Secure Management of Personal Health Records by Applying Attribute-Based Encryption

Luan Ibraimi, Muhammad Asim, Milan Petković

Abstract—**The confidentiality of personal health records is a major problem when patients use commercial Web-based systems to store their health data. Traditional access control mechanisms have several limitations with respect to enforcing access control policies and ensuring data confidentiality. In particular, the data has to be stored on a central server locked by the access control mechanism, and the data owner loses control on the data from the moment when the data is sent to the server. Therefore, these mechanisms do not fulfill the requirements of data outsourcing scenarios where the third party storing the data should not have access to the plain data, and it is not trusted to enforce access policies. In this paper, we present a new variant of ciphertext-policy attribute-based encryption (CP-ABE) scheme which is used to enforce patient/organizational access control policies. In CP-ABE, the data is encrypted according to an access policy over a set of attributes. The access policy specifies which attributes a user needs to have in order to decrypt the encrypted data. Once the data is encrypted, it can be safely stored in an untrusted server such that everyone can download the encrypted data but only authorized users who satisfy the access policy can decrypt. The novelty of our construction is that attributes can be from two security domains: social domain (e.g. family, friends, or fellow patients) and professional domain (e.g. doctors or nurses).**

I. INTRODUCTION

IN recent times, the healthcare delivery has gradually extended from acute institutional care to outpatient care and home healthcare. Healthcare services can now be availed at a distance due to the advances in communication and information technology. Besides these, there are a number of initiatives for adoption of electronic health records (EHRs) from different governments around the world as well as from the private sector for adoption of personal health records (PHR). While EHR systems function to serve the information needs of health care professionals, PHR systems [1] capture health data entered by individuals and provide information related to the care of those individuals. There are number of web services that an individual can use to store her PHRs including the prominent examples of Microsoft HealthVault, Google Health or WebMD. They allow individuals to enter, store and share their own health data, upload health measurements from their devices, but also to import their health records from hospital EHR systems.

Despite numerous initiatives by industry and a number of standards under development to provide the interoperability across different PHR and EHR services, confidentiality of patient's health information remains a major obstacle with respect to the adoption of the PHRs by the individuals. Access-control mechanisms are very important to protect the confidentiality of electronic health records. They comprise a very large set of technologies, which include mechanisms to authenticate and authorize individuals or systems to access resources. Many consumers hesitate to upload their health data to commercial PHR systems since they do not trust access control mechanisms provided by these companies. Next to that, in modern healthcare, where a lot of IT functionality gets outsourced, patients are worried if their health data will be treated as confidential by companies running data centres.

The problem addressed in this paper is the confidentiality of PHRs. Patients records contains sensitive information such as details of a patient's disease, drug usage, sexual preferences, etc. Inappropriate disclosure of a record can change patient's life, and there may be no way to repair such harm financially or technically. Therefore, it is crucial to protect patient's health records when they are uploaded and stored in commercial Web-based systems. In this paper we propose a new variant of a ciphertext-policy attribute-based encryption (CP-ABE) scheme. CP-ABE is an encryption scheme which can be used to cryptographically enforce patient or organizational access control policies. Our scheme allows a patient to store her PHRs in an encrypted form on a commercial PHR system and share them securely with other users who belong to two different security domains: (a) professional domain (PD) - a group of healthcare providers e.g. doctors, nurses, or (b) social domain (SD) - her family, friends, or fellow patients.

The rest of this paper is organized as follows. In Section II we discuss how to enforce access policies using cryptographic techniques and give some background information about CP-ABE. In Section III we describe the proposed system architecture and introduce a new variant of a CP-ABE scheme. Section IV concludes the paper.

II. ENFORCING ACCESS POLICIES USING CRYPTOGRAPHY

Access control mechanisms can be grouped into four main classes: *discretionary*, *mandatory*, *role-based and attribute-based.* In these mechanisms the receiving end of the information must provide a set of credentials to the Access-Control Manager (ACM) who is responsible to

Luan Ibraimi is with the Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, Enschede, The Netherlands (e-mail: ibraimi@ewi.utwente.nl).

Muhammad Asim is with Philips Research, Eindhoven, The Netherlands (e-mail: muhammad.asim@philips.com).

