Privacy Protocol for Linking Distributed Medical Data

Daniel Janusz, Martin Kost, and Johann-Christoph Freytag

DBIS Group
Humboldt-Universität zu Berlin
Unter den Linden 6
10099 Berlin, Germany
{janusz,kost,freytag}@informatik.hu-berlin.de

Abstract. *Health care* providers need to exchange medical data to provide complex medical treatments. In general, regulations of *privacy protection* define strong constraints for exchanging such personal data within a distributed system. Privacy-preserving *query protocols* provide mechanisms for implementing and maintaining these privacy constraints. In this paper, we introduce a new two-phase protocol for protecting the privacy of patients. The first phase implements a private record linking. Thereby, the queried data provider links the received query with matching records in his data base. In the second phase, a requestor and a data provider perform an authorized exchange of matched patient data. Thus, our protocol provides a method for health care providers to exchange individual medical data in a privacy preserving manner. In contrast to other approaches, we actively involve patients in the exchange process. We apply the honest-but-curious adversary model to our protocol in order to evaluate our approach with respect to complexity and the degree of privacy protection.

1 Introduction

Health care providers use Hospital Information Systems (HIS) such as *Orbis* [1] or *i.s.h.med* [17] to manage patient data. In the course of a patient's treatment, lots of medical data is collected by various health care providers, e.g., hospitals or medical laboratories. In general, the HIS of these providers are not connected. Every provider manages a separate database to store the patient records.

Some medical treatments may involve the cooperation of several health care providers. Therefore, the involved health care providers exchange the required medical data, e.g., access blood analysis data from medical laboratories. Privacy concerns arise from communicating and processing such data. Moreover, exchanging medical records is often impossible due to legislative privacy reasons. For example, the privacy regulations of the European Union [8] restrict automatic exchange of personal data. A common way to overcome these restrictions is to hand over the records to the patient and he turns them over to the subsequent health care provider. Thus, patients implicitly authorize the receiving party to access this data. This procedure protects privacy, because the patient decides who should receive his/her data. However, the *data avoidance principle* is not addressed sufficiently. The approach described misses to check whether the receiver requires all record information for the intended treatment. Besides its benefits in terms of privacy, a manual data transfer of patient data may be too slow, e.g., in emergency scenarios.

In this paper, we present a protocol for exchanging medical data of one patient electronically while protecting the patient's privacy. Our approach protects patient's privacy with the same quality as the ”manual approach”. In addition, we address the data avoidance principle and provide an authorized remote data access mechanism. In our protocol, we use a two-phase approach. During the first phase, we identify patient records in the remote databases that may belong to the queried patient. Therefore, we utilize the observation that health care providers often collect the same medical attributes by means of family predispositions or baseline examinations. In the second phase, the required patient data is exchanged.

Our protocol protects privacy by guaranteeing the following two principles. First, the protocol enables health care provider to query remote data sources about a specific patient while not disclosing the identity of this patient. We call such queries *private fuzzy queries*. Second, the queried party requires that the query result can only be read if the patient is present and if the patient agrees to reveal the result data to the requestor. We call this principle an *authorized data exchange*.

In our approach, we guarantee that only those requesting parties get patient data that have the corresponding permission for receiving them. Moreover, we assume that health care providers do not trust each other; however we assume that they act honestly; i.e., all health care providers return accurate patient data. Such an approach resembles the *honest-but-curious adversary model* [11]. In a medical scenario, this adversary model is a proper assumption, as health care providers have no motivation for faking patient data. We will show that if only accurate patient data is submitted there is no attack on our protocol that may harm the privacy of patients.

As discussed before, our protocol aims at implementing and improving the manual approach of transfering patient data. Our approach includes that we do not implement more complex privacy principles such as limited data retention. There are still more open issues such as to update broken encryption keys at all data sources. Another challenge is to optimize our protocol in terms of communication overhead.

Outline The remainder of this paper is organized as follows. Section 2 introduces preliminaries and basic definitions. In Section 3 we present our protocol, which we evaluate in terms of privacy and complexity in Section 4. Section 5 provides an overview of related work before Section 6 concludes the paper by outlining future work.

