Hippocratic Databases

Rakesh Agrawal **Jerry Kiernan** **Ramakrishnan Srikant** **Yirong Xu**

IBM Almaden Research Center
650 Harry Road, San Jose, CA 95120

Abstract

The Hippocratic Oath has guided the conduct of physicians for centuries. Inspired by its tenet of preserving privacy, we argue that future database systems must include responsibility for the privacy of data they manage as a founding tenet. We enunciate the key privacy principles for such Hippocratic database systems. We propose a strawman design for Hippocratic databases, identify the technical challenges and problems in designing such databases, and suggest some approaches that may lead to solutions. Our hope is that this paper will serve to catalyze a fruitful and exciting direction for future database research.

1 Introduction

> "And about whatever I may see or hear in treatment, or even without treatment, in the life of human beings – things that should not ever be blurted out outside – I will remain silent, holding such things to be unutterable" – Hippocratic Oath, 8[1]

The explosive progress in networking, storage, and processor technologies is resulting in an unprecedented amount of digitization of information. It is estimated that the amount of information in the world is doubling every 20 months, and the size and number of databases are increasing even faster [37]. In concert with this dramatic and escalating increase in digital data, concerns about the privacy of personal information have emerged globally [15] [17] [37] [51]. Privacy issues are further exacerbated now that the Internet makes it easy for new data to be automatically collected and added to databases [6] [10] [58] [59] [60].

Privacy is the right of individuals to determine for themselves when, how and to what extent information about them is communicated to others.[2] Privacy concerns are being fueled by an ever increasing list of privacy violations, ranging from privacy accidents to illegal actions. Of equal concern is the lax security for sensitive data. See Appendix A for some examples of recent privacy violations. Database systems, with their ubiquitous acceptance as the primary tool for information management, are in the middle of this gathering storm.

We suggest that the database community has an opportunity to play a central role in this crucial debate involving the most cherished of human freedoms[3] by re-architecting our database systems to include responsibility for the privacy of data as a fundamental tenet. We have been inspired by the privacy tenet of the Hippocratic Oath, and propose that the databases that include privacy as a central concern be called Hippocratic databases. We enunciate the key principles for such Hippocratic database systems, distilled from the principles behind current privacy legislations and guidelines. We identify the technical challenges and problems in designing Hippocratic databases, and also outline some approaches that may lead to solutions. Our hope is that future database research will convert the Hippocratic database vision into reality.

We recognize that technology alone cannot address all of the concerns surrounding a complex issue like privacy. The total solution has to be a goulash of laws, societal norms, markets, and technology [32]. However, by advancing what is technically realizable, we can influence the proportion of the ingredients and the overall quality of the solution. We also recognize that all of the world's data does not live in database systems. We hope the Hippocratic databases will provide additional inducement for privacy-sensitive data to move to its right home. If nothing else,

[1] Translation by Heinrich Von Staden. In a Pure and Holy Way: Personal and Professional Conduct in the Hippocratic Oath. *Journal of the History of Medicine and Applied Sciences* **51** (1966) 406–408.

[2] This definition is attributed to Alan Westin, Professor Emeritus of Public Law and Government, Columbia University.

[3] Samuel Warren and Louis Brandeis. The right to privacy. *Harvard Law Review* **4** (1890) 193–220. See also [2].

Proceedings of the 28th VLDB Conference,
Hong Kong, China, 2002

Hippocratic databases can provide guidance for incorporating similar principles in other types of data repositories.

The structure of the rest of the paper is as follows. Section 2 discusses current database systems, focusing on the closest related work: statistical databases and secure databases. We define the founding principles of Hippocratic databases in Section 3, and sketch out a strawman design for a Hippocratic database in Section 4. We give a set of technical challenges in Section 5, and conclude with some closing remarks in Section 6.

2 Current Database Systems

In [53], the following two properties are considered fundamental for a database system:

1. The ability to manage persistent data.
2. The ability to access a large amount of data efficiently.

In addition, the following capabilities are said to be found universally:

1. Support for at least one data model.
2. Support for certain high-level languages that allow the user to define the structure of data, access data, and manipulate data.
3. Transaction management, the capability to provide correct, concurrent access to the database by many users at once.
4. Access control, the ability to deny access to data by unauthorized users and the ability to check the validity of the data.
5. Resiliency, the ability to recover from system failures without losing data.

Other database text books also provide a similar list for the capabilities of a database system [16] [40] [48]. For instance, in [48], the primary goal of a database system is said to be providing an environment that is both convenient and efficient to use in retrieving and storing information. The control of redundancy is stated as an additional capability in [16]. Interestingly, access control is not mentioned in [48], although they do discuss integrity constraints.