Milan Petković is with Department of Mathematics and Computer Science, Eindhoven University of Technology, and with Philips Research, Eindhoven, The Netherlands (e-mail: milan.petkovic@philips.com).

enforce access control policies. The ACM checks whether user credentials satisfy the access control policy. If so, the user can read the resource, otherwise not. The drawback of this approach is that the data has to be stored on a central server locked by the access control mechanism. Furthermore, the data owner loses control on the data from the moment when the data is sent to the requester. This is also not suitable for data outsourcing scenarios where the third party storing the data should not have access on the plain data, and where the third party is not trusted to enforce access control policies (for example, patients hesitate to upload their PHRs to Google Health or Microsoft HealthVault). Therefore, recent proposals on enforcing access control policies exploit the use of cryptography to enforce access control policies. In such systems, there is no need for an ACM to check user credentials, and every user can get the encrypted data, but only users who have the right credentials (decryption keys) can decrypt the encrypted data.

A. Ciphertext-Policy Attribute-Based Encryption

CP-ABE is a type of attribute-based encryption scheme which can be used to enforce access policies cryptographically. In CP-ABE, the data owner encrypts the data according to an access control policy P defined over a set of attributes, and the receiving end can decrypt the encrypted data only if his secret key associated with a set of attributes satisfies P. For example, suppose Alice encrypts her data according to an access policy P= (a_1 AND a_2) OR a_3. Bob can decrypt the encrypted data only if his secret key is associated with a set of attributes that satisfy the access policy. To satisfy P**,** Bob must have a secret key associated with at least one from the following attribute sets: (a_1, a_2), (a_3) or (a_1, a_2, a_3). In general, CP-ABE scheme consists of four algorithms [2,3]:

- Setup algorithm **(MK, PK) ← Setup (1^k):** is run by the trusted authority or the security administrator. The setup algorithm takes as input a security parameter k and outputs a master secret key MK and a master public key PK.
- Key Generation algorithm **(SK) ← Key Gen (MK, ω)**: is run by the trusted authority, and takes as input a set of attributes ω and MK. The algorithm outputs a user secret key SK associated with the attribute set ω**.**
- Encrypt algorithm **(CT) ← Encrypt (m, PK, P)**: is run by the encryptor. The input of the algorithm is a message *m*, a master public key PK and an access control policy P, the output of the algorithm is a ciphertext CT encrypted under the access control policy P.
- Decrypt algorithm **(m) ← Decrypt (CT, SK):** is run by the decryptor. The input of the algorithm is a ciphertext CT to be decrypted and a user secret key SK. The output of the algorithm is a message *m*, if the attribute set of the secret key satisfies the access policy P under which the message was encrypted, or an error message if the attribute set of the secret key does not satisfies the access policy P under which the message was encrypted.

III. Our Proposal

We propose a variant of a CP-ABE scheme where the patient can encrypt her health records according to an access policy which has attributes issued by two trusted authorities: the trusted authority (TA_1) of the professional domain (PD) and the trusted authority (TA_2) of the social domain (SD). In our proposal the patient himself takes the role of TA_2. TA_1 will authenticate users of the professional domain, and issue secret keys based on their attributes, while the patient will use the reputation of the users of the social domain to generate appropriate secret keys. Our scheme is suitable for the healthcare setting and has the following benefits. 1) Allows a patient to store her PHRs in a protected form on an un-trusted commercial PHR server such that the access control policy is fully enforced. The patient encrypts the health data according to her access policy such that only the users who satisfy the access policy can decrypt the protected data. 2) Helps the patient to share securely their PHRs with users from different security domains. This is because the access policy under which the data is encrypted can contain attributes issued from different trusted authorities. 3) Removes the need for the patient to know the identity of the data recipient. The patient specifies only the attributes the recipient needs to have in order to access patient's data.

A. Proposed System Architecture

Fig. 1 depicts the architecture of the proposed system where the patient can securely manage her health records using the proposed CP-ABE scheme (the details of the proposed CP-ABE scheme are given in the next section).

