medical facts along with the patient’s details. An EHR system provides several benefits such as accurate documentation, disease tracking, data sharing, statistical analysis, and so forth. Consequently, security and privacy concerns have hampered the spread of the EHR system, and they have received increasing focus in current years [35]–[37], [48]. Narayan et al. [45] recommended using ABE to protect the privacy of EHR data from outside threats, as well as the CSP. Fatos et al. [60] originally presented a multi-user fuzzy keyword search method that supported fine-grained permission restriction over encrypted data.

Unfortunately, most proposed solutions fall short in providing controlled access, encryption device, searchable encryption, and attribute revocation. Furthermore, the majority of the accessible application is licensed, making them costly to use. In this circumstance, our research effort tries to develop an open-source, low-cost EHR management system that can provide advanced data privacy and protection levels.

B. Regulatory Policies

Patient data is secured in the United States under some statutes; the most notable is the HIPAA Act. Electronic health information (ePHI) [11] is the name given to the information about one’s health that is protected by these rules. The Health Information Technology for Economic and Clinical Health Act (HITECH) allows sharing ePHI while still requiring HIPAA privacy and protection laws to be applied more strictly and thoroughly [1]. These rules, on the other hand, make no mention of encryption principles or algorithms. Furthermore, data encryption in data access control and transfer is defined as addressable rather than mandatory. This left space for different definitions and then became a source of debate regarding sharing ePHI. Cloud-based EHR services in the United States are required to comply with these regulatory standards and ensure enhanced data protection combined with the seamless user experience that cloud services offer. This also requires implementing strict access control mechanisms to provide unauthorized access by any user is prohibited by their EHR.

C. Semantic Web Technologies

We have used Semantic Web technologies to develop our system’s knowledge graph and the reasoning component of our system. These enable us to build the schema using W3C standardized languages that support our design requirements, including interoperability, sound semantics, Web integration, and the 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 of incorrect contexts. Semantic Web technologies include languages such as Resource Description Framework (RDF) [26] and Web Ontology Language (OWL) [43] for defining ontologies and describing meta-data using these ontologies as well as tools for reasoning over these descriptions.

Our most fundamental requirement is for a representation that supports interoperability at both the syntactic and semantic levels. OWL has 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 is 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 and Cloud, which is becoming the dominant technology for today’s digital health systems. These technologies can be used to provide common semantics of service information and policies enabling all agents who understand essential Semantic Web technologies to communicate and use each other’s data and Services effectively. 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.

D. Attribute-Based Encryption

ABE [17], introduced by Sahai and Waters, has been one way to ensure data security and eliminate risks. In ABE [17], the data is encoded using a set of attributes, and the private key is defined using a different set of attributes. Based on the threshold parameter, the ciphertext can only be deciphered if the two sets of attributes overlap. One of the EHR system security developments was known as ABE [3], [6], and [45]. It has been further divided into ciphertext-policy ABE (CP-ABE) [8] and key-policy ABE (KP-ABE) [4] due to lack of expressibility. The secret key is coupled with an attribute set in CP-ABE [8], and the ciphertext is paired with an access policy. In most cases, the policy is defined as a Boolean formula with a specific set of attributes. A secret key may decrypt a ciphertext if the attributes set match the access policy, while the whole scenario is reversed in the KP-ABE scheme.

CP-ABE [8] is considered more effective for authentication in the cloud because an individual ciphertext defines a policy that explicitly specifies attributes that data users must hold for the encryption process. Joshi et al. [24] developed attributed-based access control (ABAC) that is semantically enriched in accessing data leveraging CP-ABE [8]. Their model evaluates access categories based on user attributes and EHR fields. The national hub controls both EHR secure entry and distribution.
E. Attribute-Based Encryption With Attribute Revocation

