Towards Trusted eHealth Services in the Cloud

Antonis Michalas
Security Lab
Swedish Institute of Computer Science
Stockholm, Sweden
antonis@sics.se

Rafael Dowsley
Institute of Theoretical Informatics
Karlsruhe Institute of Technology
Karlsruhe, Germany
rafael.dowsley@kit.edu

***Abstract*—As adoption of eHealth solutions advances, new computing paradigms – such as cloud computing – bring the potential to improve efficiency in managing medical health records and help reduce costs. However, these opportunities introduce new security risks which can not be ignored. In this paper, we present a forward-looking design for a privacy-preserving eHealth cloud system. The proposed solution, is based on a Symmetric Searchable Encryption scheme that allows patients of an electronic healthcare system to securely store encrypted versions of their medical data and search directly on them without having to decrypt them first. As a result, the proposed protocol offers better protection than the current available solutions and paves the way for the next generation of eHealth systems.**

***Index Terms*—eHealth, Security, Cloud Computing, EHR Protection, Storage Protection, Searchable Encryption**

I. Introduction

Not many years ago, eHealth was seen as an expenditure rather than an investment. During the last decade this has been changed so drastically that eHealth has moved to the top of the development agenda not only for private organizations but also for public administration bodies that have spurred the development of eHealth. To this end, we have seen a steady increase in research focus and funding aiming to modernize existing healthcare systems and to provide reliable and cost effective eHealth services. As a result, nowadays we are in front of a major technological upturn of an industry that for years relied on handwritten records and now is expanding at a phenomenal rate.

During the early development stages, the main idea around eHealth was the modernization of current medical systems by digitizing the existing personal health records. The transition from the handwritten to digital records revealed the importance and the benefits of the overall project and made a great impact to the healthcare industry. As a consequence, researchers focused on finding ways to further develop the area by effectively creating and efficiently delivering healthcare services that goes beyond the digitization of data. More precisely, scientists started envisioning eHealth systems where patients would be able to access their personal records from anywhere and digitally share their medical summary with health professionals. Furthermore, researchers in the eHealth sector started getting advantage of the new technologies such as smart-phones in order to build services that allows patients to collect daily insights and find out more about healthy behaviors.

eHealth started becoming widely available not only thanks to the impact that such services have on the day-to-day health experiences for millions of users but also due to the advancement of major technologies such as the fast Internet and mobile technology. As the eHealth industry started following and adopting new technologies and developments in interrelated areas the advent of cloud computing created the foundations for an even bigger evolution. By taking advantage of the flexibility that cloud services offer, scientists realized that the combination of cloud computing with the eHealth has the potential to rapidly expand both the development and the adoption of eHealth worldwide. More precisely, cloud computing can help us create more reliable healthcare services that will provide us with the necessary functionality to improve the management of our health and optimize our care services.

Nevertheless, the adoption of cloud computing by the eHealth sector is not an easy task. The main reason for that, is the fact that there is a natural resistance to the idea of handing over sensitive information to external storage. And when we refer to sensitive data, we need to consider that our health data is the most personal and private information we have which makes them our most precious and unique data ever. Actually, far more sensitive than our financial data. Therefore, building trustworthy eHealth services that will provide banks strength security features is of paramount importance.

The importance of building secure eHealth services as well as the benefits from adopting cloud computing and migrating existing healthcare systems to the cloud has been extensively studied [1]. However, such an attempt raises many challenges and risks that needs to be tackled. A typical eHealth scenario usually involves many parties (e.g special equipment that collects data from the patients' body, equipment for authentication, the patient, the doctor etc.) that needs to be trusted in order to provide the necessary security guarantees to the end-user who is going to hand-over her personal health data to a third party – the cloud.

One step towards a trustable eHealth platform is to make sure that the stored data will always remain encrypted and access will be granted only to legitimate users (e.g the therapist or the doctor of a patient). Keeping patients' data encrypted in every step, reduces the risk of someone stealing or misusing it to a minimum and increases the patient privacy to a maximum. However, having a closer look at the problem we can observe that encrypting the patients' data needs to be treated in a more

sophisticated way. For example, if the entity that encrypts users data is the cloud service provider (CSP), then it means that CSP has access to the encryption/decryption key. Thus, in the case of a corrupted CSP patients data can be exposed to unauthorized access.