2 Background

In this section, we provide existing preliminaries and new definitions for introducing our approach.

2.1 Problem Definition

We implement a two-phase protocol that links distributed medical records. In the first phase, we identify health care providers that collect data of one specific patient. In the second phase, we exchange medical examination results of this patient that are relevant to the patient's intended medical treatment. Linking records stored on different databases has been widely discussed as the *record linkage problem* [9]. Record linkage is the process of identifying records belonging to the same entity, across two or more data sets. In contrast to general record linkage, our approach only localizes records belonging to one specific entity.

We call two records of different data sets *matching records* if these records belong to the same individual. In our scenario, we use *medical attributes* to find candidates for matching records. Medical attributes describe characteristics of a patient that belong to a patient's anamnesis, e.g., family predisposition, or biometric attributes, e.g., eye color.

Simple demographics like zip code and birth date are often sufficient to identify single individuals in a statistical database [18]. Medical attributes have similar characteristics [12]. Therefore, medical attributes are so-called *quasi-identifier*. To deal with measurement inaccuracies we can use ranges instead of exact values, e.g., if the height of a patient is 1,84m we search for a patient who's height is between 1,80m and 1,90m.

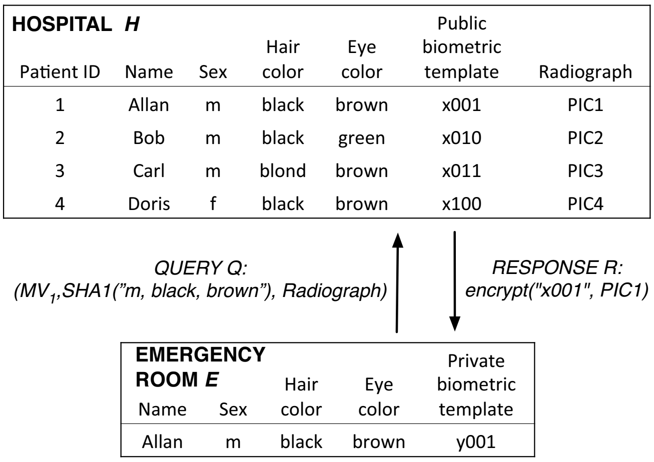
DEFINITION 1 (Table) *A table $\mathcal{T}$ is a set of tuples (records) with a schema $S = (A_1, ..., A_t), t \in \mathbb{N}$ and $\{A_1, ..., A_t\}$ is a set of attributes.*

DEFINITION 2 (Key) *Given a table $\mathcal{T}$ including a set of attributes $\mathcal{A}$. A key $\mathcal{K}$ of table $\mathcal{T}$ is a subset of the table attributes $\mathcal{K} \subseteq \mathcal{A}$ that uniquely identifies every record of the table $\mathcal{T}$.*

DEFINITION 3 (Quasi-Identifier) *Given a database table $\mathcal{T}$ including a set of personal attributes $\mathcal{A}$. A quasi-identifier in $\mathcal{T}$ is a subset of attributes $QID \subseteq \mathcal{A}$ that can be joined with external information to re-identify individual records with sufficiently high probability [19].*

We use quasi-identifying attributes to identify the searched patient. In order to hide the real identity of the searched patient from the queried party, we perform only fuzzy record matching. Fuzziness means not uniquely identify one person or in our context not less than k people. We call an identifier that performs a fuzzy record matching a *fuzzy matching pattern*. An example for these concepts is illustrated in Figure 1. In the table of hospital H, the attribute set $\{$*sex, hair color, eye color*$\}$ forms a quasi-identifier. Consequential, *(sex, hair color)* forms a fuzzy matching pattern.