Since the user’s attribute can vary significantly over time, attribute revocation is crucial in ABE frameworks. Perretti et al. [46] were the first to implement attribute revocation, which they accomplished by a timed keying process. Each attribute had an expiration time in the system, so authority centers had to reprint revised keys regularly. The authority center had to cease releasing and modifying the current edition of the attribute to revoke an attribute in the scheme. Bethencourt et al. [8] later expanded Perretti’s work where there was a single expiration time connected with the user’s private key. Boldyreva et al. [9] proposed a revocable KP-ABE scheme that improved on their previous revocable IBE. Wang et al. [57], [56] presented two explicitly revocable CP-ABE frameworks based on bilinear and multilinear maps, accordingly. Several ABE systems involving instant attribute revocation were suggested in current history. Yu et al. [62] and Ibraimi proposed et al. [21] the CP-ABE scheme that employs a semi-trusted proxy server to execute instant attribute revocation. Their approach shifted the authority’s responsibilities to the proxy server, significantly reducing the authority’s burden. They have, nevertheless, been unable to obtain fine-grained access control. Furthermore, as the number of users increases rapidly, the proxy server’s update work skyrocket. Li et al. [31] devised an effective CP-ABE scheme of user revocation with a lower computing expense. Several other schemes can be seen in [66], [47], [63], and [39].

Computational efficiency is another consideration in the latest ABE schemes. Outsourced decoding technologies can help decrease the user’s computing load. Green et al. [18] first proposed an effective ABE scheme that facilitates outsourced decoding. The bulk of decryption activities are done by the CSP using the users’ keys. Zhou et al. in [65] suggested an optimized data management system centered on mobile devices, in which portions of the encryption and decryption processes were safely delegated to the CSP without sensitive data leakage. Li et al. [30] proposed an ABE scheme including complete verification for outsourced decryption, which addresses the problem of ensuring the accuracy of outsourced decryption for unauthorized individuals. The scheme implemented in our systems seems to be perfect compared to [20], [61], [54], and [41].

F. Keyword Search Over Encrypted Data

Fast and efficient searchability is required for any EHR system, particularly in the movement of evidence-based healthcare, because doctors have a limited time in which to make judgments. Dawes et al. [13] mentioned that time constraints are the most significant factor impeding computer systems in medical practice. Physicians indicated that response time is one of the obstacles to EHR system adoption in another study by Holden et al. [19]. Searchable encryption (SE) thus remains to be of the utmost important feature in EHR systems.

SE is an encryption technique that allows users to scan for keywords in cyphertext without revealing the keywords. Song et al. [52] first devised a realistic SE scheme focused on symmetric cryptography, establishing a significant standard for keyword search on encrypted data. Boneh et al. [10] later pioneered SE research into public-key cryptography. Following that, numerous SE schemes were developed to improve search performance, security issues, and search functionality [12], [15], [27], [32], [53]. Attribute-based keyword search, which combines ABE and SE properties, has seen many hypes in current history that can be seen in [28], [29], [33], [44], [58], [64], and [38].

III. PRELIMINARIES

Let $\lambda$ be the security parameter.

A. Revocable, Searchable ABE

In this section, we describe revocable, searchable attribute-based encryption scheme.

Syntax. Let $\mathcal{X}$ be the attribute universe. A revocable, searchable ABE consists of the following algorithms:

1. Setup($1^\lambda$, $\mathcal{X}$) $\rightarrow$ $(mpk, msk, msvk)$. The setup algorithm gets as input the security parameter $\lambda$, the attribute universe $\mathcal{X}$. It outputs the public parameter mpk, the master secret key msk, and the master secret version key.

2. KeyGen(msk, $\mathcal{X}$, msvk, $\Delta$) $\rightarrow$ $(sk^1_\mathcal{X}, sk^2_\mathcal{X})$. The key generation algorithm gets as input msk, $\mathcal{X}$, msvk and a set of attributes $\Delta$. It outputs a pair of secret keys $(sk^1_\mathcal{X}, sk^2_\mathcal{X})$.

3. Enc(mpk, msk, $\mathcal{X}$, $\mathcal{I}$, $\mathcal{W}$, $\mathcal{F}$, $\mathcal{M}$) $\rightarrow$ $ct_f$. The encryption algorithm gets as input mpk, a boolean formula $\mathcal{F}$ over $\mathcal{X}$, a set of keywords $\mathcal{W}$ and a message $\mathcal{M}$. It outputs a ciphertext $ct_f$.

