A Cloud-Based eHealth Architecture for Privacy Preserving Data Integration

Alevtina Dubovitskaya, Visara Urovi, Matteo Vasirani, Karl Aberer, Michael I. Schumacher

A Cloud-based eHealth Architecture for Privacy Preserving Data Integration

Alevtina Dubovitskaya[1,2], Visara Urovi[1], Matteo Vasirani[2], Karl Aberer[2], Michael I. Schumacher[1]

[1] *AISLab, HES-SO VS, Switzerland, {Firstname.Lastname}@hevs.ch*
[2] *LSIR, EPFL, Switzerland, {firstname.lastname}@epfl.ch*

Abstract. In this paper, we address the problem of building an anonymized medical database from multiple sources. Our proposed solution defines how to achieve data integration in a heterogeneous network of many clinical institutions, while preserving data utility and patients' privacy. The contribution of the paper is twofold: Firstly, we propose a secure and scalable cloud eHealth architecture to store and exchange patients' data for the treatment. Secondly, we present an algorithm for efficient aggregation of the health data for the research purposes from multiple sources independently.

Keywords: Access Control, Interoperability, Point-of-Care System.

1 Introduction

While building an anonymized database from multiple sources of individuals' sensitive data the privacy of a person may be violated. Even if the data are locally anonymized, their aggregation can still reveal sensitive information, especially if the data about an individual are distributed between different local databases [2, 3]. Several models in the area of distributed privacy-preserving data publishing have already been proposed (i.e., pseudonymization [9, 26], secure multi-party computations (SMC) [6], microaggregation [22], cloning [2]). However, those models significantly affect the utility of the data, and, therefore, an efficient independent release of the data from multiple sources and their aggregation without violation of privacy remains an open problem [10].

This problem is of great interest especially in the case of secondary use of medical data. This includes the analysis of patients healthcare data in order to enhance their health care experiences and the expansion of knowledge about different diseases and appropriate treatment. Datasets containing health related information about an individual are increasingly becoming "open". In this paper, we focus on the medical data to address the following question: How is it possible to share and aggregate medical data for research purposes?

Collecting medical data raises privacy concerns as these data are of a personal nature to the patient. Additionally, in medical settings, the following requirements have to be considered: the ability to update the data about a patient

(without creating multiple entries corresponding to the same person), and the possibility to recontact the patient through the caregiver that uploaded the data.

Our aim is to create an infrastructure for medical data management that allows the healthcare professionals to release patients' data for research purposes while insuring patients' privacy. To achieve this we employ generalization and pseudonymization techniques. We use binary trees to represent the data generalization and multi-key searchable encryption for generating pseudonyms.

The contribution of this paper is the following: we propose a secure framework for independently and asynchronously sharing, aggregating and searching health data in the cloud, therefore without trust to the server that stores the health data. We have chosen the cloud-based approach because it allows patients and caregivers to access aggregated healthcare data from everywhere anytime (according to the access control policy specified by the patient). Moreover, it facilitates the aggregation of the data and the creation of the database for the research purposes (*RSDB*).

The rest of the paper is organized as follows. In Section 2 we present a use case scenario, in Section 3 we provide the knowledge about encryption scheme and anonymity approach that we use in our work, and we compare our solution with the existing approaches. In Section 4 we describe the architecture of our proposed eHealth system, the protocol for sharing and accessing patients' data in the cloud, and our algorithm for constructing research database. In Section 5 we discuss privacy threats and countermeasures. We present the conclusion and future work in Section 6 of the paper.

2 Use Case Scenario

The treatment of certain diseases, such as cancer, HIV, or other serious medical conditions, relies on the administration of critical drugs used to keep those life-threatening diseases under control. Those drugs (e.g. *Efavirenzum, Imatinib*) have a narrow therapeutic range and a poorly predictable relationship between the dose and the drug concentration in the blood, which may greatly vary among individuals. Therapeutic Drug Monitoring (TDM) aims at improving patient care by monitoring drug levels in the blood and adjust a dosage individually.

In order to ensure a better prediction of the relationship between dose and drug concentration a Bayesian TDM approach [12] has been developed. This approach requires population health data to be collected and analyzed by researchers, therefore, building databases for medical research is of a high importance [8]. We consider a patient, P, who visits several caregivers during the treatment (e.g., when there is a need for a consultation from particular specialists, in case of traveling, or if patient has moved). We expect that the patient P is able to access his healthcare information and to decide with whom to share it. Storing data in the cloud allows an access from anywhere anytime. However, the question of privacy has to be addressed.

