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

Luan Ibraimi[#*1], Muhammad Asim[#2], Milan Petkovic[#3]

[#]*Philips Research Eindhoven*

The Netherlands

[1]luan.ibraimi@philips.com

[2]muhammad.asim@philips.com

[3]milan.petkovic@philips.com

[*]*Faculty of EEMCS, University of Twente*

The Netherlands

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, such as Role-Based Access Control, 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 requester. Therefore, these mechanisms do not fulfil 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 control policies. In this paper, we describe a new approach which enables secure storage and controlled sharing of patient's health records in the aforementioned scenarios. A new variant of a ciphertext-policy attribute-based encryption scheme is proposed to enforce patient/organizational access control policies such that everyone can download the encrypted data but only authorized users from the social domain (e.g. family, friends, or fellow patients) or authorized users from the professional domain (e.g. doctors or nurses) are allowed to decrypt it.**

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. An increasing number of telehealth services (remote patient monitoring, teleradiology, etc.) are becoming available. 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 that is striving towards adoption of personal health records (PHR). While EHR systems function to serve the information needs of health care professionals, PHR systems capture health data entered by individuals and provide information related to the care of those individuals. Besides providing the repository services to the individuals to store their health related data, PHR systems also include other functionality such as decision support functionality. All of these changes in the healthcare are expected to result in the following benefits: 1) reduction in the healthcare cost, 2) increasing the patient safety, 3) improving the quality of care, 4) involving and empowering patients to more actively manage their health. However there are still some challenges that have to be overcome before the full potential benefits of these new healthcare technologies are realized. One of the very important challenges PHR and EHR systems are facing is the confidentiality of patient's health information. Before going into the detail discussion of how to address the confidentiality issue, let us introduce PHRs. According to [1], the PHR is defined as:

"*An electronic application through which individuals can access, manage and share their health information, and that of others for whom they are authorized, in a private, secure, and confidential environment.*"

There are number of web services that an individual can use to store his/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. In realization of the full benefits of these solutions, interoperability plays a key role. Continua health alliance is an industry initiative in this direction which aims at providing interoperability standards across the whole eco system of personal healthcare.

Despite numerous initiatives by industry and a number of standards under development to provide the interoperability across different PHR and EHR services, security and privacy remain major obstacles with respect to the adoption of the PHRs by the individuals. Many consumers do not trust commercial companies to manage their PHRs. 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. To address these issues related to security and confidentiality of individual's health information, we propose a new variant of a ciphertext-policy attribute-based encryption (CP-ABE) scheme which enables patients to securely store and share their health records on a commercial PHR system. Our CP-ABE scheme allows the patient to store her PHRs in an encrypted form, and cryptographically enforces patient or organizational access policies. The scheme enables sharing of

patient's data with users from different domains, based on attributes certified by multiple authorities. For example, a patient can encrypt her data as such that it can be accessed by an individual from a social domain (e.g. his/her adult children) as well as from the professional domain (e.g. doctors or nurses).

The rest of this paper is organized as follows. In Section II we describe traditional access control mechanisms. In section III we discuss how to enforce access policies using cryptographic techniques, and give some background information about CP-ABE. In section IV we introduce our security requirements for securing personal health records. In section V we describe the proposed system architecture and introduce a new variant of a CP-ABE scheme. The last section concludes the paper.

II. Access Control

Access-control mechanisms comprise a very large set of technologies, which include mechanisms to authenticate and authorize individuals or systems to access resources. The main objective of access-control mechanisms is to provide data confidentiality. There are many authentication and authorization mechanisms, ranging from simple username-password combinations to federated role-based access schemes (where a claim of one institution that a certain person has some specific role, e.g. nurse, may be sufficient to provide that person access to some resource at another institution).