DEFINITION 4 (Fuzzy Matching Pattern) *Given a database table $\mathcal{T}$ including a set of personal attributes $\mathcal{A}_P$ and a quasi-identifier $QID \subseteq \mathcal{A}_P$ in $\mathcal{T}$. A fuzzy matching pattern $FMP = (A_y, ..., A_z)$ is an ordered list of attributes with: $\{A_y, ..., A_z\} \subset QID$.*



HOSPITAL *H*					Public	
			Hair	Eye	biometric	
Patient ID	Name	Sex	color	color	template	Radiograph
1	Allan	m	black	brown	x001	PIC1
2	Bob	m	black	green	x010	PIC2
3	Carl	m	blond	brown	x011	PIC3
4	Doris	f	black	brown	x100	PIC4

QUERY Q:
(MV$_1$,SHA1(”m, black, brown”), Radiograph)

RESPONSE R:
encrypt("x001", PIC1)

EMERGENCY ROOM *E*				Private
		Hair	Eye	biometric
Name	Sex	color	color	template
Allan	m	black	brown	y001

Fig. 1. Hospital data and query example

2.2 Biometric Templates

Encryption methods [11] prevent unauthorized parties to access private data. Strong encryption keys are required to provide a secure encryption. By means of *biometric templates* such encryption keys can be generated. A biometric template is a digital description of distinct human characteristics that have been generated from biometric samples. After the generation of a biometric template the input samples must be deleted. Biometric templates are fault-tolerant against imprecise biometric input samples. In [3] the authors introduce a method to protect biometric templates based on *pseudonym identifier.* A pseudonym identifier is a diversifiable, protected binary strings, which does not reveal any information about the originally biometrics or the real identity of the related person.

Biometrics has been proofed to be valid for strong encryption [5]. We will utilize protected biometric templates to generate *public/private key-pairs* [11]. In the first version of our protocol, we will use a fixed set BIO of biometric sample attributes. In general, every party could use a different input for generating unique pseudonym identifier. In our approach, every patient holds a private and public encryption key. The private key is equal to the pseudonym identifier generated from BIO and will be called private biometric template BIO_{pr}. As public key and private key are interlinked, the public key must be generated using the private key, which we will call public biometric template BIO_{pu}.

2.3 Privacy Issues in Data Linking Scenarios

Usually, record linking is performed across data sets from different data sources. Privacy concerns arise whenever personal data is exchanged across companies. *Private record linkage* [10] addresses these concerns. However, the most private linking approaches do not consider one important privacy principle. That is, individuals have no control over the linkage process. In our approach, the individuals agree or disagree to the intended linkage process.

In the first phase of our protocol, we use special queries to identify matching records. We want even the query to be protected. Thus, the query must not leak any personal information. In order to realize such privacy-preserving queries, we define the notion of k-disguisebility. A *k-disguised* query performs fuzzy record matching. Let q be a query and ds a data source than we write $q(ds)$ to denote the result of the query q applied at the data source ds.

DEFINITION 5 (k-disguisebility) *Let $\mathcal{D}$ be a set of data sources in a distributed database setting, where every data source stores the same set of keys and (quasi-)identifying attributes. Let q be a query originated at one of the data sources $dr \in \mathcal{D}$ on keys or identifying attributes. The query q is said to satisfy k-disguisebility if and only if:*

$$|q(dr)| \geq k \qquad \textit{and} \qquad \left| \bigcup_{d \in \mathcal{D} \setminus dr} q(d) \right| \geq k, \qquad (k \in \mathbb{N}).$$

The first property that must hold for k-disguisebility means that the result of query q at dr consists of at least k records. The second property states that the union of the results of query q at each of the remaining data source in $\mathcal{D} \setminus dr$ also consists of at least k records.

For an example consider the databases of hospital H (see Figure 1) and hospital G (see Figure 2). For these two databases a query on all tuples having ”Hair Color = black” conforms to 2-disguisebility.

3 Privacy-Preserving Linking Protocol

In this section, we introduce our protocol. First, we exemplify the concepts of the approach before describing our protocol formally.

3.1 Example Scenario

We illustrate our approach using the following emergency scenario. After a serious car accident has occurred, a patient named Allan arrives at the emergency room E and needs a treatment. The physician wants to compare the current radiograph of his neck with an older image. Unfortunately, there is no prior record of Allan in the local HIS. The physician PE at E is allowed to access the remote database of hospital H (Figure 1) and starts a request for radiographs of Allan.

