Discussion Paper: Privacy-Preserving Distributed Queries for a Clinical Case Research Network

Gunther Schadow, Shaun J. Grannis, Clement J. McDonald

Regenstrief Institute and
Indiana University School of Medicine
1050 Wishard Blvd., Indianapolis, IN 46202, USA.

{gschadow,sgrannis,cmcdonald}@regenstrief.org

Abstract

We present the motivation, use-case and requirements of a clinical case research network that would allow biomedical researchers to perform retrospective analysis on de-identified clinical cases joined across a large scale (nationwide) distributed network. Based on semi-join adaptive plans for fusion-queries, in this paper we discuss how joining can be done in a way that protects the privacy of the individual patients involved. Our method is based on a cryptographically strong keyed-hash algorithm (HMAC.) These hash values are truncated and the resulting hash-collisions in semi-join filters are exploited to limit the ability of an apprentice-site to re-identify patients in the filter. As a measure of privacy we use likelihood ratios. Since the join key is based on real person identifiers, we need to apply the methods of record linkage to hashing and semi-join filters. We find that multiple disjunctive rules as used in deterministic matching, lead here to a higher privacy risk than rules based on a single identifier vector.

Keywords: distributed databases, record linkage, privacy, semi-join.

1 Introduction

Increasingly, individual healthcare provider institutions are capturing data on all of their patients in a computer analyzable form. This data has many valuable research uses (Tierney, W. M. and McDonald, C. J., 1991). We have used the Regenstrief Medical Record System (RMRS, McDonald C.J., Overhage J.M., et al., 1999) at Wishard and IU hospital to show relationships between erythromycin use in newborns and pyloric stenosis (Mahon B.E., Rosenman M.B. and Kleinman M.B., 2001) and between ibuprofen and renal failure (Murray M.D., Brater D.C. et al., 1990). But the real opportunities come when data of different sources is joined together to link outpatient visits, hospitalizations, pharmacy data, cancer registry abstracts, death records and other data. Such linked data allows following medical histories and linking treatments to outcomes, which is important to do research on risk factors for diseases and on the risks and efficacy of treatments.

Previous approaches to performing research on such large bodies of data include (1) collecting all data at one site, such as in the Regenstrief Medical Record System, and (2) collecting a portion of medical data about cases in registries, such as cancer registries. Collecting either all data or specific data sets at one site has the disadvantage that one can always only cover a subset of a large population or a subset of data about that population. In addition, large accumulations of individual patient data are commonly viewed as threatening patients' privacy, and therefore may not be scalable.

In this paper, we are investigating approaches to performing research queries against many disparate and autonomous systems (instead of previously accumulated data in one system). This massively distributed query processing must be able to join individual cases across all source systems and still protect patient privacy. We propose that a particular set of real patient identifiers used for the joining are one-way-encrypted by a keyed hash-function (HMAC, cf. NIST 2002). In addition we propose exploiting hash-collisions to protect against re-identification attacks that discover known identifiers in semi-join filters.

1.1 Use Cases

Consider the example for a retrospective cohort study such as by Mahon and Rosenman (2001), assessing the adverse effects of erythromycin on the development of pyloric stenosis in newborns. To do a retrospective cohort study, we need to find a cohort of cases with prescriptions of *erythromycin* (including time, and dose) during the first three months of age, and then find the subset of this cohort that later had a diagnosis of *pyloric stenosis*. About these cases we need the gender, date of birth, and the earliest date of the diagnosis, and perhaps some other clinical information items. In the cited study 14876 cases were included in the cohort of which 43 (0.29%) developed the disease.

If the studied disease is very rare, it would be hard to find a cohort large enough to get significant cases of the disease. For these situations, case-control studies are used. In a case-control study, we would find two cohorts, one with the diagnosis of pyloric stenosis, and another cohort without that diagnosis (but otherwise similar structure) Data about exposure (*erythromycin* in our example) is collected for both cohorts and then compared between the cohorts.

Cross-sectional queries are used to find out the prevalence of a condition in a certain population. For example, the National Cancer Institute (NCI) is funding a Shared Pathology Information Network (SPIN) that should help

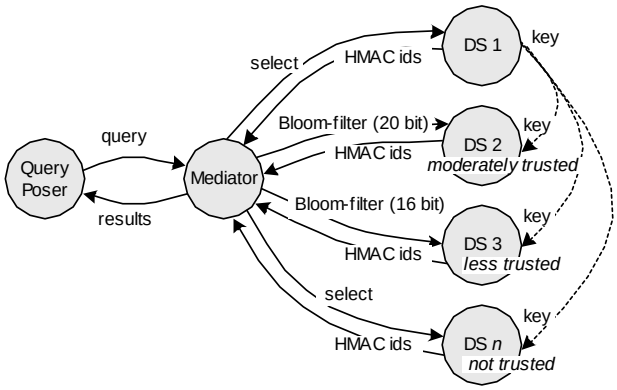

Figure 1: Essential architecture with query poser, query mediator and multiple data sources (DS). The resolution of the hash-codes depends on how much the data source is trusted not to attempt re-identification.

cancer researchers locate rare specimens and clinical cases among multiple data sources nationwide. In these queries we look for certain anatomic-pathological observations. These queries may well consider clinical history before the tissue diagnosis as well as the course of the disease, treatments and outcome (e.g., survival.)

These queries would be processed in a possibly very large network of data source nodes, including hospitals, pharmacies, cancer and other specialized registries, benefits management databases, death records etc. across the community, the state, or the entire country. The SPIN project for example, would have a nationwide scope with collaborating sites in Boston Mass., Pittsburgh Penn., Indianapolis Ind., and Los Angeles Cal.

Two major impediments to such endeavours are the lack of standardized medical information and concerns about patient privacy. We have written much about the standardization issues (e.g., McDonald C.J., Schadow G. et al., 2001) and hence will not raise these issues here. Agreeing on a global schema for biomedical information is not trivial, and some of the terminology issues involved are daunting. However, there are also solutions to these problems, that involve standardization, mapping and translation, and we have long been working on these matters.

1.2 Outline of the Approach

For this paper we consider the architecture illustrated in Figure 1. A query poser submits a query to a mediator that queries the data sources (DS). Depending upon the query, the mediator may first collect a seed for a cohort from DS 1 (or from multiple data sources) and then retrieve additional information about the cohort members from the other data sources.