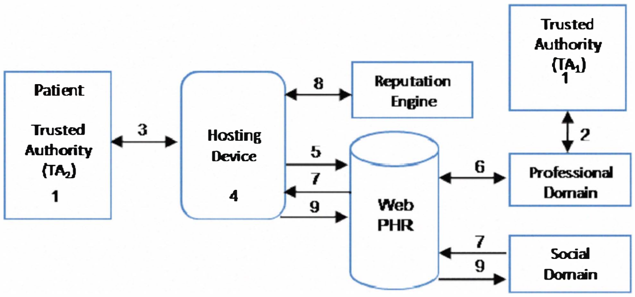

Fig.1. Architecture of the proposed system

In the following we explain the interactions that occur in the system. In the 1st step, TA_1 and TA_2 run the Setup algorithm of the CP-ABE scheme. In the 2nd step, users from the professional domain get their secret keys related to their attributes they possess from the TA_1. In the 3rd step, the patient uses a number of healthcare devices and creates measurement data and forwards them to the application hosting device which can be patient's personal computer, mobile phone or any other trusted device. In the 4th step, the application hosting device categorizes the measurement data. For example the measurement data $MD_{2/1}$, is the second measure taken by the patient which belongs to the data category 1, and the measurement data $MD_{1/3}$ is the first measure taken by the patient which belongs to the data category 3. Besides the fact that each measurement data belongs to a data category (DC), we assume that each measurement data belongs to an administrator category (AC). The hosting device encrypts the data according to an access policy P=P_1 OR P_2, which consists from two sub policies P_1 and P_2. Either P_1 or P_2 must be satisfied in order to decrypt the ciphertext. The first part of the access policy P_1 is intended for the social domain, therefore, the patient would be responsible to generate secret keys associated with attributes in P_1, and the second part of the policy P_2 is intended for the professional domain, therefore TA_1 would be responsible to generate secret keys associated with attributes in P_2. The structure of P_1 is: $P_1 = \hat{a}_{MD}$ OR $\hat{a}_{DC}$ OR $\hat{a}_{AC}$. This implies that in order to access the measurement data, the receiver must have a secret key SK_{MD} (associated with attribute $\hat{a}_{MD}$), or a secret key SK_{DC} (associated with attribute $\hat{a}_{DC}$), or the secret key SK_{AC} (associated with attribute $\hat{a}_{AC}$). Note that, P_1 contains attributes related to the resource (In CP-ABE a policy contains attributes which identify the user), in which the attribute $\hat{a}_{MD}$ identifies the measured data MD, the attribute $\hat{a}_{DC}$ identifies the data category DC, and the attribute $\hat{a}_{AC}$ identifies the administrator category AC. The motivation behind this categorization is that if a patient wants to allow the recipient to decrypt all measurement data belonging to the category 1, then the decryption (secret) key SK_{DC1} is given to the recipient. The secret key $SK_{MD1/1}$ can be used to decrypt only one measurement data $MD_{1/1}$, and the secret key SK_{AC} can be used to decrypt all measures, therefore, this key is known only to the patient, or to someone with whom the patient has a special relation. The structure of P_2 is dynamic and depends on patient preferences and contains attributes associated with users from the professional domain. In the 5th step, the encrypted data is sent to the web PHR repository. When the doctor from the professional domain wants to see patient data, it downloads the encrypted data from the server, and decrypts them locally using their secret key, as shown in step 6th. In the 7th step, the patient receives a request from a user from the social domain with whom the patient may have no pre-arranged trust relationship, to see his/her data. In the 8th step, the patient makes a decision regarding whether to issue or not the secret key to the requesting user from the social domain. The patient bases his decision on the requester's reputation score generated by the reputation evaluation engine. The reputation evaluation engine may take as input the ratings given by other users, and outputs the reputation of the requester [4]. Note that, the patient uses the reputation evaluation engine only when the requester does not have a digital certificate. If the requester has a digital certificate which shows his claimed identity, role or affiliation, then the patient generates the requester's secret key based on the received digital certificate. In the 9th step, the patient runs the key generation algorithm to generate the secret key associated with a set of attributes related to the document. The patient could generate different types of secret keys with different decryption power. If the user has high reputation, he will get a secret key with higher decryption power and vice versa. The requesting user uses the secret key to decrypt the encrypted data.

B. The Scheme

The proposed scheme uses bilinear maps between groups. Let G_0 and G_1 be two multiplicative groups of prime order p, and let g be a generator of G_0. A bilinear map $e : G_0 \times G_0 \rightarrow G_1$ satisfies the properties of: a) bilinearity - for all $u, v \in G_0$ and $a, b \in Z_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$, and b) non-degeneracy: $e(g,g) = 1$. G_0 is said to be a bilinear group if the group operation in G_0 and the bilinear map $e : G_0 \times G_0 \rightarrow G_1$ can be computed efficiently. We now present our proposed two-authority CP-ABE scheme.