A. Contribution

Our contribution in this paper is two-fold. First, we describe a forward-looking design for an electronic healthcare system that offers reliable and privacy-preserving mechanisms for placing and maintaining sensitive data in a cryptographic cloud storage. Second, the proposed scheme is solely based on the promising concept of Searchable Encryption (SE). Even though SE schemes have gained a lot of popularity lately, there are no proposed solutions with their main aim to tackle the security and privacy challenges that raise by the use of cloud in the healthcare industry. Hence, we hope that this work will provide essential knowledge to organizations, researchers and key-players in the area in order to design and build more reliable and efficient privacy-preserving eHealth services.

B. Organization

In Section II, we present the current state-of-the-art regarding eHealth services that provide secure cloud storage mechanisms, as well as the main cryptographic tools that have the potential to provide a reliable solution. In Section III, we describe the threat model under consideration and present the problem statement for this paper, while in Section IV we describe the proposed protocol. In Section V, we provide a security analysis and in Section VI we conclude the paper.

II. Related Work

Although there are many proposed electronic healthcare systems only some of them deal with the problem of protecting the privacy of users [2] when patients data is stored in the cloud [3]. However, the challenges that need to be tackled as well as the benefits from migrating an existing eHealth system to the cloud has been the focus of many studies [1], [4], [5], [6]. Furthermore, the absence of solutions that are based in searchable encryption schemes is even greater. In this section, we describe some of the most important works regarding secure cloud storage for privacy-preserving eHealth systems and we also describe the fundamentals of searchable encryption – a promising concept that we believe will serve as the bases for future healthcare systems that wish to effectively protect user privacy.

Paladi et al. [7] introduced a data confidentiality and integrity protection framework for eHealth systems based on Infrastructure-as-a-Service (IaaS) clouds. The proposed solution relies on trusted computing principles to provide transparent storage isolation between IaaS clients. In addition to that, the authors address the absence of reliable data sharing mechanisms, by providing an XML-based language framework which enables clients of IaaS clouds to securely share data and clearly define access rights granted to peers. The proposed improvements were prototyped as a code extension for Openstack – a popular cloud platform. Even though the proposed framework is secure and it has been also implemented as part of an existing eHealth platform it does not provide the same flexibility as our design since patients data are stored in an encrypted form based on different domains. Even though this technique is considered as a good practice that takes advantage of the trusted computing principles, explicitly targets the IaaS model. Our solution, gives the option to Platform-as-a-Service (PaaS) services to store users data in an encrypted form based on a unique key for each user and not for a whole domain. However, the combination of the two protocols, can be proved as a concrete solution that has the potential to cover both the IaaS and the PaaS layer.

In [1], authors presented an overview of major requirements that must be considered when migrating eHealth systems to the cloud. This overview was based on their experience with deploying part of a Swedish electronic health records management system in an infrastructure cloud. Furthermore, they described a new attack vector inherent to cloud deployments and presented a novel data confidentiality and integrity protection mechanism for infrastructure clouds.

Most of the current approaches in the area, deal with the problem of securely sharing patients data based on defined policies and access rights. To this end, there are many proposed solutions [3] that are based on attribute based encryption [8]. However, even though these techniques can offer granular access control protocols are different than our approach and can be used in combination with our proposed protocol. More precisely, such an attempt has the potential to offer more concrete privacy-preserving eHealth platforms for the cloud.