The data sources return keyed-hashed patient identifiers to the mediator along with clinical data. The key is generated for each query and shared between all data sources, but is not disclosed to the mediator; hence the mediator cannot associate individuals of interest with the hashed identifiers send to it. The mediator is trusted, not to attempt re-identification from the clinical data.

The data sources are queried with select queries or Bloom-join queries using a varying hash-code range that depends on the number of ids in the join and the level of trust in the data source not to attempt re-identification and inference from the query.

Upon completion of the distributed join query, the mediator usually aggregates the clinical data and only returns statistics to the query poser. Upon special proof of authorization, individual data may also be sent, but will be subjected to scrubbing, i.e., data is removed or altered, that would otherwise permit re-identification from the clinical data. The hashed patient ids are never returned.

1.3 Organization of this Paper

In this paper we mainly focus on the privacy implications of the distributed join query. In many of the other functions of the system, query plan generation, record linkage, and privacy-protection in de-identified data disclosures (scrubbing), we rely on existing work and our own future research as discussed in the sequel.

The further organization of this paper is as follows. In Section 2 we discuss distributed join queries, how common optimization of distributed join queries affect privacy, and how we propose to control the privacy risk involved with such optimizations.

Section 3 discusses issues related to record linkage, which is the technique used to implement the privacy protecting equi-join's equivalence operation.

Section 4 discusses our findings and relates our approach to other work that we have not yet mentioned up to that point. We will conclude by outlining our next steps.

2 Privacy Protecting Distributed Joins

2.1 Prior Work

Over 3 decades of prolific research and practical applications in distributed databases exists, as outlined by Kossman (2000) and Özsu and Valduriez (1990). In summary, the two major problems of distributed databases are schema translation and distributed join optimization. Distributed join query planning must decide what data is transferred between two sites. The site where the query originates is referred to as the *master-site*. The master-site requests data from another site, called the *apprentice-site*. The main approaches to this interaction are:

1.) Selection-only queries, where the master-site sends a query criterion to the apprentice-site, which responds with all records matching the criterion, which the master-site in turn joins with its local data. This method is optimal if most matching records at the apprentice-site can be expected to join with the master-site's local data.

2.) Semi-join queries (Bernstein and Chiu, 1981), where the master-site sends a set of join-keys to the apprentice-site (along with query criteria) and the apprentice-site responds uses this set of join keys to filter out all records that cannot possibly join at the master-site (and hence would be a waste to send.)

3.) Bloom-join queries (Bloom, 1970; Babb, 1979), which are semi-joins that are optimized using Bloom-

filters. Bloom filters are hash sets (instead of enumerated lists) of join keys, encoded in bit fields for efficiency. As all hash-based methods, Bloom filters may contain hash-collisions that reduce the power to filter out records that will not complete the join at the master site. Mullin (1990) describes the trade-off between filter size and filter efficacy.

Most of the research and practical applications of distributed databases work under the premise of only a few data sources and with good knowledge of a fragmentation schema that specifies how data is distributed. With such fragmentation schema a query optimizer can develop a very good execution plan from knowledge at a single site. However, when the number of data sources becomes very large and the systems are rather autonomous and loosely coupled, a fragmentation schema that would assist in global query planning may not be known at any one site.

Abiteboul (1997) and Yerneni (1998), and co-workers have introduced the notion of *fusion queries*, and their use cases are very similar to ours. The authors consider a simple relation that is distributed over many autonomous data sources without a known fragmentation schema. They then consider simple queries that can be decomposed into simple selects and semi-join driven selects. The results are assumed to be projected on the subject-identifier, such that joins of two selections are the same as an intersection of the result set of each selection. These queries are planned and executed by a mediator node in the network, and this planning depends on only very limited knowledge about the data sources and no knowledge at all of a data fragmentation schema. The mediator node creates "semi-join-adaptive" plans choosing between simple select and semi-join-select for each node. The mediator executes the query through sequential interaction with each node possibly several times.

The issue of schema translation can be managed with schema-mapping metadata that the query-planner uses to produce the appropriate query steps for the different data sources, and at the same time to perform the translation of the results to a common schema. This has been described for example by Özsu and Valduriez (1990) and Sheth and Larson (1990). Such global schema translation is important to provide full transaction support, but may not be needed (or is not even feasible) if a large number of independent systems are only queried but not updated. In these cases one can assume a global schema and make the schema translation the task of each data source using "wrappers" such as proposed by Roth and Schwarz (1997).

2.2 Data Schema

For purposes of this paper we use a very simple data schema of a single relation $R(p, e, t, v)$ in which each tuple is the record of one clinical event (observation, treatment, etc.) for one patient at one time. The relation's attributes are patient identifying information p, the event class e, event time t, and event value v. We will scrutinize the patient identifier attribute p in Section 3 below. Suffice it to say here that the domain of p is defined as an abstract data type with an equivalence relation. The equivalence classes generated by this equivalence relation can be mapped to a scalar value space.

We consider the domain of event class e as a scalar with an equivalence relation, the event time t is a totally ordered scalar (order and equivalence relation), and event value is a variant data type that, depending on the event class e, has at least an equivalence relation if not an order relation or additional properties of a quantity.

The tuples of this relation $R(p, e, t, v)$ are scattered across many data sources. But we have no knowledge about how the data is partitioned, what patients nor what kinds of events any given data source stores information about. In particular, we cannot assume that all data about one patient is local to a single data source, nor can we assume that a single data source has data about a certain event for all patients. Also, the same tuple in R may be recorded at multiple data sources. In other words, we have to assume partitioning to be completely random and with possible replication that occurs likewise randomly.

2.3 Distributed Joins

Distributed joins rely on primary key attributes to flow from the data source to the mediator (select queries) and from the mediator to the data sources (semi-join queries.) Without further arrangements, each query would effectively broadcast many patient identifiers among all data sources. We do not consider the exact nature of the patient identifiers until later; but we note that identifiers that can be equal for the same person across all data sources must necessarily be sensitive real identifiers. While the patient identifiers may not need to be broadcast with any explicit sensitive medical data, the fact that a set of identifiers is sent for a semi-join, allows the receiver of such data to infer that most likely a strong condition of the query (e.g., a certain disease) must hold for the patients in this set.

Protecting privacy in the distributed join means to make certain (1) that neither party discloses real identifiers that would allow another party to re-identify individual patients; (2) that neither party must be able to over time gather significant information about individual patients; and (3) that identifiers exchanged should not allow any party involved to infer significant probabilistic information about who the real persons behind those identities might be.