Clearly, a Hippocratic database will need the capabilities provided by current database systems. However, given the design goals of current database systems, it is not surprising that they fall short in providing a platform for privacy sensitive applications. In fact, efficiency – though it will continue to be important – may not be the central focus of Hippocratic databases. They may place greater emphasis on consented sharing rather than on maximizing concurrency. The need for the database system to completely forget some data beyond the purpose for which it was collected has interesting implications on the current resiliency schemes. There will be new demands on the data definition and query languages, query processing, indexing and storage structures, and access control mechanisms. In short, Hippocratic databases will require us to rethink almost all aspects of current database systems.

2.1 Statistical Databases

The research in statistical databases was motivated by the desire to be able to provide statistical information (sum, count, average, maximum, minimum, pth percentile, etc.) without compromising sensitive information about individuals [1] [47]. The proposed techniques can be broadly classified into query restriction and data perturbation. The query restriction family includes restricting the size of query results [13] [18], controlling the overlap among successive queries [14], keeping audit trails of all answered queries and constantly checking for possible compromises [8], suppression of data cells of small size [9], and clustering entities into mutually exclusive atomic populations [61]. The perturbation family includes swapping values between records [12], replacing the original database by a sample from the same distribution [33] [42], adding noise to the values in the database [52] [57], adding noise to the results of a query [4], and sampling the result of a query [11].

Hippocratic databases share with statistical databases the goal of preventing disclosure of private information, and hence some of the techniques developed for statistical databases will find application in Hippocratic databases. However, the class of queries that Hippocratic databases have to contend with is much broader.

2.2 Secure Databases

Whenever sensitive information is exchanged, it must be transmitted over a secure channel and stored securely to prevent unauthorized access. There is extensive literature on access control and encryption that is relevant [12] [38] [45] [46]. Hippocratic databases will also benefit from the work on database security [7] [30]. Of particular interest is work on multilevel relations in the context of multilevel secure databases [23] [24] [50]. It allows multiple levels of security (e.g., top secret, secret, confidential, unclassified) to be defined and associated with individual attribute values. The security level of a query may be higher or lower than that of individual data items. A query with a lower level of security cannot read a data item requiring a higher level of clearance. On the other hand, a higher security query cannot write a lower security data item. Two queries having different levels of security can thus generate different answers over the same database. Many of our architectural ideas about Hippocratic databases have been inspired by this work.

3 Founding Principles of a Hippocratic Database

We first present a summary of some of the current privacy regulations and guidelines. Our founding principles are motivated by, and based on the principles underlying these regulations.

3.1 Privacy Regulations and Guidelines

The United States Privacy Act of 1974 set out a comprehensive regime limiting the collection, use, and dissemination of personal information held by Federal agencies [43] [44]. The Act requires the agencies to

i) permit an individual to determine what records pertaining to him are collected, maintained, used, or disseminated;

ii) permit an individual to prevent records pertaining to him obtained for a particular purpose from being used or made available for another purpose without his consent;

iii) permit an individual to gain access to information pertaining to him in records, and to correct or amend such records;

iv) collect, maintain, use or disseminate any record of personally identifiable information in a manner that assures that such action is for a necessary and lawful purpose, that the information is current and accurate for its intended use, and that adequate safeguards are provided to prevent misuse of such information;

v) permit exemptions from the requirements with respect to the records provided in this Act only in those cases where there is an important public policy need for such exemption as has been determined by specific statutory authority; and

vi) be subject to civil suit for any damages which occur as a result of willful or intentional action which violates any individual's right under this Act.

The concepts underlying the Privacy Act have come to be known as Fair Information Practices [55], and have contributed to the development of important international guidelines for privacy protection. The most well known of these are the OECD guidelines, which set out eight principles for data protection: collection limitation, data quality, purpose specification, use limitation, security safeguards, openness, individual participation, and accountability [43]. Consent and disclosure limitation are covered under collection limitation and use limitation respectively. Countries around the world have used OECD guidelines to develop legal codes [5].

The Canadian Standard Association's Model Code for the protection of Personal Information builds upon the OECD guidelines and suggests standards for the design of information systems. The CSA Model Code is quite similar to the OECD guidelines; the main differences are that the CSA makes consent and disclosure limitation separate principles, and adds retention limitation as a new principle [43].

The Japanese legislature is currently debating a bill aimed at regulating the acquisition, custody, and use of personal information, with the aim of putting the law into effect in early 2003. The proposed bill stipulates five basic principles regarding the collection and use of personal information: i) information must not be used other than for clear, specified purposes; ii) information must be collected properly; iii) information must always be correct and up-to-date; iv) information must be kept secure and safe from leakage; and v) information must be handled in a transparent manner that properly involves individuals (www.fpcj.jp/e/shiryo/jb/0113.html).

