Cloud based Secured Health Record Management and Sharing Using cryptographic and stenographic technique

Abstract— The use of cloud computing becomes popular and widespread. So, its systematic management should be applied. The utilization of modern cloud technology in the delivery of health information management is to enhance the availability and reliability of improved healthcare services to patients at a reduced cost and reliable health information sharing/exchange. However, there have been wide privacy concerns as personal health information could be exposed to those third party servers and to unauthorized parties. To assure the patients’ control over access to their own health information, it is a promising method to encrypt the information before outsourcing.

Steganography is the art of hiding the fact that communication is taking place, by hiding information in other information. Many different carrier file formats can be used, but digital images are the most popular because of their frequency on the Internet. Steganography is the art and science of invisible communication, which plays a vital role in information security.

Yet, issues such as risks of privacy exposure, scalability in key management, flexible access and efficient user revocation, have remained the most important challenges toward achieving cryptographically enforced data access control. In this paper, I propose a mechanism for data access control to health information stored in semi-trusted servers. To achieve secure and scalable data access control for patient health information, I leverage advanced encryption standard (AES) techniques to encrypt each patient’s data file.

The project process involves converting a secret image into a text document, then encrypting the generated text into a ciphertext using a key (password) based encryption algorithm, and finally embedding the ciphertext on to a cover image(using AES algorithm). This embedding process is carried out using a threshold based scheme that inserts secret message bits into the cover image only in selected pixels. The security to maintain secrecy of message is achieved by making it infeasible for a third person to detect and retrieve the hidden message.

I. INTRODUCTION

While it is exciting to have convenient HRMS services for everyone, there are many security and privacy risk which could impede its wide adoption. The main concern is about whether the patients could actually control the sharing of their sensitive personal health information (PHI), especially when they are stored on a third-party server which people may not fully trust. On the one hand, although there exist healthcare regulations such as HIPAA which is recently amended to incorporate business associates, cloud providers are usually not covered entities. On the other hand, due to the high value of the sensitive personal health information (PHI), the third-party storage servers are often the targets of various malicious behaviors which may lead to exposure of the PHI. As a famous incident, a Department of Veterans Affairs database containing sensitive PHI of 26.5 million military veterans, including their social security numbers and health problems was stolen by an employee who took the data home without authorization. To ensure patient-centric privacy control over their own HRMSs, it is essential to have fine-grained data access control mechanisms that work with semi-trusted servers.

A feasible and promising approach would be to encrypt the data before outsourcing and applying steganography. Basically, the HRMS owner herself should decide how to encrypt her files and to allow which set of users to obtain access to each file. A HRMS file should only be available to the users who are given the corresponding decryption key, while remain confidential to the rest of users. Furthermore, the patient shall always retain the right to not only grant, but also revoke access privileges when they feel it is necessary. However, the goal of patient-centric privacy is often in conflict with scalability in a HRMS system. The authorized users may either need to access the HRMS for personal use or professional purposes. Examples of the former are family member and friends, while the latter can be medical doctors, pharmacists, and researchers, etc. We refer to the two categories of users as personal and professional users, respectively. The latter has potentially large scale; should each owner herself be directly responsible for managing all the professional users, she will easily be overwhelmed by the key management overhead. In addition, since those users’ access requests are generally unpredictable, it is difficult for an owner to determine a list of them. On the other hand, different from the single data owner scenario considered in most of the existing works in a HRMS system, there are multiple owners who may encrypt according to their own ways, possibly using different sets of cryptographic keys. Letting each user obtain keys from every owner who’s HRMS she wants to read would limit the accessibility since patients are not always online. An alternative is to employ a central authority (CA) to do the key management on behalf of all HRMS owners, but this requires too much trust on a single authority (i.e., cause the key escrow problem). In this paper, we endeavor to study the patient centric, secure sharing of HRMSs stored on semi-trusted servers, and focus on addressing the complicated and challenging key management issues. In order to protect he personal health data stored on a semi-trusted server.
A. Abbreviations and Acronyms