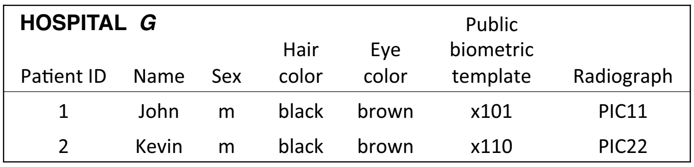
In order to protect the privacy of Allan the query Q conforms to k-disguisebility. Therefore, the query Q is generated as follows: First, PE selects a fuzzy matching pattern FMP consisting of the attribute list $(sex, hair\ color, eye\ color)$. Next, the strings of Allan's values of the fuzzy matching pattern's attributes are concatenated. The result is hashed $h_{Allan} = \mathit{SHA1}(\text{”}male, black, brown\text{”})$ using *SHA1* [16]. Finally, the query $Q = (FMP, h_{Allan}, \{Radiograph\})$ is transmitted to hospital H. In section 4, we present an analysis verifying that Q now conforms to our notion of private fuzzy matching queries.

When hospital H receives the query Q from E, Q is processed as follows: For every record in H's database the *SHA1* hash h_{temp} of the respective combined values of the attributes in FMP is generated. If h_{temp} equals h_{Allen}, the considered record matches. Thus, the image in the attribute *Radiograph* is a candidate for being the searched prior radiograph. The algorithm adds all images of the matching records to the result set R. As seen in the table of Figure 1 h_{Allen} only matches the record with $PatientID = 1$. Therefore, *PIC1* may be a radiograph of Allan.

The database of H contains the attribute *Public biometric template*. Public biometric templates BIO_{pu} are generated and stored at the first time a patient visits hospital H. The public biometric template is now used to encrypt the corresponding radiographs of the matching records in the response R. In case of Allan's record *PIC1* is encrypted using his public biometric template BIO_{pu}=*”x001”* as encryption key.

After receiving the response R, E tries to decrypt all images in R using the private biometric template of Allan as decryption key. As Allan is physically present the emergency room gets Allan's permission to generate his private biometric template BIO_{pr} in order to access his previous medical data.

We now consider an example that includes an additional hospitals G (Figure 2). For G's database there will be two matching hashes for query Q. Therefore, the response of hospital G will contain the two encrypted radiographs *PIC11* and *PIC12*. As Allan's private biometric template BIO_{pr} will only decrypt radiographs belonging to Allan, it is not possible to decrypt radiograph belonging to other patients such as John and Kevin.



HOSPITAL *G*

Patient ID	Name	Sex	Hair color	Eye color	Public biometric template	Radiograph
1	John	m	black	brown	x101	PIC11
2	Kevin	m	black	brown	x110	PIC22

Fig. 2. Data of the second hospital G

3.2 General Approach and Protocol

We now introduce our protocol formally. As mentioned before, our protocol protects privacy by implementing two selected privacy requirements (see Section 1). We introduce the following new type of private queries in order to realize our first privacy requirement, i.e., use private fuzzy matching queries. Let $\mathcal{D}$ be a set of independent medical data sources, e.g., hospitals or medical laboratories. Every data source in $\mathcal{D}$ stores or has access to the public biometric template BIO_{pu} of every individual in its database. Consider a patient $\mathcal{P}$ being at the health care provider $\mathcal{H}$. Let $\mathcal{A}$ be a set of all medical attributes of patient $\mathcal{P}$.