The Australian Privacy Amendment (Private Sector) Act 2000 is based on ten principles (www.privacy.gov.au/publications/npps01.html). The first six principles are quite similar to other guidelines: collection, use and disclosure, data quality, data security, openness, and access. The next two principles are: vii) identifiers: companies may not adopt government-issued identifiers in lieu of issuing their own identifiers; and viii) anonymity: individuals should have the option of remaining anonymous whenever lawful and practicable. The ninth principle restricts trans-border information flows to companies that follow similar principles, and the last principle requires extra care with sensitive (e.g., health) information.

Finally, we mention two recent industry-specific privacy regulations passed in the U.S. The 1996 Health Insurance Portability and Accountability Act (www.hhs.gov/ocr/hipaa/) gives patients control over how their personal medical information is used and disclosed. The 1999 Gramm-Leach-Bliley Financial Services Modernization Act (banking.senate.gov/conf/) requires financial institutions to disclose their privacy policies and allows consumers to opt-out of sharing of personal information with nonaffiliated third parties.

3.2 The Ten Principles

We now enunciate what we believe can become the founding principles of Hippocratic database systems. These principles are rooted in the privacy regulations and guidelines described above. They articulate what it means for a database system to responsibly manage private information under its control. They also define what a donor of private information can expect if a database system advertises itself to be Hippocratic.

1. **Purpose Specification** For personal information stored in the database, the purposes for which the information has been collected shall be associated with that information.

2. **Consent** The purposes associated with personal information shall have consent of the donor of the personal information.

3. **Limited Collection** The personal information collected shall be limited to the minimum necessary for accomplishing the specified purposes.

4. **Limited Use** The database shall run only those queries that are consistent with the purposes for which the information has been collected.

5. **Limited Disclosure** The personal information stored in the database shall not be communicated outside the database for purposes other than those for which there is consent from the donor of the information.

6. **Limited Retention** Personal information shall be retained only as long as necessary for the fulfillment of the purposes for which it has been collected.

7. **Accuracy** Personal information stored in the database shall be accurate and up-to-date.

8. **Safety** Personal information shall be protected by security safeguards against theft and other misappropriations.

9. **Openness** A donor shall be able to access all information about the donor stored in the database.

10. **Compliance** A donor shall be able to verify compliance with the above principles. Similarly, the database shall be able to address a challenge concerning compliance.

4 Strawman Design

We now outline a strawman design of a Hippocratic database system. The purpose of this exercise is not to provide a complete blueprint for implementing a Hippocratic database, but rather to show the directions in which we can proceed in order to create a Hippocratic database.

We first present an example that we will use to illustrate the strawman design.

4.1 A Use Scenario

Mississippi is an on-line bookseller who needs to obtain certain minimum personal information to complete a purchase transaction. This information includes name, shipping address, and credit card number. Mississippi also needs an email address to notify the customer of the status of the order. Mississippi uses the purchase history of customers to offer book recommendations on its site. It also publishes information about books popular in the various regions of the country (purchase circles).

Alice is a privacy fundamentalist who does not want Mississippi to retain any information once her purchase transaction is complete. Bob, on the other hand, is a privacy pragmatist who likes the convenience of providing his email and shipping address only once by registering at Mississippi. He also likes Mississippi's recommendations and does not mind Mississippi using his purchase transactions to suggest new recommendations. However, he does not want Mississippi to use his transactions for purchase circles.

Mississippi is an enlightened merchant who deploys a Hippocratic database to support the privacy wishes of its customers. Trent is Mississippi's privacy officer who is responsible for ensuring that the information systems comply with the company's privacy policies. Mallory is a Mississippi employee with questionable ethics.

4.2 Architecture

Figure 1 shows the strawman architecture of a Hippocratic database system. In this design, we use *purpose* as the central concept around which we build privacy protection.

4.2.1 Privacy Metadata

The *privacy metadata* tables define for each purpose, and for each piece of information (attribute) collected for that purpose:

- the *external-recipients*: whom the information can be given out to,
- the *retention-period*: how long the information is stored, and
- the *authorized-users*: the set of users (applications) who can access this information.

Conceptually, we split the above information into two separate tables, whose schemas are as shown in Figure 2.[4] The external-recipients and retention attributes are in the *privacy-policies* table, while the authorized-users attribute is in the *privacy-authorizations* table. The former captures the privacy policy, while the latter captures the access controls that support the privacy policy.

Continuing our example, Figure 3 shows the schema of the two tables, *customer* and *order*, that store the personal information collected by Mississippi. Notice that we have added a special attribute, "purpose", to each table, which is similar to attaching security level with records in secure databases [24] [50]. Mississippi's privacy policy for the purchase, registration, recommendations and purchase-circles purposes are shown in Figure 4. For the purchase purpose:

- all the attributes have a retention period of 1 month,
- the name and shipping-address are given to the delivery company, and
- the name and credit-card-info are given to the credit-card company.

The authorizations that support this policy are shown in Figure 5. In our example, the shipping application is authorized to access the name, shipping-address, email, and set of books. Similarly, the charge application is authorized to access the name and credit-card-info. Notice that these

[4] For ease of exposition, we have set-valued attributes in these tables, rather than normalizing the table.

