Multi-Part File Encryption for Electronic Health Records Cloud

Xiali Hei and Shan Lin
Department of Computer and Information Sciences
Temple University
19122 Philadelphia, USA
xiali.hei, shan.lin@temple.edu

ABSTRACT

The rapid advancements of mobile technologies promote many applications for public health, such as continuous health monitoring. The inherent mobility of these applications imposes new security and privacy challenges. Since mobile devices usually use public network, such as WiFi, to transfer patient data, patient data is exposed to various security breaches. Moreover, patient data stored on cloud servers are also exposed to malicious attacks. Therefore, it's crucial to encrypt patient data for secure transfer and storage. To address this problem, we present a new access control model for managing patient data. Our approach utilizes a key server for key assignment, which associates a key with each user based on his specific role in medical applications. The doctors, nurses, family members, and insurance companies of a patient can access different sets of patient data from cloud given their keys. Different from existing attribute based encryption, which protects data from inappropriate disclosure for individual files, our design provides a fine-grained access control scheme that protects any specified part of a file. Our role-based access control provides high security, accuracy, and update flexibility for patient data management. Performance evaluations of our solution are stated in the paper.

Categories and Subject Descriptors

K.6.5 [**Management of Computer and Information Systems**]: [Security and Protection-access control]; J.3 [**Computer Applications**]: [Medical information systems]

General Terms

Algorithms, Security

Keywords

Field-level security; Attributed based encryption; SubSet Sum problem, electronic health records, mHealth

1. INTRODUCTION

The unprecedented spread of mobile devices and their applications for public health have evolved into mHealth. According to the International Telecommunication Union (ITU), there are now close to 5 billion mobile phone subscriptions in the world, with over 85% of the world's population now covered by a commercial wireless signal [1]. Wireless health market had risen from 2.7 million in 2007 to 9.6 billion in 2012, and more than 80% physicians in U.S. use smartphones. 95% physicians using handheld devices/smartphones download applications for medical info [2]. Many hospitals allow the use of mobile and wireless devices without formal policies and procedures in place.

The growing mobility in health applications imposes a lot of security and privacy challenges. For example, patient data is usually stored on third party cloud server and transferred over public network to smart phones. Additionally, many mobile devices transfer data using airdrop over near field communication, Bluetooth, or WiFi in plain text directly. There are many vulnerabilities such malwares in existing mHealth systems [3] [4]. On the other hand, a patient's health information may contain sensitive information such as identity data (ID), sexual health (SH), mental health (MH), dermatology health (DH), addictions to drug or alcohol, abortions, etc. To suppress security and privacy breaches on such sensitive information, patient specific data protection should be enforced flexibly for various mobile health applications.

In our design, patient data is encrypted and stored as files on cloud server. Different users can access specific information about a patient across files with their keys associated with their roles in the application. More specifically, authentication is the initial stage of validation of the users to determine whether they are who they claim they are. The data owner utilizes a key server to generate keys and assign them to different users based on their specified access rights. Once authenticated, each user gets a unique key to decrypt the encrypted file downloaded from the cloud server to his/her mobile devices. After decryption, users can view the specific health data of the patient. The architecture of this process is shown in Figures 1.

MobileHealth'14 Aug 11-14, 2014, Philadelphia, USA

http://dx.doi.org/10.1145/2633651.2637473

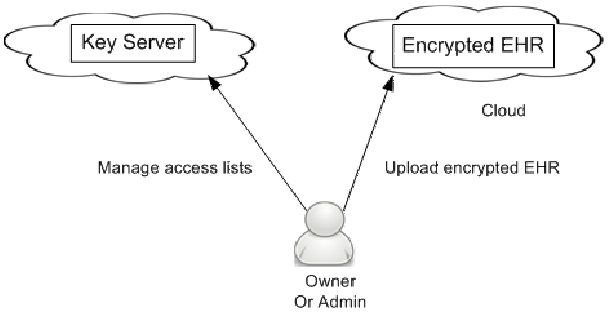

Figure 1: The overview of the system

In health care applications, a patient usually shares specific parts of his/her health data (may overlap from multiple record files) to multiple healthcare providers, each healthcare provider should only view the parts that they are allowed to access. For example, patient A's files containing ID, SH, MH and DH are encrypted, his different healthcare providers can decrypt them and they could see only the parts they have been given authorization to see: doctor Sandra can see the ID and SH, another doctor Bill can see the ID and MH, and at the same time the other doctor Matt can see the ID and DH, as shown in Figures 2 and 3.