DEFINITION 6 (Privacy-Preserving Linking Query) *A privacy-preserving linking query $Q_{pr} = (\mathcal{FMP}, hash_{FMP}, X)$ on medical data of patient $\mathcal{P}$ is a triple with:*

1. *$\mathcal{FMP} \subset \mathcal{A}$ is a fuzzy matching pattern containing n attributes that are present in every schema of the databases in $\mathcal{D}$.*
2. *$hash_{FMP}$ = SHA1($val_1, ..., val_n$) with val_i, $1 \leq i \leq n$, being the concrete value of the patient $\mathcal{P}$ for the i-th attribute in the fuzzy matching pattern $\mathcal{FMP}$.*
3. *$X \subset \mathcal{A}$ are the attributes of the patient $\mathcal{P}$, the health care provider $\mathcal{H}$ is interested in.*

If the fuzzy matching pattern $\mathcal{FMP}$ includes attributes that are not present in the schema of a queried databases, the query cannot be answered. However, in a medical scenario it is a realistic assumption that health care providers store a similar set of patient attributes. In our observations, we found that health care providers often collect the same attributes by means of family predispositions or baseline examinations. For example, every patient has to fill in an anamnesis questionnaire during the admission to a hospital. In many cases, such data may uniquely identify this patient.

In the first phase of our protocol the query Q_{pr} is sent to every data source in $\mathcal{D}$. Thereby, we assume that a global authentication method exists that identifies $\mathcal{H}$ as a valid health care provider, which is allowed to pose queries. Moreover, we assume communication is supposed to be end-to-end encrypted. A receiver $d \in \mathcal{D}$ of the query Q_{pr} processes a query as follows: For every record in d's database the *SHA1* hash h_{temp} of the corresponding combined values of the attributes in the fuzzy matching pattern $\mathcal{FMP}$ is generated and compared to $hash_{FMP}$. If h_{temp} equals $hash_{FMP}$, then the record is marked as a matching record.

In order to realize our second privacy requirement, i.e., perform only authorized data exchange, every response R_{pr} of a data source $d \in \mathcal{D}$ to the query Q_{pr} is constructed as illustrated in Algorithm 1. It is important to highlight that for the encryption in line 5 the public biometric template BIO_{pu} of the corresponding individual must be used as encryption key. This guarantees that the encrypted items in the response R_{pr} can only be decrypted, if the corresponding indivial has agreed and provides his private biometric template BIO_{pu}. After constructing the response, R_{pr} must be returned to $\mathcal{H}$.

Algorithm 1. Generating the query response R_{pr}.

```
$r_k = \emptyset$;
for each record in $d$ marked as a matching record, do:
    for each $x$ in $X$, do:
        if $x$ present in $D$, do:
            encrypt the concrete value/item of $x$
              using the public biometric template $BIO_{pu}$
              of the processed record as encryption key;
            add the encrypted value/item to the set $R_{pr}$;
        end if
    end for each
end for each
```

During the last step, the health care provider $\mathcal{H}$ tries to decrypt the items in every response using $\mathcal{P}$'s private biometric template BIO_{pr}, which can only be gained directly from the patient $\mathcal{P}$. After all responses are evaluated the private biometric template BIO_{pr} of the patient $\mathcal{P}$ is deleted from $\mathcal{H}$'s databases.

4 Evaluation

To evaluate the proposed protocol, we analyze our protocol in terms of privacy and complexity. For the latter, we briefly compared our protocol with those of others.

4.1 Privacy Evaluation

As we stated in our introduction (Section 1), faking patient data is the only attack on our protocol that might harm the privacy of patients. In order to proof this statement, we apply the honest-but-curious adversary model to our protocol and look at attacks an initiator or a receiver of a query may execute.

An attacker may pose fake queries within our approach, i.e., a requester who is not authorized by the patient poses a query. A query may be generated, if the requester knows some medical attribute values of the patient. If the patient is not present the requester has no access to the private biometric template BIO_{pr}. Therefore, the requester will be unable to decrypt any response.

The worst case scenario for our protocol is, if the private biometric template BIO_{pr} of a patient gets leaked, i.e., someone gets access to the private biometric template BIO_{pr} who is not authorized by the patient. Such leakage of the private biometric template BIO_{pr} does not harm the privacy of the patient as biometric templates do not reveal any information about the originally biometrics or the real identity of the related person. In case of a leakage the public biometric template BIO_{pu} for this patient can no longer be used and must be changed. An infinite number of different protected biometric templates of a patient is available [3]. Thus, the challenge is to find out if a private biometric template got leaked and to update the public biometric template at all data sources. In the current state of our protocol, we do not handle such leakage.