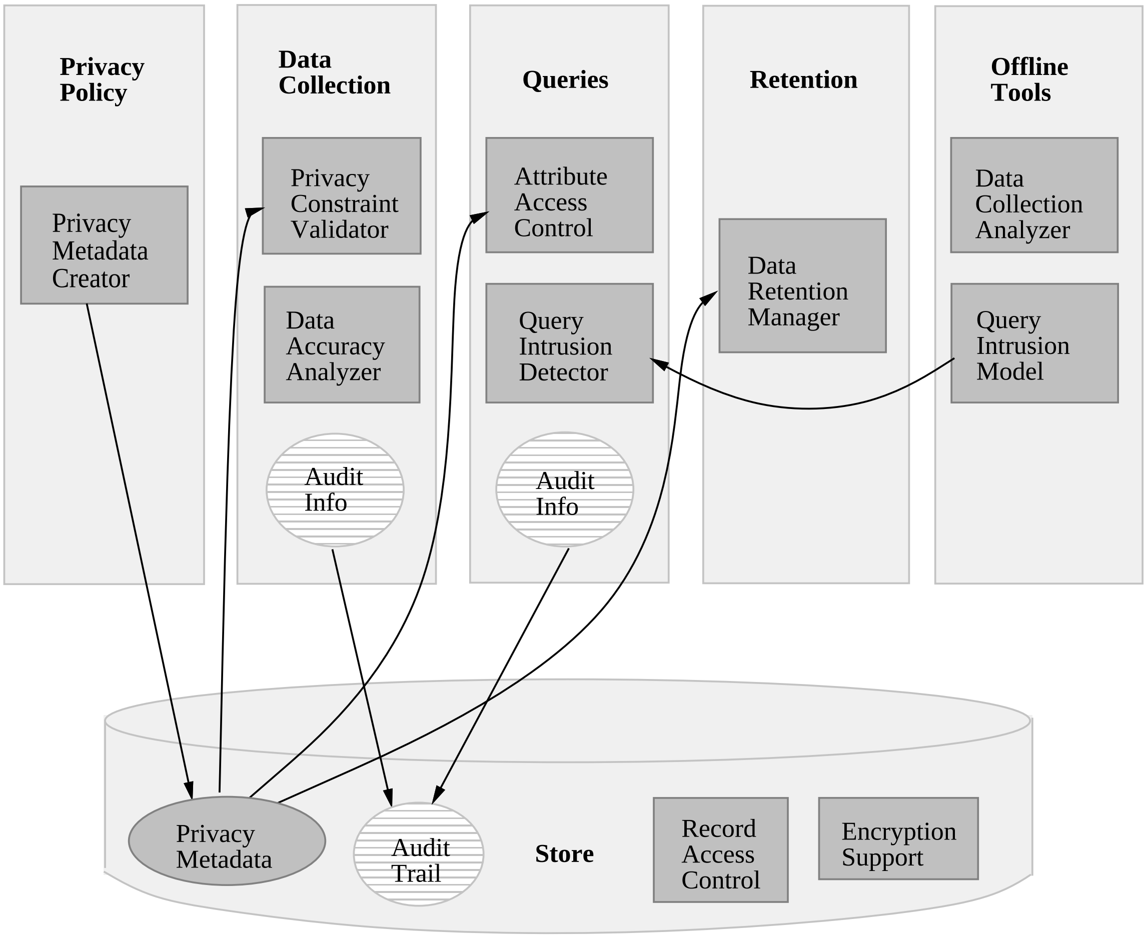

Figure 1: Strawman Architecture

authorizations implement many of the external-recipients constraints in the privacy-policies table: shipping cannot access credit-card-info, and charge cannot access book-info or shipping-address.

Trent first designs the privacy policy, and then uses the *Privacy Metadata Creator* to generate the privacy metadata tables. The mapping from the privacy policy to the privacy-policies table makes use of automated tools. Creating the privacy-authorizations table requires understanding of who should have access to what data, which in turn is constrained by the privacy policy.

Alternate Organizations The above design assumes that purpose together with attribute completely determine the set of recipients and retention period. There is also an implicit assumption that the set of attributes collected for a purpose is fixed. These assumptions can be limiting in some situations. Consider a scenario where Bob agrees to give his business phone number for “contact” purpose. He also consents to “telemarketing” purpose for his home phone number, but opts out of his business phone number for this purpose. Now if purpose opt-ins or opt-outs are tracked per person, not per ⟨person, attribute⟩ pair, a query will be able to incorrectly obtain Bob’s business phone number for telemarketing purposes. These limitations can sometimes be circumvented by splitting a conceptual purpose into multiple database purposes. In the above example, we would split “telemarketing” into “telemarketing-home” and “telemarketing-business”.

