A New Architecture for Secure Storage and Sharing of Health Records in the Cloud Using Federated Identity Attributes

Lucas de Melo Silva, Roberto Araújo, Felipe Leite da Silva, Eduardo Cerqueira
Laboratório de Segurança e Criptografia Aplicada (LabSC)
Department of Computer Science
Universidade Federal do Pará
Belém, Brazil
{lucasmelo, rsa, fsilva, cerqueira}@ufpa.br

***Abstract*—Cloud computing is a technological service that has become a trend. This paradigm adds many benefits to storage of personal health records (PHR) and electronic health records (EHR), such as availability and on-demand provisioning. It also facilitates sharing of health records among doctors, family members, friends and clinicians in general. However, this technology increases the risk of leaking sensitive health data. Aiming at mitigating this problem, we present here a new architecture that takes advantage of identity federation for secure storage and sharing of PHR and EHR in the cloud environment.**

***Index Terms*—cloud computing, security, privacy, health records, federated identity, attribute-based encryption**

I. Introduction

The deployment and maintenance of large data storage infrastructures is costly. As a result, data storage is usually outsourced to third party providers. Cloud computing is a paradigm that provides this kind of service. It became a trend and is one of the fastest growing technological services [1].

In e-health environments, cloud computing may lower costs and increase system availability. This is the case of electronic health records (EHR) storage scenarios. EHRs are alternatives to the traditional paper health records and store patient's medical data, such as laboratory results, prescriptions, diagnoses, and medical history. It is easier to manage, update, share, and access. Healthcare providers or clinicians manage EHRs, controlling who can access and edit its data.

Differently from EHR, personal health records (PHRs) are patient centered. This means that patients themselves can input information in their records as well as import from the traditional EHR. The use of PHR may help patients to have a better vision of their health history and progress, centralize information when consulting with different doctors, and share health information with family and friends. In PHR, the patient controls over who can access and edit information on his records. PHR can also benefit from storing its data in the cloud. In fact, commercial systems of these outsourced service providers have emerged in the health context, e.g., Microsoft Health Vault [2].

Cloud computing, however, has many challenges with regards to data storage security. In particular, the provision of safe storage and the protection of user's privacy. This is directly related to health records as the cloud may store EHR or PHR data. Privacy and system security is a major barrier to the e-health development in outsourced storage [3].

A malicious cloud provider could undermine user privacy in an EHR and PHR scenario. It could obtain information from stored health records to use as input for advertisement, data mining, business research, etc. For instance, with the help of a cloud provider, a bank could obtain a client's health history to evaluate a possible loan.

Traditionally, cloud computing manages and authenticates its users in order to provide its services. This is awkward, requiring different user accounts and credentials for each service as well as different databases for managing these accounts. As an alternative, cloud services can employ a federated identity management (FIM). This technology outsources user identity and attribute management and delivers single sign-on (SSO). FIM also allows the creation of collaborative networks between institutions that have some kind of affinity and want to share services between its users. Examples of these networks are academic associations, such as the Federated Academic Community (CAFe) [4] of Brazil's National Research Network (RNP) and InCommon [5] of United States' Internet2 network; and federated networks for e-governance, as initiatives in New Zealand, Australia, Canada and the United States [6].

Integrating cloud and FIM affords benefits from both paradigms to health record storage. This increases availability as well as makes collaboration and management easier. In addition, it permits patients to change affiliation (e.g., change healthcare provider) while maintaining their health records.

Taking into account both these technologies, this paper introduces a new architecture for secure storage and sharing of PHR and EHR in clouds. It makes possible a secure cloud storage with file sharing managed by the user in accordance with an access policy. Also, it allows access revocation and does not require users to continuously store encryption keys. To the best of our knowledge, there are no other architectures that leverage FIM and cloud to achieve these benefits.

This paper is organized as follows. The next section presents

978-1-4799-6644-8/14/$31.00 ©2014 IEEE

the technologies necessary for our proposal. After that, Section III introduces related work. Section IV shows the new architecture. Then, Section V sketches a discussion of the new proposal. Finally, Section VI concludes this work.

II. Preliminaries

Our solution is based on two technologies: Federated Identity Management and Attribute-Based Encryption. In order to clarify these technologies, we now present them.

