Three Levels of Access Control to Personal Health Records in a Healthcare Cloud

Gabriel Sanchez Bautista
and Ning Zhang

School of Computer Science
The University of Manchester
Manchester M13 9PL, United Kingdom
Email: {sanchezg, nzhang}@cs.man.ac.uk

***Abstract*—We present a novel access control framework (3LAC), which supports multiple levels of access privileges. 3LAC is aimed to tackle the privacy issues in existing access control solutions to access patients' records in cloud computing environments. In 3LAC, we propose an access control framework that extends the Ciphertext-Policy Attribute-Based Encryption (CP-ABE) with the integration of secret sharing in a way that different number of shares are needed to reconstruct a level-key. In this research work, we introduce the idea of level-keys. Level-keys are used to authenticate users when requesting the generation of private keys to decrypt patients' data. Level-keys are split into different shares and users will request the shares to different level-key authorities (LKAs). The number of shares needed to reconstruct the level-key depends on the level of access privilege of the user. As the level of access privilege increases, the number of shares needed also increases. In a healthcare cloud context, 3 levels of access privileges have been identified, L_1- Access to de-identified data-objects, L_2- Access to individual data-objects and L_3- Access to a large set of data-objects of a patient. The 3LAC framework incorporates a CP-ABE based 3-level access control model and the design of 4 protocols: 1- Upload data-object (UDO) , 2- Share acquisition (SAc), 3- Private key acquisition (PrKAc) and 4- Access to data-objects (ADO).**

Keywords–Privacy; eHealth; Attribute-based encryption; Secret sharing; Access control; Multilevel.

I. INTRODUCTION

The implementation of Personal Health Records (PHRs) over cloud computing environments assumes advantages to the access of data. Different users may access data anytime and anywhere in a flexible and scalable manner. Advantages of cloud computing are scalability, flexible access to data and on-demand use of resources [1].

However, cloud computing raises an important issue in terms of privacy. Typically, PHRs are stored on the servers of the cloud service providers and patients have no control on how their records are accessed [2]. Patients have to trust that the cloud service providers only grant access to users for legitimate access purposes. Unauthorized access of data can be used to collect information of patients for other purposes such as marketing, commercial or research without the patients' consent.

Concerns about privacy in access to PHRs have been raised by different organizations and communities around the world. For example, the U.S. Department of Health and Human Services (HHS) has issued the Health Insurance Portability and Accountability Act (HIPAA) [3], which provides a set of standards and regulations for the privacy protection of PHRs.

PHRs are electronic records of patients' health data [4]. They are different from Electronic Health Records (EHRs) in the sense that PHRs are patient-centric while EHRs refer only to health records transmitted electronically [5]. PHRs are the aggregation of data-objects generated by different healthcare providers that may collaborate and contribute to the generation of data [6]. In this context, there may be different purposes to access patients' data. It may be for regular medical treatment, for secondary use (e.g., marketing or research) or for emergency situations in which access to a large set of data-objects of a patient may be needed. A data-object is defined as the most granular piece of data in a patient's records.

The rest of this paper is organised as follows: Section II presents the privacy issues of PHRs in a healthcare cloud context. Section III specifies the design requirements. Section IV gives the notations and introduces the high-level ideas in the design of 3LAC. Section V describes the 3LAC framework. Section VI presents the conclusion and future work.

II. PRIVACY ISSUES

Privacy issues of PHRs in a healthcare cloud are described as follows.

- **Unauthorised access to patients' PHRs.** Healthcare cloud service providers are responsible for the management and storage of patients' health records in the cloud. PHRs are usually processed on servers and machines in which patients have no control. This increases the possibility of theft or misuse of their data. Access to sensitive information on patients' records represents a risk of privacy when data is accessed by unauthorized users.
- **Lack of fine-grained access control.** Patients' records are often treated as a unitary piece of data. When users are granted access, they can usually see all the information on the patient's records even if the records contain information they do not need to perform their job functions. This constitutes a privacy issue. For example, a patient may not want to disclose certain information (e.g., sexual abuse) when it is not necessary for her current treatment.
- **Linkability to patients' identity when PHRs are accessed for secondary purposes.** Data contained in a patients records can become a source of information that may be used to track the identity of the patient for malicious purposes. If identifying information is

 ISBN: 978-1-61208-470-1

not removed, it can represent a risk to the privacy of patients' data. We have identified that there are certain access purposes in which users do not need to know the identity of the patient to perform their job functions.

III. REQUIREMENTS

This section specifies the set of requirements that 3LAC is aimed to address.