2.4 Protecting Privacy against the Mediator in Selection-Queries

We propose using cryptographically strong hashing of identifiers. Such hashing is irreversible; hence, identifiers could not be decrypted. However, with only a hash encoding, one can test for any given person of interest whether that person is a likely member in the set of identifiers exchanged, which is not much less of a risk than straight-forward decryption. Hence we will use a keyed-hash scheme when sending identifiers from the data source to the mediator.

The keyed-hash (HMAC) transform (NIST, 2002) only allows testing for membership of a real individual if the

key is known, and given that we can protect the key from disclosure to the mediator, the mediator cannot test real patients for membership in the hash sets exchanged.

The HMAC key must be shared between all of the data sources in order for all of them to generate hash codes comparable to the mediator. Since there are potentially thousands of data source sites, this HMAC key must be regarded semi-public. Hiding the hash key from the mediator is thus possible only if we can trust the mediator not to seek knowledge of this key.

To mitigate the risk that the mediator could accumulate and refine a large body of knowledge about hashed identifiers to the point where these data can be re-identified easily or where re-identification with a hash key discovered later will reveal a large set of data, we require that the hash key be randomly generated for each query. For example, one data source could generate the key and distribute it to all other data sources.

Let *N* be the total number of individuals covered by this research network. Assuming that it would include a large fraction of the U.S. population, both those who live and those who have died in the last few decades, we use $N=10^9$ (approximately 4 to 5 times the current U.S. population) as an estimate for our total population size. (We use numbers throughout to illustrate the orders of magnitudes that we are dealing with.)

A typical HMAC algorithm produces hash codes of 128 bit or more; hence, the hash range has cardinality beyond 10^{38} which is many orders of magnitude beyond our total population size. Hence we can consider the HMAC coding to be a one-to-one mapping of patient identifiers. Since all data sources use the same HMAC key and the mediator only knows of patient identifiers as HMAC coded, for simplicity of notation we speak of patient identifiers *p* in the following to generally mean the HMAC coded identifiers.

2.5 Protecting Privacy against the Data Sources during Semi-Joins

For semi-joins, identifiers contributed by data sources need to flow from the mediator back to other data sources. Since all data sources use the same HMAC key, HMAC will not protect privacy. In our model, there are large numbers of data sources, whose compliance to the network's policies we cannot sufficiently ascertain; hence we cannot generally trust the data sources. The risk in sending semi-join filters to an untrusted data source lies in the possibility that that data source may, in violation of the network's policies, attempt to infer information from the context of the query.

For instance, if the query includes a search for patients with a rare disease, and is followed by occupational or insurance data, the sites holding these occupational or insurance data, but should not know of the disease, can infer that patients in their data base who match the semi-join filter must have the disease.[1]

We propose reducing the resolution of the hashing code to limit the ability of a data source to infer information about individuals of their interest.

Let $R=\{p_1, p_2,..., p_m\}$ be as set of (HMAC coded) patient identifiers returned from data sources to the mediator in previous steps of a query-execution.

Let $f^b : H \mapsto H^b$ be a function that truncates the HMAC codes to *b* bits by clipping bits in a defined manner. Assuming the HMAC coding is pseudo-random with a uniform distribution, the truncated HMAC codes will have these same properties.

For a semi-join set to be sent from the mediator to a data source, the mediator builds a set of truncated patient identifier values $F^b(R)=\{f^b(p): p\in R\}$. This set is called the filter and can be represented either as an enumerated list of hash codes, or as a bit field where the $f^b(p)$-th bit is set if $p\in R$ and cleared otherwise.

This filter $F^b(R)$ is sent to the other data source, which has *M* number of patients represented by a set of patient identifiers $S=\{q_1,q_2,...,q_M\}$. This set is filtered to $S'=\{q_j : 1\le j\le M \wedge f^b(q_j)\in F^b(R)\}$ and filtered again by the clinical query criteria, and returned to the mediator along with the clinical data requested for each element.

The purpose of the truncation of the HMAC codes is two fold. For one, it reduces the size of the filter to send to the data source, which is important to make semi-joins efficient. On the other hand, the truncation increases the hash collisions in the filter, which we will actually exploit to hinder the data sources from inferring information about individuals from the join filters.

For the truncated semi-join filter we derive the number of hash collisions analogously to Mullin (1983), as the probability $P(F)$ of any one of the $n=2^b$ possible truncated hash codes to be in the filter after all *m* identifiers have been encoded (which is the same as the fraction of hash codes used.)

$$P(F)=1-P(\bar{F})$$

$$P(\bar{F})=(1-1/n)^m$$

$$P(F)=1-(1-1/n)^m, \qquad (1)$$

which for $n \gg m$ converges to m / n.

So, the expected number of patient records that pass through the filter is $M\times P(F)$. Of these patient records, however, only *m* could possibly be true members of the semi-join set $S\ltimes R$, and none of them need actually be a true member if $S\cap R=\varnothing$. Since a semi-join is sent with a condition that must also be true besides the identifiers matching the filter, we can only provide some boundary estimates of how many records we will retrieve and what our false-positive probability is. In the best case, the data

[1] For a case control studies such as in this example, one could certainly dilute the cohort with the disease with the control cohort, thus reducing the maximum likelihood of a match implying the disease to ½.

source contains all of the true elements of the set, then the false positive probability is the one Bloom (1970) gives

$$P_{\min}(f(q)\in F \mid q\notin R) = \frac{M\times P(F)-m}{M-m} \quad (2)$$

for $M\times P(F)\geq m$

However, usually the intersection between S and R from the mediator may be quite small, and will often be empty. In the worst case all retrieved elements would not be members of the set and would all pass the selection criterion that went along with the semi-join:

$$P_{\max}(f(q)\in F \mid q\notin R) = \frac{M\times P(F)}{M} = P(F) \quad (3)$$

Note that this maximum false-positive probability is what Mullin (1983) assumes.

When using semi-joins, our primary concern besides the privacy issue is efficiency, i.e., we want the cost of sending the hash code set from the mediator to the data source to outweigh the costs of transmitting all records that match the select criterion from the data source to the mediator. To estimate the worst-case performance of this semi-join, we take the upper bound for the false-positive probability. To consider a realistic example, with a hash set of $m = 10^5$ elements against a database of $M = 10^6$ patients, with a 20 bit hash code ($n \approx 10^6$) we would recall about 10^5 patients (of which a significant fraction may not pass the selection criteria sent with the semi-join.)