A. Federated Identity Management

Federated Identity Management (FIM) is a paradigm where several organizations share a common set of policies, practices and protocols to manage digital identities in a collaborative network. It performs as an authority for identity and attributes of users from various organizations. FIM also thrives towards cross organization services, allowing managed users to be authenticated and access online resources.

In FIM, identity providers (IdP) manage affiliated users and their specific attributes. It allows users to authenticate using the IdP on various service providers (SP), such as clouds. A FIM network can also grow with the creation of new partnerships and thus the entry of different SPs and IdPs.

An e-Health scenario can benefit from FIM. In a health centered federation, health related institutes (e.g., healthcare providers, hospitals, pharmacies, certain non-profit organizations, etc.) and governmental agencies that regulate health professionals establish IdPs for clinicians. In a similar way, governmental agencies that regulate citizen identification and health related institutes that manage patient registries establish IdPs for patients.

A FIM deployment has different approaches and standards. In this work we employ SAML 2.0 (Security Assertion Markup Language). SAML is an open standard managed by OASIS [7] and uses the XML format in HTTP messages. These messages are signed to prove authenticity to the receiver side (e.g., prove a message was really sent from a specific IdP or SP).

SAML enables authentication between different security domains in the web environment and makes possible Single Sign-On (SSO). This standard exchanges tokens between a SAML authority in the role of IdP and a consumer in the role of the SP. Tokens contain assertions, i.e., sets of statements about a user. Two important assertions types are: authentication, which states whether the subject was or not authenticated; and attributes, which declares attributes associated with the subject.

The most common way to authenticate in FIM begins by the user requesting the service of a SP. As the SP does not manage authentication, it creates a request through a SAML message of the type <samlp:AuthnRequest>. The SP sends the message via HTTP Redirect to the IdP selected by the user.

Upon receiving the SAML request message, the IdP performs the authentication with the user, typically via username and password, or alternatively through the use of a token, biometric means, etc. In case of correct credentials, the IdP builds a SAML response message <samlp:response> containing both authentication and attribute assertions. This message passes through the user to the SP on another HTTP redirect.

Finally, after the SP receives the assertions, it is able to create a security context in the form of user session. Based on the attributes, it can also impose access control, deciding to release or not the desired resource. A cloud acting in this manner controls the authorization. If the cloud provider wishes to disclosure or access user's data, it might very well break user privacy. For more details on SAML, see [7].

B. Attribute-Based Encryption

Attribute-based encryption (ABE) is a cryptographic protocol that enforces access control through encryption, as introduced by Sahai and Waters [8]. It allows a more intuitive data sharing by associating user's cryptographic keys to attributes that represent true characteristics, e.g., roles, organization membership, and professions. The data encryption process also uses attributes to determine which keys are able to decrypt it.

In order to accomplish this, ABE requires an attribute authority (AA). This entity is responsible for the protocol's initial setup (e.g., creation of AA's public and secret keys), management of attributes and distribution of ABE keys. It is trustworthy, otherwise it could distribute keys and secret information in an unauthorized manner.

Bethencourt, Sahai, and Waters [9] proposed an adaptation to the original ABE, called CP-ABE (Ciphertext Policy Attribute-Based Encryption). During the encryption process, the CP-ABE uses a policy based on attribute relations, instead of a list of attributes.

The CP-ABE encryption policy is a logical equation built with AND or OR operators and attributes. For instance, (HospitalB AND Neurosurgeon) OR (HospitalA AND Pharmacist) OR (ID:3132). A user can define different sets of combinations of attributes that must be satisfied in a key for the decryption to occur.

A centralized AA responsible for all attributes and the issuance of keys may represent a risk to the use of ABE. If the AA does not prove reliable, it could issue keys in an improper way, this is known as a key escrow problem. There are proposals for decentralization that role, among them the proposal of Lewko and Waters [10] allows multiple authorities on a CP-ABE model (Multi-Authority ABE - MA-ABE). Thus, different AAs are responsible for different attributes and, together, emit keys for users. This decentralized form of ABE is used in the proposal of this paper.

The MA-ABE proposed by Lewko and Waters encompasses five algorithms: global setup, AA setup, keygen, encrypt, and decrypt. Our new architecture proposal uses this decentralized form of ABE and its algorithms are detailed throughout Section IV.

