Design of a Secure Privacy Preserving Cloud Based Framework for Sharing Electronic Health Data

by

Munachiso Ilokah

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of

Masters of Applied Science

in

The Faculty of Engineering and Applied Science
Electrical and Computer Engineering

University of Ontario Institute of Technology

Oshawa, Ontario, Canada

November 2019

Abstract

The abstract will be rewritten at the end and should provide a complete narrative of the thesis i.e like a summary. The current abstract is completely focused on the ABE element while the focus of the thesis is on a framework for the secure sharing of health data through third party storage services such as cloud computing.

The need to protect user data from unauthorized access and malicious use by authorized users, especially in the case of a system that includes the use of a third party storage service, which limits the amount of control that the data owner has, continues to present itself. The use of encryption for secure data storage has continued to evolve in order to meet the need for flexible and fine grained access control which led to the development of Attribute Based Encryption (ABE) based on the concept of Identity Based Encryption (IBE). ABE gives more control to the data owner and has continued to evolve to enable users make use of the technological advancements available to them.

The use of ABE to ensure the security and privacy of health data has been explored with multiple solutions proposed. This thesis aims to develop a platform that applies an improved ABE scheme which allows for the secure outsourcing the more computationally intensive processes for data decryption to the cloud servers, reducing the amount of time needed for decryption to occur at the user end and also reducing the amount of computational power needed by users who require access to the data stored in the cloud in its encrypted form.

For/Dedicated to/To my...

Acknowledgements

- Acknowledge supervisor
    - External academic support or guides
    - External professional support (slc, ogs staff)
    - Colleagues (lab mates, lokendra,etc)
    - Personal Support (Liz, Tim, Almey, Tessa, Catherine, Margot, Aida, etc)
    - Family (Mum, Dad, Brothers, etc)

Contents

List of Figures

List of Tables

List of Abbreviations

ABE	**A**ttribute **B**ased **E**ncryption
CP-ABE	**C**iphertext **P**olicy - **A**ttribute **B**ased **E**ncryption
KP-ABE	**K**ey **P**olicy - **A**ttribute **B**ased **E**ncryption
MA-ABE	**M**ultiple **A**uthority - **A**ttribute **B**ased **E**ncryption
EHR	**E**lectrobic **H**ealth **R**ecords
GPP	**G**lobal **P**ublic **P**arameters
GPK	**G**lobal **P**ublic **K**ey
GSK	**G**lobal **S**ecret **K**ey
uid	**U**nique **ID**entifier
CA	**C**ertificate **A**uthority
AA	**A**ttribute **A**uthorities
CSP	**C**loud **S**ervice **P**rovider

Chapter 1

Introduction

1.1 Motivation

Secure storage of electronic health data by health institutions such as hospitals is important in order to make this data available to other parties while ensuring that they prevent unauthorized access to user data and protect the privacy of their patients. These institutions typically store this data on physical hardware in a secure location on their premises and so are able to prevent unauthorized access through the use of adequate network security infrastructure. They are also able to prevent authorized users from gaining access beyond their levels of control through the use of proper access control mechanisms which ensures accountability. The deployment of more efficient and modern information technology such as cloud computing extends the trust boundary of the Information Technology (IT) infrastructure of institutions beyond their facilities while giving them multiple advantages such as a potentially unlimited amount of storage and computational capabilities. This new infrastructure exposes data to both internal and external threats as the institutions no longer have physical control of the infrastructure on which their data is stored as that is now in the control of the Cloud Service Providers (CSP).

One way of ensuring that user data is protected from both the CSPs and from unauthorized users is to encrypt the data before it is stored in the cloud using available encryption schemes. This grants only authorized users access to the required information (usually a key) to gain access to data in its original form. This system becomes a challenge when there is a need to provide fine-grained access to data to third parties in order to ensure that privacy of users are not

violated in relation to privacy laws such as Personal Health Information Protection Act (PHIPA) [10] which governs the collection, use and disclosure of personal information in the health sector in the province of Ontario, Canada. Sharing of data with third parties such as universities and research institutes becomes a challenge as the health institutions need to ensure that privacy of their patients are not violated while they provide these parties with the data necessary for conducting research which could lead to advancements in the health industry and improve the service delivery for patients while saving more lives.

The need to meet the privacy and security requirements have led to the creation of systems for the sharing of electronic health data based on a system of encryption called Attribute Based Encryption (ABE) [21][32][2][4][3][20][19][27]. ABE, developed by Sahai and Waters [34], provides the opportunity for fine-grained access control to data as either the user secret keys or the corresponding ciphertexts contain descriptive attributes with the other containing an access structure that specifies what attributes need to be present for decryption. This means that, if there is no match between the attributes of a user's key and the ciphertext, decryption is prevented. This enables the health institutions to be able to grant third parties access to only the specific data that they require to carry out their research while ensuring that Personal Identifiable Information (PII) of patients are not exposed in line with privacy laws and requirements.

1.2 Thesis Contributions

In this thesis we have developed a framework for the sharing of health data among multiple parties even when the parties are using untrusted third party storage services. The platform employs symmetric encryption for the security of the actual data sets while a variant of the Multi-Authority Ciphertext Policy Attribute Based Encryption (MACP-ABE) is applied to encrypt the secret keys that users need to gain access to data in its plain form. This is because of the high amount of computation required for encryption and decryption algorithms of the different available Public Key encryption schemes, a category that includes the MACP-ABE scheme. The security of this protocol had been proven adequately in the

original work [41] and as a result, we have limited our analysis to the basic security requirements that are to be met by the framework.

The major contribution of this thesis is a framework for the exchange of electronic health data with significant decrease in the computational requirements in terms of time and resources.

1.3 Thesis Outline

The rest of this thesis is organized as follows: Chapter 2 provides some background technical information on related subjects such as security and privacy, electronic health data, cloud computing, cryptography and Attribute Based Encryption (ABE). It also provides a review of literature outlining work done in the area of ABE in order to improve the security and performance original schemes while adding features such as revocation. A review of systems that have been built based on ABE for the handling of health related data is also included. Chapter 3 introduces our secure privacy preserving framework for electronic health records. It provides the description and mathematical construction for the underlying ABE scheme and also a description of the architecture of our framework together with use cases. Chapter 4 provides details of the evaluation of our framework and also the results. It describes our experimental setup, provides a detailed performance evaluation with relation to computation and also provides a security analysis of the framework. Chapter 5 concludes the thesis, highlighting our contribution and also providing information on possible areas for future improvements.

Chapter 2

Background and Literature Review

2.1 Security and Privacy

This section provides a description of the concepts of security and privacy, two key features of the framework described in this thesis. They are both important features of any system that involves the flow of data among multiple parties or entities.

2.1.1 Security

Security according to the National Institute of Standards and Technology (NIST) [23] is defined as any condition that leads to the creation and maintenance of defensive measures to ensure than an information technology infrastructure continues to perform it basic or critical functions irrespective of the risks posed by the threats to its normal operation.

The major considerations when analyzing the security of any system are confidentiality, integrity and availability. These objectives are an essential condition for any system to be considered secure.

Confidentiality [23] assures that only those entities in the system that are authorized have access to data that is either being stored, processed or transferred in the system.

Integrity [23] relates to the verification of the authenticity of data. This means ensuring that data has not been manipulated in any form either while in transit or

while in storage. This ensures that any unauthorized manipulation of data in the form of addition, deletion or substitution is detected.

Availability [23] is a measure of the level of accessibility and usability of a particular system upon request by an authorized user. This means that the system should be able to at all times carry out the various functions in order to meet the demands of its users. This also covers the ability of the infrastructure to remain functional even when some individual makes attempts to compromise its integrity.

Cryptography

Cryptography[23] is the field of study which represents the principles, means and methods used for transforming data in order to hide their original content and prevent unauthorized use or modification. This typically involves the study of several mathematical techniques. Cryptography can be broadly divided into secret key and public key cryptography also known as symmetric and asymmetric schemes.

(1) Secret Key Cryptography (Symmetric) - This type of cryptographic system involve the use of a single secret key which is usually agreed upon by both parties who want to keep their communication secret. This secret key is used to encrypt the original message typically described as the plaintext (i.e encode the plaintext into a ciphertext that cannot be read by a party without the secret key). The receiving party if authorized and in possession of the secret key is able to decrypt the ciphertext and gain access to the original message. Examples of secret key schemes include the Ciphers (Caesar, Monoalphabetic and Polyalphabetic cyphers), Data Encryption Standard (DES) and Advanced Encryption Standard.

(2) Public Key Cryptography (Asymmetric) - This type of cryptographic systems was developed as a result of the challenges in secret key cryptography which include the problem of key management and lack of secure channel for users to exchange keys. Public key cryptography involves the use of two separate keys, a public and private key, which are used to perform complementary operations such as encryption and decryption or signature generation and

verification. Examples of public key schemes include the Diffie–Hellman key exchange protocol, RSA, Elgamal and Elliptic Curve Cryptography.

Cryptographic Adversary Models

Attacks on cryptographic systems aimed at recovering the plaintext from the ciphertext or the secret key can be classified into four broad categories.

(1) Ciphertext Only Attack [31] - A ciphertext only attack is a class of attacks in which the adversary only has access to some ciphertext without any knowledge of the corresponding plaintext. This is the weakest type of attack because the adversary has the least amount of information to work with and any encryption scheme vulnerable to this class of attack is considered to be completely insecure.

(2) Known Ciphertext Attack [31] - A known plaintext attack is a class of attacks in which the adversary has access to some plaintext and ciphertext pairs. The adversary is unable to create more pairs and in only able to gain access to these by eavesdropping on the comunication channel between parties. These types of attacks are only marginally more difficult to mount.

(3) Chosen Plaintext Attack (CPA) [31] - A chosen plaintext attack is a class of attacks where the adversary is able to select the plaintext and request for the corresponding ciphertexts. This is typically done through the use of a black box system, typically called an oracle, that is able to produce the corresponding ciphertext when given any plaintext without revealing the key or any information about the plaintext of the original ciphertext that the adversary is trying to decrypt. A variation of this is the adaptive chosen plaintext attack where the adversary chooses the new plaintext based on the ciphertext received for earlier submitted plaintexts.

(4) Chosen Ciphertext Attack (CCA) [31] - A chosen ciphertext attack is a class of attacks where the adversary selects any ciphertext and requests for the corresponding plaintext. This is the direct opposite of the chosen plaintext attack class. This class of attacks are considered to be the strongest model of

attacks when classifying encryption schemes based on their level of resistance. An adaptive chosen ciphertext attack just like the adaptive version of CPA, involves the adversary deciding on what ciphertext to submit based on the plaintext received for earlier requests.

Note that some of the attack types described above are mutually exclusive (for instance, an attack cannot be both chosen plaintext and known plaintext). And also the chosen plaintext/ciphertext attacks are somewhat exclusive to the modern era of cryptography.

2.1.2 Privacy

Privacy is commonly equated with the concept of confidentiality although they are both distinct. While confidentiality is mainly concerned with ensuring that only users who are authorized have access to data which is being stored, processed or transferred within a system. Privacy, on the other hand, is concerned with ensuring that users have more control over the collection, use and storage of information that is related to them. Therefore, while maintaining the confidentiality of a system aids in preserving privacy, confidentiality does not completely ensure privacy as an authorized user may abuse that privilege by violating the privacy of user information [33].

The range of what is considered private information significantly varies in scope depending on the application area. For instance, in the health sector, private information can be regarded as any oral or written information that meets any of the following criteria: relates to the health of the individual, including their family history; relates to health care provision, including the source of care; constitutes as service for individuals who require long term care; relates to payment for health care. More importantly, private information is any information that can be used to identify an individual, either alone or when related to another piece of available information [11].

The Personal Health Information Protection Act (PHIPA)[10] is an act that establishes the guidelines for the collection, use, and disclosure of personal health information in the Ontario province of Canada. The guidelines are to be followed

by individual and organizations that work with health information, both the custodians and those who receive information from the custodians. The guidelines also protects the right to privacy of individuals with respect to their Personal Health Information (PHI) by giving them the authority to have access and request modifications to their PHIs. It also requires that individuals provide consent before their PHI can be collected, user or disclosed.