2.6 A Model of Privacy

In order to understand the impact of a semi-join on patient privacy we ask what inference a data source can make about a certain individual of interest given the semi-join filter. Assume, for example, that the data source knows that the semi-join filter passed to it contains a list of patients that are positive for a sensitive characteristic C (e.g., HIV positive). We can use Bayes' theorem to describe the posterior probability $P(C(q) \mid f(q)\in F)$ of a person to be positive for that disease after that person matches a hash code in the filter (for brevity we will leave the argument to C and f off):

$$P(C \mid f\in F) = \frac{P(f\in F \mid C)\,P(C)}{P(f\in F \mid C)\,P(C)+P(f\in F \mid \overline{C})\,P(\overline{C})} \quad (4)$$

Since the post-test probability will depend on the prior probability $P(C)$, we can isolate $P(C)$ in this term choosing the odds notation, where $O(\xi) = P(\xi) / (1 - P(\xi))$

$$O(C \mid f\in F) = \frac{P(f\in F \mid C)}{P(f\in F \mid \overline{C})}\times O(C) = L\times O(C) \quad (5)$$

with L being the likelihood ratio. The prior probability $P(C)$, is set to the prevalence of a disease if no further information is known. Note that for probabilities $P(\xi) <$ 0.1, which is common for most diseases (e.g., P(HIV) = 0.006, P(cancer) = 0.03), $O(\xi) \approx P(\xi)$ is a very good estimate. Hence we can intuitively think of L as an amplification factor for the prior probability.

If the abusing data source (intruder) suspects a person of interest to be positive and seeks confirmation, it has a considerably higher prior probability. As the prior probability rises above 0.1, however, the same likelihood ratio contributes less to the posterior probability. Hence, given a set of significant "clues" that a data source may already have to suspect an individual being positive, a reasonably slight contribution of the semi-join may be small relative to the other clues.

To estimate a likelihood ratio, we will assume that if the person of interest were indeed positive the hash set would contain that person, i.e., we set $P(f\in F \mid C)=1$. This assumption is of course never realistic, but it becomes more likely as the network grows in size, achieving more and more complete coverage of all patients of interest to the intruder. Since a low posterior probability is good for privacy overestimating the false-negative probability is safe.

We can estimate the false-positive ratio $P(f\in F \mid \overline{C})$ analogously to our thoughts above, but this time we want to be safe and consider the minimum false positive probability $P_{\min}$ as in equation 2, which depended on the size of the covered population M. This time, however, we must assume M to be the number of all possible patients, in which case $P_{\min}$ converges to $P_{\max}$. So, we can write

$$L = \frac{1}{P(F)} = \frac{1}{1-(1-1/n)^m} \quad (6)$$

In our example of $n = 10^6$ and $m = 10^5$ we have $L \approx 10$, which is a moderate increase of the likelihood. Obviously, we cannot completely avoid inference from semi-join filters. The question then is, can we dimension the filter such that the likelihood ratio is kept acceptably low (closer to 1)?

If m is small, we have to reduce n accordingly to keep L acceptably low. When m is high, we can increase the resolution of the hash code. This intuitively works in the same sense as we need to increase the selectivity of the hash set if we have large sets in order to reduce needless data transfer.

To find the required hash function range n for any target likelihood ratio we rearrange equation 6 for n:

$$n = \frac{1}{1-\left(1-\frac{1}{L}\right)^{\frac{1}{m}}} \quad (7)$$

We can then merge this into equation 1 to get a false positive retrieval rate that we entail for achieving a low likelihood ratio L

$$P(f(q)\in F \mid q\notin R) = 1-\left(1-\left(1-\left(1-\frac{1}{L}\right)^{\frac{1}{m}}\right)\right)^m = \frac{1}{L} \quad (8)$$

Given this very simple relationship[2] we can adjust the likelihood coefficient according to how much trust the network community assigns to a data source site. Intuitively, a data source site that has large amounts of patient data (millions of patients) has a lower relative gain from specific new information it might infer from semi-join filter. Furthermore, the mere fact that such site is entrusted with such large amounts of patient data is a measure of trustworthiness. Conversely, a small participant with little locally stored information may have a greater relative gain in information inferred from semi-join filters, and so, we can buy a lower likelihood ratio for a reduced effectiveness of the semi-join filter. At the extreme, the data source is so small that it is never entrusted with a semi-join filter, and instead will always send all its records that match the selection criterion.

3 Record Linkage with Real Identifiers

The most critical piece in fusion queries is the equi-join on the subject primary key (patient identifiers.) As noted in Section 2.2, our schema defines the domain of the patient identifying attribute p as an abstract data type with an approximate equivalence relation $\cong$.

Since there is no simple globally valid surrogate key for patients, this patient identifying attribute must be a vector of "real identifiers", including last name, first name, date of birth, social security number, etc., i.e., person identifying information that all data sources commonly know about a patient.

Real identifiers are subject to error and inconsistencies throughout, including misspellings, mismatch between formal-name and nickname, swapped digits, and patient/beneficiary (spousal) mix-up of social security numbers. Thus, we have to define an equivalence operation that can compensate for some error. Study of this equivalence relation is known as record linkage.

3.1 Prior Work

The theory of record linkage was incepted by Newcombe et al (1959) and formalized in Fellegi and Sunter's (1969) seminal work. The Fellegi-Sunter's method of statistical record linkage defines the linkage operation as for each pair of patient identifier vectors (p, q) determining a comparison vector $\gamma(p, q)$ and comparing the likelihood ratio

$$\frac{P(\gamma(p,q) \mid p \cong q)}{P(\gamma(p,q) \mid p \ncong q)} = \frac{m(\gamma(p,q))}{u(\gamma(p,q))}$$

against two decision levels (thresholds) T_μ, above which the equivalence is accepted and T_λ, below which the equivalence is rejected. For likelihood ratios between T_μ and T_λ, the linkage is undetermined and is usually deferred to review by a clerk.

[2] The reader may find this rather lengthy derivation through equations 7 and 8 unnecessary, because equation 6 and 3 already contain this result quite clearly. However, we need this derivation further below for a generalization.

Quantin, Bouzelat et al. (1997) describe a public health registry system that uses the Fellegi-Sunter method on keyed-SHA hashed identifier data. Similarly, Michaelis and Miller et al. (1995) describe a cancer registry system that uses keyed MD5 hashing together with DES encryption on the components of a patient identifier vector which are the used for matching according to Fellegi and Sunter.