**HMS**: HEALTH RECORD MANAGEMENT SYSTEM

**ACTUARY**: THIRD PARTY WHO HAVE ACCESS TO PHR

**JSP**: JAVA SERVER PAGE

**PUD** - PUBLIC DOMAINS

**PSD** - PERSONAL DOMAINS

**AES** - ADVANCED ENCRYPTION STANDARD

II. Benefits of Health Record Management System

The benefits of HRMSs are defined by the types of functions they support. There are a variety of potential HRMS infrastructures and application functions. The HRMS infrastructure functions consisted of sharing of complete test results and sharing of complete medication lists. The HRMS application functions were smoking cessation management, congestive heart failure (CHF) monitoring, appointment scheduling, medication renewals, preen counter questionnaires, and e-visits. The impact of these functions is derived from reducing administrative costs and direct healthcare utilization costs.

a. Sharing of Complete Test Results

Providers may re-order tests when necessary results are not available at the time of the patient visit. Typically, this occurs because the current provider does not have the results of tests ordered by another provider. In a non-EHR world, the test results may be lost or unavailable and must be conducted again when results are imperative to clinical decision making, thus resulting in unnecessary spending by the healthcare system for the duplicate test. For providers with an EHR, there is a reduced chance of losing a test result, and clinical decision support systems (CDSS) can prompt the provider of the prior test result during the ordering of a test. This benefit can also be achieved through HRMSs. The HRMS infrastructure includes a data repository where test results, among other healthcare data, are stored. Unfortunately, there have been no studies to date that examine the impact of HRMSs on the reduction in redundant tests. Issue and area of potential savings in healthcare. 38-42 Many types of ADEs exist, and HRMSs have the potential to enhance avoidance of some preventable ADEs: those ADEs that are not caught during the medication management process but are foreseeable. 43 CITL modeled the impact of HRMSs on preventable ADEs due to DDI. When a provider prescribes a medication for a patient, there is a chance that it will interact with another drug that the patient is taking. Typically, the provider knows which medications he or she has prescribed for the patient. However, the patient may be on medications from another provider or taking over-the-counter medications that the provider is unaware of. Therefore, an accurate medication list is necessary to avoid preventable ADEs from outpatient medications. While no research has investigated the use of a HRMS in this capacity, a patient-maintained medication list that can be shared with a provider through a HRMS could be beneficial in reducing preventable DDI ADEs. Existing research shows that current medication lists are not 100% complete or accurate. 44-47 For this analysis, CITL assumed that a HRMS with a patient-maintained medication list would ensure a patient’s medication list to be 100% complete, thereby eliminating all preventable ADEs and their associated costs due to outpatient, preventable drug-drug interactions, as detected by a provider’s EHR error-checking capabilities. The analysis also assumes that all providers have an EHR with medication decision support capability.

b. Benefits to patients

For patients, HRMSs add value in four ways: increasing access to education and care, increasing patient control, enabling greater patient choice, and improving the quality of care. Patient access to education and care HRMS systems offer several tools that improve patient access to education and care. Medical history: Access to clinical records can give patients the facts with which to gain a deeper understanding of their conditions and help them make better health plans. Educational resources: HRMSs can automatically provide relevant background and treatment information, based on patient histories (e.g., a diabetic who logs on to a HRMS system will see links to the latest literature and resources on diabetes). Social networking and patient provider interaction: Patients can interact with caregivers without visiting a hospital or primary care center through interactive chat tools, e-visits or emails. Patient control: Most HRMSs provide dashboards that allow patients to monitor their health,
potentially preventing certain illnesses and conditions. For example, consumers who used a health tracking system that issued weekly email reminders about exercise targets increased their average exercise time by 55 minutes a week and reduced sedentary periods by 2 hours a week. These systems can also be used to extend the reach of public health initiatives (e.g., public service announcements, education). Patient choice in their health The reference resources often attached to a HRMS (e.g., hospital and physician rankings, medical articles) provide the basis for more informed decisions about care.