A widespread use of the the electronic identity cards and the cards provided by the insurance companies shows that having a smartcard is not a burden in

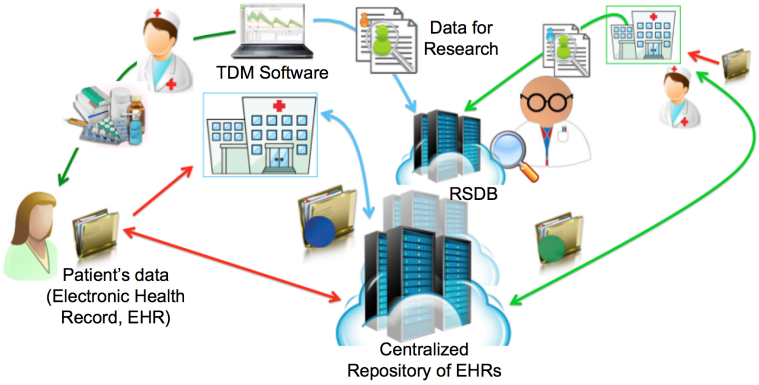

Fig. 1. Use case scenario.

everyday life. Therefore, we assume that a patient could use a smartcard to store a set of cryptographic keys for encryption/decryption of the sensitive data (contained in *EHR*) in order to prevent an un-authorised access.

For example, as it is shown on Figure 1, a set of the Electronic Health Records (*EHR*s) may belong to the same patient but could be generated by different caregivers. Each *EHR* then will be encrypted with the key shared between the patient and the caregiver that generated this *EHR*. The access control policy can be based on sharing the keys with the caregivers to allow access to the data for the treatment. Patients' data can also be collected for the secondary use. Anonymization algorithms are required to preserve patient privacy.

3 Related Work

In this section, first, we recall the details of multi-key searchable encryption scheme [20] that we employ in our solution to generate pseudonyms and annotate *EHR*s. Second, we describe (k, k^m) – *anonymity* property [21] that we impose on the research database and preserve while updating *RSDB* in the distributed environment. Finally, we present an overview of the related work and specify how they differ from our approach.

3.1 Multi-key Searchable Encryption

Without loss of generality we can assume that the server stores documents (a set of *EHR*s) encrypted with m different keys $k_1, ...k_m$, and a user (patient, or caregiver) that possesses n keys ($n \leq m$) wants to search for T words (e.g., diagnosis, date of visit, etc.) $w_1, ...w_T$ in the documents.

According to the prior work [5, 23], in order to perform the search of a word over the documents encrypted using different keys a user has to compute a token for each word under every key. In this case the complexity of the search will

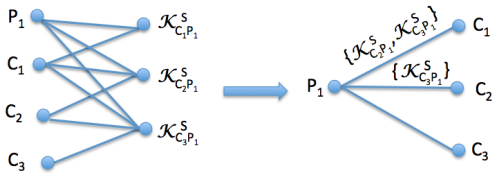

Fig. 2. An example of the access graph.

be $\mathcal{O}$ (nT). However, with an approach proposed in [20] the complexity of the multi-key search over encrypted data does not exceed $\mathcal{O}$ $(n+T)$. The multi-key searchable scheme is constructed using bilinear maps on elliptic curves [4]. The pseudocode for the multi-key searchable encryption scheme and its implementation can be found in [13].

According to the scheme if a user has an access to the keys $k_1, ...k_m$ in order to search for a word w he needs to compute only a single search token for this word: $tk_w^{k_i}$ using the key k_i, and deltas, $\{\Delta_{k_i \to k_j}\}$, an additional information that allows to adjust the token $(tk_w^{k_i})$, computed with the key k_i, to the tokens corresponding to the keys $k_1, ...k_m$ $(\{tk_w^{k_j} : j \neq i, j \in \{1...m\}\})$. These deltas represent the user's access to the documents, and, most important, these deltas can be reused for every search, so the user needs to generate them only once. Efficiency of the scheme has been evaluated and it was shown that performance overheads of using multi-key searchable encryption scheme are modest [13].

In the paper [20] the authors use graphs to represent an access to the shared key. We modify the structure of the access graph by using the labeled graph instead. This allows us to reduce the complexity of the graph. For p patients and c caregivers access graph according to the approach used in [20] will contain at most $p+c+p*c$ nodes and $p*(1+c)$ edges, while in case of using labeled graph it will take as most $p+c$ nodes and $p*c$ edges for the complete access graph. This makes access control policy easier to interpret and manage. Each node of the graph represents a patient or a caregiver. Figure 2 shows an example of access graph. The edge (between P_i and C_j) shows that P_i visited C_j (e.g., P_1 visited caregivers C_1, C_2 and C_3), therefore patient P_i and caregiver C_j connected with the edge share the key that C_j will use to create a pseudonym and encrypt the data about P_i. Labels on the edge shows the keys P_i shared with the caregiver C_j, (e.g., P_1 shares with C_1 the keys P_1 generated together with C_2 and C_3, therefore allowing C_1 access the data about P_1 generated by C_2 and and C_3,), however no label on the edge between P_1 and C_3 indicates that the only data about P_1 that C_3 can access are the data generated by C_3.

3.2 Anonymity of medical data

A variety of models, (e.g., ϵ–differential privacy, $k - anonymity$ [24], $(k^m) - anonymity$, $l - diversity$, etc. [11]) can be used for privacy preserving data publishing. However, Poulis et al. show that all these methods are not appropriate

for the anonymization of the datasets containing both relational (i.e., single-valued) and transaction (i.e., set-valued) attributes, such as medical datasets that contain patient demographics and diagnosis information together [21].