The privacy of data that is stored in the cloud faces multiple challenges as a result of the different ways in which the data are stored or processed on a machine that is usually owned by a different organization, the CSP. The major issues that exist in this area of privacy relate to trust as users are not completely certain that: their data is not being used for other purposes other than that for which it was collected; that data is destroyed properly in the end; privacy breaches have occurred which may have exposed their information; their information is retained even after they have stopped using a particular service [33].

2.2 Electronic Health Data

There are three broad classes in relation to the electronic collection and storage of health information and according to Canada Health Infoway, established for accelerating the adoption and use of digital health solutions across Canada, can be defined as:

- Electronic Health Record (EHR)[37] - these records usually contain information about an individual's health and their health care history. Typical information contained in these records include lab results, medication profiles, clinical reports, and diagnostic images. The EHR is made available electronically to authorized health care institutions

- Electronic Medical Record (EMR)[37] - this refers to the digital form of the information acquired during an individual's visit to a health institution. This allows the doctor at the facility to gain access to information about the individual, including potentially information stored in the EHR.

- Personal Health Record (PHR)[37] - this is simply a compete of partial record containing information about an individual's health and usually in their custody. The health care institution has no control of this and it is managed by the individual.

The focus of this work is on electronic health records which are typically shared among multiple parties and under the control of the medical institution.

2.3 Cloud Computing

Cloud computing [30], is "a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." Cloud computing offers considerable advantages to both government and private organizations, which has led to its growth and world wide acceptance in recent years. Some of the advantages offered by the cloud include: easy and fast deployment of IT systems; reduction in the cost of installation and maintenance of infrastructure; easy accessibility; improved flexibility of systems; and a heavy reduction in the responsibilities of the user as most of the traditional tasks will be handled by the provider of the cloud based service, the CSP.

The different service models for cloud computing as Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). These delivery models are distinct based on what services the CSP provides and the amount of responsibility that falls on the user in terms of control and management of resources. The IaaS model gives users more responsibilities as they have control over their operating systems, storage and applications which have been deployed while the SaaS offers the least amount of responsibilities which are limited to some application configuration settings. More detailed information about the different service models can be found at [30].

The deployment models available in cloud computing are the private, community, public and hybrid cloud models. These models are based on the number of parties that share the available deployed infrastructure. The private cloud is

typically setup for use for a single organization while the community and public cloud models usually involve multiple parties with the former involving parties that share similar interest and requirements, while the latter is typically provisioned and available for use by the general public. The hybrid cloud model is basically a combination of the any of the other models and is typically a combination of the private and public models with the aim of benefiting from the strengths of the models while eliminating individual model weaknesses. More detailed information about the different deployment models can be found at [30].

The cloud computing deployment model this thesis considers is the public cloud model as this is the model mostly used or a hybrid model potentially involving a private and public cloud model focusing on the vulnerabilities of the public facing infrastructure. Also, users of the private model have more control over their infrastructure and are able, to a certain degree, to ensure that the security and privacy of stored data is assured.

2.4 Technical Preliminaries

This section provides a detailed description of concepts that form the foundation of Attribute Based-Encryption which is a pairing based form of cryptography that is a fundamental building block of the framework described in this thesis. This includes the different mathematical tools, foundational concepts and a description of the complexity assumptions.

2.4.1 Bilinear Maps

The concept of bilinear maps, or pairings, are the foundation of pairing based cryptography which allowed of the creating of cryptosystems with a great variety of functionalities. Cyclic groups with efficiently computable bilinear maps form the basis of bilinear maps.

There are two general types of bilinear maps, namely:

1. **Symmetric Pairing[29]** - In this type of pairing we have two cyclic groups G, G_T of prime order p with g as the a generator of G. The efficiently computable bilinear map in this case is represented as

$$e\colon G \times G = G_T$$

2. **Assymmetric Pairing[29]** - In this type of pairing we have three cyclic groups G_1, G_2, G_T with G_1 and G_T of order p and G_2 a group with each element having an order dividing pG. The efficiently computable bilinear map in this case is represented as

$$e\colon G_1 \times G_2 = G_T$$

The type of pairing used in this work is symmetric. Below is a definition of bilinear maps as well as the properties that make it efficient for use in Attribute Based Encryption.

Definition 2.4.1. (Bilinear Maps[29]) Let G, G_T be two cyclic groups (multiplicative or additive) of prime order p. Let g be a generator for G and $e\colon G \times G = G_T$. The bilinear map e has the following properties:

1. Bilinearity: for all $u, v \in Ganda, b \in Z_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
2. Non-degeneracy: $e(g, g) \neq 1$

2.4.2 Access Structures

The definitions of access structures and linear secret sharing schemes used in this thesis have been adapted from [5].

Definition 2.4.2. (Access Structures [5]) *Let $\{P_1, \ldots, P_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\{P_1,\ldots,P_n\}}$ is monotone if $\forall B, C$: if $B \in \mathbb{A}$ and $B \subseteq C$, then $C \in \mathbb{A}$. An access structure, i.e monotone is a collection $\mathbb{A}$ of non-empty subsets of $\{P_1, \ldots, P_n\}$ i.e., $\mathbb{A} \subseteq 2^{\{P_1,\ldots,P_n\}} \setminus \{\}$. The sets in $\mathbb{A}$ are called the authorized sets, and the sets not in $\mathbb{A}$ are called the unauthorized sets.*

In ABE, attributes play the role of parties in the access structure and the scheme in this thesis only considers access structures that are monotone.

Access policies based on monotone access structures could be represented as either a Linear Secret Sharing Scheme (LSSS) Matrix or with the use of monotonic boolean formulas which could be represented as an access tree in which the core nodes are used to represent the AND and OR gates with the attributes represented by the leaf nodes.

Access policies based on monotone access structures could be represented in two different ways for ABE schemes. The two widely used methods are:

2.4.3 Secret Sharing Schemes

Secret sharing schemes which was first created by Shamir in [35] allows for the division of data among multiple parties in such a way that the original data can only be reconstructed if a party is in possession of at least a fixed number of division, usually the threshold, and possession of a number of pieces less than the threshold reveals no information about the original data. Other earlier works in secret sharing include works by Barkley [8], Benaloh [6] and Ito, Saito and Nishizeki [22]. LSSS are secret sharing schemes in which the reconstruction of the original secret is done using a linear function of the available pieces [5].

2.4.4 Linear Secret Sharing Schemes

Definition 2.4.3. (Linear Secret Sharing Schemes [5][40]) A secret sharing scheme $\prod$ over a set of parties $\mathcal{P}$ is called linear (over Z_p) if

1. The shared for each party form a vector over Z_p.

2. There exists a matrix M with ℓ rows and n columns called the share-generating matrix for $\prod$. For all $i = 1, /ldots, \ell$, with M_i representing the i'th row of M. The function ρ is defined as the party labeling row i as $\rho(i)$. Consider a column vector $\vec{v} = (s, r_2, \ldots, r_n)$ where $s \in Z_p$ is the secret to be shared and $r_2, \ldots, r_n \in Z_p$ are chosen randomly, then M_v is can be described as the

vector ℓ shares of the secret s according to $\prod$. The share $(M_v)_i$ belongs to party $\rho(i)$.

The linear reconstruction property as described in [5] shows that suppose that $\prod$ is an LSSS for an access structure $\mathbb{A}$. Let $S \in \mathbb{A}$ be any authorized set, and let $I \subset 1, \ldots, \ell$ be defined as $I = i : \rho(i) \in S$. Then there exits constants $w_i \in Z_{p_{i \in I}}$ such that, if ρ_i are shares of the secret s according to $\prod$, then $\sum_{i \in I} w_i \lambda_i = s$.

Note that access structures represented as boolean formulas which are typically represented by binary trees can be converted into a LSSS form using the techniques described in [38] with the number of rows in the corresponding matrix equal to the number of leaf nodes in the access tree.

2.4.5 Cryptographic Complexity Assumptions

Let G be a cyclic group of prime order p. Let g be a generator for G represented as $G = \langle g \rangle$ and let $x, y, z \in G$. The different complexity assumptions used to show the security of the different pairing based schemes have their foundation in the following core hardness problems[29]:

- **Discrete Log DLog) Problem** - Given g and g^x, compute x.
- **Computational Diffie-Hellman (CDH) Problem** - Given g, g^x, and g^y, compute g^{xy}.
- **Decisional Diffie-Hellman (DDH) Problem** - Given g, g^x, g^y and g^z , determine if $xy = z$.

2.5 Attribute Based Encryption (ABE)

ABE is a pairing based cryptogaphic scheme that was developed based on Identity Based Encryption (IBE) which was originally proposed by Shamir in 1984 [36]. Shamir proposed a scheme which allowed for the encryption and decryption of information between two different users without the need for any exchange of keys between both parties. His proposal assumed the existence of a trusted key generation service similar to Certificate Authorities (CA) which were responsible for

registration of users as they join a network and also for subsequent verification of their identity. Personal information unique to several users, such as their address, email address or a combination of this information, was used as the public key in the system. This allowed for the encryption of data meant for UserB by UserA using the email address of UserB, e.g "userB@gmail.com". UserB on receiving this encrypted data would then contact the key generation service and, after successful authentication, receives a secret key granting him access to the original data. The scheme proposed by Shamir was further developed and the first practical and secure IBE scheme was presented by Boneh and Franklin in [9], who developed a fully functional IBE scheme which made use of groups for which there existed an efficiently computable bilinear map such as the Weil pairing.

Sahai and Waters in [34] developed a new scheme that improved on the existing IBE schemes by creating a system in which the user identity is viewed as a set of descriptive attributes, allowing a user to encrypt data for all users who have a certain set of attributes. Decryption in this case is only permitted if the identity of a user, and the identity for which the ciphertext was encrypted, were close enough based on their individual attributes. It is in this work that the notion of Attribute Based Encryption is first mentioned.

Goyal et al. in [17] developed an ABE scheme that was more robust than the original ABE scheme proposed by Sahai and Waters [34]. In their scheme, termed Key-Policy Attribute-Based Encryption (KP-ABE), each ciphertext created by the user contains a set of descriptive attributes. Secret keys of individual users are associated with an access structure which specifies the attributes that need to be contained in a ciphertext for successful decryption. The access tree structure could be made up of interior nodes that consist of AND and OR gates with the leaves containing different attributes. For example, if UserA's key in KP-ABE contains "C AND D" as the access policy, the only ciphertexts he should be able to decrypt are those that contains both attributes C and D. A ciphertext with only attribute C or D could not be decrypted by UserA as the requirements for access would not be satisfied. The keys generated for users in this scheme are also collusion resistant just like the original scheme, meaning that no two users with different attributes could combine their keys to create an overlap of attributes that would give them

the ability to decrypt files which they would not normally be able to decrypt.

The authors in [17] mentioned a variant to the KP-ABE scheme known as the Ciphertext-Policy Attribute-Based Encryption (CP-ABE) scheme which they left as an open problem that was solved by [7]. In CP-ABE, the ciphertexts are associated with the access policy while the user keys contain a set of descriptive attributes. This would mean that, for a key to decrypt a particular ciphertext, its attributes need to match the access structure of the access policy of the ciphertext. This scheme, unlike the KP-ABE scheme, gives the user encrypting data more control as the user is able to control who can have access to data being encrypted by making sure the access policy in the ciphertext specifies what attributes need to be possessed for access to be granted.

The four basic algorithms of any ABE based system includes the following:

1. Setup - The setup algorithm is responsible for the selection of the bilinear group and the definition of a bilinear map that has the properties of bilinearity, computability and non-degeneracy. The setup algorithm takes as its input the security parameter which specifies the size of the attribute set and generates a public key (PK) and a master key (MK) as output.

2. Keygen - The keygen algorithm takes as its input two parameters, the MK generated during setup and the set of attributes that the user possesses, and generates a secret key (SK) for the user in CP-ABE. The input for KP-ABE includes the MK, PK and an access structure and outputs SK.