A malicious receiver of a query $Q_{pr} = (\mathcal{FMP}, hash_{FMP}, X)$ may try to learn personal data from the query Q_{pr}. $\mathcal{FMP}$ and X include only sets of attribute types and include no personal information. Thus, the hash value $hash_{FMP}$ is the only part of the query that may be attacked. Brute-force is the only known attack on hash values created with *SHA1*. This attack may decrypt the original values of $hash_{FMP}$, if the attributes in $\mathcal{FMP}$ cover only small domains, e.g., the attributes in the fuzzy matching pattern $\mathcal{FMP} = (sex, hair\ color, eye\ color)$ in our example in section 3.1 cover small domains. In the worst case, an attacker gains the same knowledge that a query receiver having some matching records learns. As the query Q_{pr} must be k-disguised, the original values of $hash_{FMP}$ do only identify a group of at least k individuals. Moreover, a malicious receiver cannot learn from the query Q_{pr} whom of the individuals in the identified group the query poser was looking for. Finally, an attacker cannot even learn, if the queried patient is part of this group.

The challenge for a requester is to select an adequate fuzzy matching pattern for the query Q_{pr}. To generate a statistical large number of matching records, the requester may use local statistics about its database or global statistics of medical attributes.

Within the honest-but-curious attacker model our approach protects the privacy of patients. At last, we give an example of what may happen under a less restrictive attacker model where parties may deviate from the protocol. A malicious receiver of a query $Q_{pr} = (\mathcal{FMP}, hash_{FMP}, X)$ may harm the patient in two ways. First, matching records may be withheld. In this case, no private information is leaked. Second, if there are matching records, a malicious receiver may response false results. In both cases we end up having false or insufficient data of the patient, which may result in false-treatment. Faking query results can seriously harm the patient's health. Thus, our protocol may only be applied in scenarios where honest-but-curious behavior is a proper assumption.

4.2 Costs and Overhead

Most other approaches for privacy-preserving query protocols use secure multi party computations or homomorphic encryption (see Section 5 for related work). Current approaches use blocking strategies, which help to reduce costs. Nevertheless, the complete dataset of the queried party has to be encrypted to find the matching records. In our protocol, we use hashing to find matching records. Hash algorithms such as *SHA1* are much less costly than secure multi party computations or homomorphic encryption. Moreover, once generated the hash values may be stored and reused for further query execution.

In our approach, responses can include false positive record matchings. This protects privacy but also increases communication complexity. In the worst case, the response includes the complete dataset of the queried party. To prevent such scenarios, the queried parties could demand a new query that uses a different fuzzy matching pattern. In general, we assume that a requester will try to reduce communication time and size. Therefore, requesters try to select adequate fuzzy matching patterns.

5 Related Work

Several approaches exist which provide methods for privacy-preserving query processing in distributed database settings. Exemplary, we compare our protocol with three recently proposed approaches. Subsequently, we discuss general concepts used in most of the existing approaches.

Chow et. al. [4] introduce a two-party computation model for privacy-preserving query execution. In their approach, they utilize the concept of two non-colluding parties. One party obfuscates the query and another party computes the query result (set intersection) on encrypted data. The advantage compared to our approach is that the query requester does not learn anything about the data sources. In our approach, we do not rely on non-colluding parties. Moreover, we do not require to hide the queried data sources as the patient explicitly permits the query requester for accessing his data from these remote data sources.

Allman et. al. [2] proposed an approach that resembles our query methodology to some extend. In this paper, the authors evaluate private queries for detecting attacks on network infrastructures. The queries include a hashed communication pattern. Every queried data source must try to rebuild the hash with its local data. Disregarding the different application domain of this approach, our approach is much more general and secure. Instead of a fixed fuzzy matching pattern we use a variable fuzzy matching pattern. Moreover, the proposed private matching approach is vulnerable as it suffers from a small domain of the fuzzy matching pattern (see section 4.2 for a privacy evaluation of our approach).