$(k, k^m) - anonymity$ proposed in [21] ensures that for any record r in the dataset and any set of m or less items in transaction attribute of r, there should be at least $(k-1)$ records that are indistinguishable from record r. However, $k - anonymity$ for relational attributes (i.e., existence of at least $(k-1)$ records that are indistinguishable from record r with respect to relational attributes of the record r) and $(k^m) - anonymity$ for transaction attribute do not imply $(k, k^m) - anonymity$. Poulis et al. developed two frameworks that produce $(k, k^m) - anonymous$ datasets with bounded information loss in one attribute type (relational or transaction) and minimal information loss in the other (transaction or relational). Our algorithmic solution (presented in the Section 4) addresses the problem of maintaining $(k, k^m) - anonymization$ property in a distributed environment.

3.3 Existing Approaches

Using encryption combined with pseudonymization techniques [17, 9, 26] has been proposed recently for building eHealth system in the cloud. There exist also a number of architectures that employ Attribute-Based encryption (ABE) scheme [27, 14–16, 19]. However, these approaches have several limitations. ABE still can leak information from the access control policy. Encryption, in general, may affect the system performance especially when there is a need to search over encrypted data for a particular information. In our work we use multi-key searchable encryption scheme [20], for which it was shown that performance overheads of using this scheme are modest [13]. In [26, 17] the authors suggest a patient-centric architecture and propose to use the smartcards for the key management. if the smartcard is lost it is very difficult to recover the keys. However, in our solution the keys can be recovered through the caregivers.

Urovi et al. in [25] proposed a secure mechanism for *EHR* exchange over a Peer to Peer (P2P) agent based coordination framework. In this approach the encrypted heterogeneous data are exposed over a P2P network. The authors provide the algorithms for searching and for publishing the *EHR*s in the untrusted P2P network without compromising the privacy, integrity and the authenticity of the shared data. This work, however, does not cover the aggregation of the data for the research purposes, as we propose here.

Using unambiguous pseudonym for the patient [18] allows one to infer additional information about a patient by linking the data from different sources. In case of using multiple pseudonyms, as in [26], their efficient management is problematic. To solve these issues we generate patients' pseudonyms with the means of the multi-key searchable encryption scheme proposed in [20]. We also use this scheme to enable efficient search over the EHRs.

In [6] the authors describe privacy-preserving distributed $k - anonymity$ algorithm that allows merging two local $k - anonymous$ datasets while preserving $k - anonymity$ property in the resulting dataset. However, the solution is not

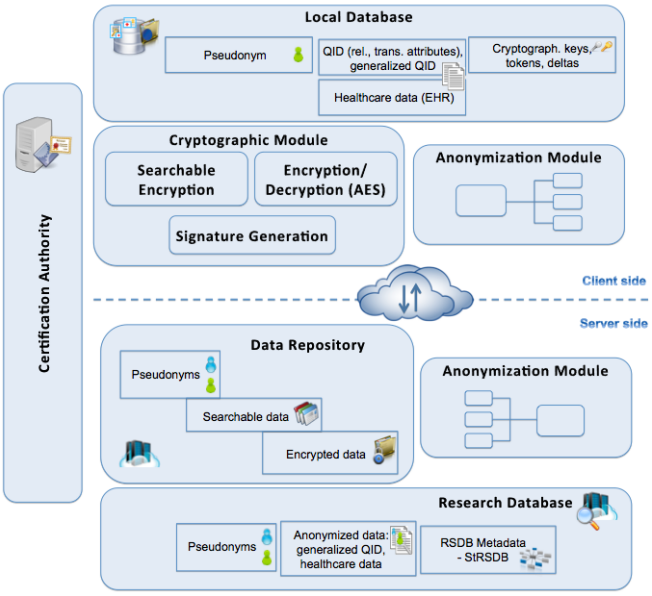

Fig. 3. Architecture overview.

scalable and requires using SMC, sharing data is not independent among different sources contributing to the *RSDB*. Baig et al. [2] suggest a model called ϵ-cloning for privacy protection in multiple independent data publications. However, it cannot be applied in our settings because it significantly affects the utility of the data. In [3] the authors proposed an architecture that allows collecting the patients consents for sharing their data for the research in an anonymous way. However, the authors assume that the data are already anonymized.

4 Proposed e-Health Architecture

In this section, we describe our proposed eHealth system. Figure 3 shows an architecture that consists of the following entities: Databases (Local Database, on the client side and Data Repository and Research Database, both hosted on the cloud server); Cryptographic Module on the side of the client; Anonymization Module (on both sides); and standalone Certification Authority. Local Database, *LDB*, belongs to the caregiver and contains healthcare data about the patients that receive treatment from this caregiver. Data Repository, *DR*, is hosted on the untrusted cloud server and stores *EHR* generated in different medical institutions. Anonymized patients data for the research purposes are stored on the cloud server in Research Database, *RSDB*. Cryptographic Module consists of three parts and its functionalities are the following: to perform multi-key searchable encryption; to encrypt *EHR* before uploading to the cloud server in order to share with the other caregivers, as well as to decrypt when accessing *EHR* according to the access control policy specified by the patient; and, to generate

the signature to ensure the authenticity of the data. Anonymization Module is a realization of the algorithm for medical data anonymization presented further in this paper. Certification Authority, *CA*, is a service that is responsible for issuing certificates of public keys and smartcads for storing private keys that are protected with the PIN known only to the owner of the smartcard.

