Master of Science in Software Engineering
February 2017

Attribute Based Encryption of Electronic Health Records

Comparative study of existing algorithms

Arun Tej Seethamraju

Faculty of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona Sweden

This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering. The thesis is equivalent to 20 weeks of full time studies.

Contact Information:
Author:
Arun Tej Seethamraju
Email: arse15@student.bth.se

University advisor:
Dr. Emiliano Casalicchio
Department of Computer Science and Engineering

Faculty of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona, Sweden

Internet	: www.bth.se
Phone	: +46 455 38 50 00
Fax	: +46 455 38 50 57

ABSTRACT

Context: Cloud Computing today, is an evolving technology which features large Data Storage and ready-to access from any device. The Healthcare Industry stores large Databases of patient's records, considering at the advantages of Cloud Computing it is looking forward to move on from the traditional, proprietary Database Management Model into an Open Source Cloud DBMS Model. To complete this transition, it is of primary importance to provide Privacy and Security for Electronic Medical Record / Electronic Health Record. There are several researches being done on how to mitigate these privacy issues using algorithms like Attribute Based Encryption and Identity Based Encryption. In this study, we compare the performance of these two attribute based encryption methods.

Objectives: This thesis compares the performance of the state-of-the art Attribute Based Encryption Schemas for Electronic Medical Record / Electronic Health Record Systems. Performance evaluation is conducted in local and cloud environments.

Methods: A Literature Review has been performed to identify the existing Cloud based Electronic Health Record Systems which uses the attribute based encryption as a mechanism to mitigate the privacy issues and realization in Cloud. Two algorithms have been selected by performing snowballing from the IEEE Research Articles. Experimentation was performed on the two algorithms in a local machine and on Amazon Web Services Cloud Platform to compare the performance.

Results: Verification of performance in each stage of the execution of the algorithms, in both local machine and Cloud environment was done.

Conclusions: It is concluded that a combination of both the selected algorithms would lead to a better performance and security.

Key Policy – Attribute based encryption defines the access structure while creating keys.
Cipher Policy – Attribute based encryption defines the access structure while encrypting the file.

Keywords: Attribute Based Encryption, Electronic Health Records,
Cloud Computing, Privacy.

ACKNOWLEDGEMENTS

I would like to take this opportunity to sincerely thank my Thesis Supervisor, **Dr.Emiliano Casalicchio** for his kind support and erudite guidance during this preparation. If not for his support and feedback this thesis would not have been completed.

None of this could have happened without my parents, Mr. Dasaradhi Seethmaraju and Mrs. Padma Latha Seethamraju for their unconditional love and support throughout my thesis. I would also like to thank my friends for their support and suggestions.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

Over the past few years' Cloud Computing became a phenomenon because of its characteristics such as Elasticity, Pay-per-use, and Scalability, etc., offering cost-efficient and a large pool of virtual resources which can be deployed for varied Applications / Services to the users [1]. Users can access their Data stored in Cloud from anywhere on-the-go through the web. Thus, making it easier to maintain Cloud Databases compared to Traditional Databases [1]. Cloud Computing Solutions are less expensive than their traditional counterparts as the Cloud pricing is based on pay-per-use. Cloud Computing can be integrated into the Healthcare Sector as the Healthcare Providers are facing a momentous task in handling the Electronic Health Records (EHR) for their patients (healthcare can leverage Cloud for its application). EHR can be easily maintained in Cloud so that the patients can easily access their records, book appointments, etc. if they are hosted in Cloud.

Since Cloud Computing is an emerging technology which is currently in development, it has its downsides such as Privacy and Security concerns. Privacy is very essential when it comes to storage of personal data such as EHR. Per the definition set out in the Health Insurance Portability and Accountability Act (HIPPA): "The confidential section of the electronic medical record needs to be protected. Thus, a mechanism to protect the patient's privacy is needed during electronic medical record exchange and sharing" [2]. Also, Health Insurance Portability and Accountability Act (HIPPA) defined the following categories for ensuring privacy of EHR: "*Privacy Protection Mechanism can be categorized into four types; namely Anonymity, Pseudonymity, Unlinkability, and Unobservability"* [3].

Medical Records Institute, US, recommends five stages for medical records digitization [3]:

- Automated Medical Records (AMR): A Basic Medical Record, without containing extensive Data about the Medication, Stage of disease, etc.
- Computerized Medical Records (CMR): This Record is provided by a Specific Department with in the Hospital. These records are currently being used by most of the Hospitals and Clinics [3].
- Electronic Medical Records (EMR): It is a Digital Chart which consists of the Medical Record of the Patient form a Single Practice [4].
- Electronic Patient Records (EPR): These Records are the collection of all the Medical History of a Patient.
- Electronic Health Records (EHR): These are same as EPR but these provide access to Tools which can be helpful to make decisions on patients' problems[5].

Extensive research is being done for the realization of EHR; there has been a breakthrough by several researchers which allow sharing of the EHR on Cloud. Many researchers have proposed the usage of Attribute Based Encryption (ABE) Techniques to solve the privacy issues. ABE algorithms have been mostly chosen because the EHR consists of lots of data which can be converted into Cipher Keys and can be used to construct ABE algorithms having two to three layered encryption and decryption process both at the user-end and server-end.

1.1 Problem Outline

There is extensive research going on to develop a compatible ABE, addressing the privacy and security issues of EHR in Cloud. There are several research papers published on different approaches to use the ABE such as using Cipher-text, Role-based, Trust-based, etc.

Hence, there is a need to find the most appropriate method that is more efficient and can help in development and testing of the algorithms. Taking this into consideration the following research gap has been identified:

- Need for a comparative study of the existing ABE schemas to implement EHR on Cloud Platforms.

1.2 Aim and Objectives

Aim:

Aim of the thesis is to conduct a Quantitative Study on comparison of existing ABE schemas used in Cloud based EHR.

Objectives:

- Review the privacy issues in Cloud based e-health systems.
- Review the implementation of ABE based Mechanism in Cloud based EHR.
- Identification of performance metrics to evaluate EHR Cloud based Systems.
- Quantitative Comparison of 2-3 different ABE algorithms which can be realized with EHR privacy in Cloud.

1.3 Research questions

Following research questions were formulated to accomplish the Aim and Objectives of this thesis:

1. What are the state-of-the-art ABE based Algorithms / Schema for EHR Systems in Cloud?

   Motivation: Identifying different state-of-the-art ABE Algorithms / Schemas is the key to this quantitative study. Identifying the ABE Algorithms, which are available and would help for further research in developing sustainable ABE Algorithms for Access Control or Data encryption and decryption.

2. What are the existing Performance Metrics to evaluate ABE Algorithms / Schemas?

   Motivation: Analyzing the performance metrics such as user time, system time, CPU utilization and No. of read and write operation performed on the disk.

3. How the selected ABE Algorithms / Schema compare based on metrics found in RQ2 in local and cloud environment?

   Motivation: Comparing the existing ABE Algorithms / Schema, firstly, in a Local Machine and then on Amazon Web Service Cloud Platform. This would help us to analyze how the algorithms perform and would vary in both the environments to develop a sustainable ABE Algorithm / Schema.

1.4 Contribution

Following are the contributions achieved during the thesis:

- Identifying various Privacy issues in Cloud Computing.
- Identifying various state-of-the art ABE Schemas in EHR scenario.
- Comparing the Performance of state-of-the-art ABE Schemas.

1.5 Structure of thesis

Thesis is documented and presented in the following manner:

- Chapter 2: Related work done in the field of study, here we discuss various studies to fill in the research gap of the thesis.
- Chapter 3: Research methodology, a detailed description of the methods which were used in the thesis.
- Chapter 4: Experimentation results and Analysis, in this chapter we evaluate the selected algorithms with metrics obtained from literature review and perform paired sample t-test to test the significance of the results obtained.
- Chapter 5: Discussion, in this chapter we discuss the answers to research questions, contributions made in the thesis and threats to validity of the.
- Chapter 6: Conclusion and Future work, in this chapter we conclude with the findings of the thesis and further scope of improvisation for this study.

2 RELATED WORK

This chapter exhibits the background and related work done in the field of study. Section 2.1 describes why Researchers are considering Cloud for storing EHR’s. Section 2.2 describes how researchers are tackling the privacy issue of EHR's in the cloud.

2.1 EHR’s in Cloud

In eHealth application model, users/patients can login into system and create or upload their EHR. These EHR are maintained by healthcare provider/hospital, users’ credentials and these EHR is stored in their data store. Figure 2-1 is a depiction of a basic traditional eHealth system. In recent years, healthcare providers are moving these data stores into cloud. Cloud Computing provides large amounts of data storage capability that can be accessed on-the-go and further lowers the costs. Hence, can be used for EHR storage scenarios [6]. EHR’s are easy to Manage, Update, Access, and Share compared to the traditional paper alternatives[6]. EHR’s can be very useful in emergency scenario’s where it can help the doctor and the medical staff to access previous healthcare records of a patient, thus improving the decision-making process [7].

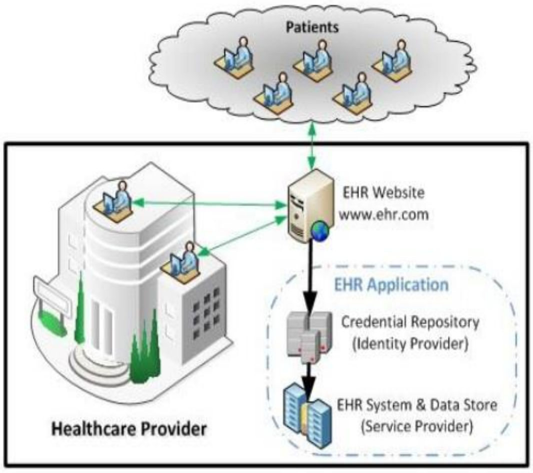


Figure 2-1 Basic eHealth system [14].

Using EHR, a healthcare provider can view the entire medical information about the patient without the need for tracking older medical records[8]. Authors [3] stated that sharing and the management of traditional EHR systems are slow and expensive. The inefficient process can be mitigated if EHR’s are maintained in the cloud. Authors [9] also stated that interoperation and sharing of traditional EHR’s among various healthcare institutions have been extremely slow, which has been the cited as the biggest obstacle in the adoption of health IT, specifically EHR systems. Many countries are planning to migrate their traditional healthcare services into cloud-based EHR systems, to improve the quality and delivery cost for healthcare services [10]–[12].