III. IMPLEMENTATION

Implementation is the most crucial stage in achieving a successful system and giving the user’s confidence that the new system is workable and effective. Implementation of a modified application to replace an existing one. This type of conversation is relatively easy to handle, provide there are no major changes in the system.

a. Modules Description

![Diagram of Modules Description]

b. Registration

In this module normal registration for the multiple users takes place. There are multiple owners, multiple AAs, and multiple users. The attribute hierarchy of files – leaf nodes is atomic file categories while internal nodes are compound categories. Dark boxes are the categories that a PSD’s data readers have access to.

c. AES Encryption Algorithm

In this module AES to realize fine-grained access control for outsourced data especially; there has been an increasing interest in applying ABE to secure electronic healthcare records (EHRs). An attribute-based infrastructure for EHR systems, where each patient’s EHR files are encrypted using a broadcast variant of CP-ABE that allows direct revocation. However, the cipher text length grows linearly with the number of un revoked users. In a variant of AES that allows delegation of access rights is proposed for encrypted EHRs applied cipher text policy AES (CP-AES) to manage the sharing of HRMSs, and introduced the concept of social/professional domains investigated using AES to generate self-protecting EMRs, which can either be stored on cloud servers or cell phones so that EMR could be accessed when the health provider is offline.

AES is an iterated block cipher with a fixed block size of 128 and a variable key length. The different transformations operate on the intermediate results, called state. The state is a rectangular array of bytes and since the block size is 128 bits, which is 16 bytes, the rectangular array is of dimensions 4x4. (In the Rijndael version with variable block size, the row size is fixed to four and the number of columns varies. The number of columns is the block size divided by 32 and denoted Nb). The cipher key is similarly pictured as a rectangular array with four rows. The number of columns of the cipher key, denoted Nk, is equal to the key length divided by 32

\[
\begin{align*}
\text{AES Key:} & \quad a_{1}, a_{2}, a_{3}, a_{4}, a_{5}, a_{6}, a_{7}, a_{8},
\text{AES File:} & \quad k_{1}, k_{2}, k_{3}, k_{4}, k_{5}, k_{6}, k_{7}, k_{8},
\text{Steganography:} & \quad b_{1}, b_{2}, b_{3}, b_{4}, b_{5}, b_{6}, b_{7}, b_{8}.
\end{align*}
\]

\[
\begin{bmatrix}
a_{1} & a_{2} & a_{3} & a_{4} \\
a_{5} & a_{6} & a_{7} & a_{8} \\
k_{1} & k_{2} & k_{3} & k_{4} \\
k_{5} & k_{6} & k_{7} & k_{8}
\end{bmatrix} +
\begin{bmatrix}
b_{1} & b_{2} & b_{3} & b_{4} \\
b_{5} & b_{6} & b_{7} & b_{8} \\
k_{1} & k_{2} & k_{3} & k_{4} \\
k_{5} & k_{6} & k_{7} & k_{8}
\end{bmatrix} =
\begin{bmatrix}
a_{1} + b_{1} & a_{2} + b_{2} & a_{3} + b_{3} & a_{4} + b_{4} \\
a_{5} + b_{5} & a_{6} + b_{6} & a_{7} + b_{7} & a_{8} + b_{8} \\
k_{1} + b_{1} & k_{2} + b_{2} & k_{3} + b_{3} & k_{4} + b_{4} \\
k_{5} + b_{5} & k_{6} + b_{6} & k_{7} + b_{7} & k_{8} + b_{8}
\end{bmatrix}
\]

d. Steganography

All of the approaches to steganography have one thing in common that they hide the secret message in the physical object which is sent. The following figure shows the steganography process of the cover image being passed into the embedding function with the message to encode resulting in a steganographic image containing the hidden message. A key is often used to protect the hidden message. This key is usually a password, so this key is also used to encrypt and decrypt the message before and after the embedding. Secrets can be hidden inside all sorts of cover information: text, images, audio, video and more. However, there are tools available to store secrets inside almost any type of cover source. The most important property of a cover source is the amount of data that can be stored inside it, without changing the noticeable properties of the cover. Secrets can be hidden inside all sorts of cover information: text, images, audio, video and more. Most steganographic utilities nowadays, hide information inside images, as this is
relatively easy to implement. However, there are tools available to store secrets inside almost any type of cover source. It is also possible to hide information inside texts, sounds and video films for example. The most important property of a cover source is the amount of data that can be stored inside it, without changing the noticeable properties of the cover. When an image is distorted or a piece of music sounds different than the original, the cover source will be suspicious and may be checked more thoroughly.