Setup (1^k):

Run by TA_1. It selects a bilinear group G_0 of prime order p and generator g. Next to this, it selects randomly $\beta, x_1, x_2, \cdots, x_n \in Z_p$. For a set of attributes $\Omega_{PD} = \{a_1, a_2, \cdots, a_n\}$, it sets $T_j = g^{x_j} (1 \le j \le n)$. The public key is published as:

$$\mathbf{PK}_{PD} = \left(g, Y_{PD} = e(g,g)^{\beta}, \{T_j\}_{j=1}^{n} \right)$$

The components of the master secret key are:

$$\mathbf{MK}_{PD} = \left(\beta, \{x_j\}_{j=1}^{n} \right)$$

Run by TA_2. The bilinear group G_0 of prime order p and generator g is selected. It also selects randomly $\alpha, \hat{x}_1, \hat{x}_2, \cdots, \hat{x}_n \in Z_p$. For attribute set $\Omega_{SD} = \{\hat{a}_1, \hat{a}_1, \cdots \hat{a}_n\}$ which has three types of attributes: administrator category attribute, data category attributes, and measurement data attributes, it sets $\hat{T}_j = g^{\hat{x}_j} (1 \le j \le n)$. The public key is published as:

$$\mathbf{PK}_{SD} = \left(g, Y_{SD} = e(g,g)^{\alpha}, \{\hat{T}_j\}_{j=1}^{n} \right)$$

The components of the master secret key are:

$$\mathbf{MK}_{SD} = \left(\alpha, \{\hat{x}_j\}_{j=1}^{n} \right)$$

Key Gen (MK, ω):

Run by TA_1. The algorithm takes as input the attribute set $\omega_{Alice} = \{a_1 \ldots a_k\}$ which identify the requesting user (e.g. Alice). It picks a random value $f \in Z_p$ and computes the secret key for Alice which consists of the following components:

$$\mathbf{SK}_{\omega \mathrm{Alice}} = \left(D^{(1)} = g^{\beta - f}, D^{(2)} = \left\{ g^{\frac{f}{x_j}} \right\}_{a_j \in \omega \mathrm{Alice}} \right)$$

Run by TA_2. Suppose Bob, who is part of the social domain, asks for a secret key for the attribute set $\omega_{Bob} = \{\hat{a}_1 \ldots \hat{a}_k\}$ (Note that these attributes identify the resource and not the requesting user). The TA_2 picks a random value $r \in Z_p$ and computes the secret key which consists of the following components:

$$\mathbf{SK}_{\omega \mathrm{Bob}} = \left(\hat{D}^{(1)} = g^{\alpha - f}, \hat{D}^{(2)} = \left\{ g^{\frac{r}{\hat{x}_j}} \right\}_{\hat{a}_j \in \omega \mathrm{Bob}} \right)$$

Encrypt (*m*, PK, P):

As mentioned before, the scheme is designed to help patients to share securely their personal health records. Therefore, we describe only the encryption algorithm run by the patient.

Run by the patient (TA_2). In the proposed scheme, the patient encrypts the data according to the access policy P = P_1 OR P_2, where $P_1 = \hat{a}_{MD}$ OR $\hat{a}_{DC}$ OR $\hat{a}_{AC}$, and P_2 is the access policy over the attributes from the professional domain. To encrypt the measurement data *m,* the patient chooses at random $s \in Z_p$ and computes the following components:

$$\mathbf{CT} = \left(\begin{array}{l} C^{(1)} = g^s, C^{(2)} = m \cdot (Y_{SD})^s = m \cdot e(g,g)^{\alpha s} \\ C_j^{(3)} = \left\{ g^{\hat{x}_j s} \right\}_{\hat{a}_j \in P_1} \\ C_j^{(4)} = \left\{ g^{x_j s_i} \right\}_{a_j \in P_2} \end{array} \right)$$

The s_i values are generated using Benaloh and Leichter [5] secret sharing scheme. The scheme takes as input the secret to be shared *s*, and generates the shares s_i of the secret *s* in the following fashion:

- Transforms P_2 into an access tree where the interior nodes represent an AND or OR Boolean operators, and the leaf nodes represent attributes. The scheme, recursively, for each un-assigned non-leaf node does the following:

a) If the node is AND, it assigns a share s_i to each child node, such that the sum of all shares is s. Mark this node as assigned.

b) If the node is OR, it assigns the same value s to each child node. Mark this node as assigned.

In addition, the patient computes the helper data W which helps the users from the professional domain to decrypt the data: $W = e(g,g)^{\gamma s} = e(g,g)^{\alpha s} \cdot e(g,g)^{\beta s}$ thus, $\alpha = \gamma - \beta$.

At the end, the patient uploads the ciphertext CT along with the helper data W to his/her PHR.