4.1 Data Structure

Hereafter we describe the structure of the data that are stored in the databases.

- *Pseudonym(s)* – a set of uniquely identifiable patient data, ID_P, (such as combination of date of birth, place of birth and the name) that is encrypted using multi-key searchable encryption scheme proposed in [20], stored in all databases: *DR*, *LDB* and *RSDB*.
- *QID* – quasi-identifiers – a set of the attributes ($\{qid\}$) that in combination can uniquely identify the person (e.g., single-valued *qid*, such as age, gender, address (i.e., ZIP code) and set-valued *qid*, such as diagnosis codes), *gnrlQID* – a combination of generalized *qid* (in a form of a binary string), with which the data about P have been uploaded to the *RSDB*.
- *Healthcare data* – drug intakes (time, dosage, drug name), co-medications, concentration measurements (time, measurement) – multiple attributes, that can be set-, or single-valued)).
- *Cryprographic keys* and *deltas* – a set of the deltas for the keys ($\mathcal{K}^S_{P,C_j}, j \in \overline{1,\mathcal{N}}, i \neq j$) related to the patient and shared with C_i (see Subection 3.1 for more details).
- *Encrypted data* – health data, or, *EHR*, encrypted with symmetric cipher (e.g., AES).
- *Searchable data* – *EHR* or a list of the attributes that describe the content of the *EHR* (encrypted using multi-key searchable encryption scheme).
- *Anonymized data* – consist of generalized *QID* (*gnrlQID*) and a subset of *healthcare data* from *LDB*.

StRSDB – is a table that characterizes the current state of the $(k, k^m)-$ *anonymous RSDB*. For each combination of *qid* that are presented in *RSDB*, *StRSDB* stores the following information: *PsNumber* – a number of different *pseudonyms* from *RSDB* associated with the same *QID* set and the sources of data (C_i that uploaded the data, and $PsNumber^i$, a number of *pseudonyms* associated with this *QID*). One has to notice that as *RSDB* is $(k, k^m) - anonymous$, $PsNumber \geq k$ and $\sum_{i \in \overline{1,\mathcal{N}}} PsNumber^i = PsNumber$. Figure 4(c) presents an example of *StRSDB*.

We also assume that each database stores date/time of inserting a record; in *DR* and *RSDB* the signature of every record is stored together with $\mathcal{PK}_C$ of a caregiver that uploaded the data and sigh them. Figure 4 shows the examples of *LDB* (a), *RSDB* (b) and the representation of the metadata of *RSDB*, *StRSDB*, on Figure 4(c). The data from this particular example show the dosage of the drug and its actual concentration in the blood for a group of patients.

PS_P	QID		gnrlQID		healthcareData			$\mathcal{K}^S_{C,P}$	deltas
	age	gender	age	gender	analyte	dose, mg	concentr., mg/l		
Pseudo1	39	f	[38-50)	f	Efavirenzum	600	321	F498…	{A54…, 345…}
Pseudo2	26	m	[25-38)	m	Efavirenzum	550	257	4252…	{779…, 7B2…}
Pseudo3	30	m	[25-38)	m	Efavirenzum	600	354	76B1…	{57C…, 7FA…}
Pseudo4	12	m	[0-25)	m	Efavirenzum	450	214	32C4…	{7B3…, 48A…}
Pseudo5	45	f	[38-50)	f	Efavirenzum	600	319	6812…	{242…, DA4…}
Pseudo6	5	m	[0-25)	m	Efavirenzum	450	214	AB45…	{72F…, 36D…}
…	…	…	…	…	…	…	…	…	…

(a)

$prevPS_P$	PS_P	gnrlQID		healthcareData		
		age	gender	analyte	dose, mg	concentr., mg/l
	Pseudo1	[38-50)	f	Efavirenzum	600	321
	Pseudo2	[25-38)	m	Efavirenzum	550	257
	Pseudo3	[25-38)	m	Efavirenzum	600	354
	Pseudo4	[0-25)	m	Efavirenzum	450	214
Pseudo15	Pseudo5	[38-50)	f	Efavirenzum	600	319
	Pseudo6	[0-25)	m	Efavirenzum	450	214
Pseudo17	Pseudo7	[38-50)	f	Efavirenzum	650	320
	Pseudo8	[25-38)	m	Efavirenzum	600	210
	Pseudo9	[25-38)	m	Efavirenzum	600	315
	Pseudo10	[0-25)	m	Efavirenzum	450	201
	Pseudo11	[25-38)	m	Efavirenzum	550	270
	Pseudo12	[0-25)	m	Efavirenzum	500	300
…	…	…	…	…	…	…

(b)

gnrlQID	PsNumber	data source $\{(C_i, PsNumber_{C_i})\}$
{00; 0}	4	$(C_1, 2); (C_2, 2)$
{010; 0}	5	$(C_1, 2); (C_3, 1); (C_3, 2)$
{011; 1}	3	$(C_1, 2); (C_3, 1)$
…	…	…

(c)

Fig. 4. Example of data representation in *LDB* (a) *RSDB* (b) and *StRSDB* (c).