Authors have [13] discussed that their Cloud-based EHR Systems, if implemented on a National Scale, would help in providing a cost-effective alternative for patients in rural areas. By motivating and encouraging people in rural areas to upload their records to Cloud, and thus providing them better health care alternatives, which are cheaper and better with an improved supervision and support during all emergency scenarios.

2.2 Mitigating Privacy issues for EHR in Cloud

Per HIPPA, Healthcare Data needs high Security and Privacy. Most of the Researchers used Attribute-Based Encryption (ABE) Schemas for securing their research to protect EHR in Cloud. Authors [14] introduced ABE as a new encryption schema for access control [15]. In ABE Systems data is usually encrypted for more than one user compared to the traditional public-key cryptography. ABE encrypts data with a set of attributes or a policy over attributes [15]. Any user with a key which matches the attributes or the policy can be able to decrypt the data and access it [15].

The following are the state-of-the art algorithms currently being used by researchers in their articles to mitigate privacy issues for cloud based EHR systems:

Identity Based Encryption (IBE):

IBE is a public key encryption schema. Where the public key consists of some information about the key holder like email address. The admin / key-authority issues a private/secret key which is tied to the pubic key. The owner of the public key can only decrypt then encrypted message.

Role Based Access Control (RBAC):

In this approach the access is granted to a specific role rather than individuals. Any individual who is assigned this role will automatically inherit the privileges assigned the role [39].

Attribute Based Access Control (ABAC):

It is a relatively newer and simpler to implement than RBAC. In this paradigm if a user has a set of attributes which satisfy the object they want to access, then they can retrieve that object [39].

Attribute Based Encryption (ABE):

ABE is an encryption schema which can perform a fine-grained access control with encryption where a user with certain attributes can read the data or parts of the data which the attributes grant access to [16] . When compared with other schemas ABE has a complex access control and decryption process. The public is generally hosted in the cloud which can be accessed by the users in the cloud to create their private keys. Functioning of ABE based eHealth systems are discussed in Page 6. Using attributes, we can make use of both IBE and RBAC algorithms using ABE. This motivated me to pursue a study based around ABE schemas.

Author [16] states that ABE is suitable for EHR's as there are many users in the cloud, and multiple users can access EHR with their keys and access the part of the EHR which they have access to. Several research papers are published on ABE-based cloud schemas for EHR. Authors [17] proposed an ABE model which can enforce fine-graded access control in EHR outsourced into the cloud. Authors [15], [18]–[20] based their solutions using Cipher Policy – Attribute-Based Encryption (CP-ABE) schema.

Authors [21], [22] base their models on Key Policy-Attribute Based Encryption (KP-ABE) schemas. There are other cryptographic schemas other than ABE which is also being used by researchers, which is an Identity-Based Encryption (IBE) Schema. Authors[23] applied this IBE Schemas so as to reduce the hassle of key-management which was proposed in various ABE schemas. There are few types of research [16], [24] which make use of both of the ABE and IBE schemas.

Generally, e-health applications are web applications with its backend supported by database which is used to store the EHR's. In a single hospital system consider that, there are multiple users/patient and doctors who make use of this application. When a user/patient uploads his EHR into the cloud, he can select the doctors treat the patient for his illness to make changes to his EHR, for this he sets certain attributes to parts of his EHR so that the doctor can access this part of the file rather than complete EHR. This encryption process can be assisted using ABE. Server shares the EHR over the nodes along with its access policies [16]. When the doctor tries to access, it would throw up an error saying he can't access the part of the file as your key does not have attributes to access it. The same can be understood by the Figure 2-3. Key sharing process in ABE based EHR system is depicted in Figure 2-3. Server stores the keys and has the same capabilities of traditional EHR systems [16] which is depicted in Figure 2-1.

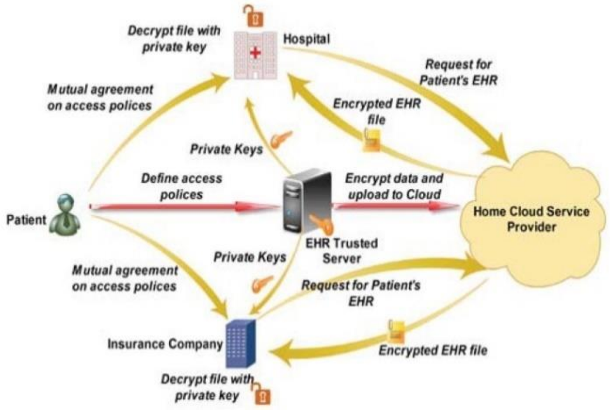

Decrypt file with private key
Hospital
Mutual agreement on access polices
Request for Patient's EHR
Encrypted EHR file
Private Keys
Define access polices
Encrypt data and upload to Cloud
Patient
Home Cloud Service Provider
EHR Trusted Server
Private Keys
Mutual agreement on access polices
Request for Patient's EHR
Insurance Company
Encrypted EHR file
Decrypt file with private key

Figure 2-2 Key Sharing in ABE systems [18].

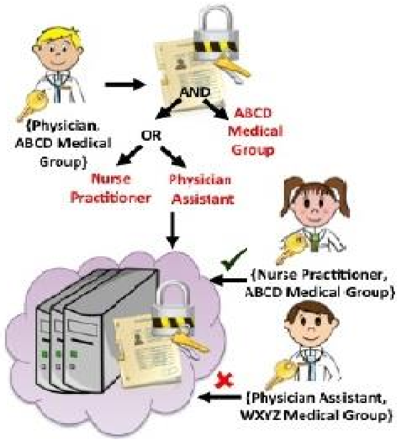

AND
{Physician, ABCD Medical Group}
OR
ABCD Medical Group
Nurse Practitioner
Physician Assistant
{Nurse Practitioner, ABCD Medical Group}
{Physician Assistant, WXYZ Medical Group}

Figure 2-3 ABE based ehealth system [19].

Based on the findings in the research field, this thesis is motivated to compare the performance of the ABE Schemas, CP-ABE and KP-ABE, since most of the researchers are trying to make use of the ABE schemas or attempting to improve the schemas. There exists little research which compares performance of both ABE Schemas.

3 METHODOLOGY

Descriptions of the Methodologies followed in this thesis are being presented. For Evidence-Based Software Engineering Studies (EBSE) the following guidelines were adopted [25]:

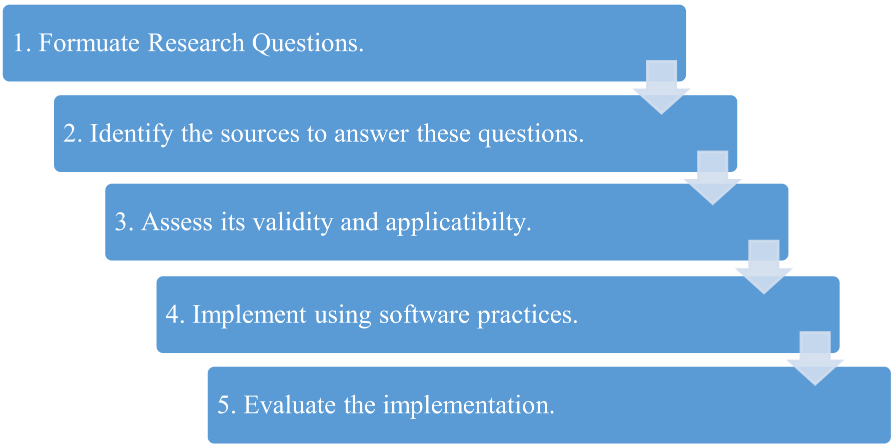

1. Formuate Research Questions.
2. Identify the sources to answer these questions.
3. Assess its validity and applicatibilty.
4. Implement using software practices.
5. Evaluate the implementation.

Fig 1. Guidelines for EBSE.

Steps 2 & 3: Literature Review was done, as described in Section 3.2,
Step 4: Experimentation was done, as described in Section 3.3,
Step 5: Results and Analysis is performed, as described in Section 4.

3.1 Literature Review

Literature review was conducted by the guidelines set by [26] and is the process of identifying and interpreting all the relevant research studies to a research question or a broad research field or a field of interest. Literature review has been conducted to learn about the following:

1. Existing Research on Utilization of Cloud Computing in Healthcare Industry,
2. Trying to identify the various Privacy Issues.
3. Learning on how the Researchers and Practitioners are trying to solve these Privacy issues.

Mapping Study has been performed as it helps to identify and categorize the existing literature among the broad research topics into a smaller, yet specific, Categories [26].

3.1.1 Mapping study

For mapping study IEEE Explore was used. Initially to find the privacy issues regarding the implementation of Cloud Computing in healthcare industry, *(((EHR in Cloud) AND Privacy) AND Security)* search string was used. Year range was set from 2010 to 2015. A total of 25 research articles were identified out of which 14 research articles were chosen based on the inclusion and exclusion criteria found in section 3.2.1.1 and 3.2.1.2. Search string was modified to *((Attribute Based Encryption) AND health care systems)* to learn about the existing ABE solution which was used my most of the 14 research articles obtained from the above string to solve the privacy issue regarding implementation of Cloud in healthcare industry. It yielded 18 research articles, after filtering the papers to primarily focus ABE in Cloud Computing the papers were reduced to 6.

3.1.1.1 Inclusion and Exclusion Criteria

Inclusion Criteria

1. Studies based on EHR / PHR.
2. Privacy and security issues were addressed in the proposed solution.
3. Papers published between the years 2010 to 2015.
4. Articles described/published in English.

Exclusion Criteria

1. Studies which are not based on EHR / PHR.
2. Studies which did not use ABE in their solutions.

3.1.2 Snowballing

Snowballing is a technique where additional resources can be identified by going through references and citations of the selected research articles [26]. Using this methods types of ABE algorithms were understood and CP-ABE algorithm used in the experimentation was discovered. Start set for the process were derived from the mapping study. During this process, few papers which were published in 2006 and 2007 were also considered, as they were directly related to the algorithms which were found during this process.