Decrypt (CT, *W*, SK):

Run by a user from the PD. The decryption algorithm takes as input the secret key $\mathbf{SK}_{\omega_{\text{Alice}}}$ of user Alice, the ciphertext CT, and the helper data W. It checks if the secret key $\mathbf{SK}_{\omega_{\text{Alice}}}$ related to the attribute set ω_{Alice} satisfies the access policy P_2. If not, then it outputs $\perp$. If yes, then the algorithm chooses the smallest subset ω' that satisfies P_2 and computes:

(a) $$Z^{(1)} = \prod_{a_j \in \omega'} e\left(D^{(2)}, C_j^{(4)}\right) \cdot e\left(C^{(1)}, D^{(1)}\right) = e(g,g)^{\beta s}$$

(b) The measurement data *m*, is recovered by computing:

$$m = \frac{C^{(2)} \cdot Z^{(1)}}{W} = \frac{m \cdot e(g,g)^{\alpha s} \cdot Z^{(1)}}{e(g,g)^{\gamma s}} = m$$

Run by a user from the SD. The decryption algorithm takes as input the secret key $\mathbf{SK}_{\omega_{\text{Bob}}}$ of user Bob, and the ciphertext CT. To decrypt the ciphertext (assuming that the user has a secret key associated with at least one attribute from P_1), the decryptor first computes:

(a) $$Z^{(2)} = e\left(\hat{D}^{(2)}, C_j^{(3)}\right) \cdot e\left(\hat{D}^{(1)}, C^{(1)}\right) = e(g,g)^{\alpha s}$$

(b) The measurement data *m* is recovered by computing:

$$m = \frac{C^{(2)}}{Z^{(2)}} = \frac{m \cdot e(g,g)^{\alpha s}}{e(g,g)^{\alpha s}}$$

C. Security Intuition

To decrypt the ciphertext and reveal *m*, without having necessary attributes that satisfy the policy, the adversary has to compute $e(g,g)^{\beta s}$ or $e(g,g)^{\alpha s}$, and then divide the product of $C^{(2)}$ and $e(g,g)^{\beta s}$ with W, or divide $C^{(2)}$ with $e(g,g)^{\alpha s}$. Thus the adversary must compute $e(g,g)^{fs}$ or $e(g,g)^{rs}$, which can be computed by pairing the components of the secret key $D^{(2)}$ or $\hat{D}^{(2)}$ with the components of the ciphertext $C_j^{(4)}$ or $C_j^{(3)}$. To perform such operations the adversary has to use only the secret key components received in the key generation phase. Therefore, the adversary cannot compute $e(g,g)^{fs}$ or $e(g,g)^{rs}$ without having enough attributes which satisfy the access policy. The very important security property of our scheme is collusion safe, different users can not combine their secret keys and satisfy the access policy. This is because each user gets a secret key which is randomized with a different value (*r* and *f*). In a full security proof of our scheme we will follow the security model presented by Bethencourt et. al. [2] (in our security model the adversary can choose to decrypt a ciphertext associated with an access policy which contains attributes from two trusted authorities)

IV. Conclusion

The paper presents a new approach for secure management of personal health records which are stored and shared from an un-trusted web server. The CP-ABE scheme has shown to be a useful tool in a healthcare setting since the access policy is enforced by virtually associating the access control policy to the protected data. This removes the need for involving a trusted entity which has to enforce access policies. A possible future work is to provide a formal security proof for the proposed scheme.

References

[1] Markle Foundation. Connecting for Health -The personal health working group final report. 2003 July 1. [Online]. Available: http://www.connectingforhealth.org/resources/final_phwg_report1.pdf .

[2] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In D. Shands, editors, *Proceedings of the 28th IEEE Symposium on Security and Privacy (Oakland)*, pp. 321-334. Citeseer, 2007.

[3] L. Ibraimi, Q. Tang, P. Hartel, and W. Jonker. Efficient and provable secure ciphertext-policy attribute-based encryption schemes. In F. Bao, H. Li, and G.Wang, editors, *Proceedings of Information Security Practice and Experience*, volume 5451 of *LNCS*, pp. 1–12. Springer, 2009.

[4] Jøsang, R. Ismail, and C.Boyd. A survey of trust and reputation systems for online service provision. *Proceedings of Decision Support Systems,* pp. 618-644. Elsevier , 2007.

[5] J. Benaloh and J. Leichter. Generalized secret sharing and monotone functions. In S. Goldwasser, editors, *Proceedings of Crypto 1988,* volume 403 of *LNCS*, pp. 27-35. Springer, 1990.