4.2 Sharing and Accessing Patient's Data for the Treatment

Hereafter we present a protocol for storing and accessing patients' data in *DR*.

- **Step 1.** Patient generates a shared key (using a hash-function H and a random number $r_{1'}$) with Caregiver (C_1) he visits, this key will to be used for multi-key searchable encryption scheme: $\mathcal{K}^S_{P,C_1} = \mathrm{H}\left(\mathcal{SK}_P \parallel r_{1'}\right)$.
  Since this scheme does not support decryption, the data need to be encrypted twice: once for searching, and once with a traditional encryption scheme like AES, for decryption. Unique AES encryption key ($\mathcal{K}^D_{P,C_1}$) also has to be generated for the caregiver visited by the patient:
  The keys are generated from the Patient's secret key with the use of a smart-card or a mobile device.
- **Step 2.** At the Caregiver's office Patient has to transmit the keys to Caregiver's machine using card-reader device in the Caregiver's office or through a secured channel in the encrypted form: $\mathcal{CT}_1 = Enc\left(\mathcal{K}^S_{P,C_1}, \mathcal{K}^D_{P,C_1}\right)_{\mathcal{PK}_{C_1}}$.
  To ensure integrity Patient's signature ($Sign\left(\mathcal{CT}_1\right)_{\mathcal{SK}_P}$) and Patient's public key ($\mathcal{PK}_P$) for verification are also required.
- **Step 3.** When the *EHR* is generated the content is encrypted with the shared keys ($\mathcal{CT}_2$) and signed with the secret key of Caregiver ($Sign\left(\mathcal{CT}_2\right)_{\mathcal{SK}_{C_1}}$): $\mathcal{CT}_2 = \langle Enc^S\left(EHR\right)_{\mathcal{K}^S_{P,C_1}}, Enc\left(EHR\right)_{\mathcal{K}^D_{P,C_1}} \rangle$.

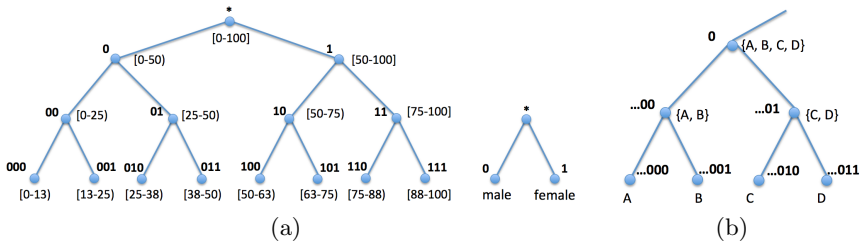

Fig. 5. Binary trees for *qid*s: age and gender (a), and any set-valued *qid* (b).

To improve efficiency the *indexable* version of the encryption scheme proposed in [13] can be used. One can also apply encryption algorithm for searching only to the list of the keywords that describes the content of the *EHR*, such as patient's *ID* (ID_P), caregiver's qualification, date of the visit, symptoms, etc. Pseudonym, with which *EHR* can be associated in *LDB* of C_1: $PS_P^1 = (Enc^S(ID_P)_{\mathcal{K}^S_{P,C_1}})$ is generated using patient's set of uniquely identifiable patient data encrypted with the shared key for search. Therefore, a Caregiver will be able to find and update the data about Patient in an efficient way.

Visiting a caregiver Patient can decide *what data* stored at *DR* he wants to share. For instance, to provide Caregiver 1 an access to the *EHR* generated by Caregiver 2, he only needs to share with Caregiver 1 the keys shared between Caregiver 2 and Patient: $\left(\mathcal{K}^S_{P,C_2}, \mathcal{K}^D_{P,C_2}\right)$. To be able to retrieve Patient's *EHR*(s) based on the pseudonyms (or an attribute of *EHR*) a caregiver has to submit to Cloud Platform, *CP*, a token generated for the ID_P (or an attribute), as well as the deltas for other keys related to the patient, in order to let *CP* adjust the token. Token and deltas are to be computed according to the scheme described in [20].

4.3 Anonymization of Patients' data for Research Purposes

In this subsection we present a description of the algorithm that allows to release medical data for the research purposes from different *LDB*s independently, while preserving the anonymity property of *RSDB*. We ensure that given the consent of the patient caregivers will be able to update *RSDB* with the data about the patient without creating multiple entries that correspond to the same person. Our solution also provides a possibility to recontact the patient through a caregiver that uploads the data.

We consider $\mathcal{N}$ Caregivers that may upload the data to *RSDB*. We assume that *RSDB* is initialized as $(k, k^m) - anonymous$, i.e., an algorithm to achieve $(k, k^m) - anonymity$ proposed in [21] had been applied to the local dataset to build the initial version of *RSDB*. For each *qid* there exist a binary tree, according to which generalization is performed. Figure 5 presents an example of binary trees that are constructed for the single-valued *QID*: age and gender (Figure 5(a)) and also shows an example of representing a set-valued attribute (Figure 5(b)). Our algorithm scales for any number of *qid*s.