3.2 Experimentation

Experimentation for the comparison is carried out in two stages,

1. Experiment in Local Machine and
2. Experiment in Amazon Cloud (AWS).

Independent and Dependent Variables

Variable Algorithm	Independent Variable	Dependent Variable
KP-ABE	Dataset, keys	Performance Metrics
CP-ABE	Dataset, keys	Performance Metrics

Table 3-1 Independent and Dependent variables

3.2.1 System configuration

Local Machine:

Configuration of the local system was setup as a Virtual machine in VirtualBox application for experimentation. In this application, the configuration was set to match the configuration of AWS Instance. Configuration of the local virtual machine is provided in Table 3-2.

System Configuration	***Environment Value***
Operating system	Ubuntu 16.04 LTS x64
No. of CPU	1 CPU
Processor Speed	2.4 GHz
RAM	1 GB
Programming Language	C

Table 3-2 Local System Configuration

AWS Instance:

Configuration of the AWS Instance used:

System Configuration	***Environment Value***
Instance Type	General Purpose Free-Tier T2. micro
No. of CPU	1 CPU
Operating system	ubuntu-xenial-16.04-amd64-server-20161020 (ami-40d28157)
Processor Speed	2.4 GHz
RAM	1 GB
Programming Language	C

Table 3-3 AWS Instance Configuration

t2. micro general purpose instance was used for the experimentation. The t2 instance is a burstable performance instance where the which provide a baseline CPU performance and the ability to increase this baseline CPU performance occasionally when required [27]. The use case of the t2 instances include web applications, code repositories, building servers, etc. [27]. In this study, we are going to compare CP-ABE and KP-ABE using their code repositories, hence t2. micro instance was chosen. Also, the configuration of t2. micro instance is closer to "i5" PC which is widely used by the computing community.

3.2.2 Setting up the Experimentation Environment

For ease of understanding divided this section in the following manner:

- Section 3.2.2.1: Procedure to set up the testing environment for CP- ABE Toolkit,
- Section 3.2.2.2: Procedure to set up testing environment for KP-ABE Toolkit,
- Section 3.2.2.3: Procedure to setup the AWS Instance.

3.2.2.1 CP-ABE toolkit

Following steps were followed to set up the CP-ABE toolkit:

- CP-ABE toolkit has been split into two tar packages libsware (consists of crypto functions) and **cpabe** (user interface). Library should be installed first.
- Libsware uses PBC library for its library algebraic functions, it should be installed first.
- Now, we need to build the libsware directory using configure and make commands.

- If the configuration fails and asks for further dependencies, like M4, etc., then they should also be installed so that it can successfully configure.
- Those dependencies can easily be installed using synaptic package manager.
- Once libsware is successfully configured and installed then we need to unpack and configure cpabe.tar, using the same commands.
- Repeat the same steps to successfully configure and install cpabe.tar. If there are any problems during the installation, in the file policy_lang. y go to line 64 add a ";" at the end of the following line : final_policy = $1. Also, check the makefile go to line 19, in LDFLAGS add "-lgmp" after "-lcrypto".

3.2.2.2 KP-ABE toolkit

Following steps were followed to set up the KP-ABE toolkit:

- Like cpabe, kpabe toolkit is also split into two packages **libcelia.tar** and **kpabe-master.tar**.
- Primarily, we need to install the "autoconf" package to configure this toolkit.
- After getting the autoconf, unpack libcelia.tar and in a terminal, then configure and install.
- It may ask to install dependencies like python-3, etc. find the required packages using synaptic package manager and install them.
- After the successful installation of libcelia, use the same steps to compete the configuration and installation of kpabe.tar.
- If there are any problems in installation, check the makefile. Go to line 19, in LDFLAGS add "-lgmp" after "-lcrypto".

3.2.2.3 Setting up AWS Instance

Following steps were followed to setup the AWS Instance in AWS Console:

- Go to the EC2 service page, to create a new instance press Launch Instance button.
- It asks you to choose an Amazon Machine Image (AMI), go to community AMI and search for "ami-40d28157" with root device type: EBS. In this type of machine, we can add additional disk space if needed, install additional software's and have root access.
- In the Security Groups tab, find Inbound Rules and add the following rule:

Type	Protocol	Port Range	Source	
Custom TCP/IP Rule	TCP	5901	custom	0.0.0.0/0

Table 3-4 AWS Instance Security Group Inbound Rules

- Finish the setup and download the ".pem" file to access your ec2 instance. Now download puttykeygen and putty sshclient. Convert the pem file into ppk file using puttykeygen.
- Now connect using the information given at the AWS console.

A detailed description for setting up of GUI for AWS instance is described in Appendix 2. After setting up GUI for AWS instance then, follow the steps in Sections 3.3.2.1 and 3.3.2.2 to setup both the toolkits in the EC2 instance.

4 RESULTS AND ANALYSIS

This chapter is presented in the following manner, Section 4.1: Literature Review, in this section presented synthesis of the research articles gathered using mapping study and snowballing. This section is further sub divided into Section 4.1.1, here we discuss the flavors of ABE schemas with examples. In sub section 4.1.2, we discuss the privacy issues in cloud computing which were identified from literature review and sub section 4.1.3, we discuss the metrics which were selected to carry on experimentation of ABE algorithms.

In Section 4.2: Experimentation, we discuss how the experimentation is performed on algorithms. This section is further divided into two sections, Section 4.2.1, here results observed in local environment are discussed and Section 4.2.2 here results observed in cloud environment are discussed.

In Section 4.3: Analysis, we discuss t-Test: Paired Two Sample for Means briefly and formulate hypothesis to test the significance of the observed results. This section is further sub divided into two sections, Section 4.3.1, here we compare results observed in local environment in all the phases of the experiment and Section 4.3.2 here results observed in cloud environment in all the phases of the experiment.

4.1 Literature Review

This Section contains the data which was used to identify state-of-the-art Cloud EHR Systems and the issues faced. Synthesis of the research articles found using the first search string mentioned in Section 3.2, are as follows:

Authors Coats et al.,[28] propose, a trust based framework to integrate EHR of healthcare organizations to cloud with identity validations for compliance with security and privacy guidelines [28]. Also, they define 3 levels of Level of Access (LOA) to data in the cloud, level 1 being lowest and level 3 being the highest level of access. All these levels have authentication bits form 14, 20 and 64 bits respectively [28].

Authors Yu et al.,[7] propose, a watermarking method in cloud computing to reduce the risk of insider disclosures [7]. It was designed and implemented using MapReduce. The authors insert a watermarking process into software layer of the cloud. Whenever the EHR is accessed at different nodes there is a small change to the watermark. This change can be identified and traced down to the nodes where the EHR was accessed [7].

Authors Ramasamy et al.,[29] propose, to split the incoming EHR data stream into frames, where each frame would consists a part of the medical record with a bookmark for identification [29]. They made use of renowned "*DGIM (Datar-Gionis-Indyk-Motwani) algorithm which proves to be a promising technique for the transfer of healthcare data from the cloud to the offline storage unit without considering previous inputs*" [29]. Using this technique, EHR file is broken down into pieces to from chunks of data called buckets. The current bucket can be downloaded using its bookmark. This bucket of data can now be used offline; the remaining buckets of data would be downloaded using the same mechanism. Authors stress the fact that using this mechanism the transfer and sharing of EHR's would be efficient and less tedious [29].

Authors Huang et al., [16] propose a secure and scalable framework for EHR data sharing which combines Identity-based Encryption (trusted private key generator) and Attribute-based Encryption (Both KP-ABE and CP-ABE.) together to enforce access control policies[16].

The framework workflow is described below:

- The authors assume there is a secure network established between EHR owner and the trusted server. EHR owner sends his EHR to the trusted server over the secure network, and different domains across the network are allowed to read different parts of the EHR if they satisfy the policy or roles [16].
- Trusted server generates key information on access policies which are defined by the owner [16].
- The server performs KP-ABE encryption on the EHR with the public key over the set of attributes defined by the owner. After this process, EHR is then uploaded to the home Cloud [16].
- For each domain server in the network the server generates a decryption key and shares it across the network using IBE to ensure the domains which should receive the key would get it [16].
- When a domain server receives the EHR and the key, it performs CP-ABE decryption on the EHR [16].
- Each domain server generates a decryption key for the EHR and shares it with its entities [16].

Authors Melo et al., [6] propose, an architecture which makes use of identity federation for secure storage and sharing of PHR (Personal Health Record) and EHR in the cloud [6]. Author's solution is composed of the following parts: "*Attribute Authorities (AA), Identity Providers (IdP), cloud service providers (SP), and a set of users. The users have file owner or collaborator (with whom the file is shared) roles*" [6].

Authors Chen et al., [30] addresses security and privacy issues in healthcare cloud by framework CPRBAC (Cloud-based Privacy-aware Role Based Access Control) model. The model comprises of controllability and traceability of data, along with authorized access to the system resources [30].

Authors Balakrishnan et al., [8] use ABE, a technique where patient attributes are unique to each patient [8]. Secret key is generated using multiple attributes from patient's medical records [8]. The attributes which were used to generate the secret key will be changed dynamically to enhance the security [8]. Authors develop a web application where patient's records are entered and are encrypted using ABE and deployed in a cloud. They claim that the risks such as confidentiality of data, scalability and efficiency were managed [8].

Authors Abdulatif et al., [31] improved the access control for the cloud-based database by restricting parameters for each participant by using distinct encrypted parameters [31]. Also, they implement an existing secure index search algorithm and improve its efficiency of information control and its flow through the cloud-based EHR system [31].

Authors Alshehri et al., [18] propose, using CP-ABE to encrypt a EHR using healthcare provider's attributes [18]. In this method the healthcare provider's share a public key for encryption and every individual healthcare provider have their own distinctive private key [18]. This secret key can decrypt a cipher text if the attribute policy of that key satisfies the access policy of the cipher text [18].

Authors Wu et al., [3] focus on the access control issues of EHR in cloud. The propose a systematic access gain mechanism which supports selective sharing of EHR from various healthcare provider's [3]. In their approach they accommodate privacy concerns of the patient's healthcare records by enforcing access control policies specified by the patients [3].

Authors Zhang et al., [9] describe a EHR security reference model for mitigating security issues in healthcare clouds. Authors model focuses on the following three components for securing EHR in cloud [9]:

- EHR secure collection and Integration
- EHR secure storage and access management
- EHR secure usage in a model

Authors Sicuranza et al., [32] present an semantic access control model designed for fine-graded and flexible access policies in Heath Information Systems (HIS). This model enables easy management for dynamic access policies by fulfilling the identified requirements, which allows for the construction of novel access model for EHR [32]. This novel model makes use of set of security policies to grant or deny access request to the resource [32].