To share different parts of a health data to others without inappropriate disclosure, an existing approach encrypts each part of file and sends them to a cloud. Using this approach, front end encryption algorithm on mobile devices needs to differentiate each part and encrypt them separately. This approach has two drawbacks: 1) If the part is binary and with limited values, malicious attackers can use brute-force method to get these values. Thus, this approach does not provide strong security. 2) The algorithm and the system administrator need to maintain multiple parts as files and corresponding access control list for each parts. 3) Updating access rights associated with parts of files and different users is difficult.

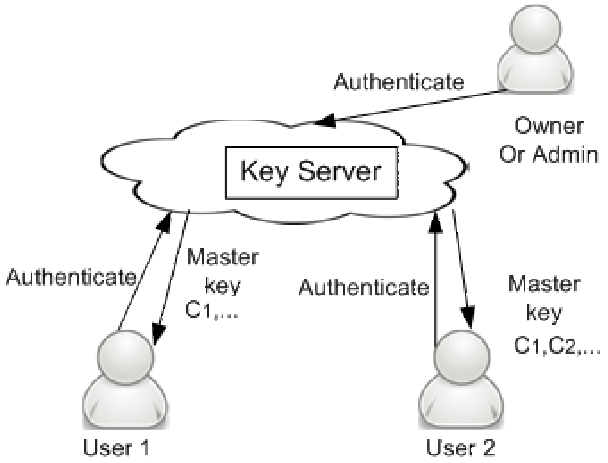

Figure 2: The key assignment process

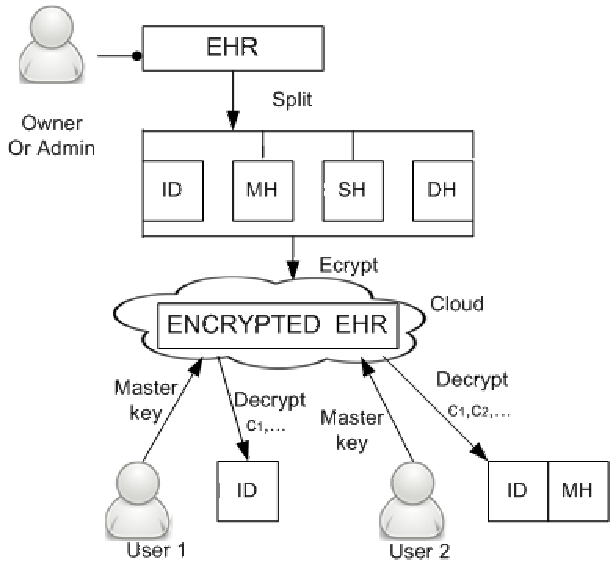

Figure 3: System model

Access control is one of the main safeguards against improper data access. Among all the access control approaches, role based access control is applied widely. Attribute based encryption (ABE) [7] achieves role based access control by associating the attributes to the cipher text and private key. It achieves file-level role protection. That means the authorized healthcare providers could decrypt the entire file with an assigned key. Regarding the example we stated above, the ABE scheme does not work in this scenario because all the authorized users get the same information after decryption of the patient file.

To address this problem, we need a fine-grained encryption scheme to control the authorized users to see only a certain part of a file. We can encrypt the whole file which consists of multiple fields or parts, and assign different keys to multiple users so that they can decrypt any fields or parts from the encrypted file. In this paper, we propose an access control method for mobile health data management. Our design employs a multi-part file encryption scheme that splits data into several parts and encrypting them with different levels of access rights for fine-grained role-based control.

