HEALTH+Z: Confidential Provider Selection in Collaborative Healthcare P2P Networks

Sergej Zerr[*], Odyseas Papapetrou[**], Elena Demidova[*]
[*]L3S Research Center, Hannover, Germany
{zerr,demidova}@L3S.de
[**]SoftNet lab, Technical University of Crete, Chania, Greece
papapetrou@softnet.tuc.gr

ABSTRACT

Many real world applications in the healthcare domain would gain a substantial advantage from sharing and search technologies available for P2P infrastructures if these technologies could provide required confidentiality guarantees. Currently, DHT-based indexes which are typically applied for effective and efficient information sharing and retrieval in P2P networks do not offer sufficient confidentiality for the patient data in a healthcare network and medical document archives. In this paper we discuss the challenges involved in securing patient data stored in a DHT-based index and discuss initial solutions to address these challenges.

1. INTRODUCTION

Patient data records in the healthcare domain are often naturally distributed over the archives of corresponding doctors and healthcare facilities. Real world applications using this data would gain a substantial advantage from using sharing and search technologies available for P2P infrastructures. The P2P paradigm enables efficient sharing and retrieval of information in distributed settings and promises unlimited scalability, easy maintenance, and robustness against network attacks and failures. A study [19] stressed the importance of P2P networks in medical informatics, especially for improving data sharing between doctors and hospitals, in the national (US) as well as international contexts. However, considering high sensibility of the personal confidential data, privacy preserving mechanisms are unavoidable in this context. In this paper we illustrate the problem of efficient and *confidential* information sharing in a healthcare network along the following scenario: In case of emergency, information about blood group, allergies and vaccinations of a patient must be accumulated from collaborative network peers and presented to an authorized emergency physician to enable rapid and informed treatment decisions. This information is naturally spread among several network peers, e.g. physicians, internists and hospitals that treated the patient in the past. In case of emergency these peers need to be efficiently identified and requested to provide required information. However, the knowledge of the content provider, in this case a doctor or a hospital, can also disclose insides in a patient's history for the interested third parties. For instance, an insurance company, a bank or a potential employer might want to find out some data about the patient history. The specific area of expertise of the corresponding specialist can give insides in the art of potential diseases or the number of medical peers corresponding to a person indirectly disclosures illness frequency. DHT-based indexes are the standard choice for efficient identification of content providers and searching information in P2P networks in general. However, an ordinary DHT-based index does not provide sufficient confidentiality guarantees for healthcare data. This index is created using the inverted index data structure, which is then distributed over the network peers. An inverted index is a sequence of posting lists, each of which contains the IDs of all peers containing information about the specific term (which corresponds to a patient ID in our scenario). Table 1 shows an inverted index with four posting lists and seven posting list elements (elements for short). For instance, for patient John Doe the index includes information on one dentist, one urologist and one general practitioner who treated her in the past. This information can be easily extracted from the ordinary inverted index and thus requires additional protection against unauthorized access. A naive solution would be to rely just on access control mechanisms on a trusted server. However, it is unlikely that all institutionally independent doctors and hospitals in a collaborative healthcare network can agree on a single trusted central authority to enforce access control on index entries. Moreover, centralized indexes are attractive targets for attack and will need additional protection even if the index would be encrypted. For example, even if the exact content of the elements is obscured, the length of the posting lists corresponds to the number of doctors the patient visited in the past. Additionally, an adversary can scan posting lists on a compromised server to collect and count the ID's of the patients of a specific doctor.