III. Related Work

Federated identity management has been considered a necessary step towards cloud federations [11]. By means of this technology, clouds cooperate to allow storage data migration

[12] or sharing of resources (e.g., when one cloud is overloaded). Our work takes a different approach and integrates FIM and clouds focusing on ABE support for storage and sharing of data. Although, there are several proposals related to cloud and ABE [13], few works employ FIM and ABE.

Tassanaviboon and Gong [14] presented an authorization scheme that resembles the FIM scenario. Their proposal uses CP-ABE to encrypt data stored on clouds and adapts OAuth [15] (an authority delegation protocol for FIM) to use access delegation tokens based on ABE. Through their scheme, a user grants a SP (e.g., a printing company) access to resources (e.g., pictures) in a cloud SP. The user is necessarily present in every access authorization procedure, having to authenticate on an authority and send an authorization token to the SP requesting access. When needing to share stored data with many SPs, users would be required to repeat these same operations several times. In contrast, our solution aims at storing health data in a SP and sharing them among several users without the direct participation of file owners on the access request operation.

Another proposal that uses FIM and ABE is due to Niwa, Kanaoka and Okamoto [16]. They introduce a generic framework that makes use of an infrastructure to supply information (e.g., user identity, attributes, etc.) for functional encryption services to create cryptographic keys (e.g., ABE keys). This allows, for an example, an entity to create user keys based on the information consulted from such infrastructures. As their solution, our proposal also makes use of a provider of user's information, i.e., the identity federation. However, their work lacks a mechanism for access revocation and does not issue keys in a scalable and decentralized way as in our solution. In addition, they do not guarantee the confidence in the information supplying infrastructure. Finally, their work is not set in sharing and using cloud resources, nor in an e-health scenario.

Besides these two solutions, our proposal is related to schemes that employ ABE to secure health records stored in the cloud. Li *et al.* [17] present such a scenario and splits the access control and key management in two domains. The first is the personal domain, in which patients manage ABE attributes and keys. The second is the public domain, where ABE attributes and key management is done by attribute authorities. Unfortunately, the personal domain adds complexity to users as they are charged with the ABE operations entitled to an AA (e.g., create and distribute keys). In our scheme IdPs work with AAs towards that task, and users still remain with full control of their records. We also take advantage of FIM to guarantee trust in user attributes informed to AAs as well as relieve users of complex operations.

Another solution was given by Alshehri, Radziszowski and Raj [18]. Their proposal uses CP-ABE to store encrypted EHR in the cloud. Unfortunately it is susceptible to the key escrow problem as it uses a centralized attribute authority (see Subsection II-B). It also does not address user's attribute proofing and key creation requests to the authority. Here, differently, IdPs prove user's attributes and interact with AAs for key creation requests.

Nzanywayingoma and Huang [19] present a scheme that uses a variation of CP-ABE. In their scheme, patients act as AAs, encrypt PHRs and manage key creation and distribution. Although this intends to offer more user control over the system, once again we believe this adds complexity to patients, that gain responsibilities for operations usually done by an AA. Also, it is important to consider that patients may not be available or capable of performing these operations (e.g., certain elderly, debilitated patients, etc.). Besides relieving patients of this complexity, our proposal allows more availability and still maintains user control over records, as there is no need to wait for patients to issue keys.

Although there are several applications of ABE on PHR and EHR in the cloud environment, most of them do not address the proof of user's attributes before AAs. This reduces the security of these applications. By using FIM to verify and manage user attributes, our mechanism presents a solution that takes advantage of standardized systems (SAML, see Subsection II-A) and the ABE protocol. In the next Section we present this scheme.

IV. The New Architecture for Secure Storage and Sharing of Health Records

As presented above, there are several proposals for access control and secure storage of health records. However, our solution has benefits when compared to them. It provides secure and user managed sharing, access revocation, and security in storage of data. In particular, it provides proof of user attributes before AAs and key management. Next, we introduce our solution.

In order to enable secure data storage and sharing in the cloud, the new solution is composed of the following parts: Attribute Authorities (AA), Identity Providers (IdP), cloud service providers (SP), and a set of users. The users have file owner or collaborator (with whom the file is shared) roles.