Our contributions are summarized as follows:

- The proposed scheme provides a role based access control to manage the electronic health record in mobile health scenarios.
- Our design allows the systems administrators to enforce field-level access control to multiple users across multiple files, significantly reducing the control overhead on mobile platforms.

The remainder of this paper is organized as follows: In Section II we describe the system model our scheme can be applied to. We analyze the encryption model and related algorithms in Section III. We evaluate our implementation in Section IV. In Section V, we review related work, and we conclude the paper and discuss future work in Section VI.

2. SYSTEM AND ATTACK MODELS

2.1 Security and privacy requirements

In this section, we discuss the security and privacy requirements of healthcare providers and patients in mobile health regarding the health data storage. Specially, we focus on the requirements that could be solved through access control schemes.

Access requirements The following access requirements of healthcare providers (both individual and the health authority) can be identified that need to be addressed in the design of a mobile health system.

- Healthcare providers need to have the capability to share patient health information with other health specialists to make well informed decisions.
- A healthcare authority should have the super user key to access the mobile health cloud. E.g. A life threatening emergency situation.

Privacy requirements A patient's health information may contain sensitive information such as sexual health (SH), mental health (MH), addictions to drug or alcohol, abortions, etc. This makes such a patient demand strong security for their EHRs. These requirements however cannot contradict those set by the healthcare providers or the healthcare authority discussed above.

We note that in the PCEHR [5] system proposed by NEHTA, all privacy settings are set by the patients. Therefore such conflicts will not arise in their proposed system. The following capabilities can be identified as requirements of a patient having an EHR in terms of access control.

- Patients need to have the capability to control access to their EHR. They should be able to allow only a preferred set of medical practitioners to access certain part of their EHR.
- Not all the medical practitioners could have full access to a patient's EHR.
- Patients need to have the capability to see the operation list of their EHR by each user who has access to it.
- The key assignment process must be easy to handle.

Note that, during emergency a patient's safety requirement outweighs the security and privacy requirement. We also need a special key for a super user who can access all the electronic health records.

2.2 System and attack models

Figure 1 illustrates the overview of the system. The data owner has a mobile device. The data owner generates a master key (MK) to assist the encryption and decryption. After finishing the encryption, the data owner sent the encrypted EHR to the cloud and have the subkey associated to each part. Then the data owner or system administrator generated a unique decryption key including the subkeys to each user and update the key column in the access control list as in Table 1.

Figure 2 illustrates the key assignment process. We can see that the data owner or the administrator authenticates himself to the key server in order to assign the keys to the users. The user's decryption key depends on his/her role. The access control list is stored on the key server. Table 1 shows the structure of an access control list. There is a key mixture algorithm running on this key server. The data owner runs it to generate different keys to the user having the same attributes. Thus, each user will get a unique key. After the user authenticates himself to the key server, he/she could access his/her unique decryption key and the master key used for a group.

Table 1: Access control list

Healthcare Practitioner	Patient's Settings	Key
Peter	<{ID}>	$c_1, ...$
Sandra	<{ID}, {SH} >	$c_1, c_2, ..$
Bill	<{ID}, {MH}>	$c_1, c_3, ..$
Matt	<{ID}, {DH}>	$c_1, c_4, ..$

Table 2: Notations description in SHipher II

Notation	Description
G	Group
a	an element in a group
b_i	a part of plain text block
a_i	a part of cipher text block
c_1	the subkey of b_1
C_k	the set of subkeys c_i
MK	master key to determine the $\{r_1, ..., r_{n-k}\}$
$\mathbb{A}$	a cipher text block
$\mathbb{B}$	a plain text block
$[n]$	$\{1, ..., n\}$
e	the unity element in group G
c_{s_i}	the ith bit in the subkey of b_s

Figure 3 describes the system model. We can see that the encrypted files are stored on the intrusted servers (cloud). The users download the encrypted files from the intrusted servers (cloud) and decrypt them using the keys they are assigned by the data owner or the administrator. Then a user is able to get the certain parts of the encrypted file he/she is authorized to access. Thus, this model achieves fine-grained access control for the medical data by assigning a unique key according to the user's access control list.