As we mentioned earlier, our approach is based on Searchable Encryption (SE). SE is an enhanced encryption technique that allows encryption while enabling search for keywords in the encrypted data (as it would be possible in the plaintexts). Its quintessential application is cloud storage. In searchable encryption it should be possible for the CSP, with the help of some search token sent by the client, to locally perform some operations and then send the relevant data to the client. The relevant data should be such that on one hand it contains the matching documents (i.e., the documents that contain the searched keyword), but on the other hand its size is not far bigger than that of the matching documents (i.e., the server cannot simply transfer a large part of the database to the client on every query). Of course the CSP should not learn the keyword that is being searched, otherwise information about the documents is revealed to a possible malicious entity. Depending on the requirements of the desired scheme, it is possible to use either public-key cryptography technologies or symmetric-key cryptography, but in general searchable public-key encryption schemes with good security guarantees do not scale well because they have a search time which is linear in the number of documents. Our protocol, relies on Symmetric searchable encryption (SSE) that was introduced by Song et al. [9], who presented a scheme that allowed a linear search time (in the number of documents) by the server. Lately, the

field of SSE has met a great development with many proposed schemes [10], [11], [12], [13], [14], [15], [16], [17].

III. Problem Statement & Definitions

In this Section, we describe the main entities that participate in our protocol and a set of cryptographic tools that the proposed solution is based on. In addition to that, we describe the threat model we consider as well as the actual problem that we try to tackle.

User (u): In our protocol, a patient that uses the eHealth application is considered as a typical user. The operations that a patient can perform are the following: *a)* register to the service, *b)* generate encryption keys to safely protect her data, *c)* store data in the cloud as well as retrieve and search over her private data that has been sent to the cloud. By u_i we denote a user with a unique identification i.

Registration Authority (RA): RA is responsible for the registration of a new user. The registration authority should be a certified and trusted organization able to successfully process the registration of a user. In our system, we envision a hospital to act also as the registration authority. Additionally, RA has a public/private key pair denoted as $\mathsf{pk}_{\mathsf{RA}}/\mathsf{sk}_{\mathsf{RA}}$.

Smart Card (SC): SC is a smart card that is given to a patient – the user – after she registers to the eHealth service. With the use of the smart card, the user can authenticate securely login to our service.

Cloud Service Provider (CSP): We consider a cloud computing environment based on a trusted IaaS provider like the one described in [7]. The IaaS platform consists of cloud hosts which operate virtual machine guests and communicate through a network. In addition to that, we assume a PaaS provider that is built on top of the IaaS platform and can host multiple outsourced databases. Furthermore, the PaaS provider offers an API through which a developer can built a privacy-privacy preserving eHealth application that offers searchable encryption functionality as we describe in Section IV.

Definition 1 (Dynamic Index-based SSE): A dynamic index-based symmetric searchable encryption scheme is a tuple of nine polynomial algorithms $\mathsf{SSE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{SearchToken}, \mathsf{AddToken}, \mathsf{DeleteToken}, \mathsf{Search}, \mathsf{Add}, \mathsf{Delete}, \mathsf{Dec})$ such that:

- Gen is probabilistic key-generation algorithm that takes as input a security parameter λ and outputs a secret key K. It is used by the client to generate her secret-key.
- Enc is a probabilistic algorithm that takes as input a secret key K and a collection of files $\mathbf{f}$ and outputs an encrypted index γ and a sequence of ciphertexts $\mathbf{c}$. It is used by the client to get ciphertexts corresponding to her files as well as an encrypted index which are then sent to the storage server.
- $\mathsf{SearchToken}$ is a (possibly probabilistic) algorithm that takes as input a secret key K and a keyword w and outputs a search token $\tau_s(w)$. It is used by the client in order to create a search token for some specific keyword. The token is then sent to the storage server.
- $\mathsf{AddToken}$ is a (possibly probabilistic) algorithm that takes as input a secret key K and a file f and outputs an add token $\tau_a(f)$ and a ciphertext c_f. It is used by the client in order to create an add token for a new file as well as the encryption of the file which are then sent to the storage server.
- $\mathsf{DeleteToken}$ is a (possibly probabilistic) algorithm that takes as input a secret key K and a file f and outputs a delete token $\tau_d(f)$. It is used by the client in order to create a delete token for some file which is then sent to the storage server.
- Search is a deterministic algorithm that takes as input an encrypted index γ, a sequence of ciphertexts $\mathbf{c}$ and a search token $\tau_s(w)$ and outputs a sequence of file identifiers $\mathbf{I}_w \subset \mathbf{c}$. This algorithm is used by the storage server upon receive of a search token in order to perform the search over the encrypted data and determine which ciphertexts correspond to the searched keyword and thus should be sent to the client.
- Add is a deterministic algorithm that takes as input an encrypted index γ, a sequence of ciphertexts $\mathbf{c}$, an add token $\tau_a(f)$ and a ciphertext c_f and outputs a new encrypted index γ' and a new sequence of ciphertexts $\mathbf{c}'$. This algorithm is used by the storage server upon receive of an add token in order to update the encrypted index and the ciphertext vector to include the data corresponding to the new file.
- Delete is a deterministic algorithm that takes as input an encrypted index γ, a sequence of ciphertexts $\mathbf{c}$ and a delete token $\tau_d(f)$ and outputs a new encrypted index γ' and a new sequence of ciphertexts $\mathbf{c}'$. This algorithm is used by the storage server upon receive of a delete token in order to update the encrypted index and the ciphertext vector to delete the data corresponding to the deleted file.
- Dec is a deterministic algorithm that takes as input a secret key K and a ciphertext c and outputs a file f. It is used by the client to decrypt the ciphertexts that she gets from the storage server.

Problem Statement: *We consider the scenario where a user u_i (e.g a patient) wishes to store and process her medical data in the cloud via an interface offered by an electronic healthcare system. The main problem is to create a protocol that will satisfy the following requirements:*

- *Data stored by u_i should be encrypted with a key that will be known only to u_i and even the CSP will not be able to access the stored data;*
- *User u_i should be able to search directly over the encrypted data that she owns. This process should be done in a privacy-preserving way so that no valuable information about the searched data is revealed;*
- *Unauthorized access to users' data should be prevented;*

Adversarial Model: Similar to existing works in the area, we make the following assumptions regarding the threat

model we consider. First, we assume physical security of the CSP as well as of the devices that patients are using in order to send/retrieve their medical summary to/from the cloud. In addition to that, we also assume that all cryptographic operations that are used throughout the protocol are semantically secure and an adversary is not able to break any cryptographic mechanism. Finally, we assume that the adversary is acting under the *semi-honest* threat model. In the semi-honest adversarial model, adversarial nodes correctly follow the protocol specification. However, nodes overhear all messages and may attempt to use them in order to learn information that otherwise should remain private. Semi-honest adversaries are also called *honest-but-curious*.

IV. Protocol Description

In this section, we introduce our protocol which satisfies the criteria mentioned in the problem statement and offers secure storage functionality for the users of an eHealth application that stores their personal health records in the cloud. Before we proceed with the actual description of the protocol we provide a high-level overview of the phases that our protocol consists. Figure 1 contains a high-level representation of the main functions that our protocol consists of (details have been omitted for clarity).

Our protocol can be divided into *five phases*: the registration phase, the login phase, the key generation phase, the secure placement of data in the cloud by a user and finally the retrieval of data by user in a privacy-preserving manner. Before accessing the eHealth application, a new user needs to first register. To do so, the registration phase requires the user to contact RA and submit her identity. Then, RA is responsible for verifying the validity of the user and can also prevent a user from creating multiple accounts/identities by simply checking if a user with the same identity has already created an account. After the verification process, RA will give the new user a smart card and a password through a secure channel. By using these, the user will be able to login to the eHealth platform every time that wishes to check her medical history, to submit fresh data or update her personal health records.

Login Phase: Lets assume that a patient u_i wishes to login to the eHealth application. To do so, she attaches her smart card to the input device that received through the registration process. Then, she types the password that she got during the registration and the smart card generates a one-time password that u_i can use in order to login to the application. The exact steps of the operations that are performed by the smart card in order to secure the login procedure is out of the scope of this paper and has been studied extensively in the literature. Hence, w.l.o.g we assume that one of the existing techniques that are used in the finance sector can be applied to our solution.