In [14] private record linkage is realized by combining secure blocking and secure matching. Before searching for matching records, all records are grouped into blocks. Blocking results in a huge decrease of record comparisons. Nevertheless, our matching method is more efficient as we do not use secure multi-party computation (see section 4.1 for efficiency and complexity of our approach). Furthermore, as in most other approaches individuals have no control over the linkage process.

Most of the existing privacy-preserving query approaches use some basic concepts. Thus, we evaluate the use of secure multi-party computation, third parties and privacy policies for our application scenario.

5.1 Secure Multi-Party Computation

Methods for privacy-preserving query execution aim to prevent any revealing of personal information about the queried individual. Usually, existing implementations apply private record matching methods for finding query results. In order to protect privacy, existing approaches [13, 14] combine record matching methods [6] with secure multi-party computations such as *secure set intersection* [10] or *private matching* [20].

Methods of secure set intersection as well as private matching methods focus on the problem of entities trying to find common data elements in their databases, without revealing the complete record pool to one another. In secure set intersection methods, the common elements are computed using *homomorphic encryption* [11]. On the other hand, private matching methods are based on commutative encryption. Both concepts suffer from high communication cost, because at least one party has to submit his complete encrypted dataset. Furthermore, computations on homomorphic encrypted data are very costly. In our approach, we only submit a very small subset of matching records and use hashing methods together with faster asymmetric encryption.

Disregarding the drawbacks, there are some benefits of using private record matching. Our matching algorithm is not capable of *approximate matching.* In order to handle typographical errors approximate matching algorithms use phonetic algorithms, string distance-based methods or bloom filters to find matching records [6].

5.2 Third Parties

The incorporation of *untrusted third parties* is another concept used for privacy-preserving queries [7]. An untrusted third party, e.g., cloud storage, may be used for storing encrypted records. The encrypted records are publicly available, but only authorized uses have access to the decryption keys. In contrast to our protocol, the integration of untrusted third parties involves more communication effort, as all parties have to upload their data. Another serious thread occurs if some encryption keys get leaked. Once encrypted records are released to an untrusted storage provider those record might stay online forever.

In complement to untrusted third parties, some approaches use *trusted third parties* to protect privacy. There are two functionalities which a trusted third party may provide. First, the trusted party may calculate the query results. In this case, all parties have to submit their data to the trusted party. Second, the trusted party may guarantee that all participating parties are honest and not curious. Many reasons exist why we should avoid using trusted third parties. For example, a trusted party may learn all input data for a query. Thus, there is a single point of failure which can be attacked. Furthermore, the approach requires to get control over the participating parties within a distributed database setting. Therefore, it is hard to guarantee an intended behavior for such distributed systems. In our approach, there does not exist a central point of failure and we require only minor behavioral guarantees of other parties.

5.3 Privacy Policies

In our protocol, the patient gives his consent for querying his data by providing his private biometric template. Using *privacy policies* is a common way to store and submit privacy preferences of individuals. Privacy policies such as *XACML*

[21] or *P3P* [15] allow to define criteria such as the purpose of use, to restrict the set of external recipients, or retention constraints. Many applications running in distributed systems do not provide direct control for enforcing privacy policies. Thus, individuals have to trust in the self regulation of services providers. Moreover, often an expert is needed to write down or understand privacy policies. Our protocol establishes a simple mechanism which enables individuals for controlling the initialization of the query process. Moreover, we offer a method to securely enforce access control in a distributed database setting. However, we do not provide to define and enforce complex privacy preferences such as limited retention criteria.

6 Conclusions and Future Work

The current development of information systems in the healthcare domain raises new challenges for the integration of appropriate privacy-preserving query protocols. Often, existing implementations use a centralized trusted party or a complex multi party computation approach for protecting privacy while query processing. In this paper, we introduced a new two-phase protocol for privacy-preserving exchange of medical data. Within this protocol, we use private fuzzy queries to link distributed medical data of a patient. In addition, we provide a simple and effective access control mechanism. In contrast to other approaches, the patients directly authorize the intended query process. We analyzed and defined requirements for protecting the privacy of patients in a medical treatment scenario. Based on these requirements, we applied the honest-but-curious adversary model to our protocol in order to evaluate the introduced approach regarding complexity and privacy protection.