3. CONSTRUCTION METHOD OF MULTI-PART FILE ENCRYPTION

The mobile health data needs to be encrypted before uploading. The suitable encryption scheme is supposed to be efficient and secure. In this section, we focus on the construction method of the multi-part file encryption. Here, we introduce the encryption and decryption of one block in detail. Then we use the counter mode (CRT) to connect all the blocks in a chunk.

For simplification, we use $[n]$ to represent $\{1, ..., n\}$. Let G be a group, such as a modular addition group or a non-Abelian group as multiplication on an invertible matrix.

We denote $G = (\mathbb{S}, \bigoplus)$, where $\mathbb{S}$ is a set, be finite or infinite, $\bigoplus$ is its operator. a^0 represents the unity element e in G , a^1 represents a itself, a^{-1} represents the inverse of an element a over group G.

We use a subset $\mathbb{A}$ of $\mathbb{S}$ to represent one cipher text block, and a subset $\mathbb{B}$ of $\mathbb{S}$ to represent one plain text block. The block size is w. The number of plain text parts is k and the number of cipher text parts is n. The relationship between them satisfy the constraints $0 < k < n$. Our goal is to construct $\mathbb{A}$ from $\mathbb{B}$, and the key set $C_k = \{c_1, ..., c_k\}$ as the byproducts. The main steps to construct $\mathbb{A}$ is divided into two parts.

3.1 Master Key Generation

In the first part, we choose $(n-k)$ random numbers $(r_1, ..., r_{n-k})$ in $\mathbb{S}$. Then we record their indices in $\mathbb{S}$ as the master key $MK = \{h_0 \| h_1 \| ... \| h_{n-k}\}$. Then we randomly assign them to $(n-k)$ elements $(a_{\alpha_1}, a_{\alpha_2}, ... a_{\alpha_{n-k}})$ in $\mathbb{A}$, also leave the other k elements to be fixed in the second parts.

In the second parts, we obtain the remaining k elements $(a_{\alpha_{n-k+1}}, a_{\alpha_{n-k+2}}, ... a_{\alpha_n})$ step by step by the following equation (1):

$$b_s = \bigoplus_{j=1}^{n} a_j^{c_{s_j}}; s = 1, 2, ..., k. \quad (1)$$

and $c_s = (c_{s_1} c_{s_2} ... c_{s_n})_2$ is a binary number as the key. where if $s_j \in T_s, c_{s_j} = 1$; otherwise $c_{s_j} = 0$; where $T_s = T'_{s-1} \bigcup \{\alpha_{n-k+s}\}$. $T'_{s-1} \subset T_{s-1}$. $T_0 = \{\alpha_1, ... \alpha_{(n-k)}\}$ includes the random positions of the first $(n-k)$ random numbers of $\mathbb{A}$ chosen in the first part and $\alpha_{n-k+s} \in E = [n] \setminus T_{s-1}$.

Since Equation (1) has only one unknown number on the right side, we can get $a_{\alpha_{n-k+s}}$ by Equation (1).

3.2 Encryption Algorithm

Algorithm 1 illustrates the encryption process.

Algorithm 1 Encryption Algorithm

1: **Input:** A plaintext block $\mathbb{B} = \{b_1, ..., b_k\}$ and $\{r_1, ..., r_{n-k}\}$.
2: **Output:** A ciphertext block $\mathbb{A} = \{a_1, ..., a_n\}$ and $\{c_1, ..., c_k\}$, where $c_s = (c_{s_1} c_{s_2} ... c_{s_n})_2$ is a binary number.
3: Randomly select $T_0 = \{\alpha_1, \alpha_2, ... \alpha_{n-k}\}$ from $[n]$, $E = [n] \setminus T_0$;
4: Set $a_{\alpha_1} = r_1, a_{\alpha_2} = r_2, ... a_{\alpha_{n-k}} = r_{n-k}$;
5: **for** each s in $[k]$ **do**
6: Randomly choose a number $p \in E$ and set $\alpha_{n-k+s} = p$, $E = E \setminus p$;
7: Randomly choose $T'_{s-1} = \{\beta_1, \beta_2, ... \beta_t\}$ from $T_{s-1} = \{\alpha_1, \alpha_2, ... \alpha_{n-k+s-1}\}$, and $t = n-k+s-1$; $T_s = T'_{s-1} \bigcup \alpha_{n-k+s}$
8:
9: **if** $s_j \in T_s$ **then**
10: $c_{s_j} = 1$;
11: **else**
12: $c_{s_j} = 0$;
13: Compute $a_{\alpha_{n-k+s}}$ according to $b_s = \bigoplus_{j=1}^{n} a_j^{c_{s_j}}$;