3. Encryption - The encryption algorithm takes as its input PK, a message M, and an access structure for CP-ABE schemes and produces a ciphertext C. It takes as input PK, M and a set of attributes and produces a ciphertext C for KP-ABE schemes.

4. Decryption - The decryption algorithm takes as input PK, C and SK and, if the attributes of either the ciphertext or secret key satisfies the access structure of the other, depending on whether the scheme is a CP-ABE or KP-ABE scheme, decrypts C and outputs M.

Both variants of ABE allows for delegation which would allow a user with a secret key for set of attributes (CP-ABE) or containing an access structure (KP-ABE) to derive a new secret key containing a set of attributes that is a subset of the attribute set of the original key or an access structure that is more restrictive than the access structure contained in the original key.

2.6 Revocation

A useful feature with ABE based systems is the ability for user revocation. This is a challenge as multiple users may share a similar attribute which could cause the revocation of one user to affect other users who share a similar attribute. Furthermore it is important that user revocation be flexible and occur at different granular levels which means that revocation could involve removal of a user completely or a partial reduction of a user's access, based on their attributes. The addition of an expiry date to the generated key has been proposed by initial ABE based systems [7] but does not offer an effective means for the revocation of user attributes.

Yu et al. in [43] proposed a scheme that enables secure, scalable and fine grained data access to a cloud based system with a great reduction in the computation overhead by delegating most of the computation intensive tasks to the cloud servers while ensuring the security and privacy of user data through the combination of KP-ABE, proxy re-encryption (PRE) and lazy re-encryption (LRE). In their scheme, the attributes that are assigned to the ciphertexts are all assigned a unique ID which serves as the version number that is stored in a list maintained by the cloud servers, together with the PRE keys used. The cloud servers also maintain a list of all the existing users in the system who are currently authorized to have access to the different stored data. Data files are encrypted using symmetric encryption with the decryption keys encrypted using ABE and appended to the encrypted data file together with a unique file ID. PRE enables the use of a proxy to convert a ciphertext which has been encrypted using the public key of a particular user into another ciphertext that can be decrypted using the private key of a different user, without revealing the contents of the underlying file. In order to revoke a user, the scheme determines the least amount of attributes that need to be updated to

prevent a user from having access and redefines the public and master keys for those attributes while generating the corresponding PRE keys. The revoked user's ID, the attribute set, the PRE keys and the new public key parameters are sent to the cloud servers. The cloud servers then remove the revoked user from the user list, store the new public key parameters and then updates its list of attributes together with the PRE keys used.

Yu et al. in [39] have applied the concept of proxy re-encryption with CP-ABE in order to enable revocation. Liang et al. in [28] have proposed the system Ciphertext Policy Attribute Encryption with Revocation (CP-ABE-R), which makes use of linear secret sharing and binary tree techniques to enable effective revocation of users with the aid of a unique identifier assigned to each user in the system and which is not needed for encryption and decryption. Cheng et al. have proposed a scheme [14] that enables effective revocation in CP-ABE by dividing the original data into multiple parts which they term *slices*, before they are stored in the cloud which allows for revocation by the re-encryption of only one slice. The data is encrypted with a symmetric key and then split into multiple parts using a secret sharing scheme. In the case of the secret sharing scheme applied here, the number of parts that are needed to reconstruct the original file is equal to the number of distinct parts. A particular slice of data is chosen as the dynamic data and it is this slice that is constantly re-encrypted to enable revocation while the static data remains the same. This reduces the computational and storage overhead while ensuring that the security of the system is not compromised.

2.7 Multi-Authority Schemes

The first Multi-Authority Attribute Based Encryption (MA-ABE) scheme was proposed by Chase in [12][15] and was based on Key Policy Attribute Based Encryption (KP-ABE). In this scheme, there are multiple Attribute Authorities (AA) in addition to a Central Authority (CA) which are responsible for generating secret keys corresponding to the attributes which they handle. Users are assigned a Global Unique Identifier (GUID) which they use to request the shares of the system-wide Master Secret Key (MSK) handled by the different authorities. The GUID is used

by the authorities to tie the shares to a particular user. The system includes a CA which is responsible for aiding users in decryption by ensuring that all the shares generated for a particular user by the different AAs sum up to the same MSK. The CA ensures this by assigning to each user a special value that cancels out all of the shares from the different authorities, providing the user with a function of the system wide MSK. In order to carry out its functions, the CA has knowledge of the MSK and as a result would also be able to decrypt any ciphertext in the system which is in contrast with the idea behind using multiple authorities, which is to distribute trust among several untrusted authorities. Also, in the original system, the users use their GUIDs to identify with the individual AAs, which means that several authorities could combine their information about a particular user and develop a profile based on the attributes that the user has acquired and be able to generate keys with the same level of access as the user.

Chase and Chow in [13] proposed a solution to the original MA-ABE problem which eliminated the use of a CA and also prevented the AAs from having the ability to combine the information about a user by allowing users to use pseudonyms for interacting with the individual AAs in the system instead of the use of their respective GUID. This solution eliminates the CA by applying a set of Pseudo Random Functions (PRF), and having every pair of authorities in the system share a secret PRF seed which allows for a combination of all their individually generated shares. To enable users to communicate with the individual AAs using pseudonyms, the authors have developed a novel Anonymous Key Issuing Protocol. Additional details about the protocol and its functions can be found in [15][13].

Lewko and Waters proposed a MA-ABE[24][38] solution in which the different authorities operate independently and do not have to share any common information with each other as in [12] e except for an initial set of common reference parameters. Their system has higher tolerance as the failure or corruption of authorities in the system will not have a direct impact on the operation of the fully functioning and uncorrupted authorities. Furthermore, in their solution, any party could become an authority by making available, to the other entities in the system, their verification key and their lift of managed attributes. This solution makes use of Linear Secret Sharing Schemes (LSSS) access structures and the authors show

that boolean formulas could easily be transformed into LSSS structures using techniques found in [38]. Lewko and Waters have used the dual system proof technique to prove the security of their system.

Kan et al. in [42] have developed a solution called DAC-MACS (Data Access Control for Multi-Authority Cloud Storage) in order to make a more efficient CP-ABE based MA-ABE solution that takes advantage of the services available in the area of cloud computing and that is more suited to this domain. Their solution includes an efficient attribute revocation method that enables both forward and backward security. In addition to providing a means for attribute revocation, Kan et al.'s system[42] has better efficiency than other similar solutions. Also, by using a decryption token for the decryption, they have been able to transfer the intensive computations over to the cloud server, thereby reducing the computational overhead on the side of the end user. A flaw in the system is the fact that the different AAs have knowledge of the GUID of the users which would give a revoked user the ability to derive the key update key that they could use to update their own keys by corrupting any AA, together with some non-revoked users.

Kan and Xiaohua[41] have developed a more effective MA-ABE solution based on CP-ABE. In their new system, the secret keys of the different users are not related to the key of the data owner and so users will only need to hold an individual secret key for an authority instead of multiple secret keys associated with multiple owners. This makes it more suitable for a multiowner setting as storage overhead for user keys is greatly reduced. They have also improved the revocation mechanism by modifying it to require that only ciphertexts associated with a revoked attribute be updated and by using a single update key for the update of both keys and ciphertexts.

2.8 Outsourcing

An important extension to the functionality of available ABE schemes is the ability to outsource the computationally intensive operations such as the decryption of the ciphertext to third party systems such as the Cloud Service Provider (CSP) while maintaining the same levels of security and privacy as the existing ABE schemes.

This enables users to have access to the required data using devices with low computational power such as mobile devices with no additional risk.

This was initially proposed by Green et al. in [18]. In their approach, they achieve outsourcing by splitting the traditional decryption algorithm into two components. The first a transformation algorithm to be run externally by the CSP, which given the necessary input transforms the ABE ciphertext into a constant-size El Gamal-style ciphertext. The second component is the decryption algorithm to be executed by the user which with the El Gamal-style ciphertext and the right input produces the original message. The second component to be run by the user is less computationally intensive as it involves a single exponentiation operation compared to the multiple pairing operation involved in the transformation algorithm. They have applied this to single authority KP- and CP-ABE schemes showing improvements with relation to size of the ciphertext at the user end and minimal impact on the bandwidth of the decryption process. Their approach has the added benefit of reducing the amount of code that has to reside on the user device and the main bulk of code which corresponding to the analysis of ABE attributes and the computation of pairings now resides on the third party device allowing for a smaller and more trusted code base on the user end.

Other authors [25][26] have extended the idea of outsourcing in ABE by outsourcing the key generation algorithm in addition to the decryption algorithm through several other methods with limited improvements on the level of overhead achieved by Green et al. in [18].

Sherman Chow has created a framework in [16] that allows for the construction of an ABE scheme with multiple authorities, revocation capability and outsourcing of the decryption algorithm from a single authority ABE scheme if certain conditions are met. These include the ability to split the ciphertext and secret keys into both attribute-dependent and attribute-independent components. The framework also requires that the structure of secret keys generated by multiple attribute authorities share a structure that makes them indistinguishable from keys generated by a single authority.

2.9 ABE Based Health Care Systems

Ibraimi et al. proposed a new variant of CP-ABE [21] in order to be able to enforce the required levels of access controls in a multiple domain based system to ensure the security of personal health records (PHR). They have distributed the group of users who normally require access to PHRs into two domains: the professional domain, which consists of the health care providers; and the social domain, made up of family members, friends and possibly fellow patients. Their proposed variant of CP-ABE allows the encryption of health records with an access policy that is made up of attributes issued by two different trusted authorities: the trusted authority in charge of the professional domain and the trusted authority in charge of the social domain.

The authors in [32] have proposed the design of a patient controlled cloud based EHR infrastructure using CP-ABE. They have based their system on the assumptions: a trusted authority (TA) exists that is responsible for the generation of keys for users and is able to store the public parameters and public keys of users in a public directory; each user is associated with a unique identifier and a set of attributes; and the cloud server used for storage is only trusted for the performance of storage operations. They have used a variant of CP-ABE, known as broadcast ciphertext-policy attribute-based encryption (bABE), which extends the traditional CP-ABE to enable revocation of users' keys. They have also provided the functionality of keyword search which allows users to search using a search term by providing a key which allows the cloud provider to perform search operations on the encrypted data without learning anything about the actual data contents.

Akinyele in [2], using ABE, has provided a detailed design and implementation of self-protecting EMR which allows EMR availability even when the providers are offline. Their system makes role- and content-based access control possible. For role-based access control, users are granted explicit access to collections of data related to their roles that match some specific criteria which the authors have termed content slices. These slices could be as specific as required by the system administrators and the content-based access is used to grant access to users such as contractors, who have no definite roles in the system but require access to records to carry out their functions. They have also implemented a policy engine

as part of their design to evaluate new or updated EMRs in order to determine the policies that are to be used for encryption. The policy engine's final decision is based on either the set of policies specified by the administrator, the identity and nature of the EHR, the annotations attached to the EHR, or in some cases the textual content of the record. They have taken advantage of the XML-based EMR standards which include the Continue of Care Record (CCR) and the Continuity of Care Document (CCD) which allows their policy encryption engine to parse each node in order to determine the appropriate access policy and subsequently the access control rule and content related attributes for which the document is to be encrypted. Users will need to be present at the initialization stage to have their mobile devices provisioned with the required decryption keys to be able to use the accompanying mobile application to access their data. CP-ABE is used to grant access to patients and health professionals using keys with fixed attributes related to their roles or responsibilities while users with no definite role are granted access through the use of KP-ABE by generating keys, which contain a specific policy that defines what data they can access and, in some cases, the time periods for which they can have access.

Barua et al. in [4] have proposed a scheme which they called Efficient and Secure Patient-centric Access Control (ESPAC), in which they have used CP-ABE to achieve patient-centric access control allowing different access privileges based on the roles of the data requester and assigning the corresponding attributes based on those privileges. They have constructed their access control policy by assigning attributes, based on the relationship between the patient and the requesting party, which is used to determine the privacy levels of the requesting party before attributes are assigned. Their system is made up of four main entities: the trusted authority (TA), the cloud service provider (CSP), the registered user, and the data-access requester. The scheme makes use of pseudo identity instead of unique identities to ensure privacy. The scheme enables message integrity checks, non-repudiation and source authentication through the use of signature verification. This scheme is able to ensure forward and backward secrecy.