Authors Yu et al., [33] discuss the design of security oriented design (SOD) framework. This frame work is indented to provide a development environment template for strengthening development tasks of eHealth systems [33]. The proposed system is three-tiered framework:

- Client Tier (web apps and mobile apps)
- Middle Tier (web servers)
- Data Tier (DB and DB servers)

The webserver and the DB server are deployed in the cloud providing integration of web based and mobile based clients [33].

Authors Piliouras et al., [34] review the definition of trust and introduce its measure. The measure was used to rank cloud-based EHR applications, the measure uses a certainty factor to evaluate EHR [34]. The model is based on a mathematical equation where the variables in their equation are decisive factors in their multiple-criteria decision model for trustworthiness [34].

Following is the synthesis of research articles form the second search string:

Authors Deepali et al., [21] make use of RSA, DES and AES for encryption using KP-ABE schema in their framework for secure sharing of EHR in cloud. Their model also enables dynamic change in access policies which supports efficient on-demand attribute revocation in emergency situations [21].

Authors Yan et al., [35] make use of both CP-ABE and KP-ABE schemas in their model. The keys are generated in KP-ABE using AES, which are stored in the Keygen package. Patients can hide their data using an attribute, only having access to that attribute other users can have access to that information [35].

Authors Zhou et al., [36] propose a system where, two or more physicians are taking care of a patient (primary and secondary) using their proposed system attributes, primary physician has full access to patient information and secondary physician has access to details such as name, etc. only. This creates a user access control over the data [36].

Authors Li et al., [37] propose MA-ABE (Multiple Authority) to define attributes. These attributes are used during the key generation to differentiate type of users. It is a patient-centric concept where the patients will have full control over their data [37].

Authors Guo et al., [38] propose a complex four level privacy framework, each level for a type of user to maintain privacy of data [38].

Synthesis of papers that were found during snowballing, are as follows:

Authors Bethencourt et al., [15] CP-ABE algorithm is being used for experimentation. Performance metrics which were found in paper are: Execution times and CPU utilization percentage.

Authors' Y. Zheng [22] KP-ABE algorithm is being used for experimentation. Authors algorithm is a modified version of CP-ABE. Performance metrics which were found in paper are: Execution times and CPU utilization percentage.

Consolidation of the data had helped me to answer the research questions RQ1, that were formulated to find state-of-the art ABE Schemas in Cloud based EHR Systems and how the researchers dealt with the privacy issues in proposing a Solution. With the above data, RQ2 has been answered and the Performance Metrics for the experimentation were chosen which are: CPU usage and Time take to encrypt and decrypt after going through the core papers whose algorithms are available for experimentation.

4.1.1 Attribute Based Encryption Algorithms

ABE were introduced in [16], for Cloud based EHR systems, where the authors introduced the idea of KPABE system. Authors [17] introduced the idea of CP-ABE systems. Following are the different flavors of ABE:

KP-ABE

In this algorithm, secret keys are generated based on an access tree which defines the scope of a user and data is encrypted over a set of attributes. An example code has been provided in Table 4-1 on the usage of KP-ABE algorithm using Linux terminal.

```
$ kpabe-setup medication_profile surgical_profile \
>'auth_level = '
$ kpabe-enc pub_key Test.pdf medication_profile \
>, 'auth_level = 3'
$ kpabe-keygen -o butch_key pub_key master_key \
> (medication_profile, surgical_profile)'
$ kpabe-keygen -o tom_key pub_key master_key cardiac_profile
$ kpabe-dec pub_key butch_key Test.pdf.kpabe
```

Table 4-1 Example on how to use KP-ABE algorithm

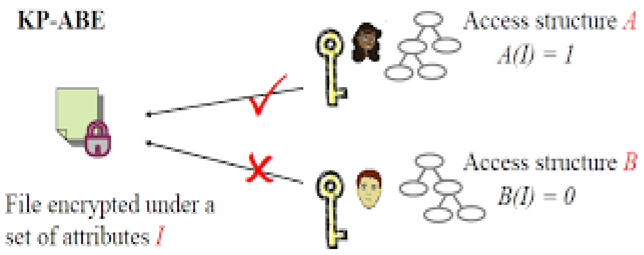


Figure 4-1 Working of KP-ABE algorithm [28]

Example: Consider two users Tina and Butch, lets encrypt a message using a set of attributes which are medication_profile and surgical_profile. The access structure consists of medication_profile and surgical_profile. Tina's key has the attributes, medication_profile and surgical_profile which matches the access structure hence she can decrypt the message. While Butch's key has the attribute cardiac_profile, which doesn't match the access structure so he can't decrypt the message.

Key idea here is that the key is associated with the policy using an access structure.

CP-ABE

This algorithm uses access trees to encrypt a message and user's keys are generated over a set of attributes. CP-ABE reverses the role of key derivation and encryption compared to KP-ABE.

A user key will be associated with a set of attributes. We encrypt a message with an access structure over a set of attributes. A user may be able to decrypt the message if the attributes of the user key pass through the access structure of the encrypted message [15]. An example code has been provided in Table 4-2 on the usage of CP-ABE algorithm using Linux terminal.

```
$ cpabe-keygen -o butch_key pub_key master_key \
   medical_staff physio_team
$ cpabe-keygen -o butch_key pub_key master_key medical_staff neuro_team
$ cpabe-enc pub_key test.pdf
   (sysadmin and (hire_date < 20110111 or admin_team)) or
   (medical_staff and physio_team)
$ cpabe-dec pub_key butch_key Test.pdf. cpabe
```

Table 4-2 Example on how to use CP-ABE algorithm

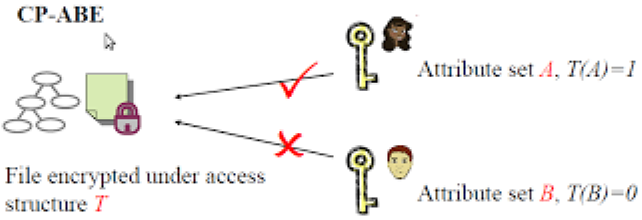


Figure 4-2 Working of CP-ABE algorithm [39]

Example: Consider two users Tina and Butch, lets encrypt a message using a set of an access structure:(sysadmin and (hire_date < 201101111 or admin_team)) or (medical_staff and physio_team). Tina's key has the attributes, medical_staff and physio_team which matches the access structure hence she can decrypt the message. While Butch's key has the attribute neuro_team, which doesn't match the access structure so he can't decrypt the message

In general CP-ABE is similar to RBAC and KP-ABE is closer to ABAC [22]. In both algorithms then encryptions are based on bilinear maps. *"Generally, bilinear maps associates pairs of elements from two algebra group to yield an element of a third algebra group that is linear in each of its arguments"* [22]. The encryption algorithm follows the "*Shamir t-out-of-n threshold scheme"* [22].

4.1.2 Privacy Issues in Cloud

Following are the privacy issues which were identified:

- *"The user may not have the kind of control over his/her data or the performance of his/her applications that he/she may need, or the ability to audit or change the processes and policies under which he/she must work."* [40]
- Data stored by one user may be manipulated by an auditor or some other person using the same hardware [40].
- As the Cloud needs to be accessed remotely, the connection may not be secure every time [40].
- User may not have control over his data or sometimes may lose data as they are locked into proprietary format suggested by the service provider or data auditor [40].
- Different laws for Data Privacy and Security set by different countries for their specific needs [40].

4.1.3 Metrics:

- User Time: Time taken by the CPU on the processor running the code, measured in milliseconds (ms).
- System Time: Time taken by the CPU running the code in OS Kernel, measured in milliseconds (ms).
- CPU Utility: The percentage of CPU which was assigned for the job, measured in percentage.
- Reads: Number of (No. of) read operations performed on the disk.
- Writes: Number of (No. of) write operations performed on the disk.

Metrics were measured in the terminal by adding this prefix to the bash commands to generate key, encryption and decryption of the file:

```
$   /usr/bin/time -v ____________________
```

4.2 Experimentation

In this section, we discuss different phases of the experiment with results observed in both the environments. Following are the three phases of experiment:

1. Key Generation: In this phase, user's private keys with a set of attributes are generated using public key, and we record the observations against metrics found in RQ2.
2. Encryption: In this phase, encryption of the selected data set is performed. In this process, we define an access structure for the encryption of the file with a set of attributes. and we record the observations against metrics found in RQ2.
3. Decryption: In this phase, we try to decrypt the encrypted file with the user's private key. If the attributes of user's key match the access structure of the encrypted file, only then we decrypt the file. Also, like in the above phases we record the observations against metrics found in RQ2.

All the above phases for experimenting in both the algorithms along with the command line arguments have been discussed in Appendix 3.

Experiments were performed in the following manner in both the environments:

1. Key Generation was performed ten times in a row,
2. Encryption on the dataset for one time,
3. Decryption on the dataset was done once,
4. Repeated steps 2 and 3 for nine times.

This Section is sub-divided in the following manner:
Section 4.2.1 Results of experimentation in Local Machine,
Section 4.2.2 Results of experimentation in AWS.

4.2.1 Results observed in local machine

In this section the results recorded while performing the experiment in local environment in all phases, Mean and Standard Deviation for 10 iterations for each metric was calculated and are presented in table 4-3.

Phase	*Algorithm*	*Mean and Standard Deviation*	**Metrics**				
			User Time (ms)	*System Time (ms)*	*CPU Utility (%)*	*No. of Reads on disk*	*No. of Writes on disk*
Key Generation	*CP-ABE*	Mean	1.36	5.10 x10^{-2}	92.70	144	56
		Std. Dev	8.01 x10^{-2}	5.39 x10^{-3}	1.35	0	0
	KP-ABE	Mean	1.53 x10^{-6}	2.13 x10^{-7}	92.80	0	8
		Std. Dev	1.47 x10^{-7}	3.58 x10^{-8}	0.98	0	0
Encryption	*CP-ABE*	Mean	4.13	1.12	74.40	62	10476
		Std. Dev	1.80 x10^{-1}	1.20 x10^{-1}	2.37	0	0
	KP-ABE	Mean	3.25	1.22	74.20	546	10476
		Std. Dev	1.94 x10^{-1}	2.12 x10^{-1}	2.71	0	0
Decryption	*CP-ABE*	Mean	5.92	1.17	74.80	84	10476
		Std. Dev	3.99 x10^{-1}	9.38 x10^{-2}	1.94	0	0
	KP-ABE	Mean	5.40	1.83	77.90	315	10476
		Std. Dev	1.10 x10^{-1}	4.20 x10^{-1}	5.15	0	0

Table 4-3 Results obtained for local machine.