- **(R1) Patients (i.e., data owners) should be in the position to decide who accesses their data.** Patients should be able to specify who accesses their data and for what specific purposes.
- **(R2) Support fine-grained access control.** Access to patients' data should be fine-grained. Users should access only the portion of data that they need to perform their job functions.
- **(R3) Support access to data in emergency situations.** Access in emergency situations should be granted to the most complete set of data of a patient. As this is a high-privileged access, there should be an increased level of protection.
- **(R4) Support access to data for secondary use.** Access to patients' data for secondary use should be granted. As this is not a high-privileged access, it should not require a rigorous level of protection that may add extra computational costs to the system.
- **(R5) Support removal of access privileges.** Patients should be in the position to remove access privileges when they consider access to a certain data-object is no longer needed.
- **(R6) Scalability.** The access control should be scalable in terms of key management and distribution. It should support a large number of users requesting access to the data-objects.

IV. DESIGN PRELIMINARIES

A. Notations

The notations used in the design of the 3LAC framework are given in Table I.

TABLE I. NOTATIONS.

Notation	*Meaning*
LK_i^1	User i's level-key 1
LK_i^2	User i's level-key 2
LK_i^3	User i's level-key 3
S_i^{2s1}	Share 1 of LK_i^2
S_i^{2s2}	Share 2 of LK_i^2
S_i^{3s1}	Share 1 of LK_i^3
S_i^{3s2}	Share 2 of LK_i^3
S_i^{3s3}	Share 3 of LK_i^3
$SKGA$	Private key generation authority
$RLKA$	Root level-key authority
LKA_1	Level-key authority 1
LKA_2	Level-key authority 2
LKA_3	Level-key authority 3

As can be seen in Table 1., in 3LAC there is a private key generation authority, a root level-key authority and 3 non-root level-key authorities. Similarly, it can be seen that level-keys are classified into three different levels and based on the level, a level-key may be split into different shares.

B. High-level ideas

In this section we describe the high-level ideas in the design of 3LAC.

- **3LAC supports multiple access privilege levels.** 3LAC supports different levels of access privileges for users requesting access to data-objects. Based on the analysis of different use-case scenarios, we have identified that 3 are the levels necessary to classify the access purposes that users have when requesting access to patients' health records. At the lowest level, there should be access to PHRs for secondary use purposes. At a medium level, there should be access to PHRs for regular medical treatment. At a higher level, there should be access to PHRs in emergency situations in which access to the most complete set of data-objects of a patient may be desired [7]. Based on the analysis of these different scenarios, in this work we propose 3 levels of access privileges, which are: L_1- Access to de-identified data-objects, L_2- Access to individual data-objects and L_3- Access to a large set of data-objects of a patient. Similarly, each level supports access to data-objects with different levels of sensitivity. L_1 supports access to data-objects with low level of sensitivity. L_2 supports access to data-objects with medium level of sensitivity. L_3 supports access to data-objects with high level of sensitivity.
- **Users are divided into different user-groups {G1, G2, G3}.** Each group corresponds to a level of access privilege. Users are intended to access data-objects at the assigned and the lower privilege levels. For example, users of G1 should access data-objects at L_1. Users of G2 should access data-objects at L_2 and L_1. Users of G3 should access data-objects at L_3, L_2 and L_1.
- **Level-keys are split into shares.** The level-keys (LK_i^1, LK_i^2, LK_i^3) are used to authenticate users according to their access privilege level. LK_i^1 is generated by LKA_1. LK_i^2 is generated by $RLKA$. LK_i^2 is split into two shares (S_i^{2s1} and S_i^{2s2}). S_i^{2s1} is distributed to LKA_1 and S_i^{2s2} to LKA_2. LK_i^3 is generated by $RLKA$. LK_i^3 is split into three shares (S_i^{3s1}, S_i^{3s2} and S_i^{3s3}). S_i^{3s1} is distributed to LKA_1, S_i^{3s2} to LKA_2, and S_i^{3s3} to LKA_3.
- **The control to the access privilege level is embedded into the level-keys.** Level-keys are reconstructed from shares obtained by different $LKAs$. The number of shares needed is based on the access privilege level. As the level of access privileges increases, the number of shares needed also increases. A level-key of L_1 (LK_i^1) does not need any share, LK_i^1 itself must be obtained. A level-key of L_2 (LK_i^2) needs two shares (S_i^{2s1} and S_i^{2s2}) to be reconstructed. A level-key of L_3 (LK_i^3) needs three shares, (S_i^{3s1}, S_i^{3s2} and S_i^{3s3}) to be reconstructed. In this way, to access higher sensitivity data-objects, one has to obtain more shares, and for the acquisition of each share, there will be an authentication process. This makes the impersonation and unauthorised access to more sensitive data-objects more difficult. Figure 1.

below illustrates the number of shares needed based on the group of access privileges of users.