While their keyed-hashing approach is very similar to ours, the setting in which linkage with hashed identifiers is employed is different. Most record linkage work, including all the authors cited above, has been done for the purpose of registries that accumulate data at one site. Because of this the Fellegi-Sunter method can afford to include an area between the two decision levels, that require human review. The use in a central registry also allows Quantin et al. and Michaelis et al. to implement a relatively complex multi-party encryption scheme. In particular dictionary re-identification attacks can be prevented because the registry is trusted not to attempt acquiring the hash key.

Conversely, it is our goal to define a mechanism for a privacy protecting distributed join that can dynamically link data from various sources for a short time to answer a specific query, and then delete the accumulated data and all key information used for this one query. The methods described by Quantin et al. or Michaelis et al. would be applicable and sufficient if we only had identifiers flowing from the data sources to the query mediator. However, particularly in querying large data bases such as hospital data bases or population registries with several million patients, optimization by a semi-join query must be used to make the distributed query feasible. To do that, identifier data is practically broadcast to the data sources, rendering the keyed-hashed identifiers vulnerable to re-identification attacks.

The dynamic nature of a distributed query system (as opposed to a static registry) also forbids any linkage decision to be deferred to human review. We therefore would at least need to reduce the Fellegi-Sunter method to using only one threshold for deciding between link and non-link.

Furthermore, we reasoned that we should not hash-encrypt each patient identifier component individually, because this would render these identifiers (and the keys) vulnerable to frequency analysis. For example, one will always expect "Smith," "Williams," and "Jones" to be one of the most frequently encountered last names, which allows one to identify these names in most samples from simply their frequency. In addition, persons with rare names could be discovered with higher likelihood in the kind of dictionary attacks that an abusing data source could mount.

3.2 Deterministic Linkage

We hypothesized that a deterministic linkage algorithm with very few (ideally only one) hash codes that cover more than one patient identifier component could be safer. Grannis et al. (2002) empirically studied a number of such identifier combinations on large files of two hos-

pital registries and the social security death index. He found that we can achieve acceptable linking results based on 4 combinations of social security number (SSN), first name (FN), last name (LN), a phonetic code of the first name (cFN), and the birth date's year (YB), month (MB), and day (DB) and gender (G) components:

1.) SSN, cFN, YB;

2.) SSN, cFN, MB;

3.) SSN, cFN, DB; and

4.) LN, FN, YB, MB, DB;

where at least one of these 4 combinations must match exactly between two records. This disjunction of for combinations produces a fairly good sensitivity and specificity (over 97%). As can be seen, the social security number is the cornerstone of the matching rules. In order to prevent linkage in the case of patient/insured mix-up, the phonetic code of the first name and a piece of the date of birth is added. This makes for very specific matching, however, since more than 30% of the patients may not have a known SSN, we use the fourth rule to include full name and birth date agreement in the match.

We calculate a hash code for each of the four identifier combinations, analogously to what was described in Section 2 above. We will now discuss how the semi-join filter can be communicated to the data source and how the privacy-properties change due to the use of multiple hash codes of the different identifier combinations.

We can support the multiple hash codes used in a disjunction in two ways. We can use a single filter for all hashed identifier combinations or we can use a different filter for each hashed identifier combinations. Mullin (1999) has shown that for the same false-positive probability one filter with multiple hash functions or multiple filters for one hash function each require the same total amount of space. This is true for Bloom-filters and for unordered enumerated hash code lists alike.[3]

We now generalize our equations of Section 2 to account for multiple hash codes used in a disjunction. Let k be the number of hashed identifier combinations (in our case $k = 4$). Then equation (1) becomes

$$P(F) = 1 - (1 - 1/n)^{km} \qquad (1')$$

which for $n \gg m$ converges to $k\ m\ /\ n$. The maximum false-positive retrieval rate is still $P(F)$.

The disjunctively used hash coding is less space-efficient, i.e., we require more space for the same false-positive probability. To consider our realistic example, with a hash filter of $m = 10^5$ elements against a database of $M = 10^6$ patients, with 20-bit hash code ($n \approx 10^6$) we would recall about 3×10^5 patients.

[3] Conversely, hash code lists where each list item is a 4-tuple of the hash codes for each identifier combinations, contain additional information about how the hash codes are associated, which would appear to be an unnecessary risk to privacy, hence, we will not pursue this method further.

The likelihood ratio becomes

$$L = \frac{1}{1 - (1 - 1/n)^{km}} \qquad (6')$$

In our example of $n = 10^6$, $m = 10^5$, and $k = 4$, we have $L \approx 3$. So, the disjunctive multi-hash code filter decreases the likelihood coefficient in our favor. To find the required hash function range n for any target likelihood ratio we rearrange equation 6' for n:

$$n = \frac{1}{1 - \left(1 - \frac{1}{L}\right)^{\frac{1}{km}}} \qquad (7')$$

But when merging into equation 1' the effects of the k number of hash codes cancel out and we still have

$$P(\bigvee_{i=1}^{k} f(q_i) \in F \mid q \notin R) = 1/L\ . \qquad (8a)$$

The reason our likelihood coefficients went down from 10 to 3 is due to the fact that each element of the set uses up 4 hash codes, hence, our rate of collision is high. However, in order to infer information about patients of interest from semi-join sets, a data source can employ a trick that we cannot employ in routine joining operation. That trick is for the data source to require that more than one of the 4 hash codes are in the set, which can be expected to be the case most of the time. If so, the false-positive probability due to hash collisions falls considerably. For instance, if a complete match is required we have

$$P(\bigwedge_{i=1}^{l} f(q_i) \in F \mid q \notin R) = (1 - (1 - 1/n)^{km})^l\ , \qquad (8b)$$

with l being the number of hash codes that must be matched. Thus we have $L_l = (1 - (1 - 1/\mathrm{n})^{km})^{-l}$. In our example the exponentiation increases the likelihood ratio to 85, which is clearly unacceptable. Even if we used only two matching rules, say rule 1 and 4, the likelihood ratio would be at 30, which is still too high.

Thus, we find that if record linkage has to be done with a disjunctive hash set, it is not possible to outweigh the exponentially greater advantage of an attack against privacy by trading off the effectiveness of the filter. This is a significant result that paradoxically suggests that some national unique patient identifier scheme that would relieve us of having to go through the complex disjunctive record linking could actually help improve privacy.