3.3 Decryption Algorithm

Algorithm 2 illustrates the decryption process.

Algorithm 2 Decryption Algorithm

1: **Input:** A ciphertext block $\mathbb{A} = \{a_1, ..., a_n\}$ and keys $\{c_1, ..., c_k\}$, where $c_s = (c_{s_1} c_{s_2} ... c_{s_n})_2$ is a binary number.
2: **Output:** A plaintext block $\mathbb{B} = \{b_1, ..., b_k\}$;
3: **for** each s in $[k]$ **do**
4: $b_s = e$;
5: **for** each j in $[n]$ **do**
6:
7: **if** $c_{s_j} = 0$ **then**
8: $b_j = b_j \bigoplus e$;
9: **else**
10: $b_s = b_s \bigoplus a_j$;

3.4 Example

Suppose the heartbeat, degree and blood pressure of Bree are 80, 100 and 90. Then $B = \{b_1, b_2, b_3\} = \{80, 100, 90\}$, $n = 6$, $k = 3$, $\alpha_1 = 2$, $\alpha_2 = 4$, $\alpha_3 = 5$, then $E = \{1, 3, 6\}$. Assume $r_1 = 20$, $r_2 = 40$, $r_3 = 38$, let $a_2 = 20$, $a_4 = 40$, $a_5 = 38$. $L = 8$. $\bigoplus$ represents modular q (137).

When $s = 1$, choose $\alpha_{3+s} = \alpha_4 = 1$, then $E = \{3, 6\}$. $T_{s-1} = T_0 = \{\alpha_1, \alpha_2, ... \alpha_{n-k+s-1}\}$=$\{2, 4, 5\}$. Choose $T'_{s-1} = T'_0 = \{\beta_1 = 4, \beta_2 = 5\}$ from $T_{s-1} = T_0 = \{2, 4, 5\}$. $T_s = T'_{s-1} \bigcup \alpha_{3+s} = \{\beta_1 = 4, \beta_2 = 5, \alpha_4 = 1\}$, then $c_1 = (100110)_2$. According to $b_s = \bigoplus_{j=1}^{n} a_j^{c_{s_j}}$, then $b_1 = a_1 + a_4 + a_5 \; mod \; 137$. So $a_1 = 2$.

When $s = 2$, choose $\alpha_{3+s} = \alpha_5 = 3$, then $E = \{6\}$. $T_{s-1} = T_1 = \{\alpha_1, \alpha_2, ... \alpha_{n-k+s-1}\}$=$\{2, 4, 5, 1\}$. Choose $T'_{s-1} = T'_1 = \{\beta_1 = 4, \beta_2 = 5, \beta_3 = 2\}$ from $T_{s-1} = T_1 = \{2, 4, 5, 1\}$. $T_s = T'_1 \bigcup \alpha_{3+s} = \{\beta_1 = 4, \beta_2 = 5, \beta_3 = 2, \alpha_5 = 3\}$, then $c_2 = (011110)_2$, According to $b_s = \bigoplus_{j=1}^{n} a_j^{c_{s_j}}$, then $b_2 = a_2 + a_3 + a_4 + a_5 \; mod \; 137$. So $a_3 = 119$.