In this paper we investigate the problem of building a DHT-based inverted index $HEALTH^{+Z}$for secure provider selection in collaborative healthcare P2P networks. This index fulfills the following conditions: (i) any information published in the DHT can be accessed only by authorized participants; (ii) each participant can easily and inexpensively access all information she has authorization for; (iii) the solution must withstand adversaries, and; (iv) the solution must be completely decentralized and stable even if some providers will not be available, to allow scalability in large P2P networks. Our contribution is summarized as follows: (i) we formalize the problem of securing provider information stored in the DHT-based index: we describe the possible threats that need to be addressed by an acceptable solution and show what characteristics each acceptable solution should adhere; (ii) we propose a solution for securing

Papetrou, O.	dentist:Peer P19, podiatrist: Peer P7
Zerr, S	dentist2:Peer P30
Doe, Joe	urologist: Peer P40, dentist:Peer P19
Smith, Joe	dentist:Peer P19

Table 1: A Patient-Doctor Inverted Index

the DHT index. The solution combines several technologies which are required to fully secure the data: k out of n encryption, encryption against statistical attacks, and policy-driven authorization; (iii) we perform a theoretical evaluation for the cost and security offered by the network. The paper is organized as follows: Section 2 discusses the threat model; Section 3 presents $HEALTH^{+Z}$index; Section 4 contains evaluation; Section 5 describes related work; Section 6 provides a conclusion.

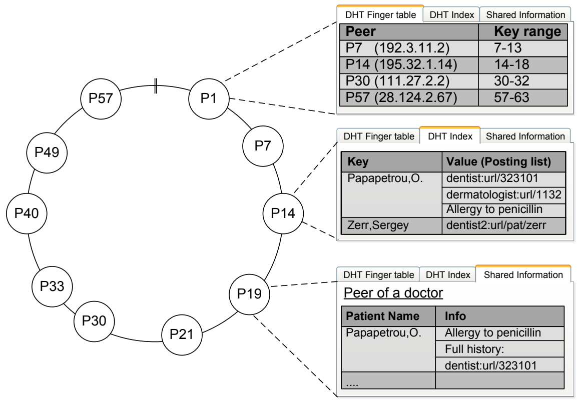
2. THREAT MODEL

$HEALTH^{+Z}$targets the problem of supporting efficient provider selection for healthcare data distributed over a set of network peers. In order to provide efficient, scalable and completely decentralized solution this network makes use of a DHT-based index which is distributed among network peers. Information stored in this index requires protection against unauthorized usage. The index needs to resist statistical attacks and achieve the privacy goals described in the following.

Attacks: To give a sense of the set of potential dangers, consider the following three goals of a potential attack on an index.

- *Determine the number of peers sharing patient's data on the network.* Aggregate number of posting elements shared about a particular patient over the network corresponds to the number of peers treated the patient in the past. For example, an adversary may observe that the number of peers sharing records of a patient exceeds average number of peers for other patients and conclude the increased illness probability.
- *Determine whether a patient record appears at a particular inaccessible site, or at any indexed site.* For example, a patient record at a specialists' peer corresponds to an increased probability of a particular disease.
- *Reconstruct the list of records shared by a particular peer on the network.* The list of posting elements shared by the peer corresponds to the list patients shared by this peer. For instance, a competitor peer may want to obtain such list of patients.

Privacy Goal: $HEALTH^{+Z}$focuses on attaining content privacy with respect to data d made searchable by some content provider p. That means that an adversary A should not be allowed to deduce that p is sharing data d unless A has been granted access to d by p. In addition, state of the art techniques auch as secure communication channels such as https should be used to provide confidentially for the content of queries and updates. Query privacy preserving techniques like [16, 7], can be used to prevent an adversary from determining which searcher issued what particular queries. An adversary could determine peers involved in the patient history by examining query logs, for this reason $HEALTH^{+Z}$does not store any query log information.



Peer	Key range
P7 (192.3.11.2)	7-13
P14 (195.32.1.14)	14-18
P30 (111.27.2.2)	30-32
P57 (28.124.2.67)	57-63

Key	Value (Posting list)
Papapetrou,O.	dentist:url/323101
	dermatologist:url/1132
	Allergy to penicillin
Zerr,Sergey	dentist2:url/pat/zerr

Patient Name	Info
Papapetrou,O.	Allergy to penicillin
	Full history:
	dentist:url/323101
....	