4 Discussion

We found that in the absence of a ubiquitous individual healthcare identifier number, it is questionable whether we can find a single identifier combination that is as sensitive as the disjunction of the four, that is still useful for effective semi-join filters, and that maximizes privacy. For instance, consulting our samples of matching data in search for a single highly link-sensitive combination, we have considered the following combination:

5.) cFN, G, YB.

The four hash codes described above would still be sent from the data source to the mediator along with this fifth hash code, and the mediator would consider the four hash codes to determine identifier equivalence. However, for semi-join filters, the mediator would only send the fifth hash code to the data sources.

In this particular example of a fifth combination, in a large database of roughly 10^8 deceased U.S. persons, we only find 10^6 different combinations (19 bit). This alone would still be adequate if the names were uniformly distributed. Yet the distribution of the data is everything but uniform such that there is only 15 bit of entropy. For example, the phonetic male first name class "JAN" (e.g., John and Jon) of individuals born in 1914 amounts to 2×10^5 ($P_i = 10^{-3}$), while there is a long tail of single occurrences ($P_i = 10^{-8}$). With such uneven distribution the selectivity of the filter will be worse while at the same time having much less privacy protection for individuals with uncommon names as opposed to their contemporaries with common names.

It is not certain if an identifier combination can be found that has a good sensitivity while providing enough entropy to allow for even selectivity and privacy. Hence, we may have to sacrifice sensitivity, allowing only more complete patient records to be gathered in a fusion query. This is still useful for finding good example cases for case research, but may be less useful for epidemiologic studies where a complete population count is desired.

4.1 Rejoinder to reviewers remarks

Can you provide a description of the practical use; for instance, can you indicate what, given a particular set of data sources and study to be performed, is the appropriate choice of a number of hash codes?

Our goal is a deterministic distributed join operation to reconstruct real patient cases. The use case is to perform traditional cross-sectional studies, retrospective cohort studies, and case-control studies. Our goals even include abstracting these individual cases, and possibly locating more information or specimen material about individual cases (in a second step that is not discussed in this paper.)

We are aiming for a linkage algorithm that has a high overall specificity. We made the statistical considerations about false-positive errors only to estimate the efficiency and privacy-safety of the semi-join operation. The effect of false positive errors in the semi-join filtering only decreases the efficiency of the semi-join filter, causing more useless records to be sent from the data source to the mediator. However, the mediator performs matching with hashed identifier vectors that are not truncated and have ranges of at least 2^{128}. Thus, the mediator will end up discarding the excess records. In other words, false positive errors through the full (not truncated) hash-coding are practically negligible in the end-result of the linkage.

For our primary use cases, we consider specificity more important than sensitivity. This is consistent with the ways in which retrospective cohort studies or case control studies are usually performed. Traditionally the cases of a cohort are collected through manual chart review which necessarily will only include samples (including considerable selection bias.)

However, our method as developed so far may not be suitable for epidemiological studies that aim for complete population counts. When querying an ad-hoc community of data sources, we can give no guarantee about the coverage one can achieve unless one can make sure that enough of the critical data sources are available for queries. This is particularly an issue if the coverage achieved with a query is unknown to the researcher.

What would be a potential strategy to minimize the effect of hash collisions? Given that we know the expected number of false positives, how can we improve the "fusion query" results? What you do propose to perform as future research to address this issue?

We need to point out again, that the false-positive error rates that we focused on in this paper, only influence the sensitivity of the linkage. The specificity of the linkage is influenced by the matching algorithm used by the mediator, not the one used for the semi-join filters. That said, we realize that more work needs to be done on the linkage algorithm. For one, we have reported in this paper about considerable problems with trading off efficiency and privacy of the kind of disjunctive deterministic linkage algorithm that we have studied so far. However, our problem overall is not false positives, but false negatives.

We may restrict our matching to a single deterministic rule similar to rule 1 (2 or 3 respectively). We know for the general population that we have studied, that we could miss over 30% of true matches due to missing social security number. However, for certain specific use cases, such as research about cancer care, the false-negative error may be much less. In the population of patients that have inpatient treatments and many follow-up visits the quality of the identifying data, including social security number, may be much better than in the general population that includes many patients with very few sporadic visits. We will assess this hypothesis more carefully.

For general clinical research, however, including risk factor analysis for cancer, the errors of omission due to sole reliance on social security numbers for linking will still be a problem.

Can you provide some indication of how this approach differs from the statistical literature on record linkage and its accuracy (e.g., the Fellegi-Sunter method), data fusion, and the statistical papers on intruder detection by Paas, Fienberg et al. ?

This question includes two issues. (1) It reminds us that we may need to revise our bias that deterministic matching is superior and rather look for ways to perform privacy-protecting joins using statistical matching. (2) It requests clarifying our position on the privacy risk involved with the disclosure of clinical data itself, even though the common person identifiers may be successfully protected.

(1) Deterministic vs. probabilistic matching and Fellegi-Sunter: We need to point out that what we called "deterministic matching" can be considered a special case that is be described by the Fellegi-Sunter theory. Notably

for each record pair we calculate a comparison vector γ, which has 4 components, one for each matching rule 1–4 listed in Section 3.2. The value space Γ is limited to 2^4 different values (true/false for each of the four rules.) our decision threshold T is set such as to accept everything as a match except the outcome where none of the four components indicate match. This threshold relates to a likelihood ratio $m(\gamma)/u(\gamma)$ of approximately 30.

We need to reiterate that for record linkage for join queries, human review cannot be afforded, and hence the two decision levels defined by Fellegi and Sunter must fall together to one point.

Because all of our comparison vector components share some of the same data elements, they are clearly dependent, hence we cannot use the simplification that most of the applications of the Fellegi-Sunter method rely on, i.e., to define γ such that its components are statistically independent:

$$\frac{m(\gamma)}{u(\gamma)} = \prod_i \frac{m(\gamma_i)}{u(\gamma_i)}$$

and with $w(\gamma) = \log m(\gamma) - \log u(\gamma)$

$$w(\gamma) = \sum_i w(\gamma_i)$$

In other words, most of the practical applications of the Fellegi-Sunter method consider the patient identifier components individually and derive the weights of matching each identifier component individually. The very attractive implication of this approach is that that the matching scores on each component are simply added and that matching on less common names can receive a higher weight than matching on more common names.