If the above assumptions do not generally hold, alternate data organizations may be more suitable. For example, one can create a table with the columns {user, table, attribute, purpose, recipient} that allows any desired combinations of attribute, purpose, and recipient for each user. The tradeoff is that the run-time overhead for checking whether a query can access a user’s data is likely to be substantially higher. It is also possible to design in-between data organizations. For instance, one can store a set of purposes for each attribute in the record, rather than once per record. In this design, purposes will only require that the set of recipients be fixed, not the set of attributes collected for that purpose.

4.2.2 Data Collection

Matching Privacy Policy with User Preferences Before the user provides any information, the *Privacy Constraint*

table	**attributes**
privacy-policies	purpose, table, attribute, { external-recipients }, retention
privacy-authorizations	purpose, table, attribute, { authorized-users }

Figure 2: Privacy Metadata Schema

table	**attributes**
customer	purpose, customer-id, name, shipping-address, email, credit-card-info
order	purpose, customer-id, transaction-id, book-info, status

Figure 3: Database Schema

purpose	**table**	**attribute**	**external-recipients**	**retention**
purchase	customer	name	{ delivery-company, credit-card-company }	1 month
purchase	customer	shipping-address	{ delivery-company }	1 month
purchase	customer	email	*empty*	1 month
purchase	customer	credit-card-info	{ credit-card-company}	1 month
purchase	order	book-info	*empty*	1 month
registration	customer	name	*empty*	3 years
registration	customer	shipping-address	*empty*	3 years
registration	customer	email	*empty*	3 years
recommendations	order	book-info	*empty*	10 years
purchase-circles	customer	shipping-address	*empty*	1 year
purchase-circles	order	book-info	{ aggregated-all }	1 year

Figure 4: Privacy-Policies Table

purpose	**table**	**attribute**	**authorized-users**
purchase	customer	customer-id	*all*
purchase	customer	name	{ shipping, charge, customer-service }
purchase	customer	shipping-address	{ shipping }
purchase	customer	email	{ shipping, customer-service }
purchase	customer	credit-card-info	{ charge }
purchase	order	customer-id	*all*
purchase	order	transaction-id	*all*
purchase	order	book-info	{ shipping }
purchase	order	status	{ shipping, customer-service }
registration	customer	customer-id	*all*
registration	customer	name	{ registration, customer-service }
registration	customer	shipping-address	{ registration }
registration	customer	email	{ registration, customer-service }
recommendations	order	customer-id	{ mining }
recommendations	order	transaction-id	{ mining }
recommendations	order	book-info	{ mining }
purchase-circles	customer	customer-id	{ olap }
purchase-circles	customer	shipping-address	{ olap }
purchase-circles	order	customer-id	{ olap }
purchase-circles	order	book-info	{ olap }

Figure 5: Privacy-Authorizations Table

Validator checks whether the business' privacy policy is acceptable to the user. The input to the validator is the user's privacy preferences (constraints). In our example, Alice's preference would be to opt out of everything except purchase, and she may have a constraint that purchase information should not be kept for more than 3 months. If on the other hand, Alice required a retention period of 2 weeks, the database would reject the transaction. Similarly, Bob may opt-in for the recommendations purpose but not for the purchase-circles purpose. This interaction may occur using a direct encrypted connection between the database and the user's client [39]. An *audit trail* of the user's acceptance of the database's privacy policy is maintained in order to address challenges regarding compliance.

Data Insertion Having checked that the privacy policy does not violate the user's privacy preferences, data is transmitted from the user and stored in the tables. Each record has a special attribute, "purpose", that encodes the set of purposes the user agreed to. In our example, Alice's records would have a single purpose: purchase, while Bob's records would have three purposes: purchase, registration and recommendations. The set of purposes combined with the information in the privacy-authorizations table will be used to restrict access.

Data Preprocessing The *Data Accuracy Analyzer* may run some data cleansing functions [19] [41] against the data to check for accuracy either before or after data insertion, thus addressing the Principle of Accuracy. In our example, Alice's address may be checked against a database of street addresses to catch typos in the address or zip code.

4.2.3 Queries

Queries are submitted to the database along with their intended purpose. For example, a query that mines associations to build a recommendation model would be tagged with the purpose "recommendations".

Before Query Execution A query is only allowed to run if the set of authorized users for that purpose in the privacy-authorization table includes the user who issued the query. Next, the *Attribute Access Control* analyzes the query to check whether the query accesses any fields that are not explicitly listed for the query's purpose in the privacy-authorizations table. In our example, if Mallory in the customer-service department issues a query tagged "purchase" that accesses credit-card-info, the query will not be allowed to run, since in Figure 5, for the purchase purpose and attribute credit-card-info, authorized-users consists of only charge, and does not include customer-service.