Currently, we are implementing a prototype of the proposed protocol. We are going to use this prototype for implementing a medical screening scenario. Next, we evaluate our implementation using real screening data.

In the future, we are going to provide a formal proof that our protocol implements our privacy requirements. Furthermore, we will include a mechanism for updating the public biometric templates in all data sources. Another challenge will be to extend the fixed schema of the fuzzy matching patterns by using individual fuzzy matching patterns for every data source. In order to improve record matching accuracy, we also want to include approximate matching capabilities in our protocol.

References

1. Agfa HealthCare, `http://www.agfahealthcare.com`.
2. M. Allman, E. Blanton, V. Paxson, and S. Shenker. Fighting coordinated attackers with cross-organizational information sharing. In Proceedings of 5th workshop on hot topics in networks (HotNets), 2006.
3. J. Breebaart, C. Busch, J. Grave, and E. Kindt. A Reference Architecture for Biometric Template Protection based on Pseudo Identities. In BIOSIG, 2008.

4. S. S. M. Chow, J.-H. Lee, and L. Subramanian. Two-party computation model for privacy-preserving queries over distributed databases. In NDSS 2009, San Diego, The Internet Society, 2009.
5. Y. Dodis, L. Reyzin, A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In EUROCRYPT, 2004.
6. A. Elmagarmid, G. Panagiotis, and S. Verykios. Duplicate record detection: A survey. IEEE Transaction on Knowledge and Data Engineering, 2007.
7. F. Emekci, D. Agrawal, A. E. Abbadi, and A. Gulbeden. Privacy Preserving Query Processing Using Third Parties. In ICDE, 2006.
8. European Union. Directive 95/46/EC of the European parliament and of the council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data; 1995.
9. I. P. Felligi and A. B. Sunter. A theory for record linkage. Journal of the American Statistical Society, 64:1183–1210, 1969.
10. M.J. Freedman, K. Nissim, B. Pinkas. Efficient Private Matching and Set Intersection. In EUROCRYPT, 2004.
11. O. Goldreich. The Foundations of Cryptography, Vol. 2, Chapter General Cryptographic Protocols. Cambridge University Press, 2004.
12. S. Gomatam, R. Carter, M. Ariet, and G. Mitchell. An empirical comparison of record linkage procedures. Statistics in Medicine, p. 1485–1496, 2002.
13. A. Inan, M. Kantarcioglu, E. Bertino, and M. Scannapieco. A hybrid approach to private record linkage. In ICDE 2008, Cancun, Mexico. IEEE Computer Society 2008.
14. A. Karakasidis and V.S. Verykios. Secure blocking + secure matching = secure record linkage. Journal of Computing Science and Engineering, p. 223–235, 2011.
15. P3P Preference Exchange Language v. 1.0 (APPEL1.0). W3C, 2002. www.w3.org/TR/P3P-preferences.
16. NIST. FIPS 180-3: Secure hash standard (SHS). Technical report, National Institute of Standards and Technology (NIST), 2008. http://csrc.nist.gov/publications/fips/fips180-3/fips180-3_final.pdf.
17. Siemens Healthcare, http://www.medical.siemens.com.
18. L. Sweeney. Simple demographics often identify people uniquely. Carnegie Mellon University, Data Privacy Working Paper, 3, 2000.
19. L. Sweeney. *k*-anonymity: a model for protecting privacy. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, p. 557–570, 2002.
20. Y. Li, J. Tygar, and J. Hellerstein. Private matching. In Computer Security in the 21st Century, p. 25–50, 2005.
21. eXtensible Access Control Markup Language (XACML) v. 2.0. OASIS Standard, February 2005. http://docs.oasis-open.org/xacml/2.0/accesscontrol-xacml-2.0-core-spec-os.pdf.