From these parts, we take into account the following security assumptions. The IdPs and AAs are trustworthy and organizations that regard user's interest have to establish them in a federation. Cloud providers have to follow the protocol correctly, i.e., they perform their role as defined. Nonetheless, providers of such services may threaten the privacy and security of users by attempting to read files. If they do not follow the scheme, the federation could cancel the contract or partnership with them. The third assumption is the use of a secure communication channel for message exchange between users and entities.

The architecture initiates in a setup phase. In this phase, AAs create ABE parameters, IdPs register users with their appropriate attributes, and clouds agree to offer services in the role of FIM SPs. After the initial phase, users are ready for storing and sharing health records securely. To store a record in the cloud, the user encrypts a health record with symmetric cryptography and ABE before the upload. Now, the health record can be shared. In order to access the record, another user requests the download from the SP that redirects it to an IdP. The IdP authenticates the user. In case the user has sufficient attributes that satisfy the ABE policy, the AAs issues an ABE key set. The user now can download the health record and use the ABE key set to decrypt it. We detail these operations as follows.

A. Setup

In order to deploy the architecture, the stakeholders first establish the FIM policies. As mentioned before, these stakeholders are organizations such as healthcare providers, governmental entities, non-profit organizations, among others. The FIM policies specify exactly which organizations are collaborating together to form the FIM, which of them will have IdPs and AAs, and also determine by contract or partnership which clouds are going to deliver services to the federation in the form of SPs.

The IdPs here register users. They associate the users to a global ID (based on social security number or any other unique identification) and their attributes. IdPs also maintain a mapping to AAs and their respective managed attributes. In turn, every AA knows and trusts IdPs that belong to the FIM. The attributes an AA is responsible to manage and the procedures to add new AAs must satisfy the federation's organizational policy.

In addition to these procedures, the parts run the following ABE algorithms. We consider here the scheme proposed by Lewko and Waters [10] (see Subsection II-B), but other ABE schemes can be used.

GlobalSetup(λ): The stakeholders run this algorithm once to establish the public global parameters for the ABE. It has as input a security parameter (λ) and outputs global parameters (GP). GP will be used throughout the rest of the scheme.

AASetup(GP): This algorithm generates the cryptographic material for each AA. It has as input GP and outputs a secret (SK) and a public key (PK). Each AA executes AASetup once.

Organizations may be responsible for both AAs and IdPs. Some attributes may be generic and common to many users from different organizations (e.g., clinician professions) and governmental or nonprofit institutions could manage AAs for them. On the other hand, some attributes are organization specific (e.g., healthcare provider's ID and clinician's workplace) and have to be managed by the organization itself. Nonetheless AAs can still be deployed independently.

B. Storage in the Cloud

In order to store a health record in the cloud, the file owner runs the EncryptHR and EncryptABE algorithms. The file owner employs these algorithms to encrypt his health record (HR) with a symmetric key, that in turn is encrypted following an ABE policy.

EncryptHR(K, HR): This algorithms has as input a symmetric key K and a health record (HR). It encrypts HR with K by means of a symmetric cryptographic algorithm, such as AES. It outputs a ciphertext (CT).

EncryptABE(K, P_{ABE}, GP, {PK}): This algorithms has as input the symmetric key K, an attribute based policy (P_{ABE}), the system's global parameters GP and a set of public keys of the related AAs (responsible for the attributes in the policy). The algorithm encrypts K according to the policy and outputs the ciphertext CT_{ABE}.

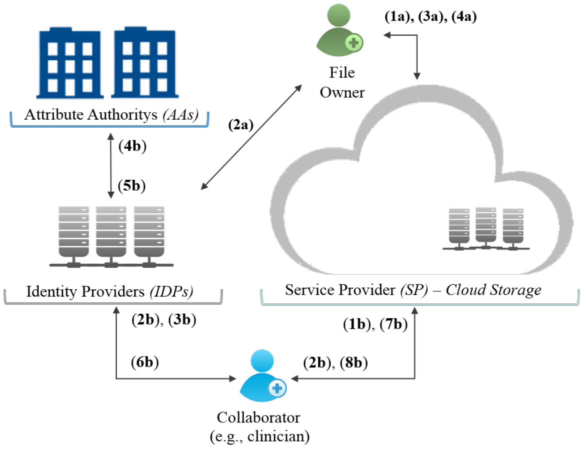
(1a), (3a), (4a)
File Owner
Attribute Authoritys (AAs)
(2a)
(4b)
(5b)
Identity Providers (IDPs)
Service Provider (SP) – Cloud Storage
(2b), (3b)
(1b), (7b)
(6b)
(2b), (8b)
Collaborator (e.g., clinician)