The work by Quantin (1997) and Michaelis (1995) is the more canonical probabilistic matching technique applied to hashed or encrypted identifiers components. However, their work is still in a setting of a static registry rather than a dynamic query. If we change HMAC keys for every query, then the weights used in the statistical matching method would have to be regenerated every time, which seems not feasible.

Winkler (1997) indicates that matching comparison vectors that consider name frequency do not necessarily deliver superior matching results as opposed to simple match/no-match outcomes. In this case, we could employ more conventional Fellegi-Sunter matching with weights for each of the independent identifier components not considering their values.

The reduction of the hash code range would increase the false positive probability $u(\gamma(p,q)) = P(\gamma(p,q) \mid p \tilde{\neq} q)$ in the Fellegi-Sunter likelihood ratios while leaving the true positive $m(\gamma(p,q)) = P(\gamma(p,q) \mid p \cong q)$ unchanged. However, in the error calculations for Bloom-filters we have relied on the assumption that the hash codes are uniformly distributed over the hash code range. This would no longer be the case (because there is much greater probability that the hash bin for the name "Smith" is used than that for the name "Schadow".) Our future work must include factoring the effect of false positives due to hash collisions from the Fellegi-Sunter theory such that we can understand the effect of hash collisions on the matching performance of the semi-join filters.

The advantage of relying more on the Fellegi-Sunter method is that this would immediately reflect our privacy model. It may be recalled that we used likelihood ratios to determine the level of privacy, and our likelihood ratios are the same as the $m(\gamma) / u(\gamma)$ ratios of the Fellegi-Sunter method. Hence, it is trivial to know the privacy level from the threshold used to accept an identifier as member of the semi-join filter.

This suggests, however, that we will run into a similar problem that we found in our disjunctive hash algorithm: at the point where the performance of the semi-join filter is good, an intruder could use different acceptance thresholds to re-identify patients with considerably better likelihood ratios if the intruder is willing to accept lower sensitivity. Worse yet, because the matching weights make it so easy to calculate the likelihood ratio for every identifier found in the semi-join filter, the intruder could decide on the level of acceptance differently for each individual. For example, a data source associated with an insurance business could use a Bayesian utility model to determine with ease whether to renew or cancel insurance contracts based on the inference from the semi-join filters with striking clarity.

This underlines the importance of adjusting the specificity of semi-join filters depending on the nature of the data source, to the point where data sources could not receive any semi-join filter. The advantage of using the Fellegi-Sunter method more directly would lie in a better understanding of the privacy risk, hence better tuned protection policies.

(2) Privacy risk from released data. Much research has been done on privacy protection in public data releases as recently summarized by Abowd and Woodcock (2001). This research is inspired by the increasing demand for public microdata-releases from governmental data collections (census data, Medicare billing data, social security data, etc.) This has become a concerning problem. For example, in many U.S. states, inpatient hospital visit data sets are publicly available and may only be weakly de-identified. This data along with other sources can be used to combine and refine knowledge about de-identified patients and eventually to re-identify patients.

Even though our Bayesian model of privacy is in principle similar to the approach used by Fienberg, Makov and Sanil (1997) we have not concerned ourselves with de-identifying clinical data sets in general (beyond the specific patient identifiers) in this paper, although we clearly require this sort of de-identification (or data scrubbing) to be done before individual patient data is delivered as a result of the query. However, since our goal is to link real patient cases, we assume by definition that the query mediator is run by an entity that is generally trusted to abide by the policies of the network that forbid any sort of re-identification and accumulation of data beyond the limited scope of a single query.

We consider dynamic querying in a distributed network of autonomous data sources an alternative to the increasing practice (and problem) of public microdata releases. The advantage of the distributed queries would be that the clinical data, on which the query results are based, need not be published, but can be deleted after each query. The role of the mediator is critical, it is much easier to control than public data releases and hence can be entrusted with more detail to generate results that could not be generated from microdata releases (particularly outcome research and research of risk factors.)

4.2 Future Work

We have much further research to do. Our program includes a more precise study of the kinds of distributed queries we have to support beyond the simple fusion queries described by Abiteboul and Yerneni. For instance, our requirements include aggregate queries and correlated sub-queries not considered by Abiteboul and Yerneni (e.g., to query for patients with a certain temporal event pattern.)

As indicated above, we may for practical purposes proceed with a simple match on a single hashed patient identifier vector based on the social security number, and we are working on a more direct integration of probabilistic matching with the semi-join filters (rather than the disjunctive deterministic rules.)

Clearly we need to apply the extensive research on disclosure control in released data to any of the results sent to the mediator and those send from the mediator to the querying researcher.

We also intend to define a more sophisticated architecture of the network, particularly such that location information from responses is hidden. Location information can have a great value in re-identifying patient information. Ideally we would also like to separate hashed identifiers completely from explicit clinical data such that the record linkage is performed in a special network node separately from the clinical data. However, these architectural concerns are secondary given the more fundamental outstanding issues.

5 Conclusion

We have shown how natural joins for fusion queries that protect privacy can be supported using keyed hash functions and a calculated reduction in hash space to increase ambiguity of the hash codes in a semi-join filter. Applied to the reality of health care databases, where patients are only loosely identified by their demographic data, there is a limitation of how much link-sensitivity and privacy and semi-join-efficiency one can achieve at the same time. Paradoxically we find that, a good globally unique health record identifier could improve the privacy conscious research use of medical data, because it removes dependency between the identifier components that can be exploited to infer information from hash-filters. Still we find it is possible to build a fusion-query mediator that uses the simple reciprocal relationship between likelihood ratio and retrieval probability to adjust the hash code size suitable for each data source that it queries, considering the cardinality of data at that source as well as the trustworthiness that the source will abide by the policy of the network.

Acknowledgements

This work has been performed at the Regenstrief Institute for Health Care, Indianapolis, IN, with support in part by the National Cancer Institute (1 U01 CA91343-01) and the National Library of Medicine (T15 LM-7117-05.) The mentioning of the SPIN collaborative in this paper is an example only and does not imply that this paper reflects the consensus of the SPIN collaborative.

References

ABITEBOUL S., GARCIA-MOLINA H., ET AL. (1997): Fusion queries over Internet databases. Technical Report, Stanford University. http://www-db.stanford.edu /pub/papers/fqo.ps

ABOWD J. M. AND WOODCOCK S. D. (2001): Disclosure limitation in longitudinal linked data. In Confidentiality, Disclosure, and Data Access: Theory and Practial Applications for Statistical Agencies, Doyle, P., Lane, J., et al. Eds. Amsterdam: North Holland:215-277.