4. Enclnd(mpk, $\mathcal{W}$) $\rightarrow$ $I_W$. The encrypted index algorithm gets as input mpk and a set of keywords $\mathcal{W}$. It outputs an encrypted index $I_W$ for $\mathcal{W}$.

5. Token(sk^1, $\mathcal{W}$) $\rightarrow$ $t_w$. The token generation algorithm gets as input the user secret key $sk^1$ and a query keyword $w$. It outputs a token $t_w$.

6. Test($sk^1_\mathcal{X}$, $I_W$, $t_w$) $\rightarrow$ 0/1. The test algorithm gets as input the cloud secret key $sk^1_\mathcal{X}$, the encrypted index $I_W$ and the user generated token $t_w$. If the embedded keyword in $t_w$ is contained in $I_W$, it outputs true; otherwise it outputs false.

Note that this algorithm can be performed by the cloud that holds the key $sk^1_\mathcal{X}$ when it receives the token $t_w$ for the user; the encrypted index $I_W$ is typically stored on the cloud server.

7. Decrypt-cloud($sk^2_\mathcal{X}$, $ct_f$) $\rightarrow$ $pd$. This algorithm gets as input the cloud secret key $sk^2_\mathcal{X}$ and the ciphertext $ct_f$. If $f(x) = 1$, it outputs partial decryption $pd$; otherwise, it outputs an error.

8. Decrypt-user($sk^1_\mathcal{X}$, $pd$) $\rightarrow$ $m$. Given the partial decryption, the user with $sk^1_\mathcal{X}$ will recover the message $m$.

9. Update-msvk(msvk, $\mathcal{X}$) $\rightarrow$ $\Delta_x$. This algorithm is run by the central authority to update the attribute $x$ when a
user with attribute $x$ is revoked. The algorithm updates the master secret version key for the attribute $x$, and also outputs $\Delta_x$ to be used for updating the master public key, the cloud secret key that is associated with attribute $x$, and ciphertexts associated with attribute $x$.

- Update-mpk($\text{mpk}, \Delta_x$). This algorithm updates the master public key $\text{mpk}$ using $\Delta_x$.
- Update-cloudkey($\text{sk}_x^2, \Delta_x$). This algorithm updates the cloud secret key $\text{sk}_x^2$ using $\Delta_x$.
- Update-ct($\text{ct}, \Delta_x$). This algorithm updates ciphertext $\text{ct}$ using $\Delta_x$.

Revocation Security. For a stateful adversary $A$ and security parameter $\lambda$, we define an experiment $\text{Expt}^\text{revoke}_A(\lambda)$ as follows:

$$\text{Expt}^\text{revoke}_A(\lambda):$$
$$f^* \leftarrow A(1^\lambda);$$
$$\text{mpk, msk, msvk} \leftarrow \text{Setup}(1^\lambda, \mathcal{X});$$
$$\langle m_0, m_1 \rangle \leftarrow A^\text{KeyGen}((\text{msk, msvk},)), \text{Update-msvk}((\text{mpk, msvk},))(\text{mpk});$$
$$b \leftarrow R\{0,1\};$$
$$\text{ct}_{f^*} \leftarrow \text{Enc}(\text{mpk, } f^*, m_b);$$
$$b' \leftarrow A(\text{ct}_{f^*})$$
If $b = b'$ output 1; otherwise output 0.

In the above, all queries $x$ that $A$ makes to oracle $\text{KeyGen}(\text{msk, msvk,} \cdot)$ should satisfy $f^*(x) \neq 1$. In addition, all queries $m_0$ and $m_1$ should have the same length.

A revocable, searchable ABE is said to be revocation secure, if for all polynomial adversary $A$, the probability $|\Pr[A(1^\lambda) = 1/2]|$ is negligible in $\lambda$.