Figure 1: An Unsecured Inverted Index over DHT

3. HEALTH+Z NETWORK

In this section we define $HEALTH^{+Z}$index structure which provides confidentiality guarantees that hold even if a given number of the network peers are compromised or malicious and analyze characteristics of the index. *DHT as a Distributed Inverted Index*: $HEALTH^{+Z}$network consists of a set of content providers $CP = \{cp_1, \ldots, cp_h\}$ (doctors or hospitals in our scenario) which share information about entities $E = \{e_1, \ldots, e_m\}$ (e.g. patients). For the ease of presentation we assume that each content provider corresponds to one network peer $P_1 \ldots P_h$. In order to enable efficient search, information about the entities is indexed using $HEALTH^{+Z}$distributed index. $HEALTH^{+Z}$distributed index is based on a Distributed Hash Table (DHT). DHT is a family of distributed algorithms typically applied in the mainstream P2P systems. As the name implies, the functionality of DHTs is similar to the functionality of traditional hash tables: they enable efficient distributed storage and retrieval of (*key*, *value*) pairs. Thereby an ordinary inverted index, like the one presented in Table 1, can be partitioned across several peers.Without loss of generality, key is a number in the range of $[0 \ldots 2^z)$ where z is a value specific to the DHT implementation (e.g., for Chord DHT[20], z is 160). In our scenario, we want to use as keys the patient names. Therefore, patient names are converted to numeric representations by using a consistent hash function. There are several suitable consistent hash functions for converting any type of data to integers. In this work we use MD5 hashing, followed by modulo with the maximum key value.

The process of retrieving all information for a patient involves two steps: (1) find all doctors that this patient has visited, and (2) contact the peers corresponding to these doctors, to retrieve all relevant information. The first step, of locating all relevant doctors, is performed using the DHT inverted index. The name of the patient is transformed to its numeric representation using a consistent hash function. Then, the peer responsible for holding this value in the DHT is located, and contacted to retrieve the list of doctors that this patient visited. The peers corresponding to these doctors contacted directly, for authorized clients (such as emergency doctors) to retrieve important information for the patient, e.g., allergies, medication, and past illnesses. The good scalability characteristics of DHTs make them suitable information sharing infrastructures for many mainstream applications. However, current DHT-based systems do not enable indexing information confidentially, or restricting information access. Everything that is published in the DHT is by default accessible to all participating peers. In the next section we show how the DHT can be secured so that only authorized peers can retrieve relevant information.

Confidential Distributed Indexing: A naive approach to locate doctors for a particular patient would be to broadcast the query to all available peers which leads to unacceptable latency in a larger network. As discussed above, an ordinary inverted index will help to precisely locate patients' medical records, but does not provide the required confidentiality guarantees. In order to index entities confidentially, $HEALTH^{+Z}$modifies index content as discussed in the following. Each posting list in this index is a bit map; like in ordinary inverted index this list corresponds to a patient; each posting element (bit) in this map corresponds to a content provider. This bit is set to one if the corresponding provider shares information about the entity and to zero otherwise. Note that in general a posting element may contain additional data shared by the content provider. Here we consider the bit map to simplify the presentation. In fact, a non-encrypted $HEALTH^{+Z}$index is an entity-provider incidence matrix which is presented in Figure 1.

More formally, given the network H, index I, a content provider cp_i, and an entity e_k,

$$cp_i \in H \Rightarrow \forall e_k \in H : cp_i e_k \in I$$

Practically, this means that index structure contains an entry for every content provider-entity pair. In order to protect the index against unauthorized usage, bit maps are encrypted using k-out-of-n encryption scheme as discussed later in the "Encryption" paragraph. The presence of an encrypted entry in the index does indicate that an entity is shared by the corresponding peer.