Store Data: After the successful login, u_i will be able to start using the eHealth application in order to store and retrieve data from the cloud provider. To do so, u_i needs to have a unique encryption key that will be used to secure her personal data. Thus, before start sending data to the CSP, u_i executes $K_i \leftarrow \mathsf{Gen}(1^\lambda)$ to generate the secret key that will be used to protect her personal records. After the successful generation of the symmetric key K_i, u_i is now ready to store encrypted data to the CSP. Lets assume that u_i wants to securely store a collection of files $\mathbf{f_i}$ to the storage offered by the CSP. To do so, u_i executes $(\gamma_i, \mathbf{c_i}) \leftarrow \mathsf{Enc}(\mathbf{f_i}, K_i)$ and outputs a collection of ciphertexts $\mathbf{c_i}$ as well as an encrypted index γ_i. Both γ_i and $\mathbf{c_i}$ are then send to the CSP via a secure channel. Upon reception, CSP stores $\mathbf{c_i}$ along with the encrytped index γ_i in a local database.

Search Over the Encrypted Data: Now that u_i has stored a collection of files in the cloud storage she can start searching directly over her encrypted data. Lets assume that u_i wishes to search over her data for a specific keyword w. First, u_i executes the $\tau_s(w) \leftarrow \mathsf{SearchToken}(K_i, w)$ and outputs a search token $\tau_s(w)$. that is sent to the CSP. Upon reception, CSP executes $\mathsf{Search}(\gamma_i, \mathbf{c_i}, \tau_s(w)) \rightarrow \mathbf{I}_w$ and outputs a sequence of file identifiers $\mathbf{I}_w$ which is a subset of $\mathbf{c_i}$ and contains a list of ciphertexts that contains the keyword w that u_i searched for. The resulted $\mathbf{I}_w$ is sent back to the user. Upon reception, u_i executes the Dec algorithm by giving as input her secret key and the sequence of encrypted files that corresponds to the list of identifiers that received from the CSP. By doing this, u_i gains access to the plaintext of data that contains the keyword w.

Update Stored (Encrypted) Data: Apart from storing data and searching over the encrypted data, the user also needs to update her current data by for example adding new files. Here, we consider the scenario where u_i wishes to add a new file f to the cloud storage. A naive approach that u_i could follow would be to run Enc algorithm again, generate the ciphertext of f and send it to the CSP. However, this would mean that u_i would also create a new encrypted index that would correspond to the encryption of file f. Such an approach is not efficient since the user would end-up with a huge list of encrypted indexes that are not related to each other and every time that wishes to perform a search over her data would require from the CSP to search over all the encrypted indexes. To avoid this, u_i needs to store her new file and instead of creating a separate encrypted index she needs to update the current one in order to also include the newly added file. To achieve that, u_i first generates an add token by executing $(\tau_\alpha(f), c_f) \leftarrow \mathsf{AddToken}(K_i, f)$ and sends it to the CSP. Upon reception, CSP executes $\mathsf{Add}(\gamma_i, \mathbf{c_i}, \tau_\alpha(f), c_f) \rightarrow (\gamma', \mathbf{c_i'})$ and outputs an updated encrypted index γ_i' and an updated sequence of ciphertexts $\mathbf{c_i'}$ that corresponds to the data stored by u_i. Thus, by running the Add algorithm, CSP stores the ciphertext of f and updates the existing encrypted index and ciphertext list of u_i.