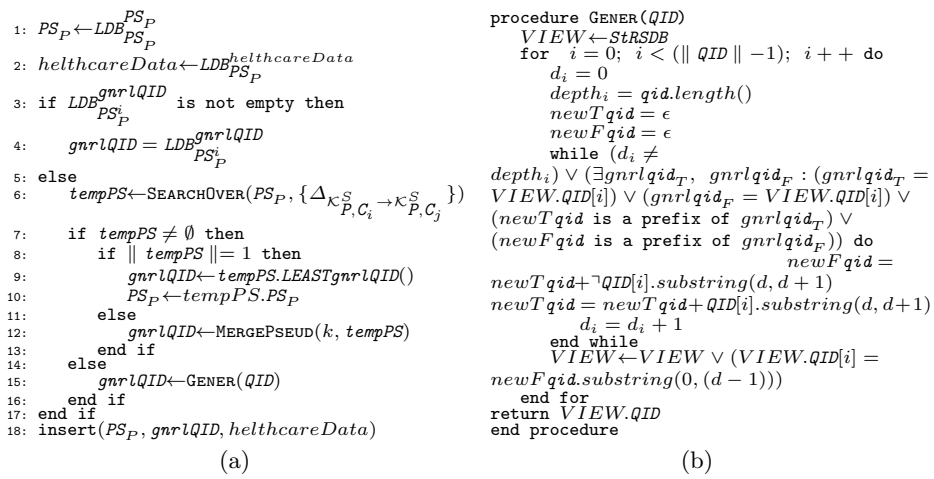

Fig. 6. Pseudocode of the RSDB Update Algorithm (a) and Generalization (b).

Figure 6 shows the pseudocode of the *RSDB* update algorithm and the generalization procedure used in the algorithm. The algorithm for *RSDB* update has to be executed every time a caregiver C_i wants to update *RSDB* with the data about patient P. First, C_i has to check whether he already uploaded the data about P to the *RSDB*. He can query his *LDB* with the patient pseudonym PS^i_P, generated using the shared between P and C_i key $\mathcal{K}^S_{P,C_i}$. If the value in a column *gnrlQID* in a raw that corresponds to the PS^i_P is not empty, than some data about P are already presented in *RSDB* with a combination of generalized *QID* that is described by the vector of binary strings, each represents *gnrlqid*. In this case C_i associates the data of P with these *gnrlQID* that corresponds to PS^i_P (*lines 3,4* of the algorithm presented in Figure 6)(a)). Otherwise, C_i has to perform a SearchOver procedure to check whether there are some data about P that had been upload to *RSDB* by another caregiver $C_j, j \neq i$ (*line 6*). However, this is only possible if P trusts C_i to check this (i.e., if P gave C_i an access to the $\mathcal{K}^S_{P,C_j}$ – key shared between P and C_j).

If SearchOver procedure returns a single pseudonym, C_i will update *RSDB* with the P' data with *gnrlQID* that corresponds to PS^j_P (*lines 8-10*). If the result of SearchOver contains more than one pseudonym, C_i checks whether there is a possibility to merge the pseudonyms related to P by applying MergePseud procedure (*line 12*). Afterwards, C_i will update *RSDB* by uploading the data of P with (the least generalized) *gnrlQID* that corresponds to PS^j_P. If SearchOver procedure returns empty set, then the Gener procedure is performed (*line 15*), and as its output, a combination of the least generalized *gnrlQID* is generated based on the *StRSDB* and the P's *QID*.

SearchOver$(PS_P, \{\Delta_{\mathcal{K}^S_{P,C_i} \rightarrow \mathcal{K}^S_{P,C_j}}\})$ procedure takes as an input the following data: patient's pseudonym (PS_P) generated with the key shared between the patient and caregiver C_i $(\mathcal{K}^S_{P,C_i})$; and a set of the deltas $(\{\Delta_{\mathcal{K}^S_{P,C_i} \rightarrow \mathcal{K}^S_{P,C_j}}\})$

– values generated for the keys ($\{\mathcal{K}^S_{P,C_j}, j \in \overline{1,\mathcal{N}}, i \neq j\}$) related to the patient and shared with the caregiver C_i. Then, according to the schema proposed in [20], a server, which hosts *RSDB*, can perform a search for all the pseudonyms ($\{PS_P\}$) generated by different caregivers with their keys ($\mathcal{K}^S_{P,C_j}, j \in \overline{1,\mathcal{N}}$) (adjusting a pseudonym generated by the caregiver C_i with key ($\mathcal{K}^S_{P,C_i}$) to the one generated by the caregiver C_j with key ($\mathcal{K}^S_{P,C_j}$) without learning neither identity of the P, nor the key $\mathcal{K}^S_{P,C}$) over the column that stores pseudonyms in *RSDB*. As a result a set of pseudonyms together with *gnrlQID* that corresponds to each pseudonyms are being returned.

MergePseud(k, *tempPS*) allows to check whether it is possible to merge pseudonyms that correspond to the same patients but generated by different caregivers. It returns the least generalized *gnrlQID* and merges pseudonyms if does not violate anonymity property of *RSDB*. The input is a parameter k and a set of pseudonyms discovered at the previous step.