Suhair et al. in [3] have proposed the design of a cloud based EHR system using CP-ABE to ensure security by using the credentials and attributes of the health

care providers as the universal attribute set. Their proposed architecture is made up mainly of three components: the EHR system hosted on the cloud; the participating healthcare providers; and the attribute authority (AA) which is in control of generating the secret keys of users which contain the appropriate attributes. The cloud is used for data storage and computation in their infrastructure. Encryption and decryption of the medical records are performed at the client end through the use of lightweight software. Suhair et al. have proposed the addition of an expiry date to the access policies used for encryption, or complete re-encryption with updated access policies, as a way to achieve revocation in order to avoid the communication overhead involved with the re-distribution of secret keys to authorized users. The use of a single AA presents a focal point of weakness for the security of the system and presents the key escrow problem.

Hupperich et al. in [20] propose an architecture that gives the patient control of the delegation of access to their EHRs, in line with the existing privacy laws. They have proposed a system that would allow patients to authorize the appropriate health care service providers to have access to their EHRs through a flexible channel that would not require the patient to be present. In order to eliminate the use of smart cards for access, and to enable health care providers to have access to EHRs their infrastructure only requires the use of the patient's smart card at the initial stage for the generation of a transaction code (TAC) which the patient can use to grant access by sending to authorized health care professionals. They have used ABE for encryption by using the patient's identity and a TAC that is specific to a particular medical record as the two main attributes for encryption and decryption. They have implemented emergency access by allowing the encryption of certain records using the attribute *"emergency"* without any TAC, with logging implemented to keep track of emergency access. The authors have not mentioned how the system would handle revocation of users and have not implemented a secure means of transmission for the generated TACs which they have stated could be transferred via traditional means such as a phone or on paper.

The authors in [19] proposed the design for a secure interoperable cloud based service for private health records (PHR) which uses the Continuity of Care Document (CCD) for the storage and exchange of information and employs several

security mechanisms using available open standards such as XACML, XML encryption, XML signature and XML key management specification. They have used CP-ABE to achieve patient controlled encryption, and the public key encryption with keyword search (PEKS) scheme to provide privacy-preserving keyword search. They have used the Secure Channel Free PEKS scheme which allows users to perform private searches over encrypted data for specific matching keywords without revealing the keywords or any partial matches to the server.

Li et al. in [27] proposed a framework titled Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute Based Encryption containing a suite of security mechanisms that aimed to solve the existing issues in cloud-based PHR storage systems which include eliminating the risk of privacy exposure, key management scalability, flexible access and effective revocation of existing users. Their work focuses on the multiple data owner scenario similar to our proposed architecture and thus they have divided the users in the system into two broad security domains similar to [21], which reduces the complexity of key management for data owners and users who require access, with the improvement being that in their scheme the public domain (PUD) is managed by multiple AAs. The personal domain is made up of users who are close to the data owners (i.e. patients) such as family members and friends while the public domain is made up of the various professionals who require access to the patient's records such as doctors and pharmacists. In order to apply ABE to the personal domain, Li et al. have employed the Key Policy ABE with efficient revocation as proposed in [39] with the data owner fully responsible for handling this particular domain. The data owner generates keys for members of this domain with the access structure corresponding to their level of access and sends this keys to the corresponding users in order to grant them access. The authors have employed in the public domain the use of the MA-ABE proposed in [12] and improved in [13]. Since the MA-ABE scheme they adopted is essentially a KP-ABE scheme with multiple AAs, in which control of access lies with the AAs who generate the keys for the different attributes therefore taking away control from the data owner, Li et al. have made a slight modification to how this scheme is used in their system. In order to grant the data owner more control, the system requires that the key access policies and the general

approach for specifying the ciphertext attributes be agreed upon in order to grant the users some level of control in specifying the access policy of the ciphertext from their end by choosing the right attributes. To improve security in the public domain, Li et al. have slightly modified the Multi-authority ABE (MA-ABE) scheme proposed by [13] to enable efficient user revocation by using the revocation technique proposed by [39] which was not a feature of the original scheme. Their system provides dynamic attribute and access policy modification together with on-demand user/attribute revocation, together with break glass access, in order to make records available for use under emergency situations.

Chapter 3

Secure Privacy Preserving Framework for Electronic Health Records (EHRs)

3.1 Overview

This chapter introduces the framework that is the basis of this thesis. We start with a description of the ABE scheme that plays an essential role in providing the features of security and privacy for the framework. The construction of the ABE scheme follows after the definition of the various algorithms. The construction shows the mathematical concepts that confirm the inner workings of the different algorithms.

A description of the entire framework follows with the architecture and its corresponding entities highlighted. We then provide use cases together with a detailed description of the individual use cases and the roles the entities play.

3.2 Multiple Authority Ciphertext Policy Attribute Based Encryption with Outsourced Decryption

This section contains the formal definition of the underlying ABE scheme for the overall framework. This formal definition provides a description of the algorithms that are part of the ABE scheme. The second part of the section provides more

detailed information in the form of the mathematical construction for the ABE scheme.

3.2.1 ABE Scheme Algorithm and Security Definition

Our ABE scheme has been adapted from the scheme developed in [41]. Using similar methods applied in [18], we have modified the scheme with the aim of outsourcing the bulk of the decryption algorithm operation to the CSP. This section contains a definition of the algorithms that make up our ABE scheme.

Algorithm Definitions

The following algorithms make up the ABE scheme. These algorithms include: the setup and system initialization algorithms, the secret key generation algorithm, the encryption, transformation and decryption algorithms as well as the revocation algorithms. They are described a follows:

(1) CASetup - The CASetup algorithm is the algorithm executed by the Certificate Authority (CA) in order to generate the Global Master Key (GMK) and Global Public Parameters (GPP) of the entire scheme. It takes as input the security parameter of the scheme. The CA makes the GPP available to all entities in the system.

   (i) UserReg - The UserReg algorithm is run by the CA whenever new users are being added to the system. The CA assigns a unique id to each new user and also generates two public-private key pairs together with user certificates and their corresponding verification keys for each new user. The CA sends one each of the generated public and private keys to the user together with the certificate.

   (ii) AAReg - The AAReg algorithm is run by the CA whenever new Attribute Authorities (AA) are joining the system. The CA assigns a unique id to each new AA and also send the other public and private keys for each registered user to the AA together with the verification keys for the generated certificates.

(2) AASetup - The AASetup algorithm is run by the individual Attribute Authorities (AAs) and generates the public-secret key pair for the attribute authority. It also generates a public-version key pair for each of the attributes under the control of the AA that is provided as input in the AA universal attribute set.

(3) SKGen - The SKGen algorithm is run by the corresponding AAs in order to provide the user with the secret key for the attributes that they have been assigned. The AA takes as input the system GPP, the users Global Public Keys (GPKs), one of the user GSKs, the set of attributes and their corresponding public-version key pairs. It provides the user with a secret key that can be used for decryption.

(4) Encrypt - The Encrypt algorithm is run by the data owner in order to encrypt the corresponding piece of data. This algorithm takes as input the system GPP, the public keys (PKs) for the AAs involved in the encryption - as they are in control of the attributes that make up the policy under which data is to be encrypted - of the data, and an access policy. It provides as output a ciphertext (CT) which implicitly contains the corresponding access policy.

(5) CTransform - The CTranform algorithm is responsible for the transformation of the CT into an El Gamal style ciphertext that can be easily decrypted by the end user. This algorithm is executed by the Cloud Service Provider (CSP) and takes as input the ciphertext, the secret keys of the attributes in the user's possession and the user's GPK. It produces as output a partially decrypted ciphertext (CT') is the user's secret keys meet the policy under which the data was encrypted, otherwise it outputs an error.

(6) Decrypt - The Decrypt algorithm is run by the end user and takes as input the partially decrypted ciphertext CT' and the user's global secret key which is not shared with any other entity in the system and produces the original data as output for the user in its plain form.

(7) UKeyGen - The UKeyGen algorithm is run by the AA under whose control the revoked attribute falls. The algorithm takes as input the SK of the AA,

the revoked attribute and the current version key for the revoked attribute. It produces as output an updated version key for the revoked attribute. It produces an update key to be used to update the corresponding secret keys of users who possess the revoked attributes and whose access is not being revoked. It also produces an update key for the affected ciphertexts whose access policies contain the revoked attribute to allow for users who still maintain access to be able to decrypt while denying revoked users further access.

(8) SKUpdate - The SKUpdate algorithm is run by each non revoked user in the system and takes as input the corresponding secret key acquired from the AA that control the affected attribute together with the update key for user secret keys generated by the UKeyGen algorithm. It outputs a new secret key for the user.

(9) CTUpdate - The CTUpdate algorithm is run the CSP and takes as input the affected ciphertexts and the update key generated by the UKeyGen algorithm for ciphertext update. it outputs a new ciphertext that can only be decrypted by users with a current version of the corresponding revoked attribute.

ABE Scheme Security Definition

The security definition of the ABE scheme is based on the correctness definition of the scheme based on the following conditions:

(1) Decrypt(Transform(Encrypt(Message, GPP, PK_aids), SK_uid, aid, GPK_uid), GKS'_uid) = Message

(2) Revocation also i.e forward decryption and backward decryption

Further analysis of the security of the underlying ABE scheme together with the security games and proofs can be found in [41], the original scheme from which this was derived.

3.2.2 ABE Scheme Construction

A. CA Setup - This procedure is run by the CA to setup the system

$$CASetup(1^{\lambda}) \longrightarrow GMK, GPP$$

where 1^{λ} is the security parameter, GMK is the Global Master Key, and GPP represents the Global Public Parameters.

The CA chooses two multiplicative groups G, G_T with the same prime order p and a bilinear map $e\colon G \times G \to G_T$ and a hash function $H\colon \{0,1\} \to G$. The CA chooses two random numbers $a, b \in Z_p$

The GMK is set as (a, b) and the GPP is set to (g, g^a, g^b, H)

(i) User Registration - This is executed for all users in the system by the CA. Each user is assigned a globally unique uid and for each uid, the CA generates two random numbers $u_{uid}, u'_{uid} \in Z_p$ as its global secret keys

$$GSK_{uid} = u_{uid},$$
$$GSK'_{uid} = u'_{uid}$$

The global public keys for each user is generated as

$$GPK_{uid} = g^{u_{uid}},$$
$$GPK'_{uid} = g^{\frac{1}{u'_{uid}}}$$

The CA generates a $Certificateuid$ for each user uid and send $(GPK_{uid}, GSK'_{uid}, Cerificate(uid))$ to the user.

(ii) Attribute Authority (AA) Registration - The CA assigns a globally unique authority identity aid to each AA and sends (GPK'_{uid}, GSK_{uid}) for all registered users to AA_{aid}. The CA also sends the verification key vk_{CA} to AA_{aid} for verifying the $Certificate(uid)$ assigned to each user.

B. AA Setup - Let X_{aid} be the set of all attributes managed by AA_{aid}. The AA chooses three random numbers $\alpha_{aid}, \beta_{aid}, \gamma_{aid} \in Z_p$ as its secret key.

$$SK_{aid} = (\alpha_{aid}, \beta_{aid}, \gamma_{aid})$$
$$PK_{aid} = (e(g,g)^{\alpha_{aid}}, g^{\beta_{aid}}, g^{\frac{1}{\beta_{aid}}})$$

For each attribute $x_{aid} \in X_{aid}$, AA_{aid} generates a public attribute key as $PK_{x_{aid}} = (PK_{1,x_{aid}} = H(x_{aid})^{v_{x_{aid}}}, PK_{2,x_{aid}} = H(x_{aid})^{v_{x_{aid}}\gamma_{aid}})$ where $v_{x_{aid}}$ is the version key of attribute x_{aid} i.e $VK_{x_{aid}} = v_{x_{aid}}$.