Delete Stored Data: The final operation that u_i needs to be able to execute, is the deletion of a file. Lets assume that u_i wishes to delete the file f that stored in the previous step. Similar to the previous case, the deletion of a file will also require the update of the existing encrypted index as well as the sequence of stored ciphertexts. To this end, u_i generates a delete token by executing

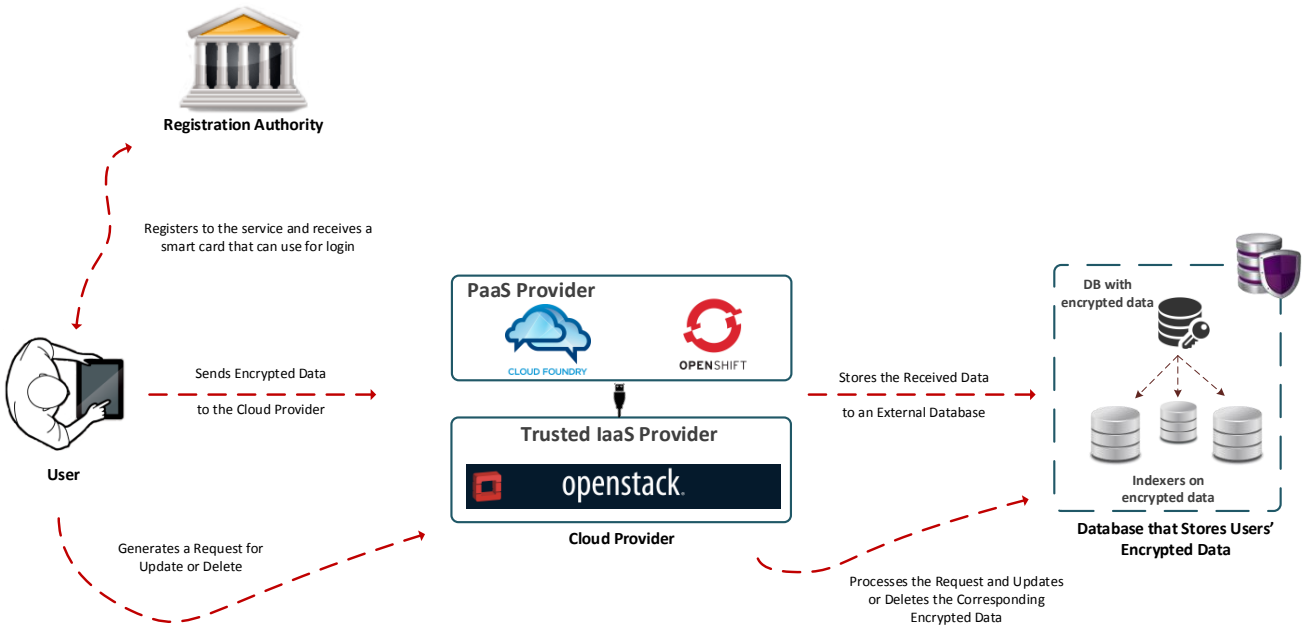

Fig. 1. High level view of the Protocol

$\tau_d(f) \leftarrow \mathsf{DeleteToken}(K_i, f)$. Then, u_i sends $\tau_d(f)$ to the CSP who executes $\mathsf{Delete}(\gamma_i', \mathbf{c_i'}, \tau_d(f)) \rightarrow (\gamma_i'', \mathbf{c_i''})$. Similar to the Add algorithm, Delete after removing the requested file f then updates both the corresponding encrypted index and the sequence of ciphertexts that are related to user u_i.

V. Security Analysis

The registration and login procedures are out of the scope of this work, but they can be secure done using standard techniques.

The main security issue in our scenario is that the user data should be kept private (even from the CSP), despite the fact that search operations are performed over the encrypted data. This security goal is achieved by using a dynamic symmetric searchable encryption scheme as a building block. In such schemes the data encryption is performed using a symmetric encryption key which is known only by the data owner (i.e., the user). In our protocol this symmetric key is kept inside the smart card, but as the smart card as well as the devices used with the smart card in order to send/retrieve the medical data are assumed to be trusted hardware, this guarantees that the symmetric key is not leaked. The security of the user data then follows from the security of the searchable encryption scheme.

To store her initial medical data the user employs the Enc algorithm of the searchable encryption scheme to produce a collection of ciphertexts and the encrypted index. Both the ciphertexts as well as the encrypted index are sent to the CSP, but the fact that they do not leak information about the user data follows straightforwardly from the security of the searchable encryption scheme. The fact that the search operations do not leak the user data also follows from the security of the searchable encryption scheme. Finally, the operations to add/update and remove data also use the respective procedures of the dynamic searchable encryption scheme and so their security follows from the security of the searchable encryption scheme. Additionally, an adversary will not even be able to extract valuable information about the searches that a user executes since the keywords that are sent by the user are also encrypted.