Figure 6(c) shows the pseudocode for Gener(*QID*) procedure that is performed to create the least generalized *gnrlQID* for the *QID* of the patient whose data have not been yet upload to *RSDB* (or the data about the patient P might have been uploaded by the caregiver C_j, but a caregiver C_i that wants to upload the data about patient P for the first time does not possess the key $\mathcal{K}^S_{P,C_j}$). Input of the procedure is *QID* – an array of binary strings, each corresponds to one *qid*. Binary strings are constructed according to the representation of the *QID* using binary trees. During the execution each *qid* is considered one after another (the order is based on the importance of the *qid*) and generalized *qid* is formed by querying first *gnrlQID* column of *StRSDB*, and then a view created based on the previously generalized *qid*. The goal is to find the least generalized set *gnrlQID* for a *QID* of the patient such that *StRSDB* already contains at least k entries with this set *gnrlQID* without disclosing the *QID*.

5 Discussion

In this section we analyze the limitations of our model and possible privacy threats. We also suggest the countermeasures against the threats.

5.1 Limitations

We assume that caregiver is trusted, meaning that he respects the medical ethic and will share the data about his patient (including the data produced by other caregivers for the treatment of this patient) only according to the access control policy specified by the patient. However, if (by any reason) the patient does not want the caregiver to be able to access patients' data that are stored in the cloud, a new key has to be created, the data have to be re-encrypted on the server side, e.g., with the means of a proxy re-encryption scheme [1]. We also require an existence of a certification authority that provides the certificates for

public keys and is able to check the identity of a caregiver to ensure that the data aggregated in *RSDB* have been uploaded by a real doctor. However, *CA* does not have an access to the patients healthcare data.

A caregiver *C* can perform a MergePseud procedure only before he makes the first update of *RSDB*. Therefore, in order to merge pseudonyms the following strategy can be applied. According to the access control policy specified by the patient a caregiver that possesses the largest number of the keys may perform SearchOver and MergePseud procedures every time after de-generalization protocol is executed. This will decrease the number of pseudonyms, with which the information about the patient had been uploaded by different caregivers.

With the proposed algorithm we only *preserve* the utility of the *RSDB*. However, *to improve* utility of the data from *RSDB*, the possibility to de-generalize the data from *RSDB* without violation of patients' privacy (during bounded time interval) need to be considered. To define the requirements and selection criteria for $gnrlqid$ to be de-generalized are the next steps in our future work.

Generalization step (procedure Gener of the algorithm) requires going through all the *qid* one by one. However, we assume that the number of *qid*s stored in the *RSDB* is not high and *qid*s are ordered based on their importance with respect to the requirements to the *RSDB*.

5.2 Possible Threats and Countermeasures

If a patient loses his smart card, all the keys can be recovered from the *LDB*s of the caregivers that treat the patient. If the smartcard was stolen it is still difficult access the data or to modify the access control policy for anybody except the patient, because the card is protected with PIN code that is known only to the owner of the card. The limit of attempts to insert a valid PIN code can be set up to prevent brute-force attack.

We assume that the cloud server, where *RSDB* and *DR* are hosted, is honest but curious (it executes protocols and the algorithm correctly but tries to learn about the patient as much as possible). For example, some additional location information can be inferred from the IP address of the device that transmits the data from *LDB*, and these data could be more precise then $gnrlqid$ that stands for the patient address. Therefore, this can violate $(k, k^m)-$ *anonymity*. A straightforward countermeasure is to hide the IP address from the cloud server, e.g., using HTTP proxies or anonymous communication service like Tor [7].

Caregivers could potentially link pseudonyms related to the same patient using the column *PrevPS* in case of pseudonyms merging. To prevent this during the procedure of merging the pseudonyms, the previous pseudonym has to be encrypted together with the information about the caregiver that had created this pseudonym. The cipher text and a parameter that will indicate how many times the pseudonym had been updated will be stored in the column *PrevPS*. Then, it will be possible to find the caregiver that initially uploaded the data (i.e., in case of legal issues), through the caregiver(s) that merged pseudonyms.

6 Conclusion and Future Work

In this paper, we proposed an architecture of a secure and scalable privacy-preserving eHealth cloud system (that allows to store and efficiently search over patient data used for the treatment), and an algorithm that allows to build a database with patients' data for the research purposes.

In future work we will focus on the implementation of the architecture proposed in this paper and on its evaluation using a synthetic dataset (http://omop.org/OSIM2), and real patient data from our medical partners in the framework of ISyPeM2 project (www.nano-tera.ch/projects/368.php). We will also work towards de-generalization of *RSDB* to improve utility of the data. Finally, we will focus on improving efficiency of proposed solution by extending representation of the *QID* (from binary trees to n-ary trees) and employing agent based coordination model for the construction of *RSDB*.

Acknowledgements

This work was supported by the Nano-Tera initiative, in the framework of an RTD project ISyPeM2: developing therapeutic drug monitoring by designing a point-of-care system to measure drug concentration in blood samples and adjust dosage accordingly.