AGRAWAL R. AND SIRKANT R (2000): Privacy-Preserving Data Mining. *SIGMOD* **29**(2):439-450

BABB E. (1979): Implementing a relational database by means of specialized hardware. *ACM Transactions on Database Systems* **4**(1):1-29.

BERNSTEIN, P. A. AND CHIU, D. M. (1981): Using semi-joins to solve relational queries. *J ACM* **28**(1):25-40.

BLOOM, B. H. (1970): Space/time trade-offs in hash coding with allowable errors. *CACM* **13**(7):422-426

DUSSERRE, L., QUANTIN, C. ET AL. (1995): A one-way public key cryptosystem for the linkage of nominal files in epidemiological studies. *Proc MEDINFO*:644-647.

FELLEGI I. P. AND SUNTER A. B. (1969): A theory of record linkage. *J Am Stat Assoc* **64**(328):1183-1210.

FIENBERG, S. E., MAKOV, E. U. AND SANIL, A. P. (1997). A Bayesian approach to data disclosure: optimal intruder behavior for continuous data. Journal of Official Statistics, **13**:75-89.

GRANNIS S, MCDONALD CJ, OVERHAGE JM (2002): Analysis of a record linkage algorithm [accepted for publication]. In *Proc AMIA Symp*.

HHS (2000): Standards for privacy of individually identifiable health information [45 CFR Parts 160 and 164, final rule]. *Federal Register* **65**(250): 82802-82829

KOSSMAN D. (2000): The state of the art in distributed query processing. *ACM Computing Surveys* **32**(4):422-469

MAHON B.E., ROSENMAN M.B. AND KLEINMAN M.B. (2001): Maternal and infant use of erythromycin

and other macrolide antibiotics as risk factors for infantile hypertrophic pyloric stenosis. *J Pediatr* **139**(3):380-384.

MCDONALD C.J., OVERHAGE J.M., ET AL. (1999): The Regenstrief Medical Record System: a quarter century experience. *Int J Med Inf* **54**(3):225-253.

MCDONALD C.J., SCHADOW G. ET AL. (2001): Data standards in health care. *Ann Emerg Med* **38**(3):303-311.

MICHAELIS, J., MILLER, M., ET. AL. (1995): A new concept to ensure data privacy and data security in cancer registries. *Proc MEDINFO*:661-665.

MULIN J.K. (1999): Optimal semijoins for distributed database systems. *IEEE Transactions on Software Engineering* **16**(5):558-560.

MULLIN J.K. (1983): A second look at Bloom filters. *CACM* **26**(8):570f.

MURRAY M.D., BRATER D.C. ET AL. (1990): Ibuprofen-associated renal impairment in a large general internal medicine practice. *Am J Med Sci* **299**(4): 222-229.

NIST (2002): The keyed-hash message authentication code (HMAC). FIPS PUB 198. Gaithersburgh, National Institute for Standards and Technology. http:// csrc.nist.gov/publications/fips/fips198/fips-198a.pdf.

ÖZSU, M.T. AND VALDURIEZ, P. (1990): Principles of distributed database systems. Englewood Cliffs, Prentice-Hall.

QUANTIN C., BOUZELAT H ET AL. (1997): Projet de protocole d'échanges sécurisés entre cabinets médicaux et établissements hospitaliers publics et privés de la région Bourgogne en vue d'études épidémiologiques. In *Informatique et Gestion Médicalisée*. Kohler F., Brémond M. and Mayeux D. (eds.). Paris, Springer-Verlag.

ROTH, M. T. AND SCHWARZ P. (1997): Don't scrap it, wrap it; a wrapper architecture for legacy data sources. *Proc VLDB*.

SHETH, A. P. AND LARSON, J. A. (1990): Federated database systems for managing distributed heterogenous and autonomous databases. *ACM Computing Surveys* **22**(3):183-236.

TIERNEY, W. M. AND MCDONALD, C.J. (1991): Practice databases and their uses in clinical research. *Stat Med* **10**:541-557.

WINKLER, W. E. (1999): The state of record linkage and current research problems. Technical Report, U.S. Bureau of the Census. http://www.census.gov/srd/papers/pdf/ rr99-04.pdf

YERNENI R, PAPAKONSTANTINOU Y, ET AL. (1998): Fusion queries over Internet databases. In *Proc EDBT*, Schek, H.J. et al. (eds.). Springer-Verlag: 57-71

Appendix: HIPAA Privacy Regulations

In the United States health care system, the primary norm for privacy of electronic medical information is the HIPAA law and its subsequent privacy regulations. Under the HIPAA privacy regulations (HHS, 2000; in the following referred to as HIPAA) *individually identified health information* generally must not be used or disclosed except (1) for treatment of that patient, or (2) with request or consent of a patient or representative, for treatment and (3) for payment, and healthcare operations and any of the numerous exceptions [§164.502]. Research use constitutes one such exception, but only if an IRB grants a waiver under the minimal risk condition [§164.512(i)].

HIPAA defines protected health information based on the *individual identifiability* [§164.501] and explicitly allows *de-identification* as a means of exempting health information from HIPAA protection. Because de-identification is never absolute, HIPAA provides for two modes of de-identification: (1) removing all of a list of 19 kinds of data about the individual, relatives, household members and employers and having *no actual knowledge* that this data could be re-identified; or (2) keeping some of those data elements and having documented scientific evidence that the risk for such re-identification is "very small" [§164.514]. A system that links records across institutions disclosing only de-identified information would comply with HIPAA. However, de-identification is by definition not consistent with distributed data joining.

Recent revisions to the HIPAA privacy regulation has eased up the need for thorough de-identification somewhat. According to the new rules, a "limited data set" [§164.514(e)] may be disclosed for research purpose, if the recipient enters into a binding "data use agreement" which does not allow attempts to re-identifying any data. A limited data set contains individual clinical data from which all face-identifiers, such as names, personal numbers, telecommunication and street address had been removed.

While in our project based in the U.S., HIPAA has to be our guiding norm; our work on a large-scale research network is not tailored to the HIPAA regulations, but instead aims at designing a network that protects the patients' privacy beyond today's written law. Particularly the implications of large scale collaborative research may have not yet been foreseen by the regulation authority. For that reason, we are proposing some provisions that might not seem necessary under HIPPAA today, that, however, we believe are prudent in order to preserve the scalability of the privacy protection as this network grows.