Encryption: In order to protect the index against unauthorized usage, posting elements are encrypted using k-out-of-n encryption scheme [17]. Application of k-out-of-n encryption to distributed indexing was first proposed in [21]. In this scheme a single posting element (secret) is spit into n parts (secret shares) such that at least k out of n parts are required in order to reconstruct the secret. These secret shares are computed at the peer holding the plain information and then distributed over the network peers, such that only encrypted information is sent over the network and even in case index holding peers are compromised/malicious, the plain text information is not available for them. The querying user needs to be authorized by at least k peers in order to obtaine enough shares to decrypt posting elements. Even if k-1 peers are compromized, it will not possible to reconstruct the initial information. Figure 3 illustrates a part of P2P network with peers P_1, P_2, P_3 and n=3. The posting list for the entity e_1 is encoded into three posting lists each represented as a random vector. Each of those vectors is stored on a separate peer (i.e., P_1, P_2 and P_3). Assume k=2; then in order to decrypt the elements corresponding to the entity e_1 the user needs to be authorized by at least two peers out of P_1, P_2, P_3.

The encryption algorithm works as follows: All the operations described later in this section are carried out in the finite field Z_p. The secret splitting algorithm starts by choosing a large prime number p, such that any posting element (secret) to be shared is in Z_p. In addition, each peer i is assigned a unique random value x_i in Z_p. We call this the x-coordinate of the peer. The numbers p and x_i are made public, so all users know them.

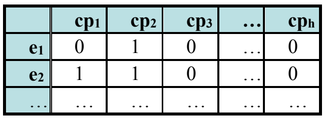
To index an element a_0 its provider generates a pseudo-random polynomial f of degree k-1. The coefficients a_i (except a_o) are randomly picked from the field Z_p. The secret share given to the i^{th} peer is $f(x_i)$. k such shares are enough to reconstruct the polynomial. To decrypt an element, a user must obtain k of its secret shares and determine the coefficients of the polynomial f by solving a system of k linear equations.

This scheme avoids complex key management and does not require re-encryption of the data unless more than k peers in the network are compromised. Moreover, if an adversary learns some of the shares, proactive sharing techniques can be used to prevent the adversary from getting k shares [11]. With this technique, the shares are updated so that those already known become useless.

k-out-of-n encryption in $HEALTH^{+Z}$replaces replication typically performed in P2P networks. Differently from the public networks, $HEALTH^{+Z}$does not store any exact copies of the index as all n parts of the encrypted secret differ. However, owing to the k-out-of-n encryption scheme the network is resistant to the failures of up to n-k peers which store any part of the index. We discuss overhead introduced by this scheme in the evaluation section.

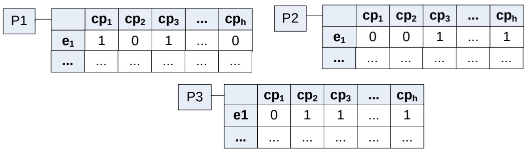
Access Control: Like in an ordinary P2P system, the index is partitioned across several peers according to entities such that each network peer stores only a part of the index. In difference to the public P2P systems, this index is stored privately on the peers and queries are answered only upon requests of the authorized users. In order to perform access control on the index entries, $HEALTH^{+Z}$makes use of standard authentication and authorization techniques.

Index Construction and Updates: Assume a network contains H content providers $cp_1, \ldots, cp_h$. At startup the index is empty. If the content provider cp_i wants to share the data of entity e_j, it first searches for the entity e_j as discussed in the following. In case the entity is not indexed, cp_i receives an empty result. An empty result corresponds to the case of a new patient, which was never indexed in the DHT by any content provider, either doctor or hospital. To insert the new entity in the index cp_i creates a new bit map of size N and sets the i^{th} bit to one and all other bits to zero. Then, cp_i encodes each posting element using k out of n encryption scheme and distributes the result over n network peers. Unlike ordinary P2P networks where the set of peers changes dynamically, set of content providers in $HEALTH^{+Z}$is rather static due to the natural properties of the healthcare network. This set can be extended by adding a new column to the index; this is a rather expensive but infrequent operation and can be further optimized, e.g., by adding columns in batches of B bits. Thus each adding of the columns will accommodate an increase of B content providers in the index and each posting list will increase by B bits. On the contrary, the bitmaps in the index require frequent dynamic updates; the bitmaps corresponding to the entities can be added and updated dynamically by corresponding content providers. Each content provider only needs to update the column that corresponds to her peer. This update can be performed inexpensively as it requires only a constant number of DHT lookups. Deletion of an entity is a rare operation which frequency in most of the cases depends on the retention period of records (e.g.