C. Secret Key Generation - The Attribute Authority with AA_{aid} assigns a set of attributes $S_{uid,aid}$ to user with uid after authentication and certificate verification. The AA chooses a random number $t_{uid,aid} \in Z_p$ and computes a secret key for the user $SK_{uid,aid}$ as

$$K_{uid,aid} = g^{\frac{\alpha_{aid}}{u'_{uid}}} g^{au_{uid}} g^{bt_{uid,aid}},$$
$$K'_{uid,aid} = g^{t_{uid,aid}},$$
$$\forall x_{aid} \in S_{uid,aid}\colon K_{x_{aid},uid} = g^{t_{uid,aid}\beta_{aid}} H(x_{aid})^{v_{x_{aid}}\beta_{aid}(u_{uid}+\gamma_{aid})})$$

If the user uid does not hold any attributes from AA_{aid}, the user secret key $SK_{uid,aid}$ only contains $K_{uid,aid}$.

D. Encryption - Data is divided into components based on the level of granularity required for access control i.e $m = \{m_i, \cdots, m_n\}$. Data components are encrypted using symmetric encryption keys $\kappa = \{\kappa_i, \cdots, \kappa_n\}$. An access structure (M_k, ρ) is defined for each content key $\kappa_i (i = 1, \cdots, n)$ and encrypted using the ABE scheme to produce the corresponding ciphertext. Let M be an ℓ x n matrix, where ℓ denotes the total number of all the attributes and the function ρ associates rows of M to attributes. The function ρ is not required to be injective which allows for an attribute to be associated with more than one row of M.

To encrypt the content key κ_i, the algorithm chooses a random element $s \in Z_p$ which is used as the random encryption exponent. It then selects a random

vector $\vec{v} = (s, y_2, \ldots, y_n) \in Z_p$ where $y_2, \ldots, y_n$ are used to share the encryption exponent s. It then computes $\forall 1 \leq i \leq \ell : \lambda_i = \vec{v}.M_i$ where M_i is the vector corresponding to the i-th row of M. It then randomly selects $r_1, r_2, \ldots, r_\ell$ and computes the ciphertext CT_{K_i} as

$$C = K_i \cdot \Big(\prod_{aid_k \in I_A} PK_{aid_k} \Big)^s, C' = g^s, C'' = g^{bs},$$

$$\forall 1 \leq i \leq \ell, \rho(i) \in X_{aid_k} : C_i = g^{a\lambda_i \cdot \left(PK_{i,\rho(i)}\right)^{-r_i}},$$

$$C'_i = g^{r_i}, D_i = g^{\frac{r_i}{\beta_{aid_k}}}, D'_i = \Big(PK_{2,\rho_{(i)}}\Big)^{r_i}$$

Then the encrypted data is uploaded to the cloud server by the owner.

E. Ciphertext Transformation - This involves the transformation of the ciphertext into an El Gamal style ciphertext referred to in this framework as a token through partial decryption while the integrity of the original message is preserved.

The transformation algorithm takes as input the ciphertext CT where $CT = (C, C', C'', C_1, \ldots, C_\ell)$ which contains an access policy (M, ρ), the user's global public key GPK_{uid}, and a set of secret keys from all the involved Attribute Authorities $\{SK_{uid,aid}\}_{aid_k \in I_A}$ i.e $(K, K', K_{x_{aid}} \forall x_{aid} \in S_{aid})$ where S_{aid} is a set of the user attributes. If the attributes provided do not meet the access policy conditions, the algorithm outputs $\perp$. The algorithm proceeds to transform the ciphertext if the set of user attributes S_{aid} meet the conditions of the access policy contained in the ciphertext.

Let I be defined as $\{I_{aid_k} \in I_A\}$, where $I_{aid_k} \subset \{1, 2, \ldots, \ell\}$ can be defined as $I_{aid_k} = \{i : \rho(i) \in S_{aid_k}\}$. Let $n_A = |I_A|$ be the number of AAs involved in the access policy of the ciphertext. The algorithm chooses a set of constants $\{\omega_i \in Z_p\}_{i \in I}$ such that if the shares of a secret represented by $\{\lambda_i\}$ are valid with respect to the matrix component of the access structure, M, then the secret can be reconstructed as $\sum_{i \in I} \omega_i \lambda_i$. The algorithm computes the partially decrypted ciphertext CT' as

$$= \prod_{aid_k \in I_A} \frac{e\Big(C', K_{uid,aid}\Big) e\Big(C'', K'_{uid,aid}\Big)^{-1}}{\prod_{i \in I_{aid_k}} \Bigg(e\Big(C_i, GPK_{uid}\Big) e\Big(D_i, K_{\rho(i),uid}\Big) e\Big(C'_i, K^{-1}_{uid,aid_k}\Big) e\Big(g, D'_i\Big)^{-1}\Bigg)^{\omega_i n_A}}$$

$$= \prod_{aid_k \in I_A} e\Big(g, g\Big)^{\frac{s\alpha_{aid}}{u'_{uid}}}$$

F. Decryption - The algorithm for decryption takes as input the partially decrypted ciphertext (CT') and the user's secret key and produces as output the original message. It performs an exponentiation on CT' in order to derive the blinding element (BE) of the original message as

$$= CT'^{u'_{uid}}$$
$$= \prod_{aid_k \in I_A} e\Big(g, g\Big)^{s\alpha_{aid}}$$

Remember the C element of the main ciphertext $= K_i \cdot \big(\prod_{aid_k \in I_A} PK_{aid_k}\big)^s$ where $PK_{aid_k} = e\big(g, g\big)^{\alpha_{aid}}$. Therefore the original message which in this case is the symmetric key K is computed as

$$= \frac{C}{BE}$$

G. User Revocation - User revocation which can also be referred to as attribute revocation as used in some of the literature on the subject should achieve two critical criteria. The revoked user should not be able to decrypt new ciphertexts which have been encrypted using the public attribute keys of attributes that the user was previously granted private keys for. This is referred to as Backward Security. Any new user who has the right attributes should

be able to decrypt the original ciphertexts that were encrypted using those public parameters. This is referred to a Forward Security. The revocation algorithm used has been gotten from the scheme by Yang et. al in [41].

(i) Update Key Generation - In order to revoke an attribute $\tilde{x}_{aid'}$ from a particular user, the attribute authority (AA) responsible $AA_{aid'}$ runs the key update algorithm taking as input the secret key $SK_{aid'}$ of the attribute authority, the revoked attribute $\tilde{x}_{aid'}$, and its current version key $VK_{\tilde{x}_{aid'}}$. The algorithm generates a new version key $VK'_{\tilde{x}_{aid'}} = v'_{\tilde{x}_{aid'}} (v'_{\tilde{x}_{aid'}} \neq v_{\tilde{x}_{aid'}})$ for the revoked attribute $\tilde{x}_{ai'd}$.
The $AA_{aid'}$ then generates a unique update key $UK_{s,\tilde{x}_{aid'},uid}$ for secret key update by each non-revoked user uid as

$$UK_{s,\tilde{x}_{aid'},uid} = H(\tilde{x}_{aid'})^{\beta_{aid'}(v'_{\tilde{x}_{aid'}} - v_{\tilde{x}_{aid'}})(u_{uid}+\gamma_{aid'})}$$

and generates the update key $UK_{c,\tilde{x}_{aid'},uid}$ for ciphertext update as

$$UK_{c,\tilde{x}_{aid'},uid} = \left(UK_{1,\tilde{x}_{aid'}} = \frac{v'_{\tilde{x}_{aid'}}}{v_{\tilde{x}_{aid'}}}, UK_{2,\tilde{x}_{aid'}} = \frac{v_{\tilde{x}_{aid'}} - v'_{\tilde{x}_{aid'}}}{v_{\tilde{x}_{aid'}}\gamma_{aid'}}\right)$$

The $AA_{aid'}$ sends the $UK_{s,\tilde{x}_{aid'},uid}$ to the non-revoked user uid and sends $UK_{c,\tilde{x}_{aid'},uid}$ to the cloud server.
The $AA_{aid'}$ then updates the public attribute key of the revoked attribute $\tilde{x}_{aid'}$ as

$$\widetilde{PK}_{\tilde{x}_{aid'}} = (PK_{\tilde{x}_{aid'}})^{UK_{1,\tilde{x}_{aid'}}}$$

(ii) Secret Key Update - This process is run by the non revoked users in the system. The algorithm produces an updated secret key using the

update key $UK_{s,\tilde{x}_{aid'},uid}$ for the individual users as

$$\widetilde{SK}_{uid,aid'} = \Big(\tilde{K}_{uid,aid'} = K_{uid,aid'}, \tilde{K'}_{uid,aid'} = K'_{uid,aid'},$$
$$\tilde{K}_{\tilde{x}_{aid'},uid} = K_{\tilde{x}_{aid'},uid}.UK_{s,\tilde{x}_{aid'},uid},$$
$$\forall x_{aid'} \in S_{uid,aid'} \backslash \{\tilde{x}_{aid'}\} : \tilde{K}_{\tilde{x}_{aid'},uid} = K_{\tilde{x}_{aid'},uid}\Big)$$

(iii) Ciphertext Update -

$$\widetilde{CT} = \Big(\tilde{C} = C, \tilde{C'} = C', \tilde{C''} = C'',$$
$$\forall 1 \leq i \leq \ell : \tilde{C'}_i = C'_i, \tilde{D}_i = D_i,$$
$$if \rho(i) = \tilde{x}_{aid'} : \tilde{C'}_i = C_i.(D'_i)^{UK_{2,\tilde{x}_{aid'}}}, \tilde{D'}_i = (D'_i)^{UK_{1,\tilde{x}_{aid'}}},$$
$$if \rho(i) \neq \tilde{x}_{aid'} : \tilde{C}_i = C_i, \tilde{D'}_i = D'_i\Big)$$

3.3 Secure Privacy Preserving EHR Framework

This section provides information about the privacy preserving EHR framework that is the main subject of this thesis. We provide a system architectural model that shows the different entities that are part of the framework and how they interact with each other to achieve the functionalities of this framework. We also provide specific use case scenarios together with a detailed description of the individual use cases to provide further information of the different settings in which the framework could be applied.

Our EHR framework is an improvement on the other existing ABE based health care exchange systems in different ways. Our underlying multi-authority scheme is a major improvement on the other single authority schemes in terms of robustness. Also the introduction of a Certificate Authority for user and authority identification allows for a decentralized system in which attribute authorities cannot collude in order to circumvent the system and gain access to data. Also our framework's use of an ABE scheme with outsourced decryption significantly improves its usability. This allows end users to use devices with low computational power to access the necessary data.

3.3.1 System Architecture/Model

The system architecture is made up of several entities that perform different functions with regards to their roles in the system.

System Entities

(1) Users - Users are the members of the system that add or request data. They could be either data owners or data users with individuals able to play both roles in general with the system.

   (i) Data Owner (DO) - The DO is responsible for the generation of data to be stored on the platform and retains ownership of data for the duration of its life cycle. The Data Owner also specifies what requirements are to be met for any user to have access by indicating what attributes they should possess though the policy used for data encryption.

   (ii) Data User (DU) - The DU represents entities that require access to stored data. They are required to possess the right attributes in order to gain access to stored data on the cloud.

(2) Certificate Authority (CA) - The CA plays the central role of assisting with the verification of the identity of the different entities to enable them to securely communicate with one another. It does this by providing the corresponding certificates and public parameters used in the verification process.

(3) Attribute Authority (AA) - The AA is responsible for providing the appropriate public and private key parameters for the different attributes under their control to the appropriate owners and users for the encryption and decryption of data.

(4) Cloud Service Provider (CSP) - This entity provides the necessary storage facilities for data on the platform and also the processing capabilities for part of the operation related to the decryption of data.