Access control mechanisms can be grouped into four main classes: *discretionary*, *mandatory*, *role-based and attribute-based*. In a discretionary access control (DAC) model [2] access is controlled based on user identities and a number of rules, called *authorizations. The authorization rules* explicitly state which subjects can execute which actions on which resources. After a user makes an access request, the access control is enforced based on the identity of the requester and on the authorization rules involving the requester and the requested object and action. In a mandatory access control (MAC) model [3], access is controlled based on mandated policies determined by a central authority. These policies are based on *classifications* associated with subjects and objects (*security levels* and a set of *categories*). In a role-based access control (RBAC) model [4], access is controlled based on user's roles and on rules defining which roles can execute which actions on which resources. Finally, in an attribute-based access control model (ABAC), access is controlled based on user's attributes. More details about the ABAC model will be given in the next section.

A. Attribute-Based Access Control

Attribute-based access control is an approach where the access decision is based on attributes (properties) of the resource, the requestor and the environment. This model provides flexibility and scalability that are essential in large distributed open environments where subjects are identified by characteristics. It can be implemented using digital credentials, that is, digitally signed assertions about credential owner by a credential issuer. More precisely, an attribute certificate can be used to support attribute-based systems. Such certificate contains attributes that specify access control information associated with the certificate holder. The decision to access a resource is based on the attributes in requestor's credentials. The attempts to provide a uniform framework for attribute-based access control and enforcement include the works of Bonati and Samarati [5] and Yu et al. [6].

XACML (eXtensible Access Control Markup Language) is an XML specification for expressing policies for data control over the Internet. It is intended to define the representation for rules that specify who, what, when, and how can access information. The ABAC model can be easily implemented using XACML due to its generic authorization architecture. Attributes can be related to the subject, resource and environment. XACML allows using the subject, resource and environmental attributes in the policy evaluation. This language supports the following features:

- Policies can be shared across different applications.
- Policies can be maintained in one or more locations.
- The application environment is isolated from the authorization process.
- Different access control mechanisms are supported, e.g. ABAC, RBAC, etc.

Based on the above features, XACML is regarded as the standard for solving complex access control problems, e.g. the access control in healthcare. The major components and actors of the XACML framework data flow are: 1) PEP (policy enforcement point) is responsible for implementing the policy evaluation decision, 2) PDP (policy decision point) is responsible for evaluation of the policies, 3) PAP (policy administrator point) writes the access control policies and makes them available to the PDP, 4) PIP (policy information point) returns the requested attributes to the PDP for the policy evaluation. Currently the OASIS Cross-Enterprise Security and Privacy Authorization (XSPA) technical committee is developing the XACML profile for healthcare enterprises. For more information and examples, please refer to the OASIS website [7].

III. Enforcing ABAC Using Cryptography

In the aforementioned access-control mechanisms the receiving end of the information must provide a set of credentials to the Access-Control Manager (ACM) who is responsible to 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 can decrypt the encrypted data.

In a public key cryptography, each user has a key pair: a private key which is kept secret, and a public key which is public. The encryptor encrypts the data using the public key of the recipient, and the recipient can decrypt the data using his secret key. In Public-Key Infrastructure (PKI), the encryptor has to use a digital certificate in order to be sure that the data is encrypted with the public key of the intended recipient. However, in ABAC model the access to data is based on user's roles/attributes, and PKI will not work for situations when the user does not know the exact identity of the recipient.

PKI is also not suitable to be applied to access control. The main problems are certificates and the key lifecycle management problem. For example, if a patient wants to send a secure email to multiple users in $Hospital_A$, the patient needs to know the digital certificate for each recipient, and then encrypt the same data many times using each user public key. Therefore, we need a crypto scheme which offers a more suitable solution for enforcing access policies based on user attributes.

A. Ciphertext-Policy Attribute-Based Encryption

Ciphertext-Policy Attribute-Based Encryption (CP-ABE) is a type of attribute-based encryption scheme which can be used to enforce ABAC cryptographically and address some of the aforementioned requirements. 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 the following four algorithms [8],[9]:

- 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.