4.2.2 Results observed in AWS Instance

In this section the results recorded while performing the experiment in local environment in all phases, Mean and Standard Deviation for 10 iterations for each metric was calculated and are presented in table 4-3.

Phase	*Algorithm*	*Mean and Standard Deviation*	**Metrics**				
			User Time (ms)	*System Time (ms)*	*CPU Utility (%)*	*No. of Reads on disk*	*No. of Writes on disk*
Key Generation	*CP-ABE*	Mean	1.43	0.05	95.80	0	144
		Std. Dev	3.74 x10^{-2}	1.18 x10^{-2}	2.60	0	0
	KP-ABE	Mean	1.01 x10^{-4}	1.62 x10^{-7}	80.40	0	16
		Std. Dev	3.00 x10^{-4}	6.33 x10^{-8}	4.22	0	0
Encryption	*CP-ABE*	Mean	7.07	1.14	89.90	6	10476
		Std. Dev	1.57 x10^{-1}	4.90 x10^{-2}	3.96	0	0
	KP-ABE	Mean	2.10	0.57	84.90	24	10476
		Std. Dev	1.82 x10^{-1}	9.92 x10^{-2}	4.53	0	0
Decryption	*CP-ABE*	Mean	12.16	2.05	91.80	112	10476
		Std. Dev	3.23 x10^{-1}	1.34 x10^{-1}	2.96	0	0
	KP-ABE	Mean	6.05	6.86 x10^{-1}	89.50	8	10476
		Std. Dev	1.44 x10^{-1}	6.61 x10^{-2}	2.97	0	0

Table 4-4 Results obtained for AWS Instance.

4.3 Analysis

For statistical significance of the results, paired sample *t*-test was performed. A paired is used to determine whether the mean differences of two paired samples defers from 0 [41]. Paired sample *t*-test was selected as there is one independent variable with two groups of matched dependent variable groups, also calculated the means for every step of execution which can be used for paired sample *t*-test. And also looking at the case example in [42], paired sample *t*-test was used for comparing the performance of two programs, which is similar to this study where we compare two code repositories of CP-ABE and KP-ABE algorithms against each other.

Null Hypothesis (H_0): The performance of both CP-ABE and KP-ABE is same.

Two-tailed hypothesis are considered as in every case; the difference of means is not equal to zero ($\mu_d \neq 0$). Here $\alpha = 0.05$ is considered, which means there would be 95% confidence level of difference in performance. An upper tailed and lower tailed alternative hypothesis are used in the thesis to increase the power of the statistical test [40]. The null hypothesis for these two, upper tailed and lower tailed alternative hypothesis remains the same. In some cases, where the value of *t* statistic is negative, then its absolute value is considered.

Alternative Hypothesis (H1: $\mu_d > 0$): The performance of KP-ABE is better than CP-ABE.

Alternative Hypothesis (H1: $\mu_d < 0$): The performance of CP-ABE is better than KP-ABE.

Performance in each stage will be compared in the following manner:

4.3.1 Comparison of results in Local Machine.
4.3.2 Comparison of results in AWS Instance.

4.3.1 Comparison of results in Local System

In this section, the results are compared in table format.

Key Generation

t-Test: Paired Two Sample for Means						
Metrics	**User Time (ms)**		**System Time (ms)**		**CPU %**	
Algorithms	*CP-ABE*	*KP-ABE*	*CP-ABE*	*KP-ABE*	*CP-ABE*	*KP-ABE*
Mean	1.36	1.53 x10^{-6}	0.05	2.13 x10^{-7}	92.70	92.80
Variance	7.13 x10^{-3}	2.33 x10^{-12}	3.22 x10^{-5}	1.43 x10^{-15}	2.01	1.07
Observations	10	10	10	10	10	10
μ_d	1.36		0.05		-0.10	
df	9		9		9	
t Stat	50.92		28.41		-0.32	
P(T<=t) two-tail	2.18 x10^{-12}		4.03 x10^{-10}		7.58 x10^{-1}	
t Critical two-tail for df=9	2.26		2.26		2.26	

Table 4-5 t-Test Key-Generation in Local system

From the table, above we can see that for user time and system time, upper-tailed alternative hypothesis is accepted, which means the performance of KP-ABE is significantly better than CP-ABE. In case of CPU %, failed to reject null hypothesis as there is insufficient evidence ($p>\alpha$) to say null hypothesis is false.

Encryption

t-Test: Paired Two Sample for Means						
Metrics	**User Time (ms)**		**System Time (ms)**		**CPU %**	
Algorithms	*CP-ABE*	*KP-ABE*	*CP-ABE*	*KP-ABE*	*CP-ABE*	*KP-ABE*
Mean	4.13	3.25	1.12	1.22	74.40	74.20
Variance	4.43 x10^{-3}	1.47 x10^{-2}	6.63 x10^{-3}	8.92 x10^{-2}	7.00	1.00
Observations	10	10	10	10	10	10
μ_d	0.88		-0.10		0.20	
df	9		9		9	
t Stat	17.85		-1.49		0.14	
P(T<=t) two-tail	2.46 x10^{-8}		1.71 x10^{-1}		8.89 x10^{-1}	
t Critical two-tail for df=9	2.26		2.26		2.26	

Table 4-6 t-Test Encryption in Local system

From the table, above we can see that for user time, upper-tailed alternative hypothesis is accepted, which means the performance of KP-ABE is significantly better than CP-ABE. In case of CPU % and system time, failed to reject null hypothesis as there is insufficient evidence (p>α) to say null hypothesis is false.

Decryption

t-Test: Paired Two Sample for Means						
Metrics	**User Time (ms)**		**System Time (ms)**		**CPU %**	
Algorithms	*CP-ABE*	*KP-ABE*	*CP-ABE*	*KP-ABE*	*CP-ABE*	*KP-ABE*
Mean	5.92	5.40	1.17	1.83	74.80	77.90
Variance	1.77 x10^{-1}	1.34 x10^{-2}	9.78 x10^{-3}	1.96 x10^{-1}	4.18	29.43
Observations	10	10	10	10	10	10
μ_d	0.51		-0.67		-3.10	
df	9		9		9	
t Stat	3.52		-5.06		-1.46	
P(T<=t) two-tail	6.47 x10^{-3}		6.79 x10^{-4}		1.78 x10^{-1}	
t Critical two-tail for df=9	2.26		2.26		2.26	

Table 4-7 t-Test Decryption in Local system

From the table, above we can see that for user time, upper-tailed alternative hypothesis is accepted, which means the performance of KP-ABE is significantly better than CP-ABE. In case of system time lower-tailed alternative hypothesis is accepted, which means performance of CP-ABE is significantly better than KP-ABE. In case of CPU %, failed to reject null hypothesis as there is insufficient evidence (p>α) to say null hypothesis is false.

Thus, for the over performance in all the three phases of the execution, based on statistical analysis it can be concluded that: For user time, upper-tailed alternative hypothesis is accepted, which means the performance of KP-ABE is significantly better than CP-ABE. In case of system time lower-tailed alternative hypothesis is accepted, which means performance of CP-ABE is significantly better than KP-ABE. In case of CPU %, failed to reject null hypothesis as there is insufficient evidence (p>α) to say null hypothesis is false.

4.3.2 Comparison of results in AWS Instance

Results were compared and tabulated, as follows:

Key Generation

t-Test: Paired Two Sample for Means						
Metric	**User Time (ms)**		**System Time (ms)**		**CPU %**	
Algorithm	*CP-ABE*	*KP-ABE*	*CP-ABE*	*KP-ABE*	*CP-ABE*	*KP-ABE*
Mean	1.43	1.01 x10^{-4}	0.05	1.62 x10^{-7}	95.8	80.4
Variance	1.56 x10^{-3}	9.99 x10^{-8}	1.56 x10^{-4}	4.45 x10^{-15}	7.51	19.82
Observations	10	10	10	10	10	10
μ_d	1.43		5.00 x10^{-2}		15.4	
df	9		9		9	
t Stat	114.90		12.68		8.88	
P(T<=t) two-tail	1.45 x10^{-15}		4.82 x10^{-7}		9.49 x10^{-6}	
t Critical two-tail for df=9	2.26		2.26		2.26	

Table 4-8 t-Test Key-Generation in AWS

From the table, above we can see that for user time, system time and CPU % upper-tailed alternative hypothesis is accepted, which means the performance of KP-ABE is significantly better than CP-ABE.

Encryption

t-Test: Paired Two Sample for Means						
Metrics	**User Time (ms)**		**System Time (ms)**		**CPU %**	
Algorithms	*CP-ABE*	*KP-ABE*	*CP-ABE*	*KP-ABE*	*CP-ABE*	*KP-ABE*
Mean	7.07	2.10	1.14	0.57	89.90	84.90
Variance	2.73 x10^{-2}	3.68 x10^{-2}	2.67 x10^{-3}	1.09 x10^{-2}	17.43	22.77
Observations	10	10	10	10	10	10
μ_d	4.97		0.57		5.00	
df	9		9		9	
t Stat	56.61		13.79		2.64	
P(T<=t) two-tail	8.43 x10^{-13}		2.33 x10^{-7}		2.68 x10^{-2}	
t Critical two-tail for df=9	2.26		2.26		2.26	

Table 4-9 t-Test Encryption in AWS

From the table, above we can see that for user time, system time and CPU % upper-tailed alternative hypothesis is accepted, which means the performance of KP-ABE is significantly better than CP-ABE.