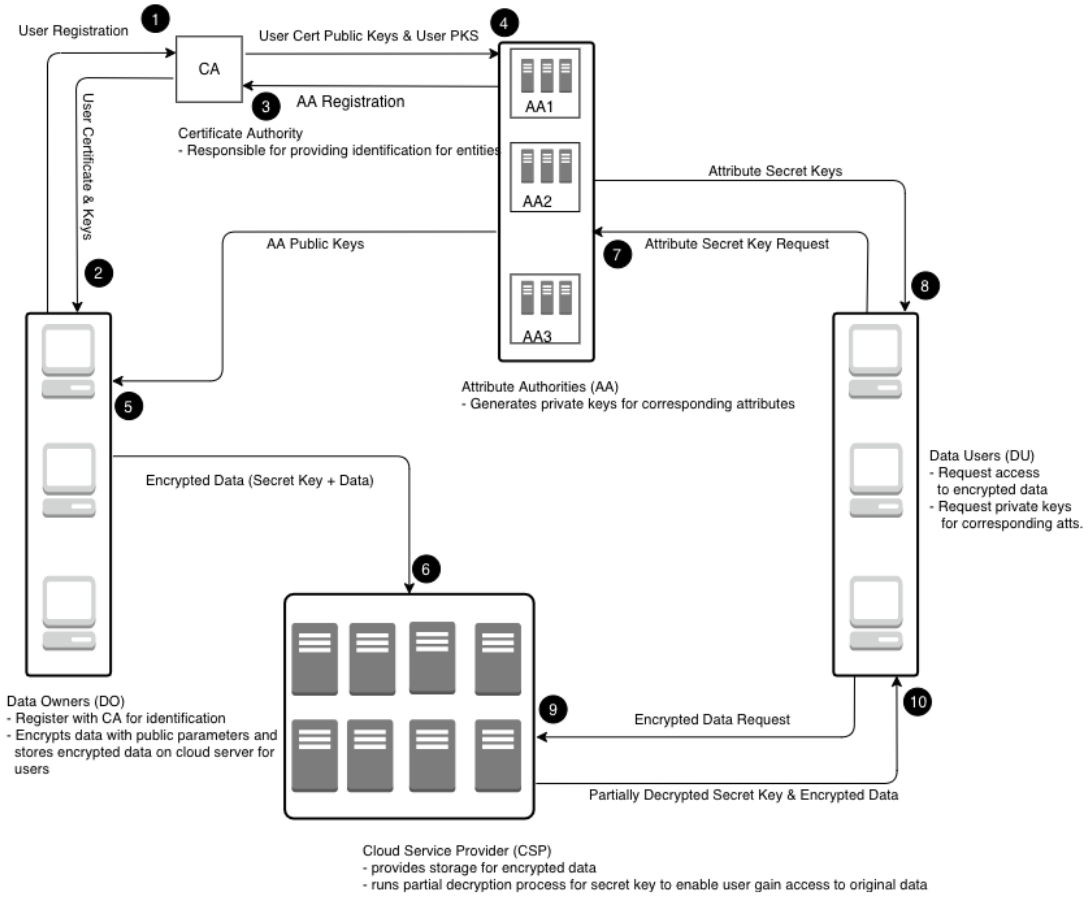

FIGURE 3.1: System Architecture Diagram

System Architecture

Figure 3.1 shows the complete architecture of the entire system framework, indicating the various entities and the different ways in which they interact with each other as part of the overall system. The numbers indicate a typical flow of the different interactions between the entities in the system.

3.3.2 Use Cases

In a real world scenario, hospitals and other medical institutions would play either the role of data owner or data user depending on where the data is collected and where the data is to be used. Hospital A would be an owner for data generated at

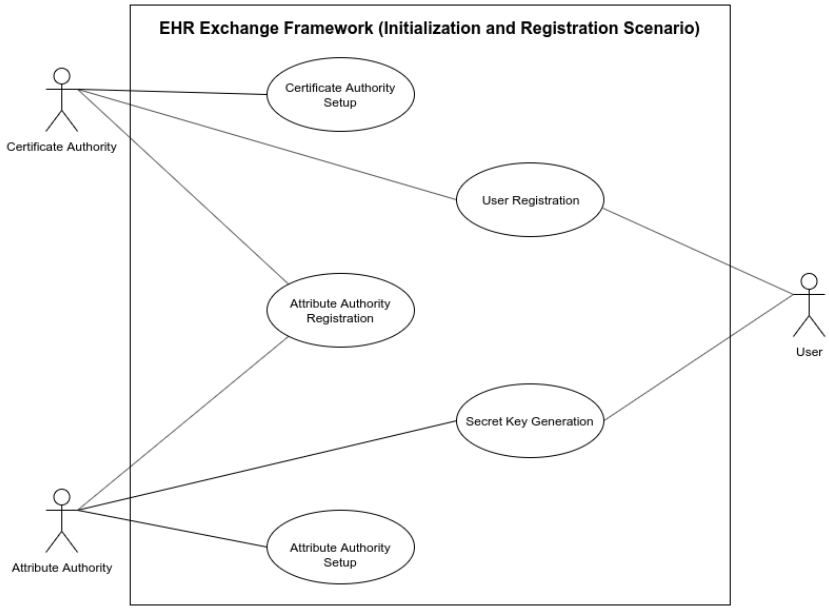

FIGURE 3.2: Use-case diagram for EHR Exchange Framework (Initialization and Registration Scenario)

its location for its patient and Hospital A would be a user when it requires access to data generated at a different hospital which it needs for a range of reasons from treatment to research.

The Attribute Authorities (AAs) in the real world would range from the administrations at different levels in the different hospitals that are part of the framework, the Research Ethic Boards, the administrations at different levels in potential research partners like academic institutions (i.e Universities), etc.

The CSP could be any of the real world cloud service providers such as the AWS cloud, Google cloud to name a few while the CA could be provincial or federal authorities responsible for oversight and regulations related to health care and privacy.

The following figures and tables show different use case scenarios which capture multiple use cases within the framework and the accompanying tables show a more detailed of the use cases captures within the use case scenario diagrams.

Use-Case	Certificate Authority Setup
Actors	Certificate Authority (CA)
Description	This use-case occurs when the CA initializes the framework. The CA generates the Global Master Key (GMK) and Global Public Parameters (GPP) for the entire framework which would be used for other processes within the framework.
Stimulus	Certificate Authority runs CASetup algorithm to initialize system
Response	Initialize system and derive GPP and GMK.

TABLE 3.1: Certificate Authority Setup use-case description.

Use-Case	User Registration
Actors	Certificate Authority, User (Data Owner & User)
Description	This use-case occurs when a user requests to be added to the system. The CA verifies their identity and then creates an entry for the new user which includes their unique id and their public parameters.
Stimulus	CA runs UserReg algorithm to register a new user.
Response	New user registered with their public information stored. The user receives a public and secret key, a certificate and the GPP for the framework.

TABLE 3.2: User Registration use-case description.

Use-Case	Attribute Authority Registration
Actors	Certificate Authority (CA), Attribute Authority (AA)
Description	This use-case occurs when an AA requests to be added to the system. The CA verifies their identity and then creates an entry for the new authority which includes their unique id.
Stimulus	CA runs the AAReg algorithm.
Response	New AA registered and the AA receives the public information of all registered users in the system and the verification keys for the generated user certificates. This triggers the Attribute Authority Setup use-case.

TABLE 3.3: Attribute Authority Registration use-case description.

Use-Case	Attribute Authority Setup
Actors	Attribute Authority (AA)
Description	This use-case occurs when a new AA is being initialized. The AA generates its public and secret keys together with the public keys for every attribute under its control.
Stimulus	Attribute Authority runs AASetup algorithm.
Response	AA initialized with its public and private keys together with the public keys for its attributes.

TABLE 3.4: Attribute Authority Setup use-case description.

Use-Case	Secret Key Generation
Actors	User (Data Owner & User), Attribute Authority (AA)
Description	This use case occurs when a user requests a secret key from an AA in the system. The AA verifies the user identity and provides the user with the appropriate secret keys for that authority and the attributes the user has been assigned.
Stimulus	User secret key request.
Response	If the user identity is verified using the available verification key for their certificate, they are provided with unique secret keys from the AA for the corresponding attributes.

TABLE 3.5: Secret Key Generation use-case description.

Use-Case	Data Encryption
Actors	User (Data Owner), Cloud Service Provider (CSP)
Description	This use case occurs when a data owner (also a user) wants to add data to the the framework. The encrypt the data using a symmetric key and then encrypt the symmetric key using the appropriate access policy as part of the underlying ABE scheme. Encrypted data is stored on the cloud.
Stimulus	User data storage.
Response	Encrypted data is stored on the cloud.

TABLE 3.6: Data Encryption use-case description.

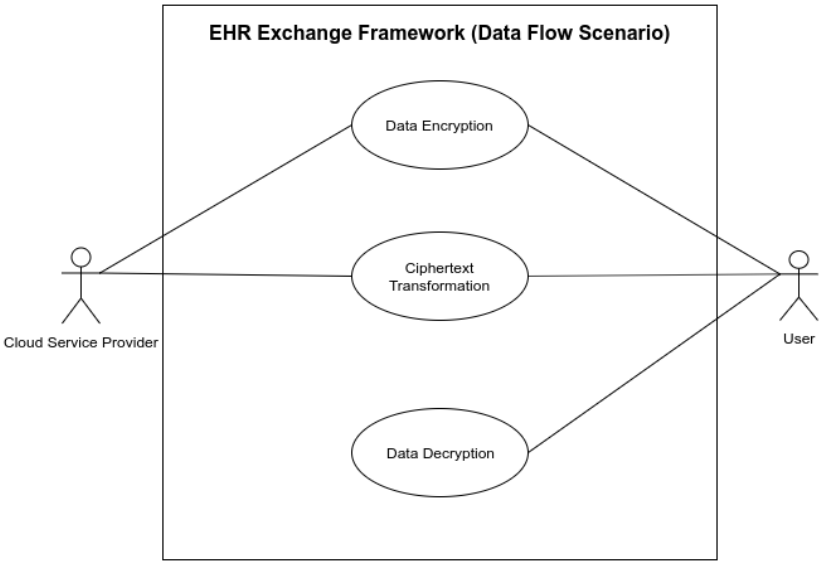

FIGURE 3.3: Use-case diagram for EHR Exchange Framework (Data Flow Scenario)

Use-Case	Ciphertext Transformation
Actors	User (Data User), Cloud Service Provider (CSP)
Description	This use case occurs when a user is requesting data from the cloud. The user provides the appropriate information and if they are valid, the CSP transforms the stored ciphertext into an El-Gamal style ciphertext
Stimulus	User data request from CSP
Response	CTransform algorithm is run and if the provided information is valid, a partially decrypted ciphertext is returned to the user.

TABLE 3.7: Ciphertext Transformation use-case description.

Use-Case	Data Decryption
Actors	User (Data User)
Description	This use case occurs when the user decrypts the partially decrypted ciphertext in order to access the symmetric encryption key which is then used to decrypt the symmetrically encrypted data.
Stimulus	User receives partially decrypted ciphertext.
Response	If successful, user gains access to symmetric key which also grants them access to the original data in plain form.

TABLE 3.8: Data Decryption use-case description.