During Query Execution For any query, the *Record Access Control* ensures that only records whose purpose attribute includes the query's purpose will be visible to the query. This is similar to the idea of multilevel relations in multilevel secure databases [24] [50]. In our example, queries tagged "recommendations" will be able to see Bob's set of books but not Alice's, since Alice's purpose attribute only lists purchase.

After Query Execution To motivate the next component, assume Mallory gives up on trying to get the credit card info, and instead decides to steal the email addresses of all registered users in Mississippi. Unlike previous attempts, neither the Attribute Access Control nor the Record Access control will be able to stop the query – customer service regularly accesses the email address in order to respond to questions about order status.

However, before the query results are returned, the *Query Intrusion Detector* is run on the query results to spot queries whose access pattern is different from the usual access pattern for queries with that purpose and by that user. The detector uses the *Query Intrusion Model* built by analyzing past queries for each purpose and each authorized-user. This problem is related to that of intrusion detection [3] [34]. In our example, the profile for queries issued by customer-service and tagged purchase might be that the query only accesses customers whose order status is not "fulfilled", and that customer-service queries cumulatively access less than 1000 records a day. Thus Mallory's queries will be flagged as highly suspicious on both counts.

An *audit trail* of all queries is maintained for external privacy audits, as well as addressing challenges regarding compliance.

4.2.4 Retention

The *Data Retention Manager* deletes data items that have outlived their purpose. If a certain data item was collected for a set of purposes, it is kept for the retention period of the purpose with the highest retention time. So Alice's information in the order table will be deleted after 1 month, while Bob's information will be kept for 10 years since Bob's purposes include both purchase and recommendations.

4.2.5 Other Features

The *Data Collection Analyzer* examines the set of queries for each purpose to determine if any information is being collected but not used, thus supporting the Principle of Limited Collection. It also determines if data is being kept for longer than necessary, and whether people have unused (unnecessary) authorizations to issue queries with a given purpose, thus supporting the Principles of Limited Retention and Limited Use. In our example, Trent may initially have given customer-service access to shipping-address; the analyzer would spot that customer-service queries never access that field and suggest to Trent that customer-service may not need access to it.

We assume the standard suite of database security features such as access control [7] [30]. Some data items may be stored in encrypted form (using the *Encryption Support*) to guard against snooping [21] [22] [39].

We did not discuss support for the Principle of Openness in this section. While supporting openness may seem easy at first glance, it in fact leads to a set of interesting problems that we discuss in Section 5.

4.3 P3P and Hippocratic Databases

Platform for Privacy Preferences (P3P), developed by the World Wide Web Consortium, is an emerging standard whose goal is to enable users to gain more control over the use of their personal information on web sites they visit. P3P provides a way for a Web site to encode its data-collection practices in a machine-readable XML format known as a P3P policy [35], which can be programmatically compared against a user's privacy preferences [31]. A major criticism of P3P has been that while P3P provides a technical mechanism for ensuring that users can be informed about privacy policies before they release personal information, it does not provide a mechanism for making sure sites act according to their stated policies [28] [44].

Hippocratic databases can go a long way in adding enforcement dimension to the P3P initiative. A P3P policy essentially describes the `purpose` of the collection of information along with the intended `recipients` and `retention period` for the collected information. The policy description uses `data` tags to specify the data items for which the policy is being stated. P3P's concepts of purpose and retention map directly to analogous concepts in Hippocratic databases. P3P lumps under recipient our concepts of external recipients and authorized users, but it is easy to map each of P3P recipient types into one of these two categories. Thus, we can take P3P policies, process them through the privacy metadata processor, and generate the corresponding data structures (i.e, the privacy-policies table in our strawman design) in the Hippocratic database system. They then can be used to assist with enforcement making use of the mechanisms provided in our architecture.

5 New Challenges

We now describe some interesting problems we identified in the course of designing the strawman system presented earlier. In some cases, we have also hinted at potential approaches for solving the problem. This list is by no means exhaustive; its purpose is to initiate discussions.

5.1 Language for Privacy Policies and Preferences

The cornerstone of a Hippocratic database is the specification of policies that are attached to data items to control their usage. As we mentioned earlier, P3P provides a standard format for encoding privacy policies [35]. P3P also has a working draft for a language, called APPEL [31], in which a user may specify privacy preferences. So, a natural question is whether P3P formats are sufficient for specifying policies and preferences in Hippocratic databases?

P3P was developed primarily for web shopping and hence the vocabulary was considerably restricted in order to reduce the complexity of the policy language. P3P has been criticized from both sides: some people think it is too complex to be usable [44] and others think it is too limiting [28]. We envision Hippocratic databases being used in a wide variety of richer environments, e.g. finance, insurance, and health care. We believe that languages for such domains should use the work done for P3P as the starting point. Developing a policy specification language for these richer environments that strikes a good balance between expressibility and usability is a difficult problem.