Fig. 1. Steps to store and share records in the new architecture

The ABE policy necessarily has the file owner's unique ID as an attribute. This ID guarantees that the file owner is able to access his file. The ABE policy can also include an expiration date attribute. An ABE policy, for instance, could be defined such as ((Attribute1) OR (Attribute2) OR (owner's ID)) AND (Expiration). This way, after the expiration date, the record will become unavailable and the file owner has to reset the policy with a new expiration date. The expiration date in policies is necessary for the access revocation mechanism described below.

After executing the algorithms, the file owner is now able to store his encrypted health record in the cloud SP, as illustrated in Fig.1. For this, he requests the storage service to the SP (step 1a) and chooses to authenticate through FIM by selecting a specific IdP. After the IdP verifies user's credentials (step 2a), it returns a SAML response to the SP redirected through the user (step 3a). This enables the user to store {CT, CT_{ABE}} in the cloud (step 4a). In conjunction to the upload, additional information is inserted to the storage metadata: the P_{ABE}, file owner's identification and the date of creation. It is also desirable that the cloud inform the URL of the stored HR. This URL is used to facilitate sharing of HR.

C. Sharing of Health Records

Once the file owner encrypts his record with a P_{ABE} policy, the ABE protocol ensures the access control. In order to safe share information, there is no need for any other operations of access delegation other than to establish attribute relations in the ABE policy. This results in collaboration between users through health record sharing.

Fig.1 presents the process of sharing health records. This process initiates when a user requests access (step 1b) to a given record on a SP (possibly through its URL). The SP then creates a SAML request of <samlp:AuthnRequest> type that is sent (step 2b) to the IdP chosen by the user. The SP also sends the P_{ABE} (saved in the metadata) to the IdP in the same message.

Upon receiving the SP's request message along with P_{ABE}, the IdP asks for user's credentials, such as username and password (step 3b). After authenticating the user, the IdP verifies whether his attributes (linked to his registry in the IdP) satisfy the ABE policy. If it does not, the IdP responds an access denied message.

Note that at first the IdP policy check is not necessary as the ABE protocol guarantees access control. However, without this check, users would only discover that they do not have enough access privileges after making the cryptographic operations and failing to read the file. This wastes time and computing resources. Therefore, by means of the IdP policy check, it avoids these unnecessary operations when the access is denied.

Once the IdP determines that the user has attributes that satisfy P_{ABE}, it requests (step 4b) corresponding AAs to each run a KeyGen algorithm. This algorithm will generate the keys according to the desired attributes that satisfy the policy. It is described as follows.

KeyGen(GID, GP, *i*, SK): This algorithm receives as input the user's global ID (GID), the system's global parameters (GP), the chosen attribute, (*i*) and the AA's secret key (SK). It generates a key corresponding to attribute *i*.

After executing the algorithm, the AAs return (step 5b) to the IdP individual keys that form a unique K_{ABE} key set (including a current date attribute). For an optimization purpose, the IdP requests a key set associated to a minimum set of attributes needed to satisfy the policy, instead of a set that encompasses all of the user's attributes.

The IdP then returns the K_{ABE} to the user (step 6b) along with the SAML response <samlp:response>. This SAML response is sent to the SP (step 7b) to prove the successful authentication. By receiving this proof, the SP allows the download of {CT, CT_{ABE}} (step 8b). To obtain the readable health record, the user needs to run the following algorithms:

DecryptABE(CT_{ABE}, GP, {K_{ABE}}): This algorithm has as input an ABE ciphertext (CT_{ABE}), the system's global parameters (GP), and a set of ABE secret keys ({K_{ABE}}). The algorithm decrypts CT_{ABE} and obtains the symmetric K key. If the K_{ABE} set of keys does not satisfy the policy, the decryption process does not returns K.

DecryptHR(CT, K): The input of this algorithm is a ciphertext (CT) and the key (K) obtained after running the DecryptABE algorithm. It decrypts CT and returns the health record.