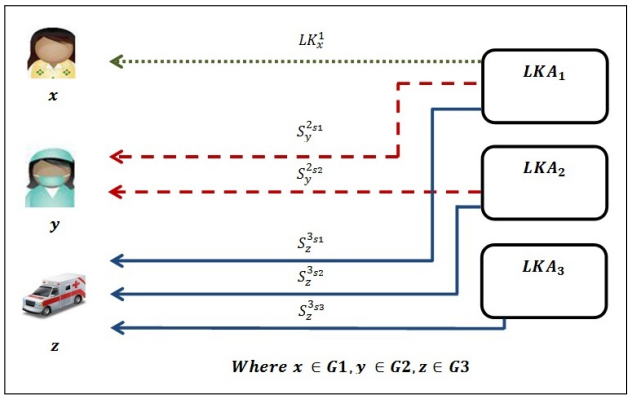

Figure 1. Different users obtaining shares from different LKAs

In the example given in Figure 1., x is a user of G1, y is a user of G2 and z is a user of G3. They obtain shares from the level-key authorities LKA_1, LKA_2 and LKA_3 in order to reconstruct their corresponding level-key that will be used for authentication when requesting the issuance of private keys. Private keys are used to decrypt patients' data-objects.

V. A Novel 3-Level Access Control Framework (3LAC)

3LAC is an access control framework that is formed of a CP-ABE-based 3-level access control model and the design of 4 protocols. 3LAC provides an access control framework that can be implemented in different applications that provide access to patients' personal health records. The front-end of 3LAC will depend on which is the application that makes use of the 3LAC framework. More details of 3LAC are given in the following subsections.

A. A CP-ABE based 3-level access control model

It is a novel access control model that supports 3 levels of access privileges (L_1, L_2, L_3). Each level supports access to data with different levels of sensitivity. L_1 supports access to low sensitive data, L_2 supports access to medium sensitive data, and L_3 supports access to high sensitive data.

3LAC also supports fine-grained access control because this is based on CP-ABE [8], an encryption scheme in which patients define access policies based on attributes to specify who has privileges to access which data-objects. Users' attributes should satisfy the access policy defined by the patient in order to decrypt the data-object. The reason it supports fine-grained access control is because users can be assigned any number of attributes. This permits a more detailed description of their identities, thus a more fine-grained access control.

Similarly, 3LAC supports revocation of access privileges because the patient can redefine the access policy at any time in order to specify the new attributes that users must have to decrypt the data-objects.

3LAC supports scalability as access to data-objects is made more efficient for different groups of users while at the same time the privacy of the data-objects is protected such that the level of protection increases with the level of sensitivity of the data-objects.

The following list presents the architectural components of 3LAC.

- $RLKA$: The root level-key authority. It is responsible to generate the level-keys (at L_1 and L_2) and split them into shares. Then, it distributes the shares to the different non-root level-key authorities.
- LKA_1: The level-key authority 1. It generates LK_i^1 and obtains the shares $S_i^{2_{s1}}$ and $S_i^{3_{s1}}$ from $RLKA$.
- LKA_2: The level-key authority 2. It obtains the shares $S_i^{2_{s2}}$ and $S_i^{3_{s2}}$ from $RLKA$.
- LKA_3: The level-key authority 3. It obtains the share $S_i^{3_{s3}}$ from $RLKA$.
- $SKGA$: The private key generation authority. It is a trusted authority responsible for generating the private keys (i.e. decryption keys) for different users based on their attributes.
- CA: The certification authority. It is a trusted authority that is responsible for signing the users' digital certificates. The public keys of the users are certified by this authority [9].
- AA: The attribute authority. It is a trusted authority that is responsible for gathering all the users' attributes. This authority passes the users' attributes to the CA. The CA takes these attributes to include them in the digital certificate of the user.
- DP: The data provider. It is the cloud service provider where the data-objects of patients are stored.

The architectural components of 3LAC are used in 5 phases as described below.