Keyword-search Security. For a stateful adversary $A$ and security parameter $\lambda$, we define an experiment $\text{Expt}^\text{keyword}_A(\lambda)$ as follows:

$$\text{Expt}^\text{keyword}_A(\lambda):$$
$$\text{mpk, msk, msvk} \leftarrow \text{Setup}(1^\lambda, \mathcal{X});$$
$$x \leftarrow A(\text{mpk});$$
$$\langle \text{sk}_x^1, \text{sk}_x^2 \rangle \leftarrow \text{KeyGen}(\text{msk, msvk,} x);$$
$$\langle W_0, W_1 \rangle \leftarrow A^\text{Token}(\text{sk}_x^2, \cdot)(\text{mpk});$$
$$b \leftarrow R\{0,1\};$$
$$I_{W_b} \leftarrow \text{EncInd}(\text{mpk, } W_b);$$
$$b' \leftarrow A(\text{EncInd}(\text{mpk, } W_b));$$
If $b = b'$ output 1; otherwise output 0.

In the above, all queries $w$ to $\text{Token}(\text{sk}_x^2, \cdot)$ should satisfy $w \notin \{W_0, W_1\}$.

A revocable, searchable ABE is said to be keyword-search secure, if for all polynomial adversary $A$, the probability $|\Pr[A(1^\lambda) = 1/2]|$ is negligible in $\lambda$.

The scheme we use. In this paper, we use the scheme in [59] that satisfies both revocation security and keyword-search security.

IV. SYSTEM ARCHITECTURE

The entire framework is based on the principles of Edge computing [51]. It is divided into two sections, with the organizational boundary comprising the Authentication Module and Data Processing Module as shown in Figure 1. Since organizations control these two modules, they are known as trustworthy bodies. All users are authenticated inside the organizational perimeters, preserving their anonymity. The other section concerns an untrustworthy CSP. Before uploading data to the cloud, a rigorous encryption approach is implemented within the organizational border to protect data integrity from privacy risks. An attacker may also be sabotaging the CSP. In our system, we assume a compromised CSP will behave in an honest-but-curious manner [42].

Our framework has a diverse set of users, authorities, and data owners from various medical fields. A single CSP stores the EHRs, encrypted index file, and user’s secondary secret keys. The Authentication module performs a thorough check on any request to the framework. Each user is granted access rights based on attributes as determined by the organization’s policies. Patients have read access to all fields of their EHR.

Use cases. Whether users choose to read, write, revoke an attribute, or browse through encrypted EHRs, our framework has multiple use cases. A user first asks for access to the EHR system. The Authentication Module reviews the application by looking over the user attributes in the user graph and ABAC rules defined according to the individual company policy. If the attributes follow the guidelines of the company, access is granted.

Whenever a user modifies an EHR, the framework uses the Data Processing Module to encrypt the updated details of the accessed fields. The Attribute Control Center in this module supplies the user attributes during the process. The Key Production Unit provides encryption keys for re-encryption. The EHR ontology housed with the CSP is then modified with the ciphertexts. A similar operation is performed during a read request.

During the search process, the user enters the search keyword as a request. The Key Production Unit provides the keys used for searching. Using the search keyword and hidden keys, the Token Origination Unit creates a trapdoor. The trapdoor is then sent to the CSP, where it is compared to the encrypted Indexes. The search operation retrieves encrypted EHRs if there is a match. The user may then choose any particular EHR to decrypt.

Attribute revocation is entirely handled in the Data Processing Module. The user gives revoked attributes to the Attribute Center, which it stores and supplies to Cryptography Unit. The Key Production Unit provides the master key. The ciphertext and the secondary secret that lies with the CSP are then updated to account for the changes.

In the following sections, we will go through each sub-module in detail.

A. Authentication Module

Any login request passes through a comprehensive investigation in this module. The key policy behind the module is the ABAC. There are also several units within the module with critical functions. The user’s login information is at first checked in the database. If it passes, the sub-modules begin to perform their functions.
The Medical Center Ontology store all attributes of every person belonging to the healthcare organization. This helps any user getting unnecessary information and only information to complete their job functions. It also preserves EHR access-related information involving the particular field level access for all users and patients of the EHR system. For example, a nurse is given access only to the data stored in the Lab Results and Doctor Notes fields of an EHR to prevent privacy leakage. The ontology is built considering the HIPAA regulations.