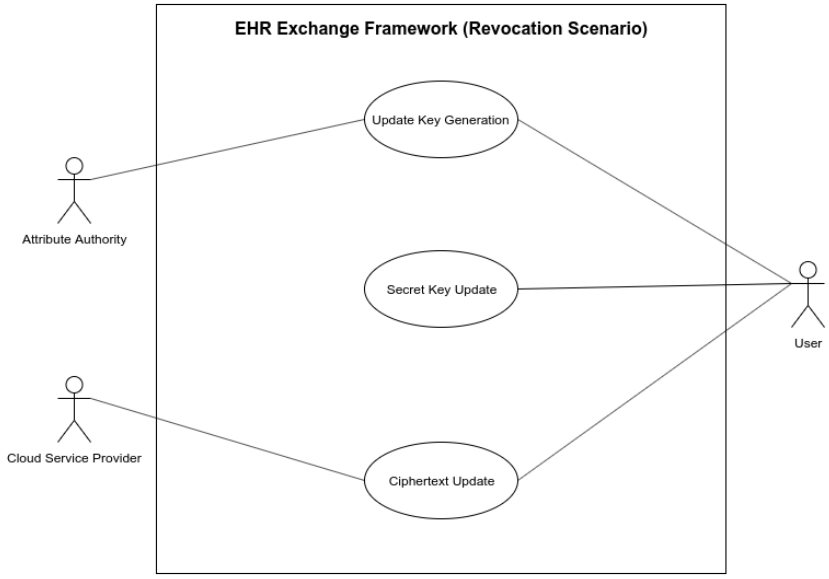

FIGURE 3.4: Use-case diagram for EHR Exchange Framework (Revocation Scenario)

Use-Case	Update Key Generation
Actors	Attribute Authority (AA), User (Data Owner & User)
Description	This use case occurs when an existing user is being revoked based on attributes they have secret keys for. The AA that has control of the attribute generates an update key pair. One to be used by users who share that attribute and still have valid access. The other to be used by the data owner via the CSP to update the affected ciphertexts.
Stimulus	The user who owns the corresponding data puts in a request with the appropriate AA who runs the UKGen algorithm.
Response	If successful, provided update keys for the secret keys and ciphertexts to the appropriate users.

TABLE 3.9: Update Key Generation use-case description.

Use-Case	Secret Key Update
Actors	User (Data Owner & User)
Description	This use case occurs when still valid users need to update their keys to retain access to ciphertext encrypted with attributes for which an existing user has been revoked. The user uses the update key it has received from the Attribute Authority.
Stimulus	Existing user is revoked and a still valid user updates their secret key for the corresponding attribute.
Response	User's secret keys matching the affected attribute is updated.

TABLE 3.10: Secret Key Update use-case description.

Use-Case	Ciphertext Update
Actors	Cloud Service Provider (CSP), User (Data Owner)
Description	This use case occurs when a data owner requests to update the current ciphertexts that are part of the system after an existing user has had their secret keys revoked. The corresponding ciphertext is updated by the CSP using the provided update key.
Stimulus	User (Data Owner) puts in a ciphertext update request.
Response	Ciphertext is updated to prevent further access by the revoked user.

TABLE 3.11: Ciphertext Update use-case description.

Chapter 4

Evaluation and Results

This chapter highlights the information about the software and hardware specifications involved in the implementation of the underlying framework and also shows the results of performance tests for the underlying ABE scheme in relation to scalability of computational overhead with regards to the number of attribute authorities and attributes involved in the system for data encryption, data decryption, and user revocation.

4.1 Experimental Setup

We evaluated the performance of our system framework by implementing the underlying ABE scheme in an environment using hardware and software tools as described below. We have focused on the performance of the underlying ABE scheme as there have been well established performance metrics for the multiple available types of symmetric encryption that could be used for the encryption of the actual data, with the ABE scheme used for the encryption of the symmetric key.

4.1.1 Hardware and Software Tools

- Linux Server - We used a Linux Server running Ubuntu 18.04.3 LTS with 12GB RAM size and an Intel Core i5-6400 CPU @ 2.70GH x 4 processor.

- Python - We wrote our underlying code using Python 3.7 as this gave us access to some other effective libraries that played a role in the robustness of our code and evaluation during our experiments.

- Charm Crypto Library[1] - We used the charm-crypto library for the implementation of the ABE components of our framework. Charm is a library implemented in python that has multiple contributors in the cryptographic field and as a result gives access to a robust tool set that provides functionalities related to pairing based public key encryption schemes such as ABE. Our scheme was implemented using 'SS512', a super-singular elliptic curve with a 512-bit base field with groups of prime order.

4.2 Performance Evaluation

We evaluated the performance of our system by evaluating the computational times for data encryption and decryption together with the time taken for users to update their attribute based secret keys. The focus of our evaluation are the processes that are run at the user end as we are working under the assumption that the Cloud Service Provider (CSP) provides in theory unlimited processing power unlike what could be available on the several devices at the user end.

We encrypt and decrypt sample symmetric keys using the underlying ABE scheme. We also use access policies that exclusively have 'AND' gates to push the limits of the maximum number of attributes that can be part of the access policy or user keys with respect to the number of Attribute Authorities (AA) involved. Our experiments have been run for 500 trials with the results being an average of the individual runs.

Our results for encryption as indicated in figure 4.1 show that our encryption system is identical in performance with the encryption algorithm of the adapted scheme for this framework [41]. Encryption requires mostly exponential operations and the amount of time scales linearly with an increase in the number of AAs and attributes that are contained in the access policy under which the file is being encrypted.

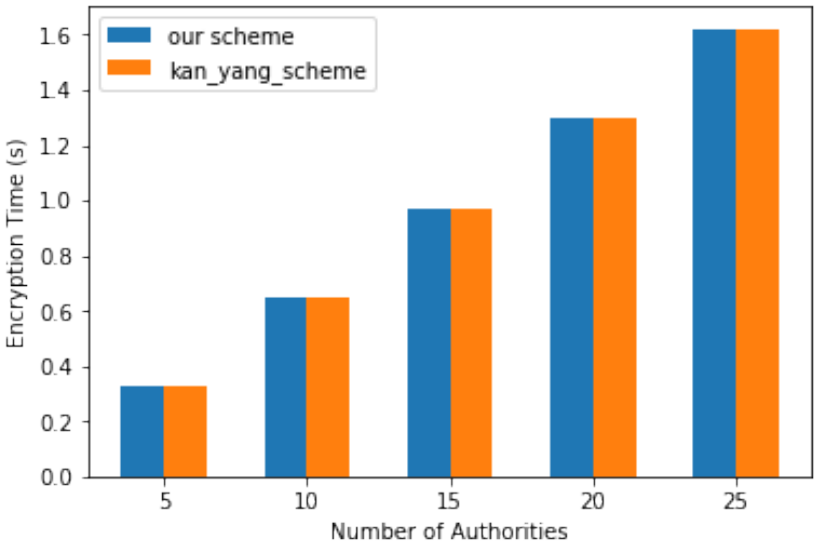

FIGURE 4.1: Encryption Computation Time

Our results for decryption show a constant decryption time irrespective of the number of attribute authorities or attributes in comparison with the increase in decryption time of the adapted scheme [41]. This can be seen in figure 4.2. These results indicate that the amount of time for decryption involving a single attribute will be the same as that involving a hundred or more attributes, keeping the computational overhead constant and allowing for the use of low computational devices for the decryption process. This is because decryption in the framework only requires a single exponentiation operation irrespective of the number of AAs or attributes. The pairing operations which would have been part of the decryption process as used in kan yang scheme[41] have been outsourced to the Cloud Service Provider (CSP).

Our setup for revocation was focused on showing how the time for revocation in terms of an existing user updating their attribute based secret keys when an existing user has been revoked scales with an increase in the number of attributes being revoked. Our results show that the time scales linearly with an increase in the number of attributes being revoked. This is as a result of the exponentiation operations that are involved in the update process. The result of our experiment can be seen in figure 4.3.

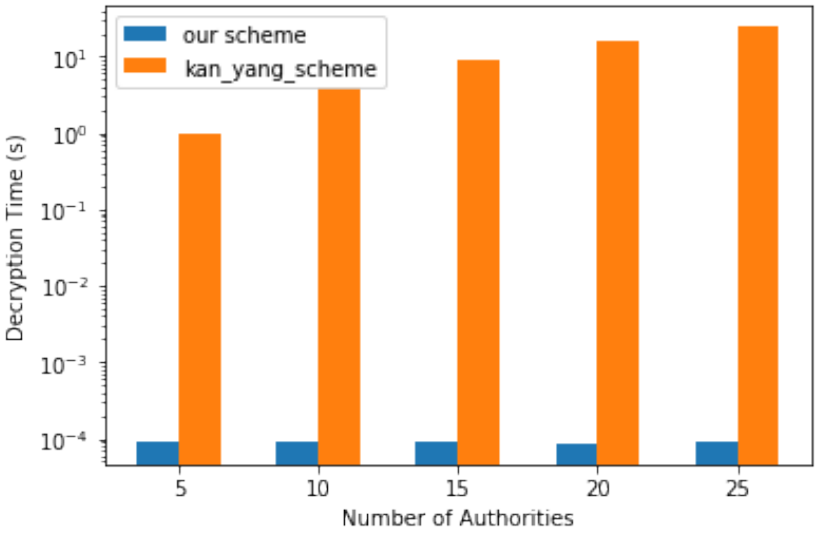

FIGURE 4.2: Decryption Computation Time

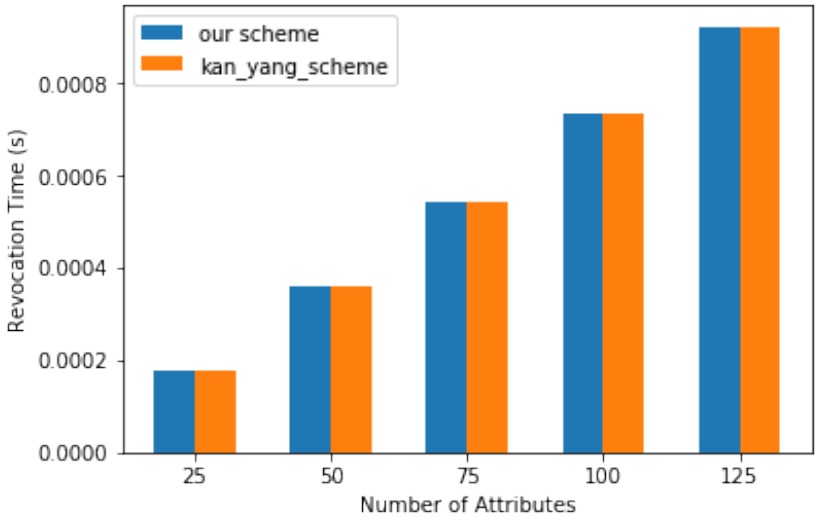

FIGURE 4.3: Revocation Computation Time

4.3 Security Analysis

In this section we analyze the different security and privacy properties of the framework. We describe how the framework is able to achieve the conditions necessary for confidentiality,integrity, and authentication including how it is able to preserve privacy through its processes.

Confidentiality and Integrity

The framework is able to ensure confidentiality by ensuring that users are only able to gain access to data that their access levels allow. Users are only able to decrypt data for which they have the required attributes that match the accompanying access structure. Also due to the randomness inserted in the generation of user secret key for attributes, multiple users are unable to combine their secret keys for individual attributes to gain access to data under a policy that they are individually unable to access. The integrity of data is ensured as users are only able to access data that they have been authorized for as there is a record of the different users who have been granted secret keys for attributes that match the policy.

The ability to revoke user access also enables the maintenance of the integrity of the system. Also the certificates generated by the Certificate Authority ensures that only valid users are able to request secret keys from the Attribute Authorities in the system. Also no single AA is able to grant itself access to data within the framework and the CA only plays the role of identity validation and does not have access to a universal key that grants it access to the stored encrypted data. Also the Cloud Service Provider (CSP) never has access to data in its plain form ensuring that confidentiality and integrity are maintained.

Authentication

Our framework has authentication built into its cryptographic processes. The fact that users need access to certain attributes which they need to request from specific authorities is in its own way a form of access control. This ensures that not only do unauthorized users not gain any level of access to the system, the authorized

users are only able to gain access to data that they have been authorized for, preventing them from being able to access data for which they do not have any permissions. The access policy acts as the rules while the attributes could be seen as the permissions that grant users access if they fit.

Privacy Preserving

Privacy is preserved as part of our framework through the fact that data owners are able to have some control over the level of access to data by specifying the appropriate access policy when encrypting data before it is stored on the cloud. Also the ability of the system to revoke access also ensures that user access can have a time limit depending on the preset rules before they are granted access. Also the strong underlying confidentiality, integrity and authentication mechanisms aid in ensuring data privacy.

Chapter 5

Conclusion and Future Work

5.1 Overview

This chapter provides a description of the contribution of this thesis and also provides recommendation fo future developments that could be implemented in extending the functionalities of the framework.