In order to update the stored record, the user encrypts the altered HR using K and uploads it to the SP. For access control update, the policy P_{ABE} needs to be changed and updated to the storage metadata. Also, K must be re-encrypted with the new policy and uploaded to the SP. Only the file owner is allowed to alter enciphered K (CT_{ABE}) and P_{ABE}. He is able to transfer ownership by changing this in the metadata. Therefore, besides sharing, clinicians can also transfer EHR ownership to other clinicians.

D. Access Revocation

The procedures presented above allow a file owner to store and share health records securely. However, in a realistic scenario these procedures are not enough. A file owner still needs a way to revoke the access to his records. A patient can use this, for example, when he changes doctors. That is, he allows the new doctor to access his records, but disallows old doctors from obtaining them. In addition, a file owner needs to revoke the access to his record in case a user (e.g., a doctor) lost attributes.

In other words, access revocation is necessary in two cases: the owner of a record wants to alter who has access to it; or a user has lost attributes that had previously and as such does not satisfy a certain health record's ABE policy anymore.

A file owner is able to revoke the access to his records in order to disallow a user to access them. For this, he needs to change the policy used in the encryption of K. That is, the file owner runs again the EncryptABE algorithm with the parameters K and new policy P_{ABE}. The algorithm outputs a new ciphertext CT_{ABE}. The file owner now uploads CT_{ABE} to the cloud.

Note that the file owner does not need to modify the encrypted health record already in the cloud when using the same K. This K can be obtained when decrypting CT_{ABE} before the update. Otherwise, the health record has to be encrypted again using a new K and the EncryptHR algorithm. This algorithm outputs a new ciphertext CT that the user uploads to the cloud.

In case a user has lost attributes, he also loses access to records whose policies are not satisfied anymore. As IdPs check whether the user has attributes that satisfy P_{ABE}, the collaborator will not receive a valid key and SAML response. As a result, he will not have access to the record in the SP. If he obtains the record in some way, the user may try to decrypt the record by using an old set of keys (received from a request before he lost the attributes). The expiration date attribute in the ABE policy used in the health record encryption process is enough to render the old key set ineffective since it wont satisfy the updated ABE policy.

V. Discussion

The solution presented in the last section makes possible secure storage of health records in a cloud. We now sketch an evaluation of the solution.

Our architecture offers services on-demand, guarantees security and privacy, minimizes complexity on to users and guarantees availability. Also, it ensures user sovereignty over his health record stored in the cloud, providing a reliable means to control access and sharing.

In order to fulfill its objectives, the architecture employs an ABE protocol. This protocol ensures security and privacy in the outsourced environment. It performs encryption and decryption operations on user's computer, which demands time and local computing resources. The major drawback is that users may ignore these safety procedures and choose to upload the readable file to the cloud. The security versus usability is still a challenger when dealing with cryptographic protocols and that is also present in our scheme.

The ABE scheme helps our solution to allow simple access revocation. That is, users may alter who has access to his health record. It performs this through policy update and an expiration date attribute. However, it is still a challenge to revoke access to an individual user that satisfies the policy, for an example, a specific orthopedist while allowing all other orthopedists. The MA-ABE and most other ABE variations do not accept the NOT logical operator. Revoking a key is a challenging task and has been a focus of research in ABE.

The most demanding operation, other than encryption, is the ABE key request (see Fig. 1 - step 3b). The key sets are created on demand and formed with the minimum set of user's attributes that satisfy the record's access policy. These keys are smaller than keys associated with all attributes from a user and created with requests to less AAs. This technique is useful as key reusability brings the undesired burden of key management for users.

The solution is also based on a FIM protocol. However, differently from the traditional FIM, here we adapt this protocol to include attribute authorities. These authorities are responsible for ABE setup and key creation in response to IdP requests. The adapted FIM reduces the ABE complexity to users as key and attribute management is delegated to the federation.

A client program abstracts some of the complexity from users by performing cryptographic operations. In order to prevent phishing attacks (redirect to fake IdPs), these clients need built in redirects to true IdPs and need to verify their digital certificates. Patient usability and minimization of operations also need to be addressed by these client programs, for example, the provisioning of policy templates and attribute descriptions.