Therefore our protocol achieves the stated goal of storing the users' medical data in the cloud without leaking the data (even to the CSP).

Nevertheless, requiring the user to execute all these operations can be proved to be inefficient depending on the size of her data. To solve this issue, an alternative approach would be to execute all the above described procedures at the CSP. Since DoS [18], [19] attacks are out of the scope of this work, we can consider a scenario where the generated symmetric key K_i for every user will be locally stored by a Trusted Third Party (TTP) and will be responsible for contacting the CSP and doing this computations for the user. Then, every time that a user u_i wishes to execute any of the aforementioned procedures, will have to send a corresponding request to the TTP. Then, the TTP will be responsible for encrypting and decrypting users data and sending back to her over a secure channel. Such a design, would make the protocol more efficient but at the same time we would need to fully trust an entity that will end-up having access to all users' data. Even though such an assumption is typical in this kind of protocols when we deal with users' personal health records, we need to make sure that such an entity is successfully protected form

both external and internal attacks.

VI. Conclusion

In this paper, we described a forward-looking design for a cloud-based eHealth platform that offers secure storage with the addition of privacy-preserving techniques. Our protocol is solely based on the promising concept of symmetric searchable encryption in order to allow the users of such a service to store data in an encrypted form and also give them the option to search directly over their encrypted data without releasing any valuable information to the cloud provider or to any other third party.

Our on-going work focuses on extending the functionality of the proposed protocol by providing access control mechanisms and sharing functionality. In addition to that, we plan to implement our design in a real PaaS environment based on the scenario described in [20]. Furthermore, we aim at combining our work with the trusted IaaS storage that was presented in [7]. Such a solution will require a detailed design and implementation but has the potential to offer a concrete secure and trusted cloud environment for various eHealth services. Furthermore, the adoption of searchable encryption in such a scenario can proved to be a valuable tool for focusing on the increasingly urgent problem of trusted geolocation [21], [22] where users of a cloud service can control the physical location of their data as well as on participatory sensing applications where privacy is of paramount importance [23]. Finally, in our future agenda is to create a privacy preserving reputation system like the ones presented in [24], [25], [26] where users will be able to vote about doctors, health professionals, services etc. Even though such a functionality would require a detailed design in order squarely fit with the scenario described in this paper we believe that has the potential to make a great impact to the existing eHealth services.

Acknowledgements

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 644814, the PaaSword project (www.paasword.eu) within the ICT Programme ICT-07-2014: Advanced Cloud Infrastructures and Services.

References

[1] A. Michalas, N. Paladi, and C. Gehrmann, "Security aspects of e-health systems migration to the cloud," in *e-Health Networking, Applications and Services (Healthcom), 2014 IEEE 16th International Conference on*, pp. 212–218, Oct 2014.

[2] T. Sahama, L. Simpson, and B. Lane, "Security and privacy in ehealth: Is it possible?," in *e-Health Networking, Applications Services (Healthcom), 2013 IEEE 15th International Conference on*, pp. 249–253, 2013.

[3] A. Abbas and S. 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, pp. 1431–1441, July 2014.

[4] L. Guo, F. Chen, L. Chen, and X. Tang, "The building of cloud computing environment for e-health," in *E-Health Networking, Digital Ecosystems and Technologies (EDT), 2010 International Conference on*, vol. 1, pp. 89–92, April 2010.

[5] G. Fern'ndez, I. de la Torre-Diez, and J. Rodrigues, "Analysis of the cloud computing paradigm on mobile health records systems," in *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on*, pp. 927–932, July 2012.

[6] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless Communications and Mobile Computing*, vol. 13, no. 18, pp. 1587–1611, 2013.