When $s = 3$, choose $\alpha_{3+s} = \alpha_6 = 6$, then $E = \{\phi\}$. $T_{s-1} = T_2 = \{\alpha_1, \alpha_2, ... \alpha_{n-k+s-1}\}$=$\{2, 4, 5, 1, 3\}$. $T'_{s-1} = T'_0 = \{\beta_1 = 4, \beta_2 = 5, \beta_3 = 2, \beta_4 = 3\}$ from $T_{s-1} = T_2 = \{2, 4, 5, 1, 3\}$. $T_s = T'_2 \bigcup \alpha_{3+s} = \{\beta_1 = 4, \beta_2 = 5, \beta_3 = 2, \beta_4 = 3, \alpha_6 = 6\}$, then $c_3 = (011111)_2$, According to $b_s = \bigoplus_{j=1}^{n} a_j^{c_{s_j}}$, then $b_3 = a_2 + a_3 + a_4 + a_5 + a_6 \; mod \; 137$. So $a_6 = 127$.

Regarding decryption, the public key is $\mathbb{A}$=$\{a_1 = 2, a_2 = 20, a_3 = 119, a_4 = 40, a_5 = 38, a_6 = 127\}$; the privacy key is $c_1 = (100110)_2$, $c_2 = (011110)_2$ and $c_3 = (011111)_2$ or a combination of them. $b_1 = a_1 + a_4 + a_5 \; mod \; 137 = 4$, $b_2 = a_2 + a_3 + a_4 + a_5 \; mod \; 137 = 10$, $b_3 = a_2 + a_3 + a_4 + a_5 + a_6 \; mod \; 137 = 7$. If a user receives $\mathbb{A}$ and c_1, then he or she can see b_1. Similarly, if a user receives $\mathbb{A}$, c_2 and c_3, he or she can access b_2 and b_3.

3.5 Key generation Algorithm

The healthcare provider Bree only can access part MH, her key includes subkey c_1. Similarly, if another healthcare Henry provider can access part SH and DH, his key includes subkeys c_1 and c_2. If we only assign the subkeys to a user, then the users work as a same role will get the same key. Also the length of key is variable. It is better to assign a user a unique key with fixed length. Since the license number of doctor Bree is fixed, we use it as a string seed to generate a random number R_{Bree} of $(k+1) * length(c_1)$ bits, where the k represents the number of divided parts of a file. Then we use the subkey c_1 to replace the most right bits of R_{Bree}. That means $Key_{Bree} = R_{Bree} << length(c_1) + c_1$. Similarly, we can

get the key of Henry is $Key_{Henry} = (R_{Henry} << length(c_1) + c_1) << length(c_2) + c_2)$. Thus, we can generate a unique key for each user.

3.6 File Update

If the file is from a database, the length of every field are fixed. That means, the data structure of a file is unchanged or the number of the divided parts is fixed. So the update of field of a file does not change the parameters n and k, then we only need to update the encrypted field to be updated. The master key and subkeys don't need to be updated.

If the data structure of an EHR file is changed, we need to update all the keys and the cipher text to update the file. At this time, we need to distribute a new master key and all the subkeys. It will need a lot of message communication messages. Since the structure of a database usually is stable, such case rarely happens.

4. PERFORMANCE EVALUATION

4.1 Security Analysis

The security of this scheme is based on the complexity to solve a general subset sum problem. Paper [8] shows the general subset sum problem is a proven NP-hard problem. So our scheme is computationally secure.

For a non-Abelian group, the mean and position (index) of the random numbers are very important. Assume the attackers have known the number of random numbers is p. Then to get c_1, the attackers can randomly choose $p+1$ numbers from $\mathbb{S}$ and add them. The numbers that the attackers need to try is $C(n, p+1)$.

If the attackers want to get $c_1, ..., c_p$, then the total numbers they need to try is $\sum_{i=0}^{p} C(n, i+1)$. If $n = 2k+1$, then when $p = k$, the total number is $2^{2k} - 1$. That means if $n = 2k$, the attackers need to take 2^{n-1} brute-force effort to break the system. If we choose n=1024, then the brute-force attack is computationally impossible.

For an Abelian group, our analysis is still correct. However, it may be vulnerable to potential attacks such as differential attacks [21]. Since we use all the k parts during the encryption, our scheme resists differential attacks. The attackers need to try $C(n, p)$ different subsets. Regarding a non-Abelian group, there are no other attack methods to apply. The number of subsets that the attackers will need to try is much larger.