- **1: Initialisation**. In this phase, the user makes a request to the CA in order to obtain a digital certificate. The CA certifies the public key of the user and signs the digital certificate. The digital certificate also includes the attributes of the user in the extension field. The attributes are obtained from the AA. Similarly, it is during initialisation that the $RLKA$ distributes the shares to the different non-root level-key authorities (i.e., LKA_1, LKA_2, LKA_3).
- **2: Shares Acquisition**. In this phase, the user makes a request to the non-root level-key authorities in order to obtain the shares needed to reconstruct his level-key. Users of $G1$ make a request to LKA_1. Users of $G2$ make a request to LKA_1 and LKA_2. Users of $G3$ make a request to LKA_1, LKA_2 and LKA_3. Once the shares have been obtained, the user can reconstruct his level-key, which will be used for authentication during private key acquisition.
- **3: Private Key Acquisition**. In this phase, the user makes a request to the $SKGA$ in order to be issued a private key. The $SKGA$ will send a challenge to the user in order to prove that the user has been able to reconstruct his level-key. The user then responds to the challenge and encrypts the response by using his level-key. Then, the $SKGA$ decrypts the challenge response by using the level-key, which is symmetrical

 ISBN: 978-1-61208-470-1

and known by both the user and the $SKGA$. Once the $SKGA$ verifies the user knows his level-key, the $SKGA$ generates and encrypts the private key of the user. The private key is encrypted by using the level-key. In other words, the level-key is used to distribute the private key to the user. Then, by using his level-key, the user can decrypt his private key, which will be used to decrypt data-objects.

- **4: Patient uploading data-objects**. This is when patients (i.e., data owners) request to upload data-objects to the databases. Patients make a request to the data provider (DP). DP is the entity that stores data on its databases. The data-objects uploaded are encrypted with an access policy that specifies the attributes that users must have to decrypt them. The access policy is specified by the patient. Patients may specify an access policy for each data-object. However, patients can also encrypt a large set of data-objects under one package. In this case, one access policy can be specified for a large set of data-objects of a patient. In other words, it depends if the patient desires fine-grained or coarse-grained access for his data-objects, so he can define an access policy per data-object or per many data-objects under one access policy.
- **5: Users requesting data-objects**.
  This is when users request to access data-objects of patients. Data-objects given to users are encrypted. However, it will depend if the user has a private key with the attributes necessary to satisfy the access policy embedded in the cipher-text. If this is true, the user can decrypt the data-object.

Three levels of access privileges

For proof of concept in this work, the three levels of access privileges are defined as follows.

L_1**: Access to de-identified data-objects:** L_1 supports access to data-objects for secondary use purposes (e.g., marketing or commercial). L_1 is defined as the lowest privilege level. Users of G1 are intended to access data-objects at L_1. Additionally, users of higher privilege levels are allowed to access L_1. Users of G1 need LK_i^1 to authenticate and obtain the private key. During private key generation, $SKGA$ verifies if LK_i^1 corresponds to or is higher than the user-group of the requesting user (i.e., user-group G1). Access at L_1 is granted only to de-identified data-objects. De-identified data-objects are in a separate database (DB^d) from the database that contains the original data-objects (DB). Data-objects at this level are assumed to be de-identified because users of G1 do not need to know the identities of the patients. However, in cases that users of G1 need to access the original data-objects, they may request access to data-objects at L_2. To accomplish this, they need to obtain 2 new shares to reconstruct the level-key for L_2 (LK_i^2).

L_2**: Access to individual data-objects:** L_2 supports access to data-objects for regular medical treatment. L_2 is defined as the medium privilege level. Users of G2 are intended to access data-objects at L_2. Additionally, users of a higher privilege level are allowed to access L_2. Access is granted to the specific data-objects that users need to perform their job functions. Access at L_2 is to the original data-objects database (DB). Users at L_2 need two shares ($S_i^{2_{s1}}$ and $S_i^{2_{s2}}$) to reconstruct LK_i^2. During private key generation, $SKGA$ verifies if LK_i^2 corresponds to or is higher than the user-group of the requesting user (i.e., user-group G2). If it is true, LK_i^2 can be used to distribute the private key. If users of G2 desire to access data-objects at L_3, they need to obtain 3 new shares to reconstruct the level-key for L_3 (LK_i^3).

L_3**: Access to a large set of data-objects of a patient:** L_3 supports access to data-objects in emergency situations. L_3 is defined as the highest privilege level. Users of G3 are intended to access data-objects at L_3. For a proof of concept in 3LAC, there is no higher privilege level than L_3. Access at L_3 is to the original data-objects database. Users at L_3 need 3 shares ($S_i^{3_{s1}}$, $S_i^{3_{s2}}$ and $S_i^{3_{s3}}$) to reconstruct LK_i^3. During private key generation, $SKGA$ verifies if LK_i^3 corresponds to the user-group of the requesting user (i.e., user-group G3). If it is true, LK_i^3 can be used to distribute the private key.