	cp_1	cp_2	cp_3	…	cp_h
e_1	0	1	0	…	0
e_2	1	1	0	…	0
…	…	…	…	…	…

Figure 2: Entity-Provider Incidence Matrix



	cp_1	cp_2	cp_3	...	cp_h
e_1	1	0	1	...	0
...	...	...	...	...	...

	cp_1	cp_2	cp_3	...	cp_h
e_1	0	0	1	...	1
...	...	...	...	...	...

	cp_1	cp_2	cp_3	...	cp_h
e1	0	1	1	...	1
...	...	...	...	...	...

Figure 3: k-out-of-n Encryption of a Posting List

10 years by German law). In order to delete an entity from the index, corresponding bitmap is simply removed. *Confidentiality Guarantees*: $HEALTH^{+Z}$index provides strong and quantifiable confidentiality guarantees that hold even if the entire index entries stored on k-1 malicious peers are made public. On her compromised peer, an adversary A can examine index entries. As all posting lists have equal length and represented as random bit vectors, she cannot determine the number of peers sharing patient's data on the network. She cannot determine at which particular site the patient record appear, although she can conclude that the patient record appears at least at one indexed site (which is not sensitive information in current setup since it corresponds to the fact that a particular person visited a doctor at least once). Similarly, she cannot reconstruct the list of records shared by a particular peer on the network as every peer corresponds to all patients in the index matrix. The k parameter in the k-out-of-n encryption defines the number of the peers that share a secret about a particular posting list and need to be compromised by an adversary in order to break the encryption of posting elements.

There is a tradeoff between confidentiality preservation and retrieval efficiency. The higher the k value, the more secure the index. However, higher k values lead to increased network traffic and response time. In the most secure case, k is close or equal to the number of providers (doctors) within the network and querying would essentially be performed by broadcasting the query. Smaller k values decrease network cost as well as security level. Thus k is a tunable parameter that can be adjusted during the index creation with respect to the trust level within the network.

The N value determines the number of peers holding a particular index entry. Since k peers holding shares of a particular index entry are needed to reconstruct the entry, N-k is the number of peers that can be offline at a time and the network would be still able to deliver enough shares.

4. EVALUATION

After discussing $HEALTH^{+Z}$architecture and confidentiality guarantees, we evaluate its storage requirements, query costs, and network usage for a network participant compared with an ordinary DHT, using a simulated data set. We created a simulated network with a reasonable size for a European country.

4.1 Experimental Data

We used the data from the World Health Organization for Europe[1] in order to estimate the potential number of doctors

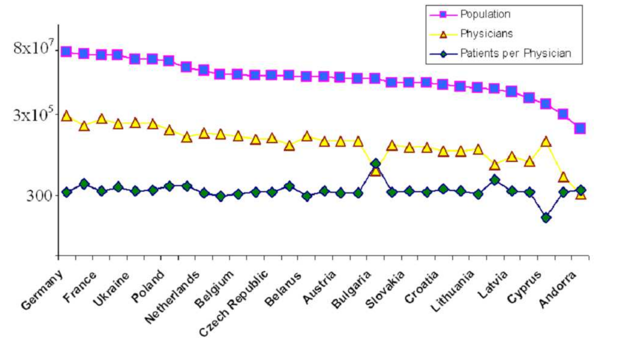


Figure 4: Population and Number the Physicians per European Country

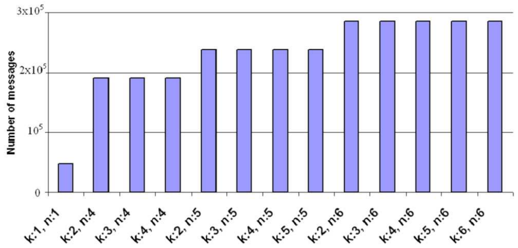


Figure 5: Network Cost for Index Construction per Participant

and patients that our system has to manage. Figure 4 shows the number of physicians per European country. According to the data, the number of physicians in the majority of the European countries does not exceed 300,000 whereas 80,000,000 is a maximal estimate for the population. Both numbers correspond to Germany. The proportion of the physicians with respect to the European population does not vary much and the proportion physician/persons can be estimated as 1/450 on average.Using these boundaries we created a matrix index. We randomly assigned patients to doctors using following estimations:

- We assumed the normal distribution of the number of doctors per patient
- We assumed that on average a person has her data by 20 doctors and used this number as a mean for the distribution
- We assumed that patients are uniformly distributed by the doctors

Thus each patient was assigned to 20 randomly chosen doctors on average, and each doctor served on average 5,333 patients. Assuming a bit of storage per patient-doctor relation, the index requires 25 kBytes for each patient's bit map. The k out of n encryption additionally increases this size by n times.

4.2 Experimental Setup

With our experiments we compared network and storage costs for an unencrypted index and for various encrypted indices. Network cost was measured as follows:

a. Network cost for creating the index from scratch. This cost occurs only once, when bootstrapping the network. This is the cost required for publishing all information of all content providers in the DHT

[1] http://www.who.int/gho/health_workforce/physicians_density/en/

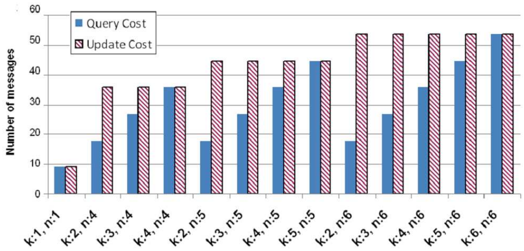


Figure 6: Cost per Query or Update, per Participant

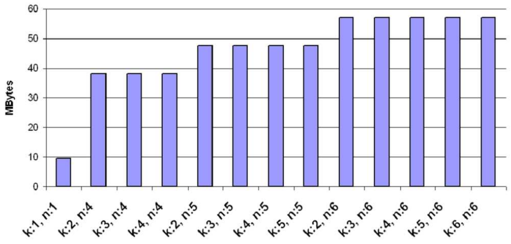


Figure 7: Storage Cost per Participant

b. Network cost for executing a query or for updating a record. This cost occurs every time a content provider needs to locate information for a patient (e.g., an emergency room doctor), or when a content provider adds a patient in her patients list

c. Storage cost. Each content provider contributes to the DHT by holding a small part of the distributed inverted index. This cost is the storage cost incurred by each peer on average

Note that our analysis does not include two additional cost factors: (a) the network overhead for maintaining the DHT connectivity between all content providers, and, (b) the cost for storing the actual medical information at the content providers. The former factor is not included because DHTs were already evaluated independently, and they were found to be scalable and extensible [20].

The latter factor depends on the information that is kept for each patient, and it is orthogonal to the application; this cost anyway occurs in current medical systems.

Parameter Selection: The parameter k determines the security level within the network. However, in order to increase k, n also needs to be increased. The parameter n, on its turn, determines the number of the times the index storage cost has to grow. We have to assume that different possibly old hardware is used by the network participants and thus each peer should hold not more around 50 Mbytes index data which corresponds to k<6 in our setup. We run the experiments for an unencrypted matrix index and compared it with 12 setups that differ in the choice of k and n.

Network Cost for Index Creation: Figure 5 (Network Cost for Index Construction per Participant) summarizes the average network cost per peer for creating the index. We measure cost with number of messages. The cost for both, unencrypted and encrypted index is at the same order of magnitude, even with high encryption parameters, e.g., k=6 and n=6. As expected, this cost grows linearly with n.

Recall that this cost occur only once, while bootstrapping the network. During this bootstrapping period, it is expected that the network will be more loaded than usual, because all content providers will be publishing information for all their patients at the same time. However, this bootstrapping process does not run under time constraints, therefore content providers can just wait for a couple of hours after installing the system, before starting to use it.

Query and Update Overhead: The number of messages needed for the retrieval of a particular posting list increases by k times compared with an ordinary DHT, because of the k-out-of-n encryption. However, even for k=6, retrieving the patientŠs information requires only 54 messages. Assuming ASDL speeds, this number of messages is negligible and can be easily executed in real-time.