Policy Controller holds the organization’s policies. It is often listed in terms of the attributes of the users. Different policies are set for various users that help to protect individual privacy.

Regulation Processor employs Semantic Web Rule Language (SWRL) to achieve access choices. It does it with the help of the Policy Controller and the output given by querying the Medical Center Ontology. The SWRL is unique for each individual in the EHR system. Regulation Processor also stores the user attributes and control policies for the data records and passes them to Cryptography Unit during the encryption-related process.

**B. Data Processing Module**

The Data Processing Module performs several critical functions in the EHR system. Several sub-modules within the module help data encryption, data decryption, search token generation, encrypted index creation, and attribute revocation. Since we have used a single scheme in all such operations compared to our previous work [55], it seems more straightforward and convenient.

The user attributes and the control policies stored within the Regulation Process are passed to the Attribute Control Center that supplies the user attributes for any operation. The Key Production Unit supplies the keys for encrypting or decrypting an EHR. Whenever a user modifies an EHR, the Attribute Control Center, and Key Production Unit function to encrypt the updated data. The EHR ontology housed with the CSP is then modified with the ciphertexts.

During a search operation, the user gives the search keyword in the form of a query. The same Key Production Unit supplies the keys for the operation. A trapdoor is later generated, as shown in Figure 2. The trapdoor allows scanning through the encrypted index, and it is given to the CSP and compared to the encrypted indexes. The user gets the corresponding EHRs if there is a match. The user may then decrypt any particular EHR with the help of the Cryptography Unit and Key Production Unit. For example, a doctor may want to find patients with covid symptoms for immediate treatment. So, the doctor puts a search query, and by processing it with the secret key, a trapdoor is generated. The trapdoor scans through the encrypted indexes and returns all the corresponding patients.

![Fig. 1. System Architecture](image1)

![Fig. 2. Token generation process](image2)
Attribute Revocation functions are also carried out within the Data Processing Module. The process is depicted as shown in Figure 3. A user provides the attribute to be revoked, and the Attribute Control Center accepts it. These operations are most common when a user leaves an organization, gets a promotion, or organization policies change, so the last attributes need to be revoked. Key Production Unit supplies the master key during the process. The process is completed when the ciphertext and the secret user key that lies with the CSP are updated. The updated private key is then used for decrypting the EHRs at any later time.

C. Cloud Service Provider

The EHR ontology, cryptographic index, and secondary secret key are stored in the CSP. The CSP is located outside the organization frontier. We consider it to be an honest-but-curious model [42]. The CSP not only runs the programs and algorithms correctly, but it can also examine the information shared in or out of the company. To address this, we enforce an authorization protocol on data inside the organizational periphery, which we refer to as the edge of our framework. As a result, users are checked only within the organization’s boundaries, preserving their privacy. While transferring data to the cloud, we introduced a robust encryption mechanism at the organizational edge to shield data from privacy risks.

The EHR ontology that lies within the CSP defines patients, users, and patient records in the medical domain. The nodes of an ontology store the EHR records. The ontology was built previously in our lab [23], and later it was updated to accommodate new enhancements: the knowledge graph stores users, patients, and EHR attributes. The EHR in our study has eight fields. Depending on the user and their attributes, different types of access are given. All such information is recorded within the ontology. The ontology further describes the roles and traits of health organization members and their various relationships.

2) Encrypted Index: The encrypted index file holds encrypted word tokens from each patient EHR along with the patient id. The file is needed for any search operation. Word tokens from each patient are extracted. The tokens are then preprocessed and then encrypted with the aid of the Cryptography Unit following the RSABE scheme [59]. The Key Production Unit supplies the public key during the process. The process is depicted in the Figure 4. The Attribute Control Center via the Regulation Processor obtains all the attributes from the Medical Center Ontology. All these functions are performed within the organization’s limit. The file is then stored within the CSP.