5.2 Contribution

The main contribution of this thesis is a framework for the exchange of electronic health data with significant decrease in the computational requirements in terms of time and resources. This is in response to the need to be able to securely exchange sensitive medical data not simply among medical practitioners but also with 3rd party partner such as researchers who contribute to developments in the area while preserving privacy and maintaining security which becomes a problem when the trust zone is extended to 3rd party computational services as provided by cloud service providers. We have modified an existing Attribute Based Encryption (ABE) scheme to enable the outsourcing of decryption in order ot allow end users use devices with low computational abilities to gain access to data securely. Our system uses symmetric key encryption to encrypt data and uses our ABE scheme to encrypt the symmetric key thereby providing an encryption scheme with in-built access control.

5.3 Future Work

- Accountability..challenge as users share attributes
- Searchable Encryption

- where it could go in terms of efficiency and accountability

Bibliography

[1] Joseph A. Akinyele et al. "Charm: a framework for rapidly prototyping cryptosystems". In: *Journal of Cryptographic Engineering* 3.2 (2013), pp. 111–128. ISSN: 2190-8508. DOI: 10.1007/s13389-013-0057-3. URL: http://dx.doi.org/10.1007/s13389-013-0057-3.

[2] Joseph A Akinyele et al. "Self-Protecting Electronic Medical Records Using Attribute-Based Encryption". In: *ePrint IACR org* 1 (2010), pp. 1–20. URL: http://eprint.iacr.org/2010/565.

[3] Suhair Alshehri, Stanisław Radziszowski, and Rajendra K Raj. "Designing a Secure Cloud-Based EHR System using Ciphertext-Policy Attribute-Based Encryption". In: ().

[4] Mrinmoy Barua et al. "ESPAC: Enabling Security and Patient-centric Access Control for eHealth in cloud computing". In: *International Journal of Security and Networks* 6.2/3 (2011), p. 67. ISSN: 1747-8405. DOI: 10.1504/IJSN.2011.043666.

[5] Amos Beimel. "Secure Schemes for Secret Sharing and Key Distribution". Doctor of Science. Israel Institute of Technology, 1996, p. 115.

[6] Josh Benaloh and Jerry Leichter. "Generalized Secret Sharing and Monotone Functions". In: *Advances in Cryptology — CRYPTO' 88* 403 (1988), pp. 27–35. DOI: 10.1007/0-387-34799-2_3. URL: http://dx.doi.org/10.1007/0-387-34799-2{_}3.

[7] John Bethencourt and Brent Waters. "Ciphertext-Policy Attribute-Based Encryption". In: (2007).

[8] G.R. Blakley. "Safeguarding cryptographic keys". In: *Afips* (1979), p. 313. ISSN: 0095-6880. DOI: 10.1109/AFIPS.1979.98. URL: http://www.computer.org/portal/web/csdl/doi/10.1109/AFIPS.1979.98.

[9] Dan Boneh and Matthew Franklin. "Identity-Based Encryption from the Weil Pairing". In: *SIAM Journal on Computing* 32.3 (2003), pp. 586–615. ISSN: 0097-5397. DOI: 10.1137/S0097539701398521.

[10] Ann Cavoukian. "A Guide to the Personal Health Information Protection Act". In: *Access* December (2004). ISSN: 1098-2752.

[11] Ann Cavoukian. "Privacy Impact Assessment Guidelines for the Ontario Personal Health Information Protection Act". In: October (2005).

[12] Melissa Chase. "Multi-authority Attribute Based Encryption". In: *Proceedings of the 4th Conference on Theory of Cryptography* 4392 (2007), pp. 515–534. ISSN: 03029743. DOI: 10.1007/978-3-540-70936-7. URL: http://www.springerlink.com/index/10.1007/978-3-540-70936-7.

[13] Melissa Chase and Sherman S M Chow. "Improving privacy and security in multi-authority attribute-based encryption". In: *ACM Conference on Computer and Communications Security* (2009), pp. 121–130. URL: papers3://publication/uuid/7B75720C-27AE-4FE4-AB73-BBA0FC0BAA87.

[14] Yong Cheng et al. "Efficient revocation in ciphertext-policy attribute-based encryption based cryptographic cloud storage". In: *Journal of Zhejiang University SCIENCE C* 14.2 (2013), pp. 85–97. ISSN: 1869-1951. DOI: 10.1631/jzus.C1200240. URL: http://link.springer.com/10.1631/jzus.C1200240.

[15] Sherman S M Chow. "New Privacy-Preserving Architectures for Identity- / Attribute-based Encryption". In: September (2010), p. 129.

[16] Sherman S.M. Chow. "A Framework of Multi-Authority Attribute-Based Encryption with Outsourcing and Revocation". In: *Proceedings of the 21st ACM on Symposium on Access Control Models and Technologies - SACMAT '16* (2016), pp. 215–226. DOI: 10.1145/2914642.2914659. URL: http://dl.acm.org/citation.cfm?doid=2914642.2914659.

[17] Vipul Goyal et al. "Attribute-based Encryption for Fine-grained Access Control of Encrypted Data". In: *Proceedings of the 13th ACM Conference on Computer and Communications Security* (2006), pp. 89–98. ISSN: 15437221. DOI: 10.1145/1180405.1180418. URL: http://doi.acm.org/10.1145/1180405.1180418$\backslash$nhttp://portal.acm.org/citation.cfm?doid=1180405.1180418.

[18] Matthew Green, Susan Hohenberger, and Brent Waters. "Outsourcing the Decryption of ABE Ciphertexts". In: *Proceedings of the 20th USENIX conference on Security* (2011), pp. 34–34.

[19] George Hsieh and Rong-Jaye Chen. "Design for a secure interoperable cloud-based Personal Health Record service". In: *4th IEEE International Conference on Cloud Computing Technology and Science Proceedings* (2012), pp. 472–479. DOI: 10.1109/CloudCom.2012.6427582. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6427582.

[20] Thomas Hupperich et al. "Flexible patient-controlled security for electronic health records". In: *Proceedings of the 2nd ACM SIGHIT symposium on International health informatics - IHI '12* (2012), p. 727. ISSN: 4503-0781. DOI: 10.1145/2110363.2110448. URL: http://dl.acm.org/citation.cfm?id=2110363.2110448$\backslash$nhttp://dl.acm.org/citation.cfm?doid=2110363.2110448.

[21] Luan Ibraimi, Muhammad Asim, and Milan Petković. "Secure management of personal health records by applying attribute-based encryption". In: *Proceedings of the 6th International Workshop on Wearable, Micro, and Nano Technologies for Personalized Health: "Facing Future Healthcare Needs", pHealth 2009* (2010), pp. 71–74. DOI: 10.1109/PHEALTH.2009.5754828.

[22] Mitsuru Ito, Akira Saito, and Takao Nishizeki. *Secret sharing scheme realizing general access structure*. 1989. DOI: 10.1002/ecjc.4430720906. URL: http://dx.doi.org/10.1002/ecjc.4430720906.

[23] Richard Kissel. "Glossary of Key Information Security Terms Glossary of Key Information Security Terms". In: *Nist* NISTIR 729.Revision 2 (2013). DOI: 10.6028/NIST.IR.7298r2.

[24] Allison Lewko and Brent Waters. "Decentralizing attribute-based encryption". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6632 LNCS (2011), pp. 568–588. ISSN: 03029743. DOI: 10.1007/978-3-642-20465-4{_}31.

[25] Jin Li et al. "Fine-grained access control system based on outsourced attribute-based encryption". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8134 LNCS (2013), pp. 592–609. ISSN: 03029743. DOI: 10.1007/978-3-642-40203-6_33.

[26] Jin Li et al. "Securely outsourcing attribute-based encryption with checkability". In: *IEEE Transactions on Parallel and Distributed Systems* 25.8 (2014), pp. 2201–2210. ISSN: 10459219. DOI: 10.1109/TPDS.2013.271.

[27] Ming Li et al. "Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute-Based Encryption". In: *Tpds* 24.1 (2013), pp. 131–143. ISSN: 1045-9219. DOI: 10.1109/TPDS.2012.97.

[28] Xiaohui Liang et al. "Ciphertext Policy Attribute Based Encryption with Efficient Revocation". In: (2011), pp. 1–21. URL: http://bbcr.uwaterloo.ca/{~}x27liang/papers/abewithrevocation.pdf.

[29] Ben Lynn. "On The Implementation of Pairing-Based Cryptosystems". PhD thesis. 2007, p. 126. DOI: 10.1007/s00145-004-0311-z.

[30] Peter Mell and Timothy Grance. "The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology". In: *Nist Special Publication* 145 (2011), p. 7. ISSN: 00845612. DOI: 10.1136/emj.2010.096966. URL: http://www.mendeley.com/research/the-nist-definition-about-cloud-computing/.

[31] A. Menezes, P. van Oorschot, and S. Vanstone. "Chapter 01: Overview of Cryptography". In: *Handbook of Applied Cryptography* (1996), pp. 1–48.

[32] Shivaramakrishnan Narayan, Martin Gagné, and Reihaneh Safavi-Naini. "Privacy preserving EHR system using attribute-based infrastructure". In: *Proceedings of the 2010 ACM workshop on Cloud computing security workshop - CCSW '10* (2010), p. 47. ISSN: 15437221. DOI: 10.1145/1866835.1866845. URL: http://portal.acm.org/citation.cfm?doid=1866835.1866845.

[33] Siani Pearson. *Privacy and Security for Cloud Computing*. 2013, pp. 3–42. ISBN: 978-1-4471-4188-4. DOI: 10.1007/978-1-4471-4189-1. URL: http://link.springer.com/10.1007/978-1-4471-4189-1.

[34] Amit Sahai and Brent Waters. "Fuzzy Identity-Based Encryption". In: (2005), pp. 457–473.

[35] Adi Shamir. "How To Share a Secret". In: *Communications of the ACM (CACM)* 22.11 (1979), pp. 612–613. ISSN: 0001-0782. DOI: http://doi.acm.org/10.1145/359168.359176. URL: http://doi.acm.org/10.1145/359168.359176.

[36] Adi Shamir. "Identity-Based Cryptosystems and Signature Schemes". In: *Advances in Cryptology* 196 (1985), pp. 47–53. DOI: 10.1007/3-540-39568-7{_}5.

[37] *Understanding EHRs, EMRs and PHRsNo Title*. URL: https://www.infoway-inforoute.ca/en/what-we-do/digital-health-and-you/understanding-ehrs-emrs-and-phrs.

[38] Allison Lewko and Brent Waters. *Decentralizing Attribute-Based Encryption*. Cryptology ePrint Archive, Report 2010/351. http://eprint.iacr.org/. 2010.

[39] Yu Shucheng et al. "Attribute based data sharing with attribute revocation". In: *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security - ASIACCS '10* (2010), p. 261. DOI: 10.1145/1755688.1755720. URL: http://dl.acm.org/citation.cfm?id=1755688.1755720.

[40] Brent Waters. "Ciphertext-Policy Attribute-Based Encryption : An Expressive , Efficient , and Provably Secure Realization". In: 02.subaward 641 (2011), pp. 53–70.

[41] Kan Yang and Xiaohua Jia. "Expressive, efficient, and revocable data access control for multi-authority cloud storage". In: *IEEE Transactions on Parallel and Distributed Systems* 25.7 (2014), pp. 1735–1744. ISSN: 10459219. DOI: 10.1109/TPDS.2013.253.

[42] Kan Yang et al. "DAC-MACS: Effective data access control for multi-authority cloud storage systems". In: *2013 Proceedings IEEE INFOCOM* (2013), pp. 2895–2903. ISSN: 1556-6013. DOI: 10.1109/INFCOM.2013.6567100. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6567100.

[43] Shucheng Yu et al. "Achieving secure,scalable ,and fine-grained data access control in cloud computing.pdf". In: *Ieee Infocom* (2010), pp. 1–9. ISSN: 0743-166X. DOI: 10.1109/INFCOM.2010.5462174.