Cost per update grows linearly with n. This happens because the content provider needs to locate the peers that hold all the n bit maps for the patient, and update one bit at each of them. For this update, the whole lists need not be retrieved. Similar to query cost, this cost is also negligible and can be executed in real time.

Storage Overhead: Unlike a DHT which is an inverted index, in $HEALTH^{+Z}$all posting lists have the same number of elements which corresponds to a number of document providers. Encryption under Shamir's k-out-of-n scheme does not change the size of the posting elements although the number of posting lists in the network increases by n times. Figure 7: (Storage Cost per Participant) shows that a storage overhead increases linearly with the growing number of n. For all the proposed setups, storage costs per peer do not exceed 60 Mbytes. This storage overhead is negligible for today's off-the-shelf personal computers.

Overall the results of the experiments prove the matrix index scalability for a given scenario and show that the network and storage costs are also reasonable.

5. RELATED WORK

The P2P paradigm promises unlimited scalability, easy maintenance, and robustness against network attacks and failures [3]. A recent study stressed the importance of P2P networks in medical informatics, especially for improving data sharing between doctors and hospitals, in the national (US) as well as international contexts [19]. $HEALTH^{+Z}$builds upon the existing work on information sharing and provider selection in P2P systems and enriches the DHT-based index structure used in P2P networks with confidentiality guarantees required in medical applications.

Encryption is a standard technique for storing data confidentially [4, 9, 13]. Other techniques include suppressing and/or generalizing released data into less specific forms, so that they no longer uniquely represent individuals [8, 12]; k-anonymity is one popular form of generalization (e.g., [2, 14, 15]). Unfortunately, it is not possible to directly apply these techniques to secure an inverted index. Even if posting list entries are encrypted, they can leak critical statistical data. The problem of sensitivity of the posting list length information was also stressed by [5]

The authors in [1, 21] considered protecting an inverted index when there is no single trusted central authority to enforce access control on posting list elements. Like μ-Serv, $HEALTH^{+Z}$addresses the problem of confidential provider selection in a network. However, μ-Serv does not provide sufficient protection for the data in the healthcare domain as the adversary can still conclude that certain percentage of posting elements in the index are true positives, which enables indirect conclusions on illness frequency of a person. Moreover, μ-Serv lengthens the querying process and wastes cycles at sites that do not contain query-relevant entries. For example, if x = 5%, the user must query 20 times as many sites to get the relevant results, which can lead to critical delays in medical emergency applications. On the contrary, $HEALTH^{+Z}$enables an authorized user directly identify corresponding peers.

Zerber [21] developed in our previous work is an r-confidential inverted index which protects indexed data by means of frequency-based merging of posting elements related to several terms in one posting list. In order to provide confidentiality guarantees for the information stored in the index it requires a training data set from which it can learn document frequency distribution. However, the terms in the $HEALTH^{+Z}$index are unique patient IDs, such that required training information is not available in this scenario. On the contrary, $HEALTH^{+Z}$enables confidential provider selection in case no training information is available.

While many other researchers have addressed aspects of data confidentiality, none of their schemes are intended for an environment with many dynamic collaboration peers. For example, researchers have suggested ways to search encrypted text or tables stored on a remote untrusted server (e.g., [10, 18]). In a situation with many collaboration peers encryption based approaches are not easy to use or manage due to the encryption key management. Data owners and/or project group managers must generate and distribute keying material for all group members.If a key is lost, stolen, or even published, the index entries encrypted with it are compromised. When a key is compromised or a member leaves a group, the key must be revoked and all the content associated with that key must be re-encrypted and re-indexed. Modern group key management schemes, such as logical key trees [6] and broadcast encryption, reduce the costs associated with giving keys to members, but still require content re-encryption. Some approaches also require that the entire index for a particular collection of documents be regenerated by the collection owner every time an entry is added to or deleted from the index. Zerber [21] proposed usage k-out-of-n encryption scheme which avoids key usage for data encryption. $HEALTH^{+Z}$builds upon this encryption scheme.