IV. Addressed Problem

The problem addressed in this paper is the confidentiality of PHRs. Patients records contain 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 consider a scenario (see Fig. 1) where a patient, an entity in a distributed system, has some sensitive personal health records which she wants to store securely in a Web-based PHR, and share them 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 scenario stresses the need for a system which has to fulfil the following security requirements:

- Protect health records from network sniffers. Therefore, the data have to be encrypted before it is sent to the web PHR.

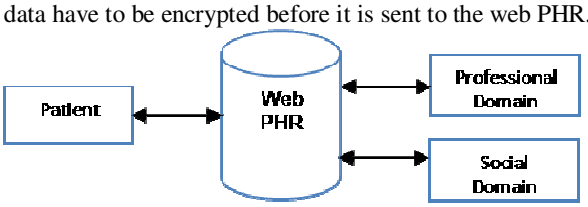

Fig. 1 System Architecture

- Protect health records from third parties who store PHRs. The third party managing web PHRs should not have access to the plain data.
- The access policy should be *sticked* to the encrypted data, such that only users who have a secret key associated with a set of attributes which satisfies the policy might be capable of decrypting it.
- Both users from the professional domain and users from the social domain need to be properly authenticated and authorized to access the data.

V. Our Solution

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). The patient himself could also take 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 might use the reputation of the users of the social domain to generate appropriate secret keys. For example, using our solution the patient can encrypt her health data such that a user who has the attribute *General Practitioner* issued from the TA_1 of the professional domain, or the attribute *friend* issued by the

patient can decrypt the encrypted data. Our scheme is suitable for the healthcare setting and has the following benefits:

- 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.
- 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.
- 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.

In the next section we demonstrate how to apply the proposed scheme to securely manage Personal Health Records (PHRs).

A. *Proposed System Architecture*

Fig. 2 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). In the following we explain the interactions that occur in the system.

- In the 1st step, the trusted authority (TA_1) from the professional domain and the patient (TA_2) from the social domain run the Setup algorithm of their 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 measurement taken by the patient which belongs to the data category 1, and the measurement data $MD_{1/3}$ is the first measurement 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.

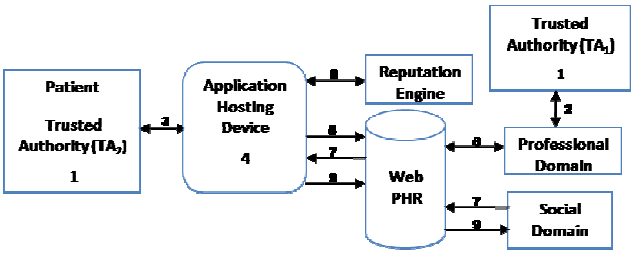

Fig. 2 Architecture of the proposed system

The structure of P_1 is as follows:

$$P_1 = \hat{a}_{MD} \text{ OR } \hat{a}_{DC} \text{ OR } \hat{a}_{AC}$$

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 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.
- From the stored data, eventually a patient could see health trends and begin to learn what lifestyle or other behaviours are affecting her glucose levels or blood pressure. This kind of information would be useful to a doctor to determine if the patient should be on a certain medication, if dosage should be adjusted, if the medication is having minimal or no affect. 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 her 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 [10]. 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. Our Construction

In this section, first we present few facts about the bilinear maps, and then we give the construction of our proposed scheme.

Bilinear Maps: 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 following properties [11]:

- Bilinear: for all $u, v \in G_0$ and $a, b \in Z$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
- Non-degenerate: $e(g, g) = 1$.