Decryption

t-Test: Paired Two Sample for Means						
Metrics	**User Time (ms)**		**System Time (ms)**		**CPU %**	
Algorithms	*CP-ABE*	*KP-ABE*	*CP-ABE*	*KP-ABE*	*CP-ABE*	*KP-ABE*
Mean	12.16	6.05	2.05	0.69	91.80	89.50
Variance	1.16 x10^{-1}	2.32 x10^{-2}	1.98 x10^{-2}	4.85 x10^{-3}	9.73	9.83
Observations	10	10	10	10	10	10
μ_d	6.10		1.37		2.30	
df	9		9		9	
t Stat	59.80		23.80		1.94	
P(T<=t) two-tail	5.15 x10^{-13}		1.95 x10^{-9}		8.39 x10^{-2}	
t Critical two-tail for df=9	2.26		2.26		2.26	

Table 4-10 t-Test Decryption in AWS

From the table, above we can see that for user time, system time upper-tailed alternative hypothesis is accepted, which means the performance of KP-ABE is significantly better than CP-ABE. In case of CPU %, failed to reject null hypothesis as there is insufficient evidence ($p>\alpha$) to say null hypothesis is false.

From the above results, it is evident that the Performance of KP-ABE Algorithm is better in all stages of execution in Cloud Platform.

The following inferences can be drawn from the above comparisons:

- Key Generation is faster in KP-ABE than in CP-ABE.
- Encryption and Decryption Process in Local Machine was relatively better in CP-ABE.
- KP-ABE is the best Algorithm in AWS Instance.

Combination of these Two Algorithms would yield an Optimal Algorithm, in terms of Performance and Privacy. Key Policy – Attribute Based Encryption defines the Access Structure while creating keys and Cipher Policy – Attribute Based Encryption defines the Access Structure while Encrypting the file, this would add an additional Security / Privacy rule. As the attribute universe of the combined algorithm would be enriched and even sophisticated attribute strings can be formed, unlike in both the individual algorithms. This can be understood further by considering the Linux commands to perform experimentation in Appendix 2.

5 DISCUSSION

In this chapter, we analyze the answers to the research questions of this study in section 5.1, in section 5.2 we revisit the contributions made in this study, in section 5.3 we discuss threats to validity of this study and in section 5.4 we discuss limitations of the study.

5.1 Analyzing Research Questions

1. *What are the state-of-the-art ABE based Algorithms / Schema for EHR Systems in Cloud?*

   Literature review has been performed to find out existing algorithms which are being used or proposed by researchers to mitigate privacy issue for EHR storage in cloud. In section 2.2, a brief discussion is provided for all the existing algorithms. This study was based on ABE algorithms, as these algorithms are improvised version of IBE. In ABE, we encrypt a file with a set of attributes, one can only decrypt the file if his key matches the attributes which were used during encryption. Using ABE, we can create identities which can also be used as attributes for encryption. There are two flavors of ABE namely, Cipher Policy Attribute Based Encryption (CP-ABE) and Key Policy Attribute Based Encryption (KP-ABE). In CP-ABE, we encrypt a file using a set of attributes, a user can decrypt the file if his key's attributes match the encrypted files' attributes. In KP-ABE, it reverses the process of CP-ABE.

2. *What are the existing Performance Metrics to evaluate ABE Algorithms / Schemas?*

   Metrics were found while performing literature review, and are discussed in section 4.1.3. These metrics were directly taken from the research papers[15] and [22] whose version of CP-ABE and KP-ABE algorithms were used respectively, also these metrics were used in most of the research articles. System Time tells us how long the CPU took for execution. User Time tells us how long the Kernel took for execution. CPU utility tells us how much percentage of CPU was allotted for the job. No. of Reads and Writes tells us how many read and write operations were performed on the disk.

3. *How the selected ABE Algorithms / Schema compare based on metrics found in RQ2 in local and cloud environment?*

   Experiments were done 10 times each during key-generation, encryption and decryption phase for both the algorithms in local and cloud environment with identical configuration. From the results, it was evident that in local system, KP-ABE algorithm performed better against System Time metric, CP-ABE algorithm performed better against User Time metric and for CPU utilization we could not reject the null hypothesis as there was insufficient evidence. In cloud environment, it was evident that the performance of KP-ABE was better against system time and user time metrics, and like in local environment for CPU utilization we could not reject the null hypothesis as there was insufficient evidence. From the above observations, it was evident that, Key Generation is faster in KP-ABE, Encryption and Decryption Process in Local Machine was relatively better in CP-ABE, KP-ABE is the best Algorithm in AWS Instance.

5.2 Contributions

- Identified various Privacy issues in Cloud Computing, these were identified by performing literature review they are discussed in section 4.1.2.

- Identified various state-of-the art ABE Schemas in EHR scenario, these were identified by performing literature review along with other existing algorithms, they are discussed in section 4.1.1.

- Compared the Performance of state-of-the-art ABE Schemas, experiment was performed on CP-ABE and KP-ABE in local and cloud environment then the results were compared for significance using paired sample t-test.

5.3 Threats to validity

In this section, we discuss different threats to validity to this thesis:

Internal Validity

Internal validity is the extent upto which these threats are migrated as these threats tend to affect the relationship between independent variable and dependent variable [42].

- The performance metrics were chosen form the literature to make sure they would yield accurate results.
- Testing was performed multiple times first to get used to procedure before the actual experimentation was done.
- Experimentation was performed multiple times and the average value of the metrics were compared to each other.

External Validity

External validity is the extent upto which results of an experiment can be generalized [42].

- Experiment was conducted based on the available environments and dataset these results cannot be generalized to other environments.

5.4 Limitations

- Only one AWS Cloud instance node was used during the experimentation as the vncserver in the instance was configured for one connection only.
- Instead of a database a ".csv" file was used for testing as there were no live EHR databases which were available for experimentation.
- These algorithms could be executed in Linux terminal, and not in other Operating Systems.
- A smaller data set was used in the experiments as the instances did not have sufficient swap space for operations.
- Data sets of different sizes could not be used as there was insufficient memory space in the AWS instance.

6 CONCLUSION AND FUTURE WORK

This chapter describes conclusions in section 6.1 and future work in section 6.2.

6.1 Conclusions

Aim of this thesis is to Perform Quantitative Comparison of Attribute Based Encryption (ABE) Schema in a Cloud based EHR Scenario. State-of-the art ABE Schemas in Cloud based EHR Systems were identified by conducting a literature review. Snowballing was performed on those research articles to find more relevant literature, from which the performance metrics were derived for the comparison of the algorithms.

Experiments were conducted in both local machine and AWS Cloud to check for the performance of these algorithms over a dataset which was taken from an American healthcare blog. From the results, we can conclude that a combination of these two algorithms would yield a more optimal solution in terms of performance and privacy. As Key Policy – Attribute based encryption defines the access structure while creating keys and Cipher Policy – Attribute based encryption defines the access structure while encrypting the file. This would add an additional security / privacy rule which is not present in the two individual algorithms.

6.2 Future Work

Experiments in Cloud could be performed in different type of AWS instance like M3 or C4 instances as they are more powerful in Performance and Memory Capabilities. Different sizes of dataset could be used to perform the experiment and to evaluate the performance. Experiment in this paper was performed on only one node of AWS. Hence, it has the scope to Perform and Analyze in Two or more Nodes. Algorithms can be tested using a Live Database of a Hospital. Combination of both the KP-ABE and CP-ABE Algorithms can be Developed and Tested on a Live Database, and a Comparative Study can be done.

REFERENCES

[1] N. Pitropakis, N. Yfantopoulos, D. Geneiatakis, and C. Lambrinoudakis, “Towards an augmented authenticator in the Cloud,” in *2014 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, 2014, pp. 000296–000300.

[2] Z.-R. Li, E.-C. Chang, K.-H. Huang, and F. Lai, “A secure electronic medical record sharing mechanism in the cloud computing platform,” in *2011 IEEE 15th International Symposium on Consumer Electronics (ISCE)*, 2011, pp. 98–103.

[3] R. Wu, G.-J. Ahn, and H. Hu, “Secure Sharing of Electronic Health Records in Clouds,” in *Proceedings of the 8th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*, 2012, no. 1, pp. 711–718.

[4] M. Thomas, “Definition of EMR.” [Online]. Available: https://www.healthit.gov/providers-professionals/electronic-medical-records-emr.

[5] healthit.gov, “Definition of EHR.” [Online]. Available: https://www.healthit.gov/providers-professionals/faqs/what-electronic-health-record-ehr.

[6] L. de Melo Silva, R. Araujo, F. Leite da Silva, and E. Cerqueira, “A new architecture for secure storage and sharing of health records in the cloud using federated identity attributes,” in *2014 IEEE 16th International Conference on e-Health Networking, Applications and Services (Healthcom)*, 2014, pp. 194–199.

[7] Z. Yu, C. Thomborson, C. Wang, J. Wang, and R. Li, “A cloud-based watermarking method for health data security,” in *2012 International Conference on High Performance Computing & Simulation (HPCS)*, 2012, vol. 16, pp. 642–647.

[8] Preethi M. and R. Balakrishnan, “Cloud enabled patient-centric EHR management system,” in *2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies*, 2014, no. 978, pp. 1678–1680.

[9] R. Zhang and L. Liu, “Security Models and Requirements for Healthcare Application Clouds,” in *2010 IEEE 3rd International Conference on Cloud Computing*, 2010, pp. 268–275.

[10] H. J. Cheong, N. Y. Shin, and Y. B. Joeng, “Improving Korean Service Delivery System in Health Care: Focusing on National E-health System,” in *2009 International Conference on eHealth, Telemedicine, and Social Medicine*, 2009, pp. 263–268.

[11] G. Gillis, “Canada Health Infoway,” *Communication*, 2006. [Online]. Available: http://www.infoway-inforoute.ca.

[12] J. Dzenowagis and G. Kernen, “Global Vision, Local Insight: Report for the WSIS,” *World Heal. Organ.*, pp. 1–41, 2005.

[13] M. R. Patra, R. K. Das, and R. P. Padhy, “CRHIS,” in *Proceedings of the 6th International Conference on Theory and Practice of Electronic Governance - ICEGOV ’12*, 2012, p. 402.

[14] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based encryption for fine-grained access control of encrypted data,” in *Proceedings of the 13th ACM conference on Computer and communications security - CCS ’06*, 2006, p. 89.

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

[16] J. Huang, M. Sharaf, and C. T. Huang, “A hierarchical framework for secure and scalable EHR sharing and access control in multi-cloud,” *Proc. Int. Conf. Parallel Process. Work.*, pp. 279–287, 2012.

[17] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based Encryption for Fine-grained Access Control of Encrypted Data,” *Proc. 13th ACM Conf. Comput. Commun. Secur.*, pp. 89–98, 2006.