[7] N. Paladi, A. Michalas, and C. Gehrmann, "Domain based storage protection with secure access control for the cloud," in *Proceedings of the 2014 International Workshop on Security in Cloud Computing*, ASIACCS '14, (New York, NY, USA), ACM, 2014.

[8] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proceedings of the 24th Annual International Conference on Theory and Applications of Cryptographic Techniques*, EUROCRYPT'05, (Berlin, Heidelberg), pp. 457–473, Springer-Verlag, 2005.

[9] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," pp. 44–55, 2000.

[10] E.-J. Goh, "Secure indexes." Cryptology ePrint Archive, Report 2003/216, 2003. http://eprint.iacr.org/2003/216.

[11] R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," pp. 79–88, 2006.

[12] M. Chase and S. Kamara, "Structured encryption and controlled disclosure," pp. 577–594, 2010.

[13] S. Kamara, C. Papamanthou, and T. Roeder, "Dynamic searchable symmetric encryption," pp. 965–976, 2012.

[14] K. Kurosawa and Y. Ohtaki, "UC-secure searchable symmetric encryption," pp. 285–298, 2012.

[15] S. Kamara and C. Papamanthou, "Parallel and dynamic searchable symmetric encryption," pp. 258–274, 2013.

[16] D. Cash, S. Jarecki, C. S. Jutla, H. Krawczyk, M.-C. Rosu, and M. Steiner, "Highly-scalable searchable symmetric encryption with support for boolean queries," pp. 353–373, 2013.

[17] D. Cash, J. Jaeger, S. Jarecki, C. S. Jutla, H. Krawczyk, M.-C. Rosu, and M. Steiner, "Dynamic searchable encryption in very-large databases: Data structures and implementation," 2014.

[18] A. Michalas, N. Komninos, N. R. Prasad, and V. A. Oleshchuk, "New client puzzle approach for dos resistance in ad hoc networks," in *Information Theory and Information Security (ICITIS), 2010 IEEE International Conference*, pp. 568–573, IEEE, 2010.

[19] A. Michalas, N. Komninos, and N. Prasad, "Multiplayer game for ddos attacks resilience in ad hoc networks," in *Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology (Wireless VITAE), 2011 2nd International Conference on*, pp. 1–5, Feb 2011.

[20] Y. Verginadis, A. Michalas, P. Gouvas, G. Schiefer, G. Hbsch, and I. Paraskakis, "Paasword: A holistic data privacy and security by design framework for cloud services," in *Proceedings of the 5th International Conference on Cloud Computing and Services Science*, pp. 206–213, 2015.

[21] N. Paladi and A. Michalas, "One of our hosts in another country: Challenges of data geolocation in cloud storage," in *Wireless Communications, Vehicular Technology, Information Theory and Aerospace Electronic Systems (VITAE), 2014 4th International Conference on*, pp. 1–6, May 2014.

[22] K. Benson, R. Dowsley, and H. Shacham, "Do you know where your cloud files are?," in *Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop*, CCSW '11, (New York, NY, USA), pp. 73–82, ACM, 2011.

[23] A. Michalas and N. Komninos, "The lord of the sense: A privacy preserving reputation system for participatory sensing applications," in *Computers and Communication (ISCC), 2014 IEEE Symposium*, pp. 1–6, IEEE, 2014.

[24] T. Dimitriou and A. Michalas, "Multi-party trust computation in decentralized environments," in *New Technologies, Mobility and Security (NTMS), 2012 5th International Conference on*, pp. 1–5, May 2012.

[25] T. Dimitriou and A. Michalas, "Multi-party trust computation in decentralized environments in the presence of malicious adversaries," *Ad Hoc Networks*, vol. 15, pp. 53 – 66, 2014. Smart solutions for mobility supported distributed and embedded systems.

[26] A. Michalas, T. Dimitriou, T. Giannetsos, N. Komninos, and N. Prasad, "Vulnerabilities of decentralized additive reputation systems regarding the privacy of individual votes," *Wireless Personal Communications*, vol. 66, no. 3, pp. 559–575, 2012.