V. IMPLEMENTATION

The open-source EHR software is built using the Python Django web-based framework. The application facilitates field-level ABE and access to patient EHR. Due to the presence of Big Data, it is often necessary to search through encrypted data within a limited time and computations; such functions are also available in the software. The user attributes keep changing with time, and it is also allowed in our open-source software. Model-View-Controller (MVC) architectural concepts have been followed while designing the application.

The EHR framework allows doctors to treat their patients securely. It also provides the necessary features that are needed for regular operation. The ABAC controls the field-level access to the EHR. The patient data, along with the encrypted index, are encrypted using the RSABE scheme. Searchable Encryption functions are also allowed using the same scheme in the system. The EHR ontology and the Medical Center Ontology have been built using the Protege [protege.stanford.edu]. Protege is a management framework for knowledge that is open-source. The ontologies are queried using SPARQL with Apache Jena library. The knowledge graph is modified using the SWRL rules. Thus, our application espouses ABE, searchable encryption, attribute revocation, and semantic web for smooth operation.

A. Dataset Description

There are more than eleven thousand patient records in our framework. A patient record has several fields, and we have considered the standard eight fields in all records in our system. There are almost thirty medical users with different attributes. The attribute defines the role and the type of access to the EHR system. All operations on the patient records are performed within the organizational limit before they are kept on the cloud.
B. Evaluation

To evaluate the EHR framework, we developed a proof of concept prototype that is described as follows. The EHR management software allows users to treat their patients supporting several critical functions. Suppose a doctor named Sarah submits an access request. The request is comprehensively evaluated in the Authentication Module; username and password are checked with the database; Policy Controller checks the policies, and Medical Center Ontology provides the unique attributes. The Regulation Processor then processes all these pieces of information. If Dr. Sarah plans to decrypt the EHR of Thomas, a patient in the EHR system, the request is processed in the Cryptography Unit by obtaining the Keys from the Key Production Unit. A similar operation is performed for encrypting an EHR. To search through encrypted EHRs, Sarah provides a search query that is processed by the Token Origination Unit by obtaining the secret keys from the Key Production Unit. To revoke the attribute of another Junior Doctor named Jennifer, Sarah submits the revoke attribute information to the Attribute Control Center. The request is processed, and later ciphertext and the secret key that lies with the CSP are updated.

To demonstrate the performance of our system, we calculated the time to produce tokens from the Token Origination Unit. It takes just 0.035 seconds on average, which is acceptable given the high security it enforces.

VI. Conclusion

In this paper, we developed an EHR system that supports field-level ABE, ABAC, searchable encryption, and attribute revocation by employing a knowledge graph that is HIPAA compliant. A knowledge graph produced by our framework accounts for all user roles and attributes in the healthcare organization. It records the attributes of the users along with EHR fields to provide meaningful access to the EHR system. Often in the presence of Big Data, doctors require to search through encrypted data within limited time and computation. Our framework allows search through the data within a short time. User attributes also keep changing with time because some users leave the organization, some may be promoted, or organization policies vary. So, all these changes require attributes to be revoked. Our framework also ingeniously addresses this issue by assigning ciphertext updates and secondary secret key updates to the cloud. The secret key that resides with the user remains firm. Usually, a system with many features has several keys that become an extra management load to the user. By using a single scheme for all the above operations, our framework seems to be more user-friendly. The knowledge graph, encrypted index file, and secondary secret key are stored in the CSP considering the HBC adversary model [42]. We have also assumed the principles of Edge Computing in our framework [51]. Users are verified within the organization's limit to protect privacy. All operations on the data were also
performed within the organization frontier before moving it to the cloud to protect against privacy threats.

ACKNOWLEDGMENT

This work has been supported by Office of Naval Research under grants N00014-18-1-2453, N00014-19-WX-00568, and N00014-20-WX01704 and by NSF grant 1955319.

REFERENCES