[18] S. Alshehri, S. P. Radziszowski, and R. K. Raj, “Secure Access for Healthcare Data in the Cloud Using Ciphertext-Policy Attribute-Based Encryption,” in *2012 IEEE 28th International Conference on Data Engineering Workshops*, 2012, pp. 143–146.

[19] M. Green, S. Hohenberger, and B. Waters, “Outsourcing the Decryption of ABE

Ciphertexts," *Proc. 20th USENIX Conf. Secur.*, pp. 34–34, 2011.

[20] M. Li, S. Yu, K. Ren, and W. Lou, "Securing Personal Health Records in Cloud Computing: Patient-Centric and Fine-Grained Data Access Control in Multi-owner Settings," in *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*, vol. 50 LNICST, 2010, pp. 89–106.

[21] D. A. Gondkar and V. S. Kadam, "Attribute based encryption for securing personal health record on cloud," in *2014 2nd International Conference on Devices, Circuits and Systems (ICDCS)*, 2014, pp. 1–5.

[22] Y. Zheng, "Privacy-Preserving Personal Health Record System Using Attribute-Based Encryption," *Master Thesis*, no. June, pp. 1–70, 2011.

[23] C.-H. Liu *et al.*, "Secure PHR Access Control Scheme for Healthcare Application Clouds," in *2013 42nd International Conference on Parallel Processing*, 2013, pp. 1067–1076.

[24] A. Lounis, A. Hadjidj, A. Bouabdallah, and Y. Challal, "Secure and Scalable Cloud-Based Architecture for e-Health Wireless Sensor Networks," in *2012 21st International Conference on Computer Communications and Networks (ICCCN)*, 2012, pp. 1–7.

[25] F. Q. B. da Silva, A. L. M. Santos, S. Soares, A. C. C. França, C. V. F. Monteiro, and F. F. Maciel, "Six years of systematic literature reviews in software engineering: An updated tertiary study," *Inf. Softw. Technol.*, vol. 53, no. 9, pp. 899–913, Sep. 2011.

[26] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering – A systematic literature review," *Inf. Softw. Technol.*, vol. 51, no. 1, pp. 7–15, Jan. 2009.

[27] E. C. Amazon, "Amazon Web Services EC2 instance types," 2014. [Online]. Available: http://aws.amazon.com/ec2/instance-types/. [Accessed: 10-Feb-2017].

[28] B. Coats and S. Acharya, "Bridging Electronic Health Record Access to the Cloud," in *2014 47th Hawaii International Conference on System Sciences*, 2014, pp. 2948–2957.

[29] P. Ramasamy, S. Venkateswaran, and S. Vidhusha, "An efficient transfer of EHRs on the cloud using Decaying Window principle," in *2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, 2015, pp. 1–7.

[30] L. Chen and D. B. Hoang, "Novel Data Protection Model in Healthcare Cloud," in *2011 IEEE International Conference on High Performance Computing and Communications*, 2011, pp. 550–555.

[31] A. Alabdulatif, I. Khalil, and V. Mai, "Protection of electronic health records (EHRs) in cloud," in *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2013, pp. 4191–4194.

[32] M. Sicuranza and M. Ciampi, "A Semantic Access Control for Easy Management of the Privacy for EHR Systems," in *2014 Ninth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, 2014, pp. 400–405.

[33] W. D. Yu, L. Davuluri, M. Radhakrishnan, and M. Runiassy, "A Security Oriented Design (SOD) Framework for eHealth Systems," in *2014 IEEE 38th International Computer Software and Applications Conference Workshops*, 2014, pp. 122–127.

[34] T. Piliouras *et al.*, "Trust in a cloud-based healthcare environment," in *2011 8th International Conference & Expo on Emerging Technologies for a Smarter World*, 2011, pp. 1–6.

[35] H. Yan, X. Li, and J. Li, "Secure Personal Health Record System with Attribute-Based Encryption in Cloud Computing," in *2014 Ninth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, 2014, pp. 329–332.

[36] J. Zhou, Z. Cao, X. Dong, and X. Lin, "TR-MABE: White-box traceable and revocable multi-authority attribute-based encryption and its applications to multi-level privacy-preserving e-healthcare cloud computing systems," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, 2015, vol. 26, pp. 2398–2406.

[37] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute-Based Encryption," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 1, pp. 131–143, Jan. 2013.

[38] L. Guo, C. Zhang, J. Sun, and Y. Fang, "PAAS: A Privacy-Preserving Attribute-Based

Authentication System for eHealth Networks," in *2012 IEEE 32nd International Conference on Distributed Computing Systems*, 2012, pp. 224–233.

[39] N. Mohamed, "Attribute based encryption (ABE) and its two flavors." [Online]. Available: http://mohamednabeel.blogspot.se/2012/03/aattribute-based-encryption-abe-and-its.html.

[40] J. Sen, "Security and Privacy Issues in Cloud Computing," *J. Netw. Comput. Appl.*, vol. 71, no. iv, pp. 11–29, Mar. 2013.

[41] "Paired Sample T-Test - Statistics Solutions," *Statistics Solutions*. [Online]. Available: http://www.statisticssolutions.com/manova-analysis-paired-sample-t-test/. [Accessed: 05-Feb-2017].

[42] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering*, vol. 58, no. 12. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

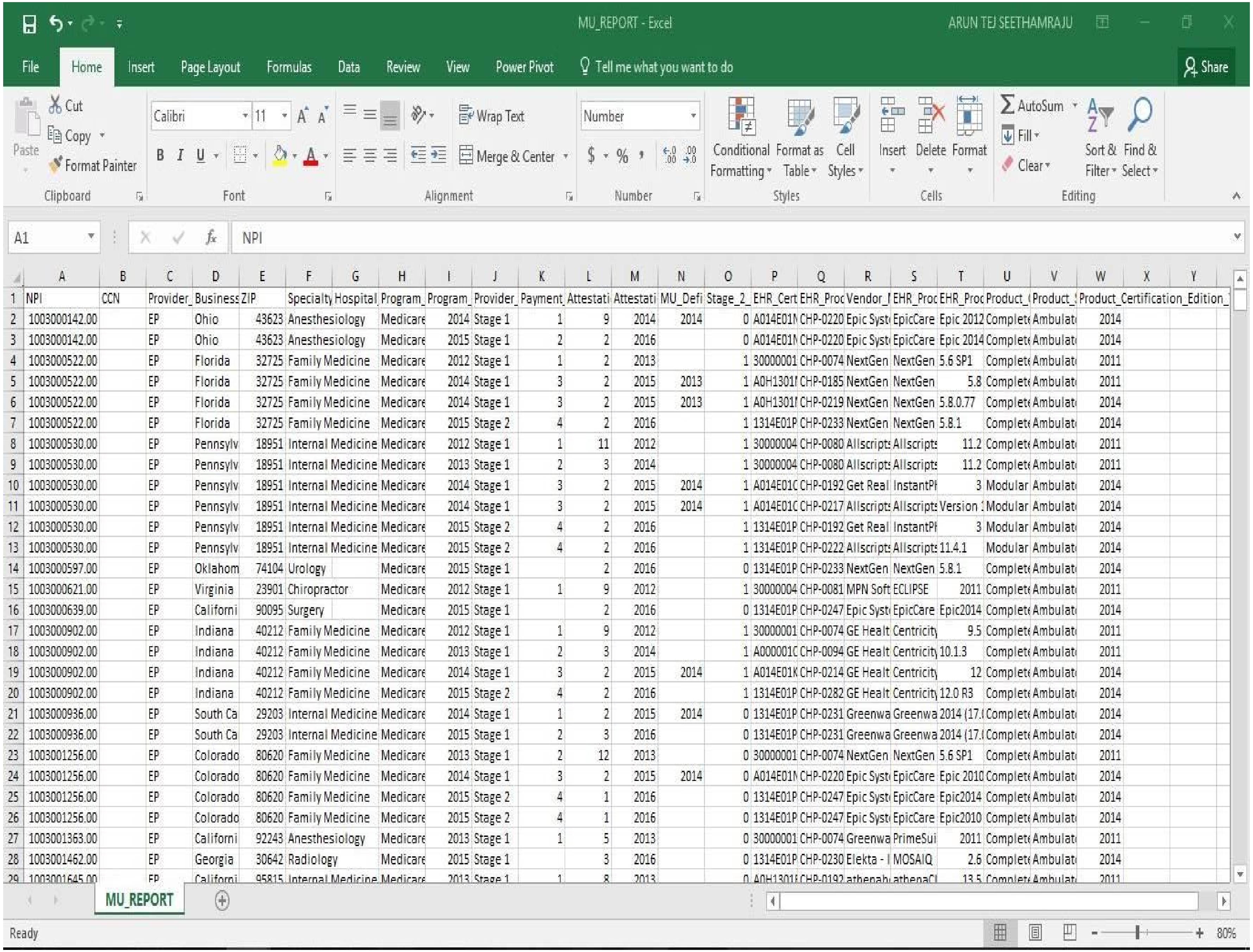
APPENDIX 1

A Sample Screen shot of the dataset with 29 records is provided in the page no 32. The complete data set consisted of more than 1 Million records, size 257 MB and has been downloaded from https://data.medicare.gov/. Adhering to the conditions and terms of use which provided at the time of download the exact URL of the file is not disclosed.

For the experiment, the file size was reduced to 10.2 MB containing the first 10476 records was used in AWS. Initially, for the first iteration of the document before the thesis defense then 267 MB file was used as the dataset for the Local System experiment.

After the change of system configuration is suggested by the examiner, the 10.2 MB data set was used for the both Local System and AWS experimentation.