6. CONCLUSION AND FUTURE WORK

In this paper we considered challenges involved in building confidential index in a P2P healthcare network and discussed initial solutions to address these challenges. Our experiments show that for a current setup it feasible to maintain an incidence matrix based index with confidentiality guaranties within a P2P like network. Such index is protected against any statistical attacks even if overtaken by an adversary. One of the requirements of DHTs is that they need to withstand unexpected peer failures and disconnections. To withstand such events without losing data, DHTs employ data replication. The integration of the replication in $HEALTH^{+Z}$keeping its' confidentiality guarantees is an interesting direction for the upcoming research.

7. ACKNOWLEDGMENTS

This work is partly funded by the European Research Council under ALEXANDRIA (ERC 339233) and by the project "Gute Arbeit nach dem Boom" (Re-SozIT) funded by the German Federal Ministry of Education and Research (BMBF) (01UG1249C). Responsibility for the contents lies with the authors.

8. REFERENCES

[1] M. Bawa, R. J. Bayardo, Jr, R. Agrawal, and J. Vaidya. Privacy-preserving indexing of documents on the network. *The VLDB Journal 2009.*

[2] R. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In *ICDE'05.*

[3] M. Bender, S. Michel, P. Triantafillou, G. Weikum, and C. Zimmer. Minerva: Collaborative p2p search. In *VLDB '05.*

[4] M. Blaze. A cryptographic file system for unix. In *CCS '93.*

[5] S. Büttcher and C. L. A. Clarke. A security model for full-text file system search in multi-user environments. In *FAST'05.*

[6] T. Cho, S.-H. Lee, and W. Kim. A group key recovery mechanism based on logical key hierarchy. *J. Comput. Secur. 2004.*

[7] N. L. Farnan, A. J. Lee, P. K. Chrysanthis, and T. Yu. Don't reveal my intension: Protecting user privacy using declarative preferences during distributed query processing. In *ESORICS'11.*

[8] B. Fung, K. Wang, and P. Yu. Top-down specialization for information and privacy preservation. In *ICDE'05.*

[9] M. Goodrich, R. Tamassia, and A. Schwerin. Implementation of an authenticated dictionary with skip lists and commutative hashing. In *DISCEX'01.*

[10] H. Hacigümüş, B. Iyer, C. Li, and S. Mehrotra. Executing sql over encrypted data in the database-service-provider model. In *SIGMOD '02.*

[11] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung. Proactive secret sharing or: How to cope with perpetual leakage. In *CRYPT0Š 95.*

[12] V. S. Iyengar. Transforming data to satisfy privacy constraints. In *KDD'02.*

[13] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu. Plutus: Scalable secure file sharing on untrusted storage. In *FAST'03.*

[14] K. LeFevre, D. DeWitt, and R. Ramakrishnan. Mondrian multidimensional k-anonymity. In *ICDE'06.*

[15] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. L-diversity: privacy beyond k-anonymity. In *ICDE '06.*

[16] S. T. Peddinti and N. Saxena. Web search query privacy: Evaluating query obfuscation and anonymizing networks. *J. Comput. Secur. 2014.*

[17] A. Shamir. How to share a secret. *Commun. ACM'79.*

[18] D. X. Song, D. Wagner, and A. Perrig. Practical techniques for searches on encrypted data. In *IEEE Symposium on Security and Privacy, 2000.*

[19] W. W. Stead and e. C. o. E. t. C. S. R. C. i. H. C. I. N. R. C. Herbert S. Lin. *Computational Technology for Effective Health Care:Immediate Steps and Strategic Directions.* The National Academies Press, 2009.

[20] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M. Kaashoek, F. Dabek, and H. Balakrishnan. Chord: a scalable peer-to-peer lookup protocol for internet applications. *Networking, IEEE/ACM Transactions 2003.*

[21] S. Zerr, D. Olmedilla, W. Nejdl, and W. Siberski. Zerber+r: Top-k retrieval from a confidential index. In *EDBT '09.*