Secured Access Control System on Cloud Disk Based on Cryptography

INTERIM REPORT

Student	:	Chow Chi Ling (3035072832)
Supervisor	:	Dr. Lucas Hui
Course	:	COMP4801 Final Year Project
Project Code	:	FYP16007
Date	:	January 22, 2017

CONTENTS

1 BACKGROUND

1.1 CURRENT SITUATION

Usage of cloud computing has been growing rapidly recently. Statistics have shown that there is a steady increase in adoption of cloud solutions for enterprises, from 57% respondents in 2012 to 72% of respondents in 2015. Despite the rising popularity, security remains one of the top challenges, such as tackling unauthorized access and data protection issues. [2]

In the enterprise situation, files and data are usually encrypted and are shared and accessible by certain groups of staffs only. This could be securely encrypted by conventional cryptography, such as the public key infrastructure. However, this could require much computation power. If there are large number of staffs sharing the same piece of information, the number of encryption computation could be very huge. As cloud is also often used together with mobile devices, with limited processing power and storage space in mobile devices, cryptographic protection is even more challenging.

Under this situation, attribute based encryption (ABE), a relatively younger cryptography, could possibly be a better alternative. Encryption and decryption are calculated based on attributes of individuals and a specified access control policy. Therefore, people with different secret keys would be able to decrypt the same ciphertext if the supplied attributes match the specified access control policy. This could allow more flexibility in information sharing while protecting from unauthorized access. Moreover, more recent researches have proposed outsourcing ABE which breaks the decryption process into stages, thereby reducing the computation power requirements.

1.2 THEORETICAL BACKGROUND

1.2.1 Types of Attribute Based Encryption

Attribute based encryption (ABE) was introduced by Sahai and Waters in 2005. The two major components in ABE are attributes and access control policy. Attribute is the characteristic of a piece of information or a user. A piece of information or a user may be identified by a set of attributes. On the other hand, access control policy is the specification of the set of attributes that users must satisfy in order to decrypt the ciphertext. It is organized in tree based data structure. [1]

ABE can be classified into two types, namely ciphertext-policy attribute based encryption (CP-ABE) and key-policy attribute based encryption (KP-ABE).

In CP-ABE, each user is associated with a set of attributes and is represented in their secret key. During the encryption, ciphertext is associated with the access control policy specified by the sender. All users possessing a secret key with attributes satisfying the access control policy specified in the ciphertext would be able to decrypt and get the message.

On the contrary, in KP-ABE, user's secret key is associated with the access control policy while the ciphertext is associated with a set of attributes specified by the sender. Users would be able to decrypt a ciphertext if the ciphertext contains attributes that satisfy the access control policy specified in his secret key.

1.2.2 Implementation

Since 2005, there are various related works in ABE. Implementation methods are slightly different, but they comprise of the four basic steps described below. [3],[4],[5]

1. Setup

Public key and the master secret key are generated.

2. Key Generation

In CP-ABE, individual secret key is generated from a set of attributes.

In KP-ABE, individual secret key is generated from an access control policy.

3. Encryption

In CP-ABE, message is encrypted by the sender from the public key and an access control policy to give the ciphertext.

In KP-ABE, message is encrypted by the sender from the public key and a set of attributes to give the ciphertext.

4. Decryption

Ciphertext is decrypted by the user who possesses a valid secret key.

2 OBJECTIVE

The project would focus on the implementation of ciphertext-policy attribute based encryption (CP-ABE) in the context of electronic health record system. The project aims at achieving the following objectives:

1. To protect electronic health records by CP-ABE
2. To support operations on mobile devices
3. To develop convenient management of access control policy and attributes for CP-ABE

3 METHODOLOGY

The project would be carried out based on an open source electronic health record system CaboLabs EHRServer version 0.7. It is a web based service-oriented system complying with the internationally accepted openEHR standard. Besides, an Android application would be developed as a client application interface. To incorporate CP-ABE into the system, the open source Libfenc Library would be used.

3.1 OPENEHR FRAMEWORK

The openEHR framework is an international open standard governing systems related to electronic health records. The standard is developed and continuously maintained by a large group of professionals. Many organizations, including commercial and research projects, are conforming to the standard and hence it is internationally accepted. The aims of the standard are as follows.