For an Abelian group, we can choose the group $G([2^{L-1}], (\oplus \, mod \, q))$ $(q > 2^L)$. According to [22], when density falls in [1, 1+$\frac{logn}{n}$], it resists low-density attacks. So we choose $n = L - 1 = w$ to make density 1.

Superincreasing sequence [11] is a case that we should avoid when we choose random numbers $\{r_1, \, ..., \, r_{n-k}\}$. We can test the random number set before we use it. q is a prime rather than 2^u and the density in this design is 1 to defend against lattice-based attacks and low-density attacks [18–20].

4.2 Computational cost

Assume the length of a message is m, the number of such messages is f. The length of the cipher text is m, and the number of such cipher texts is h. The encryption needs $f * h * m$ bit-operations, while the decryption needs $h * m$ bit-operations.

Table 3: Cost Comparison for 3-part file encryption and CP-ABE

Label	SHipherII	CP-ABE
Computation speed	1ms	66ms

4.3 Memory Expansion

Essentially, we encrypt the k parts of a plain text into n parts. If the size of the file part after encryption does not change, then the expansion ratio of this encryption scheme is $(n/k - 1)$.

4.4 Experimental Results

Table II lists the overhead comparisons between our scheme and CP-ABE scheme when encrypting a file. A 3-part file encryption costs 1ms, while encryption the same file using CP-ABE 3 times costs 66ms on an Ubuntu Linux PC. ABE operations are about 100-1000 times slower than those of RSA [23]. RSA could not provide fine-grained access control.

5. RELATED WORK

To our best knowledge, our scheme is the first scheme to implement multiple-part file encryption. It also provides field level protection. We believe our scheme is similar to the ABE schemes. The concept of ABE was introduced by Sahai and Waters [9]. Later, Goyal et al. [10] further separated ABE into two categories: ciphertext policy (CP) ABE [7] and key policy (KP) ABE [10]. The Merkle-Hellman cryptosystem [12] is a first Public-key cryptosystem based on subset sum problem. The survey [11] points out most of the cryptosystems based on subset sum problem were broken. Literature [12–14] are all public-key cryptosystem, the reason those were broken is that their trapdoors fail the randomness tests. But in this paper, we use numbers that pass the randomness tests. Also, our scheme is a symmetric encryption system. We do not have weak trapdoor design as in the aforementioned system. Okamoto et al. [15] present an subset sum problem based public-key cryptosystem which combines good features of [12] and [13] to overcome known attacks on subset sum problem based cryptosystems. However, recent lattice-based attacks [18–20] raise serious theoretical questions about the security of this system. The authors of paper [17] present a generic construction of a public-key subset sum problem based cryptosystem based on any invertible map with certain homomorphic properties. Paper [16] describes a general construction for monoid-based knapsack protocols. The paper [21] proposes a generic construction of a symmetric crypto system based on any non-Abelian group.

6. CONCLUSION

To protect the electronic health data on patient's mobile devices and cloud, we propose a framework for the data storage employing a multi-part file encryption scheme. This encryption scheme allows a user to decrypt the whole encrypted file and get certain parts of the file they are authorized to access. Our scheme achieves user-centric fine-grained access control, enabling field-level privacy protection for a single file. That means, the data owner or the patient is able to protect the data by assigning a unique to healthcare provider. The unique key is related to the healthcare provider's license and role or attributes. Also, a patient is able to grant the access of a field of his/her electronic health data to a healthcare provider. Additionally, our encryption scheme is efficient comparing with other public key encryption scheme and we provide implementation and perfor-

mance evaluation. In the future, we will work on adding searchable properties to our encryption scheme.

Acknowledgment

We would like to thank Binheng Song for comments. The authors would like to thank Shuang Liang for programming help. And he published a library of this encryption scheme online. This work was supported in part by the united states national science foundation under grants IIS-1231680, CNS-1239108, and CNS-1035715.