B. 3LAC protocol designs

In 3LAC, there are 4 different protocols: 1- UDO, 2- SAc, 3- SKAc and 4- ADO. The following list describes the protocols.

1) *Upload data-object protocol (UDO).* It is executed when a patient requests to encrypt and upload a data-object. The patient makes a request to the data provider which is responsible to accept requests for encryption of patients' data-objects and upload them to the corresponding database.
2) *Shares Acquisition protocol (SAc).* It is executed when a user requests to obtain a share from a LKA. The user may need to request different shares to different $LKAs$ depending on how many shares the user has to obtain in order to reconstruct his level-key. This protocol can be executed between any user and any LKA.
3) *Private key acquisition protocol (SKAc).* It is executed when a user makes a request to obtain a private key. The request is sent to the SKGA, the authority that is responsible for generating the private keys for users so they can decrypt data-objects of patients. In order to obtain a private key, the requesting user needs to prove he has the level-key (LK) that he was able to reconstruct from the shares. The LK has the level of access control embedded within it as the number of shares needed to reconstruct it depends on the access privilege of the user which is specified on his user-group. The level-key is used to encrypt and distribute the private key generated. In other words, if the user knows his level-key then he can decrypt his private key.
4) *Access data-objects protocol (ADO).* It is executed when a user requests access to the data-objects of a patient. The request is sent to the data provider which is responsible to authorise access to the data-objects stored in the databases.

VI. CONCLUSION

In this paper, we presented a novel access control framework (3LAC) which supports different levels of access privileges: L_1- Access to de-identify data-objects, L_2- Access to

individual data-objects and L_3- Access to a large set of data-objects of a patient.

In 3LAC, we introduce the concept of level-keys and the idea that the level of access control for a user is embedded into his level-key. 3LAC extends CP-ABE with the integration of secret sharing [10]. With the integration of secret sharing as the level of access privilege increases, the number of shares needed also increases in order to reconstruct a level-key. With a level-key a user is authenticated when requesting the issuance of a private key. A private key is used to decrypt a patient's data-objects. In 3LAC, privacy protection is increased because for the acquisition of each share there will be an authentication process. This makes the impersonation and unauthorised access to more sensitive data-objects more difficult.

Future work includes the implementation and evaluation of 3LAC.

ACKNOWLEDGMENT

This research is supported by the National Council of Science and Technology Mexico (CONACYT).

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica et al., "A view of cloud computing," Communications of the ACM, vol. 53, 2010, pp. 50–58.

[2] H. Takabi, J. B. Joshi, and G.-J. Ahn, "Security and privacy challenges in cloud computing environments," IEEE Security & Privacy, vol. 6, 2010, pp. 24–31.

[3] "U.S. Department of Health and Human Services, The Health Insurance Portability and Accountability Act," URL: http://www.hhs.gov/ocr/privacy/hipaa/understanding/special/healthit/phrs.pdf [retrieved: 03,2016].

[4] C. C. Lee, P. S. Chung, and M. S. Hwang, "A Survey on Attribute-based Encryption Schemes of Access Control in Cloud Environments," International Journal of Network Security, vol. 15, 2013, pp. 231–240.

[5] C. Wang, X. Liu, and W. Li, "Implementing a Personal Health Record Cloud Platform Using Ciphertext-Policy Attribute-Based Encryption," 4th International Conference on Intelligent Networking and Collaborative Systems (INCoS), 2012, pp. 8–14.

[6] A. Abbas and S. U. Khan, "A Review on the State-of-the-Art Privacy-Preserving Approaches in the e-Health Clouds," Biomedical and Health Informatics, IEEE Journal of, vol. 18, 2014, pp. 1431–1441.

[7] S. R. Reti, H. J. Feldman, S. E. Ross, and C. Safran, "Improving personal health records for patient-centered care," Journal of the American Medical Informatics Association, vol. 17, 2010, pp. 192–195.

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

[9] N. Leavitt, "Internet security under attack: The undermining of digital certificates," Computer, vol. 44, 2011, pp. 17–20.

[10] A. Shamir, "How to share a secret," Communications of the ACM, vol. 22, 1979, pp. 612–613.

 ISBN: 978-1-61208-470-1