1. To design better software architecture for maintainability
2. To unify data format for system scalability and interoperability

Two-level modelling is suggested in openEHR framework. In this modelling method, the structure is governed by reference model while the content is governed by archetype model. In terms of access control, there are not concrete requirements, but some guidelines summarized as below.

1. Patient consent is needed whenever his/her information is shared.
2. Different parts of EHR should be allowed to access with different rights.
3. Learning curve must not be steep for users.
4. Access control is needed for modification of access control policy of EHR.
5. Access must be declined except it is defined explicitly.

3.2 CABOLABS EHRSERVER

CaboLabs EHRServer is an open source project conforming to the openEHR standard. It is developed using Grails framework, a web framework for Groovy. Data are stored in XML files in the project directory as well as in MYSQL database.

There are two parts in version 0.7, namely an administrative web platform for direct graphical access from users and REST API for other application interfaces to access the data. The following functions are provided:

1. Register and login as users
2. Create and modify other users
3. Create and modify patients and their electronic health records
4. Create and modify documents of electronic health records
5. Query data in electronic health records

3.3 LIBFENC LIBRARY

Libfenc is an open source library with implementation of several functional encryption including both CP-ABE and KP-ABE. A few versions of ABE schemes were implemented in the library.

4 IMPLEMENTATION DETAILS

4.1 MODIFICATION TO CABOLABS EHRSERVER

4.1.1 Administrative Web Platform

Two functions were added to prepare for the implementation of ciphertext-policy attribute based encryption (CP-ABE).

Firstly, functions related to departments were added. In the original implementation, users belong to different organizations. In the actual situation, however, users are also divided into different departments. More importantly, different departments may have different access rights to a patient's electronic health record (EHR) depending on situations. Therefore, departments were implemented with creation, retrieval, update and deletion functions. Each user can be associated with the departments they are working for.

Secondly, functions related to access policies were added. Organizations and departments are used as attributes in CP-ABE to identify users. To allow convenient management of access control policy, creation, retrieval, update and deletion functions of access policies were implemented. Users can modify the access policy of a specific EHR whenever it is needed.

4.1.2 REST API

Although most of the functions in REST API has been implemented with both XML and JSON response, it is found that a few can only support XML response. Hence, two functions were modified to support JSON response as well.

Besides, a new function was created to retrieve EHR access policy by the unique identifier of the EHR.

GET /ehr/$ehrUid/ehrAccess

Parameters:

"format": specify the output format, valid values are "xml" or "json"

Result sample: (Content-Type: application/json)

{

"policy" : "((department = a1b264e7-8bdb-4882-98c5-5bf3e2fc604b) OR (department = 549f1e0b-e381-4155-9b98-be545f2d3a6a))"

}

4.2 MODIFICATION TO DATABASE

In total, four new tables were created to support the modification mentioned in the above section.

A table “department” was created to store the information of the departments. One user can work for more than one departments. Hence, a table “user_department” is created to store the 1:M relationship of users and departments.

department

“uid” is the unique identifier of the department.

“version” records the number of edition to the record.

“name” is the name of the department.

“organization_uid” is the foreign key to identify the organization the department belongs to.

user_department

“departments_idx” is the unique identifier of a record.

“user_departments_id” is the foreign key to identify the user.

“department_id” is the foreign key to identify the department.

Two new tables were created to store the access policy of EHR. Since a CP-ABE access policy is in tree structure, the table “access_policy” was created to store this information. A whole access policy is stored separately in more than one records. The table “ehr_access” was created to point an EHR to the root of its access policy in the table “access_policy”.

ehr_access

“uid” is the unique identifier of an EHR access.

“version” records the number of edition to the record.

“scheme” is the security scheme used.

“ehr_uid” is the foreign key to identify the corresponding EHR.

“settings_id” is the foreign key to identify the access policy.

access_policy

“uid” is the unique identifier of an access policy.

“version” records the number of edition to the record.

“type” defines the meaning of the value column. It can be organization, department or criteria.

“value” is the value corresponding to the type defined.