7. REFERENCES

[1] World Health Organization, “mHealth: New horizons for health through mobile technologies," *Global Observatory for eHealth series*, vol. 3, 2011.

[2] L. A. Gallagher, “Mobile computing in healthcare: Privacy and security considerations and available resources," http://csrc.nist.gov/news_events/hiipaa_june2012/day1/day1-a1_lgallagher_mobile.pdf, 2012.

[3] http://www.arxan.com/solutions/mobile-medical-device-security/

[4] National Cybersecurity and Communications Integration Center, “Attack Surface: Healthcare and Public Health Sector," 2012, http://info.publicintelligence.net/NCCIC-MedicalDevices.pdf

[5] National E-Health Transition Authority, “Draft Concept of Operations: Relating to the introduction of a personally controlled electronic health record (PCEHR) system," 2011.

[6] R. S. Sandhu and P. Samarati, “Access control: principle and practice," *IEEE Communications Magazine*, vol. 32, no.9, pp. 40-48, 1994.

[7] J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-policy attribute-based encryption," in *Proc. of SP’07*, 2007, pp. 321-334.

[8] M. R. Garey and D. S. Johnson, “Computers and Intractability: A Guide to the Theory of NP-Completeness," W. H. Freeman & Co., San Francisco, 1979.

[9] A. Sahai and B. Waters, “Fuzzy identity-based encryption," in *Proc. of EUROCRYPT 2005*, 2005, pp. 457-473.

[10] V. Goyal, O. Pandey, A. Sahai et al., “Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. of ACMCCS 2006*, 2006, pp. 89-98.

[11] M. Lai, “Knapsack cryptosystems: The Past and the Future," in Technical report, Dept. of Info. and Comp. Sci., Univ. of California, 2001, http://www.ics.uci.edu/ mingl/knapsack.html

[12] R. C. Merkle and M. E. Hellman, “Hiding Information and Signatures in Trapdoor Knapsacks," in *IEEE Trans. on Info. Theory*, vol. IT-24, 1978, pp. 525-530.

[13] B. Chor and R. L. Rivest, “A Knapsack Type Public Key Cryptosystem Based on Arithmetic in Finite Fields," in *IEEE Trans. on Info. Theory*, vol. 34, no. 5, Sep. 1988, pp. 901-909.

[14] A. Shamir and R. E. Zippel, “On the Security of the Merkle-Hellman Cryptographic Scheme," in *IEEE Trans. on Info. Theory*, vol. IT-26, no. 3, May 1980, pp. 339-340.

[15] T. Okamoto, K. Tanaka, and S. Uchiyama, “Quantum public-key cryptosystems," in *Advances in Cryptology-CRYPTO*, 2000, pp. 147-165.

[16] G. Micheli and M. Schiavina, “A general construction for monoid-based knapsack protocols," arXiv:1311.1442v1 [cs.CR].

[17] A. Kate and I. Goldberg, “Generalizing Cryptosystems Based on the Subset Sum Problem," in *Int. J. Inf. Secur.*, vol. 10, 2011, pp. 189-199.

[18] T. Izu, J. Kogure, T. Koshiba, and T. Shimoyama, “Low-density attack revisited," in *Des. Codes Cryptogr.*, vol. 43, no. 1, 2007, pp. 47-59.

[19] P. Q. Nguyen and J. Stern, “Adapting density attacks to low-weight knapsacks," in *Advances in Cryptology-ASIACRYPT*, 2005, pp. 41-58.

[20] K. Omura and K. Tanaka, “Density attack to the Knapsack cryptosystems with enumerative source encoding," in *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. E87-A, no. 6, 2004, pp. 1564-1569.

[21] X. Hei and B. Song, “SHipher: Families of Block Ciphers based on non-Ablian group," Feb. 2014, http://eprint.iacr.org/2014/103.

[22] V. Lyubashevsky, “A history of lattice-based encryption", 2012, http://crypto.biu.ac.il/winterschool2012/slides/slides-barilan15.pdf

[23] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin, “Persona: An Online Social Network with User-Defined Privacy", 2009, in *Proc. of the ACM SIGCOMM’09*, pp. 135-146.