Ideas for reducing the complexity of the policy language include arranging purposes in a hierarchy (P3P uses a flat space). See [26] for some recent work in this direction. Subsumption relationships may also be defined for retention periods and recipients. For instance, the P3P recipients can be listed in descending order of privacy sensitivity: *ours*, *same*, *delivery*, *other-recipient*, *unrelated*, *public*. Thus, if a user agrees to one recipient in the list, the user implicitly agrees to all previous recipients since these are, in some sense, more restrictive.

On an orthogonal dimension, there is recent work on quantifying the value of privacy by formulating the problem as a coalitional game [29]. How will we accommodate in the future a user who is willing to disclose some private information only if he is fairly compensated?

5.2 Efficiency

Current database systems have undergone years of tuning to make the record processing code run extremely fast. Can they afford the additional cost of privacy checking in the path length of a record fetch? Multilevel secure databases face similar efficiency issues and it will be instructive to adopt techniques from this literature [23] [25] [50]. It is easy to see that in some cases, the record level checks can be converted into meta-data level checks. We need to understand under what conditions can these checks be compiled away or their number be reduced.

We also need techniques for reducing the cost of each check. For instance, if the number of purposes is small (less than 32), we can encode the set of purposes associated with each record by setting a bit in a word. The Record Access Control check then simply requires a bit-wise AND of two words, and checking whether the result is non-zero.

Design choices that we make for efficiency will also impact both disk space and the complexity of adding checks. For example, we could have chosen an alternate implementation in the strawman design where we only tag the records in the customer table with purpose. Then, when scanning records in the order table, we do a join on customer-id to get the purpose for those records. Thus we may save significant amount of space at the cost of speed and complexity.

5.3 Limited Collection

The limited collection principle requires that a query accesses only the data values needed to fulfill its purpose and that the database store the minimal information necessary to fulfill all the purposes. The following interesting problems arise out of ensuring that this principle is being followed:

- *Access Analysis*: Analyze the queries for each purpose and identify attributes that are collected for a given purpose but not used.
- *Granularity Analysis*: Analyze the queries for each purpose and numeric attribute and determine the granularity at which information is needed.
- *Minimal Query Generation*: Generate the minimal query that is required to solve a given problem.

At first glance, access analysis may seem trivial: couldn't we simply take a union of all the attributes mentioned anywhere in the set of queries for a given purpose? However, consider this example: assets are only needed for a mortgage application when salary is below some threshold. Thus whether information is needed for one attribute may depend on the value of other attributes.

We give two examples to motivate granularity analysis:

- Salary in the mortgage application is put into one of 3 buckets, and there is no differentiation within each bucket.
- The database stores the number of children, but queries only ask "NumChildren $>$ 0" or "NumChildren $=$ 0", i.e., it can be stored as a boolean attribute.

A potential problem with both access analysis and granularity analysis is that the redundancy may be hidden in the application code. Hence it would be nice to have minimal query generation. Will the work done in the context of universal relations [54] apply here?

5.4 Limited Disclosure

Our strawman design works well at limiting disclosure when the set of external recipients is clearly defined at the time information is submitted. However, allowing the user to dynamically determine the set of recipients provides an interesting challenge.

To make this problem concrete, consider a database of credit ratings maintained by a rating agency EquiRate. Alice is concerned about identity theft and has asked EquiRate to only disclose her credit rating to companies that she has contacted. Unfortunately, Mallory has already stolen Alice's identity and has contacted EasyCredit pretending to be Alice. Today, EasyCredit would contact EquiRate in good faith and obtain Alice's credit rating without any wrongdoing on the part of either company.

One approach to solving this problem borrows from public-private key technology [46]. For example, Alice may set up categories of personal information with a different public key for each category. When Alice contacts EasyCredit, Alice encrypts EasyCredit's company ID with her private key, and provides them the result. EasyCredit then presents the encrypted ID to EquiRate, who decrypts it with Alice's public key to verify that Alice has indeed given access to EasyCredit. Working out the details of how to make this idea deployable is an interesting problem.

5.5 Limited Retention

One can conceivably delete a record from a Hippocratic database when there is no longer any purpose associated with it. However, completely forgetting some information once it is stored in a database system is non-trivial. How do we delete a record or field not just from the data table but also from the logs and past checkpoints, without affecting recovery?

A related issue is: how do we continue to support historical analysis and statistical queries without incurring privacy breaches? Will it be sufficient to limit queries as proposed in the statistical database literature [1]?

5.6 Safety

While the database system may control unauthorized accesses to tables [7] [30], the storage media on which the tables are stored might suffer from other forms of attacks. For example, Mallory might not have permission to access a table, but instead might have super user authority which enables him to access the database files using the operating system. Encryption of database files on disk or selective encryption of fields might help [22] [39].

However, encryption has serious performance implications. Encrypting a column renders it useless for searching other than exact matches. How do we index encrypted data? How do we run queries against them? See [21] [49] for some current work on searches over encrypted data.