“parent_id” is the foreign key to identify its parent access policy. A null value means it is the root of the whole access policy tree.

4.3 Development of Android Application

The Android application acts as a client application interface for users. It consists of three major functions.

Firstly, create a document. There are many cases that new clinical documents are created to record patients' situations or medical treatments they have received. Since the focus of the project is CP-ABE, the type of clinical documents supported is limited to nursing observations only for simplicity. After users complete all required inputs and click submit, the application will send a HTTP POST request to the REST API. The remaining process would be handled by REST API. Upon successful submission, the user will be redirected back to the previous page.

Secondly, retrieve a document. In diagnosis, it is essential to retrieve previously created clinical documents to learn about the patients or monitor their situations. When users click on a patient's EHR, he can read the full list of clinical documents. By clicking on one of the record, the application will send a HTTP GET request to the REST API to get the specific document and display on screen.

Thirdly, modify a document. There could be situations that some previously inputted data are found to be incorrect and thus correction is needed. After user retrieving a specific document, he can click modify button and correct any mistakes. After submission, the application will send a HTTP POST request to the REST API. The remaining process would be handled by REST API. Upon successful submission, the user will be redirected back to the previous page.

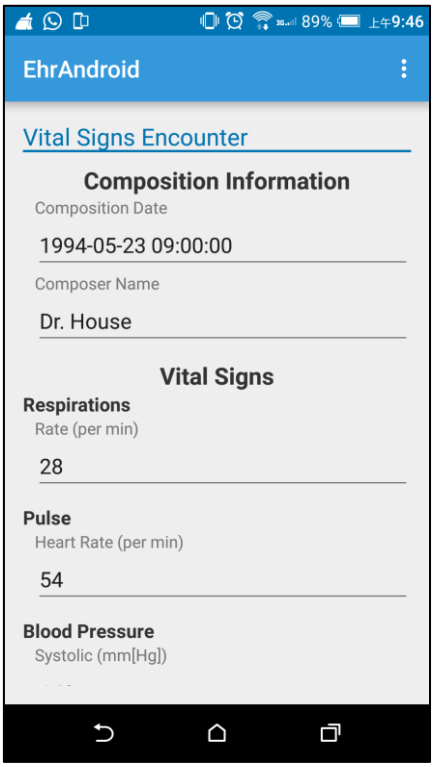

Figure 1 User Interface of Android Application

5 CURRENT PROGRESS

Date	Milestones	Deliverables
2 October 2016	Completion of Phase 1	Detailed Project Plan Project Web Page
9-13 January 2017	First Presentation	
22 January 2017	Completion of Phase 2	Preliminary Implementation Detailed Interim Report
16 April 2017	Completion of Phase 3	Finalized Tested Implementation Final Report
18-21 April 2017	Final Presentation	

In phase 1 and 2, the whole system without CP-ABE has been developed and functioned properly. Hence, the progress is satisfactory.

6 FUTURE DEVELOPMENT

In phrase 3, incorporating CP-ABE into the current system is the major focus. Incorporation would be done step by step. Simplified process would be implemented first. Finally, enhancement to the system would be done. This includes functional enhancement as well as user interface enhancement.

7 BIBLIOGRAPHY

[1] "Introduction to Attribute Based Encryption (ABE)," [Online]. Available: http://gleamly.com/article/introduction-attribute-based-encryption-abe.

[2] IDG Enterprise Marketing, "Clound Computing Survey 2015," 17 11 2015. [Online]. Available: http://www.idgenterprise.com/resource/research/2015-cloud-computing-study/.

[3] S. Banescu, H. Ideler and S. Posea, "Personal Health Record System Protected using CP-ABE".

[4] J. Li, X. Huang, J. Li, X. Chen, Y. Xiang and IEEE, "Securely Outsourcing Attribute-Based Encryption with Checkability," *IEEE Transactions on Parallel and Distributed Systems,* vol. 25, no. 8, pp. 2201-2210, 2014.

[5] F. Xhafa, J. Li, G. Zhao, J. Li, X. Chen and D. S. Wong, "Designing Cloud-Based Electronic Health Record System with Attribute-Based Encryption," Springer Science+Business Media New York, New York, 2014.