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. Note that the map is symmetric since $e(g^a, g^b) = e(g^b, g^a) = e(g, g)^{ab}$. For more details on bilinear pairing we refer the readers to [12]. We now present our proposed multi-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:

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

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:

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

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:

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

The s_i values are generated using Benaloh and Leichter [12] 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:

$$W = e(g,g)^{\gamma s} = e(g,g)^{\alpha s} \cdot e(g,g)^{\beta s} \text{ thus, } \alpha = \gamma - \beta.$$

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

In this section, we briefly discuss the security of the proposed scheme. 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 that 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 [8] (in our security model the adversary can choose to decrypt a ciphertext associated with an access policy which contains attributes from two trusted authorities), and use the generic group model, introduced by Shoup [13]. The proof in this model is based on the fact that the discrete logarithm and the Diffie-Hellman problem (DHP) are hard to solve as long as the order of the group is a large prime number. In generic group model, group elements of G_0 and G_1 are encoded as unique random strings, in such a way that the adversary cannot test any property other than equality. To perform group operations, the adversary has access to oracles which perform group operations in G_0 and G_1, and to the oracle which perform pairing operations. The adversary to break the scheme must be able to exploit the mathematical properties of the groups used in the scheme.

VI. Conclusion

This paper presents a new approach for secure management of personal health records which are stored and shared from an un-trusted web server. We gave an overview of various access control mechanisms and analyze which mechanisms are most useful for scenarios where data is stored on a commercial PHR systems or it is outsourced to a third party data center. Traditional access control mechanisms as well as traditional encryption techniques are not suitable to be used in these scenarios. The CP-ABE scheme has shown to be more useful 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.

The core contribution of this paper is the construction of a multi-authority of CP-ABE scheme. The proposed scheme allows patients to encrypt the data according to an access policy over a set of attributes issued by two trusted authorities. The scheme does not require the presence of a central authority to coordinate the work of the trusted authorities, and may support very expressive access policies, including policies written in disjunctive normal form (DNF) or conjunctive normal form (CNF). A possible future work is to formally provide a security proof for the proposed scheme.

References

[1] Connecting for Health. The personal health working group final report. 2003 July 1.

[2] G.S. Graham and P. J. Denning, "Protection-principles and practice," *In AFIPS Proc. of the Spring Jt. Computer Conference*, Montvale, NJ, USA, 1972.

[3] S. Jajodia and R. Sandhu , "Toward a multilevel secure relational data model," *In Proc. of the ACM SIGMOD Conference on Management of Data*, Denver, CO, USA, 1991.

[4] D. F. Ferraiolo and D. R Kuhn, "Role Based Access Control," *In 15th National Computer Security Conference.* http://csrc.nist.goc/rbac/, 1992

[5] P. Bonatti and P. Samarati, "A unified framework for regulating access and information release on the web," *Journal of Computer Security*, vol. 10, no. 3, pp. 241-272, 2002.

[6] T. Yu, M. Winslett, and K. E. Seamons, "Prunes: An Efficient and complete strategy for automated trust negotiation over internet," *In Proc. of the 7th ACM Conference on Computer and Communications Security*, Athens, Greece, 2000.

[7] http://www.oasis-open.org/committees/download.php/2406/oasis-xacml-1.0.pdf. Accessed in 6 July, 2009.

[8] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," *IEEE Symposium on Security and Privacy*, pp. 321–334, 2007.

[9] L. Ibraimi, Q. Tang, P. Hartel and W. Jonker, "Efficient and provable secure ciphertext-policy attribute-based encryption schemes," *Lecture Notes in Computer Science*. Berlin, Germany: Springer, pp.1-12, vol. 5451, 2009.

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

[11] D. Boneh, and M.Franklin. "Identity-based encryption from the Weil pairing," *Lecture Notes in Computer Science.* Berlin, Germany: Springer, pp. 586-615, vol. 2139, 2001.

[12] J. Benaloh and J. Leichter, "Generalized secret sharing and monotone functions," *In Proc. on Advances in cryptology.* Springer-Verlag New York, pp. 27-35, 1990.

[13] V.Shoup, "Lower bounds for discrete logarithms and related problems," *Lecture Notes in Computer Science*. Berlin, Germany: Springer, pp. 256-266, vol. 1233, 1997.