References

1. G. Ateniese, K. Fu, M. Green, and S. Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Trans. Inf. Syst. Secur.*, 9(1):1–30, 2006.
2. M. M. Baig, J. Li, J. Liu, and H. Wang. Cloning for privacy protection in multiple independent data publications. *Proceedings of the 20th ACM international conference on Information and knowledge management - CIKM '11*, page 885, 2011.
3. E. Benoist and J. Sliwa. How to Collect Consent for an Anonymous Medical Database. *HEALTHINF*, 2014.
4. I. Blake, G. Seroussi, N. Smart, and J. W. S. Cassels. *Advances in Elliptic Curve Cryptography* . Cambridge University Press, New York, NY, USA, 2005.
5. D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. *EUROCRYPT*, volume 3027 of *Lecture Notes in Computer Science*, pages 506–522. Springer, 2004.
6. C. Clifton and W. Jiang. CERIAS Tech Report 2005-134 Information Assurance and Security Privacy-Preserving Distributed k -Anonymity. 2005.
7. R. Dingledine, N. Mathewson, and P. F. Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, 2004.
8. A. Dubovitskaya, V. Urovi, M. Vasirani, K. Aberer, A. Fuchs, T. Buclin, Y. Thoma, and M. Schumacher. Privacy preserving interoperability for personalized medicine. *Swiss Medical Informatics*, September 2014.
9. B. S Elger, J. Iavindrasana, L. Lo Lacono, H. Müller, N. Roduit, P. Summers, and J. Wright. Strategies for health data exchange for secondary, cross-institutional clinical research. *Computer methods and programs in biomedicine*, 99:230–251, 2010.

10. A. Gkoulalas-Divanis and G. Loukides. *Anonymization of Electronic Medical Records to Support Clinical Analysis*. Springer Briefs in Electrical and Computer Engineering. 2013.
11. A. Gkoulalas-Divanis, G. Loukides, and J. Sun. Publishing data from electronic health records while preserving privacy: A survey of algorithms. *Journal of Biomedical Informatics*, 50:4–19, 2014.
12. V. Gotta, N. Widmer, M. Montemurro, S. Leyvraz, A. Haouala, LaurentA. Decosterd, Chantal Csajka, and Thierry Buclin. Therapeutic drug monitoring of imatinib. *Clinical Pharmacokinetics*, 51(3):187–201, 2012.
13. J. Helfer, S. Valdez, R. A. Popa, E. Stark, N. Zeldovich, M Frans Kaashoek, and Hari Balakrishnan. Building web applications on top of encrypted data using Mylar. *Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation* Pages 157-172 2014.
14. L.Ibraimi, M. Asim, and M. Petko. Secure Management of Personal Health Records by Applying Attribute-Based Encryption. *6th International Workshop on Wearable Micro and Nano Technologies for Personalized Health (pHealth)*, 2009.
15. M. Li, Sh. Yu, K. Ren, and W. Lou. Securing Personal Health Records in Cloud Computing: Patient-Centric and Fine-Grained Data Access Control in Multi-owner Settings. *Security and Privacy in Communication Networks*, pages 89–106, 2010.
16. M. Li, Sh. Yu, and Y. Zheng. Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute-Based Encryption. *IEEE Tranyactions on Parallel and Distributed Systems* 24(1):131–143, 2013.
17. Z.-R. Li, E.-Ch. Chang, K.-H. Huang, and F. Lai. A secure electronic medical record sharing mechanism in the cloud computing platform. *2011 IEEE 15th International Symposium on Consumer Electronics (ISCE)*, pages 98–103, June 2011.
18. L. Lo Iacono. Multi-centric universal pseudonymisation for secondary use of the EHR. *Studies in health technology and informatics*, 126:239–47, January 2007.
19. A. Lounis, A. Hadjidj, A.Bouabdallah, and Y. Challal. Secure Medical Architecture on the Cloud Using Wireless Sensor Networks for Emergency Management. *Eighth International Conference on Broadband and Wireless Computing, Communication and Applications*, pages 248–252, 2013.
20. R. A. Popa and N. Zeldovich. Multi-key searchable encryption. *Cryptology ePrint Archive, Report 2013/508*, 2013.
21. G. Poulis, G. Loukides, and A. Gkoulalas-Divanis. Anonymizing data with relational and transaction attributes. *Machine Learning and Knowledge Discovery in Databases*, *Lecture Notes in Computer Science*, pages 353–369. Springer, 2013.
22. A. Solanas, A. Martinez-Balleste, and J. Mateo-Sanz. Distributed architecture with double-phase microaggregation for the private sharing of biomedical data in mobile health. *IEEE Transactions onInformation Forensics and Security*, 2013.
23. D. X. Song, D. Wagner, and A. Perrig. Practical techniques for searches on encrypted data. In *Proceedings of the 2000 IEEE Symposium on Security and Privacy*.
24. L. Sweeney. K-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, 2002.
25. V. Urovi, A. C. Olivieri, A. Brugués de la Torre, S. Bromuri, N. Fornara, and M. Schumacher. Secure P2P cross-community health record exchange in IHE compatible systems. *International Journal on Artificial Intelligence Tools*, 23(1), 2014.
26. L. Xu and A. B Cremers. A Decentralized Pseudonym Scheme for Cloud-based eHealth Systems. *HEALTHINF*, 2014.
27. Sh. Yu, C. Wang, K. Ren, and W. Lou. Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing. *INFOCOM, IEEE*, 2010.