Other applications and systems can be built on to the architecture. Fine grain management, EHR to PHR conversion, patient personal devices as information sources and other functionalities related to health records are not in the center of the scope of the proposal and are examples of systems that can be coupled on to it.

VI. Conclusion and Future Work

In this paper we proposed a novel architecture for secure sharing and storage of health records in the cloud. Our solution makes possible a secure storage with file sharing managed by the user based on an access policy. Also, it allows access revocation and does not require users to continuously store keys.

The new proposal also enables patient and clinician sovereignty over PHR and EHR. In addition, it allows collaboration through health record sharing and it allows simple access revocation. It is thus a step towards secure sharing and storage of health records in the cloud. A more complex access revocation is still a challenge.

As future work, we will improve the architecture to consider different configurations. IdPs may assume the AA role and AAs may perform as SPs. In addition, we will consider other ABE protocols and will provide a proof of concept implementation for the proposed architecture.

References

[1] R. Bhadauria and S. Sanyal, "Survey on security issues in cloud computing and associated mitigation techniques." *International Journal of computer applications*, vol. 47, 2012.

[2] Microsoft healthvault. [Online]. Available: http://www.healthvault.com

[3] F. Liu, E. Rijnboutt, D. Routsis, N. Venekamp, H. Fulgencio, M. Rezai, and A. Van der Helm, "What challenges have to be faced when using the cloud for e-health services?" in *e-Health Networking, Applications and Services (Healthcom), 2013 IEEE 15th International Conference on*. IEEE, 2013, pp. 465–470.

[4] CAFe. Comunidade academica federada. [Online]. Available: http://portal.rnp.br/web/servicos/cafe

[5] InCommon. Security, privacy and trust for the research and education community. [Online]. Available: https://incommon.org/

[6] OECD, "National strategies and policies for digital identity management in oecd countries." [Online]. Available: http://dx.doi.org/10.1787/5kgdzvn5rfs2-en

[7] OASIS. Advanced open standards for the information society. [Online]. Available: https://www.oasis-open.org/standards

[8] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology–EUROCRYPT 2005*. Springer, 2005, pp. 457–473.

[9] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Security and Privacy, 2007. SP'07. IEEE Symposium on*. IEEE, 2007, pp. 321–334.

[10] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Advances in Cryptology–EUROCRYPT 2011*. Springer, 2011, pp. 568–588.

[11] D. W. Chadwick, K. Siu, C. Lee, Y. Fouillat, and D. Germonville, "Adding federated identity management to openstack," *Journal of Grid Computing*, pp. 1–25, 2013.

[12] C. Formisano, E. K. Kolodner, A. Shulman-Peleg, E. Travaglino, G. Vernik, and M. Villari, "Delegation across storage clouds: on-boarding federation as a case study," *Scalable Computing: Practice and Experience*, vol. 14, no. 4, 2014.

[13] C.-C. Lee, P.-S. Chung, and M.-S. Hwang, "A survey on attribute-based encryption schemes of access control in cloud environments," *Int J Netw Secur*, vol. 15, no. 4, pp. 231–240, 2013.

[14] A. Tassanaviboon and G. Gong, "Oauth and abe based authorization in semi-trusted cloud computing: aauth," in *Proceedings of the second international workshop on Data intensive computing in the clouds*. ACM, 2011, pp. 41–50.

[15] D. Hardt, "The oauth 2.0 authorization framework," 2012.

[16] Y. Niwa, A. Kanaoka, and E. Okamoto, "Construction of a multi-domain functional encryption system on functional information infrastructure," in *Network-Based Information Systems (NBiS), 2013 16th International Conference on*. IEEE, 2013, pp. 105–112.

[17] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 24, no. 1, pp. 131–143, 2013.

[18] S. Alshehri, S. P. Radziszowski, and R. K. Raj, "Secure access for healthcare data in the cloud using ciphertext-policy attribute-based encryption," in *Data Engineering Workshops (ICDEW), 2012 IEEE 28th International Conference on*. IEEE, 2012, pp. 143–146.

[19] F. Nzanywayingoma and Q. Huang, "Securable personal health records using ciphertext policy attribute based encryption," in *e-Health Networking, Applications and Services (Healthcom), 2012 IEEE 14th International Conference on*. IEEE, 2012, pp. 502–505.