Hiding information inside images is a popular technique nowadays. An image with a secret message inside can easily be spread over the World Wide Web or in newsgroups. The use of steganography in newsgroups has been researched by German steganographic expert Niels Provos, who created a scanning cluster which detects the presence of hidden messages inside images that were posted on the net. However, after checking one million images, no hidden messages were found, so the practical use of steganography still seems to be limited. To hide a message inside an image without changing its visible properties, the cover source can be altered in "noisy" areas with many color variations, so less attention will be drawn to the modifications. The most common methods to make these alterations involve the usage of the least-significant bit or LSB, masking, filtering and transformations on the cover image. These techniques can be used with varying degrees of success on different types of image files. The most widely used technique to hide data, is the usage of the LSB. Although there are several disadvantages to this approach, the relative easiness to implement it, makes it a popular method.

To hide a secret message inside a image, a proper cover image is needed. Because this method uses bits of each pixel in the image, it is necessary to use a lossless compression format, otherwise the hidden information will get lost in the transformations of a lossy compression algorithm. When using a 24 bit color image, a bit of each of the red, green and blue color components can be used, so a total of 3 bits can be stored in each pixel. Thus, a 800 x 600 pixel image can contain a total amount of 1,440,000 bits (180,000 bytes) of secret data. For example, the following grid can be considered as 3 pixels of a 24 bit color image, using 9 bytes of memory:

| (00100111 11101001 11001000) |
| (00100111 11001000 11101001) |
| (11001000 01001111 11101001) |

When the character A, which binary value equals 10000001, is inserted, the following grid results:

| (00100111 11101000 11001000) |
| (00100110 11001000 11101000) |
| (11001000 00100111 11101001) |

In this case, only three bits needed to be changed to insert the character successfully. On average, only half of the bits in an image will need to be modified to hide a secret message using the maximal cover size. The resulting changes that are made to the least significant bits are too small to be recognised by the human eye, so the message is effectively hidden.

IV. Sample Code

```java
<%@ page contentType="text/html" %>
<%! String user; %>
<% user = (String)session.getAttribute("user");
<% session.setAttribute("user", user);
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta name="keywords" content="" />
<meta name="description" content="" />
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>Scalable and Secure Sharing of Personal Health Records in Cloud Computing using Attribute-based Encryption</title>
<link href="http://fonts.googleapis.com/css?family=Oswald:400,300" rel="stylesheet" type="text/css">
```
V. FUTURE ENHANCEMENT

The project I have undertaken can be used as a reference or as a base for realizing a scheme to be implemented in other projects of greater level. Every application has its own merits and demerits. The project has covered almost all the requirements. Further requirements and improvements can easily be done since the coding is mainly structured or modular in nature. Changing the existing modules or adding new modules can append improvements. Further enhancements can be made to the application, so that the web site functions very attractive and useful manner than the present one.

VI. CONCLUSION

The system tries to monitor the efficient sharing of patient record or data. In this system, I have tried to make a system of secure sharing of personal health records in cloud. Considering partially trustworthy cloud servers, I argue that to fully realize the patient-centric concept, patients shall have complete control of their own privacy through encrypting their HRMS files and then applying stenographic effect. The system addresses the unique challenges brought by multiple HRMS owners and users. I utilize AES to encrypt the HRMS data, so that patients can allow access not only by personal users, but also various users from public domains with different professional roles, qualifications and affiliations. Through implementation and simulation, we show that our solution is both scalable and efficient. The project I have undertaken has helped me to gain a better perspective on various aspects related to cloud computing system, web technology and database. In addition to the knowledge I got, it is about the dream which I set for a long before. The system tries to securely share patient record. It is not the end, but it should be enhanced and updated continually. The system opens my eye and encourages me for further studies in cloud computing and web technologies.

VII. REFERENCES AND BIBLIOGRAPHY

Good Teachers are worth more than thousand books.

References Made From:
1. Data Communications and Networking, by Behrouz A Forouzan.