Figure A1-0-1 Screen Shot of dataset



A1 NPI

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1	NPI	CCN	Provider_	Business	ZIP	Specialty	Hospital_	Program_	Program_	Provider_	Payment_	Attestati	Attestati	MU_Defi	Stage_2_	EHR_Cert	EHR_Proc	Vendor_I	EHR_Proc	EHR_Proc	Product_(	Product_!	Product_Certification_Edition_		
2	1003000142.00		EP	Ohio	43623	Anesthesiology		Medicare	2014	Stage 1	1	9	2014	2014	0	A014E01N	CHP-0220	Epic Syst	EpicCare	Epic 2012	Complete	Ambulat	2014		
3	1003000142.00		EP	Ohio	43623	Anesthesiology		Medicare	2015	Stage 1	2	2	2016		0	A014E01N	CHP-0220	Epic Syst	EpicCare	Epic 2014	Complete	Ambulat	2014		
4	1003000522.00		EP	Florida	32725	Family Medicine		Medicare	2012	Stage 1	1	2	2013		1	30000001	CHP-0074	NextGen	NextGen	5.6 SP1	Complete	Ambulat	2011		
5	1003000522.00		EP	Florida	32725	Family Medicine		Medicare	2014	Stage 1	3	2	2015	2013	1	A0H1301I	CHP-0185	NextGen	NextGen	5.8	Complete	Ambulat	2011		
6	1003000522.00		EP	Florida	32725	Family Medicine		Medicare	2014	Stage 1	3	2	2015	2013	1	A0H1301I	CHP-0219	NextGen	NextGen	5.8.0.77	Complete	Ambulat	2014		
7	1003000522.00		EP	Florida	32725	Family Medicine		Medicare	2015	Stage 2	4	2	2016		1	1314E01P	CHP-0233	NextGen	NextGen	5.8.1	Complete	Ambulat	2014		
8	1003000530.00		EP	Pennsylv	18951	Internal Medicine		Medicare	2012	Stage 1	1	11	2012		1	30000004	CHP-0080	Allscripts	Allscripts	11.2	Complete	Ambulat	2011		
9	1003000530.00		EP	Pennsylv	18951	Internal Medicine		Medicare	2013	Stage 1	2	3	2014		1	30000004	CHP-0080	Allscripts	Allscripts	11.2	Complete	Ambulat	2011		
10	1003000530.00		EP	Pennsylv	18951	Internal Medicine		Medicare	2014	Stage 1	3	2	2015	2014	1	A014E01C	CHP-0192	Get Real	InstantPl	3	Modular	Ambulat	2014		
11	1003000530.00		EP	Pennsylv	18951	Internal Medicine		Medicare	2014	Stage 1	3	2	2015	2014	1	A014E01C	CHP-0217	Allscripts	Allscripts	Version 1	Modular	Ambulat	2014		
12	1003000530.00		EP	Pennsylv	18951	Internal Medicine		Medicare	2015	Stage 2	4	2	2016		1	1314E01P	CHP-0192	Get Real	InstantPl	3	Modular	Ambulat	2014		
13	1003000530.00		EP	Pennsylv	18951	Internal Medicine		Medicare	2015	Stage 2	4	2	2016		1	1314E01P	CHP-0222	Allscripts	Allscripts	11.4.1	Modular	Ambulat	2014		
14	1003000597.00		EP	Oklahom	74104	Urology		Medicare	2015	Stage 1		2	2016		0	1314E01P	CHP-0233	NextGen	NextGen	5.8.1	Complete	Ambulat	2014		
15	1003000621.00		EP	Virginia	23901	Chiropractor		Medicare	2012	Stage 1	1	9	2012		1	30000004	CHP-0081	MPN Soft	ECLIPSE	2011	Complete	Ambulat	2011		
16	1003000639.00		EP	Californi	90095	Surgery		Medicare	2015	Stage 1		2	2016		0	1314E01P	CHP-0247	Epic Syst	EpicCare	Epic2014	Complete	Ambulat	2014		
17	1003000902.00		EP	Indiana	40212	Family Medicine		Medicare	2012	Stage 1	1	9	2012		1	30000001	CHP-0074	GE Healt	Centricity	9.5	Complete	Ambulat	2011		
18	1003000902.00		EP	Indiana	40212	Family Medicine		Medicare	2013	Stage 1	2	3	2014		1	A000001C	CHP-0094	GE Healt	Centricity	10.1.3	Complete	Ambulat	2011		
19	1003000902.00		EP	Indiana	40212	Family Medicine		Medicare	2014	Stage 1	3	2	2015	2014	1	A014E01K	CHP-0214	GE Healt	Centricity	12	Complete	Ambulat	2014		
20	1003000902.00		EP	Indiana	40212	Family Medicine		Medicare	2015	Stage 2	4	2	2016		1	1314E01P	CHP-0282	GE Healt	Centricity	12.0 R3	Complete	Ambulat	2014		
21	1003000936.00		EP	South Ca	29203	Internal Medicine		Medicare	2014	Stage 1	1	2	2015	2014	0	1314E01P	CHP-0231	Greenwa	Greenwa	2014 (17.(	Complete	Ambulat	2014		
22	1003000936.00		EP	South Ca	29203	Internal Medicine		Medicare	2015	Stage 1	2	3	2016		0	1314E01P	CHP-0231	Greenwa	Greenwa	2014 (17.(	Complete	Ambulat	2014		
23	1003001256.00		EP	Colorado	80620	Family Medicine		Medicare	2013	Stage 1	2	12	2013		0	30000001	CHP-0074	NextGen	NextGen	5.6 SP1	Complete	Ambulat	2011		
24	1003001256.00		EP	Colorado	80620	Family Medicine		Medicare	2014	Stage 1	3	2	2015	2014	0	A014E01N	CHP-0220	Epic Syst	EpicCare	Epic 2010	Complete	Ambulat	2014		
25	1003001256.00		EP	Colorado	80620	Family Medicine		Medicare	2015	Stage 2	4	1	2016		0	1314E01P	CHP-0247	Epic Syst	EpicCare	Epic2014	Complete	Ambulat	2014		
26	1003001256.00		EP	Colorado	80620	Family Medicine		Medicare	2015	Stage 2	4	1	2016		0	1314E01P	CHP-0247	Epic Syst	EpicCare	Epic2010	Complete	Ambulat	2014		
27	1003001363.00		EP	Californi	92243	Anesthesiology		Medicare	2013	Stage 1	1	5	2013		0	30000001	CHP-0074	Greenwa	PrimeSui	2011	Complete	Ambulat	2011		
28	1003001462.00		EP	Georgia	30642	Radiology		Medicare	2015	Stage 1		3	2016		0	1314E01P	CHP-0230	Elekta - I	MOSAIQ	2.6	Complete	Ambulat	2014		
29	1003001645.00		EP	Californi	95815	Internal Medicine		Medicare	2013	Stage 1	1	8	2013		0	A0H1301I	CHP-0192	athenahe	athenaCl	13.5	Complete	Ambulat	2011		

APPENDIX 2

Setting up of GUI for AWS instance:

After logging into the successfully instance using putty, use the following steps to install vncserver into the instance to get GUI for it.

1. In putty terminal, gain superuser status type:
   ```
   $ sudo -s
   ```
2. Now install the Ubuntu desktop, before getting the desktop update the machine to the latest updates.
   ```
   $ sudo apt-get update
   $ sudo apt-get install Ubuntu-desktop
   ```
3. Now install vnc4server
   ```
   $ sudo apt-get vnc4server
   ```
4. Now get GNOME panel
   ```
   $ sudo apt-get install gnome-panel
   ```
5. Start vncserver
   ```
   $ vncserver
   ```
6. It asks you to create a password, after creating the password kill the vncserver.
   ```
   $ vncserver -kill :1
   ```
7. Now we need to edit the startupfile in the following way:

   ```
   #!/bin/sh
   # Remove the comments for following two lines:
   unset SESSION_MANAGER
   # exec /etc/X11/xinit/xinitrc
   gnome-session –session=gnome-classic &
   gnome-panel&
   ```

   Press ESC then :WQ to save and quit

8. Now restart vncserver and type the following command to start the desktop service:
   ```
   $ vncserver
   ```
9. Download a vncviewer or a remote connection application.
10. Give the public IP address of your EC2 Instance followed by port 5901

    Eg: 54.210.40.99::5901

APPENDIX 3

The following steps were followed to perform experiment in both environments. CP-ABE is discussed in this page and KP-ABE is discussed in the next page.

1. CP-ABE

Experiment was performed for CP-ABE algorithm, in the following manner:

i. Setting up the public and master keys by running cpabe-setup:

```
$  cpabe-setup
$  ls
   master_key pub_key
```

ii. We can setup different keys using the attributes of master key, consider a patient Tom his doctor Jerry and some other hospital personnel Butch, keys for them can be created in the following way:

```
$  cpabe-keygen -o tom_key pub_key master_key \
     medication_profile phy_team 'admission_date = '`date +%s`
$  cpabe-keygen -o jerry_key pub_key master_key \
     medical_staff phy_team 'executive_level = 7' \
      'hire_date = '`date +%s`
$  cpabe-keygen -o jerry_key pub_key master_key \
     business_staff 'executive_level = 7' \
      'hire_date = '`date +%s`
$  ls
   master_key private_key tom_key jerry_key butch_key
```

iii. Now let's encrypt the EHR file of patient Tom which contains sensitive information and should only be accessed by Tom and Jerry.

```
$  cpabe-enc pub_key tomreport.pdf
      (medication_profile , phy_team) or
     (medical_staff and 2 of (executive_level >= 5, phy_team))
   ^D (break)
$  ls
   tomreport.pdf.cpabe
```

iv. File can only be decrypted by Tom and Jerry's keys as those keys satisfies the policies and Butch's key does not. Can be decrypted in the following way:

```
$  cpabe-dec pub_key tom_key jerry_key tomreport.pdf.cpabe
$  ls
   tomreport.pdf tom_key jerry_key
```

2. KP-ABE

Experiment was performed for KP-ABE algorithm, in the following manner:

i. Setting up public and master keys, with few attributes and access level:

```
$ kpabe-setup medication_profile cardiac_profile ‘server level=’
$ ls

        master_key pub_key
```

ii. Creating private keys using keygen for users:

```
$ kpabe-keygen -o tom_key pub_key master_key 'server level < 4 and 2 of
  (medication_profile, surgical_profile, cardiac profile)'
$ ls
  master_key pub_key tom_key
```

iii. Encrypting a file using pub key:

```
$ kpabe-enc pub_key tomreport.pdf medication_profile surgical_profile and
  cardiac_profile, 'server level = 3'
$ ls
  master_key pub_key tom_key tomreport.pdf.kpabe.
```

iv. Decrypting the encrypted file:

```
$ kpabe-dec pub_key tom_key tomreport.pdf.kpabe
$ ls
  master_key pub_key tom_key tomreport.pdf
```

Metrics:

- Key-Generation time for KP-ABE was measured in Nano seconds, in the results table for ease of understanding, all the times were converted into milliseconds and recorded.

```
$ time +%s%N was used in this case.
```