5.7 Openness

At first glance, openness may appear easy: is it any different from checking a bank account online? However, consider a scenario where a user wishes to access information about her but not necessarily provided by her. For example, Alice may wish to check her credit report for errors. In this scenario, how does the database check that Alice is really Alice and not someone else?[5]

A related challenge is for Alice to be able to find out what databases have information about her. If the database does not have information about Alice, the database should not know who issued the query, and Alice should not learn anything beyond the fact that the database does not have information about her. This problem is closely related to the

[5] Currently the social security number is often used in the U.S. as identification, which is problematic given the ease with which social security numbers can be obtained.

work on symmetrically private information retrieval [20] [36]. However, the computational cost of these algorithms is still too high for large-scale deployment.

5.8 Compliance

Universal Logging Generating audit trails that are in the hands of users could provide an extremely powerful tool for protecting privacy. Consider a scenario where Mallory steals the email addresses stored in Mississippi's database. If even a small fraction of the people whose email addresses were accessed by Mallory's query wondered why their email was accessed long after they made their purchase and contacted Mississippi, Trent would know that there might have been a privacy breach. Trent could then look at the audit trail of queries and might catch Mallory.

The challenge is to provide each user whose data is accessed with a log of that access along with the query reading the data, without paying a large performance penalty. A potential approach might be to use an intermediary who aggregates logs of many users and provides them access on demand. So the database only has to send the log to a small number of intermediaries rather than to a large number of users.

Tracking Privacy Breaches Another way Mississippi might track whether it has fallen prey to privacy breaches would be to use fingerprinting [27] [56]. Trent signs up with PrivacyGuard, which inserts some number of "fingerprint" records in Mississippi's database, with emails, telephone numbers and credit card numbers. If Mallory manages to steal email addresses and sells them, PrivacyGuard would know of the privacy breach in Mississippi as soon as they receive an email sent to a fingerprinted address.

The challenge is to get maximum coverage with the minimum number of fingerprint records. For example, assume that Mallory only sold the emails of those Mississippi customers who bought a certain category of books, since those email addresses were much more valuable to spammers. The percentage of Mississippi's customers who buy books in that category may be quite small, say 1%. Thus inserting fingerprint records with random purchases might be less effective than first identifying the broad categories and then inserting fingerprints based on the category.

6 Closing Remarks

Inspired by the Hippocratic Oath, we presented a vision of database systems that take responsibility for the privacy of data they manage. We enunciated the key privacy principles that such Hippocratic databases should support and presented a strawman design for a Hippocratic database. Finally, we identified the technical challenges and problems posed by the concept of Hippocratic databases.

In the landmark book *Code and Other Laws of Cyberspace*, Prof. Lawrence Lessig observes that "code is law", and that it is all a matter of code: the software and hardware that rule the Internet. We can architect cyberspace to protect values that we believe are fundamental, or we can architect it to allow those values to disappear. The question for us in the database community is: where do we want to go from here?

Acknowledgment We would like to thank Don Haderle for suggesting that we focus on the principles underlying the current privacy laws and regulations to drive the design of the Hippocratic database.

A Privacy Violations

Examples of recent privacy accidents involving database systems include:

- Kaiser, a major US health provider, accidently sent out 858 email messages containing member IDs and responses to questions on various illnesses to the wrong members. (Washington Post, 10 August 2000).
- GlobalHealthtrax, which sells health products online, inadvertently revealed customer names, home phone numbers, bank account, and credit card information of thousands of customers on their Web site. (MSNBC, 19 January 2000).

Examples of ethically questionable behavior include:

- Lotus and Equifax considered joining their credit card and demographic data and selling the results on inexpensive CDs. Similarly, Lexis-Nexis considered making Social Security Numbers available through its online news service. (Laura J. Gurak. Privacy and Persuasion in Cyberspace. 1997).
- Medical Marketing Service advertises a database available to pharmaceutical marketers which includes the names of 4.3 million people with allergies, 923,000 with bladder control problems, and 380,000 who suffer from clinical depression. (www.mmslists.com).
- Boston University has created a private company to sell the data collected for more than 50 years as part of the Framingham Heart Study. Data collected on more than 12,000 people, including medical records and genetic samples, will be sold. (New York Times, 17 June 2000).
- The chain drug stores CVS and Giant Food admitted to making patient prescription records available for use by a direct mail and pharmaceutical company. (Washington Post, 15 February 1998).

An example of illegal action:

- Toysmart.com sold confidential, personal customer information collected on the company web site in violation of its own privacy policy. (www.ftc.gov/opa/2000/07/toysmart.htm).

An example of lax security for sensitive data:

- A researcher at the Carnegie Mellon University was able to retrieve the health records of 69% of voters in Cambridge, Massachusetts from a supposedly anonymous database of state employee health insurance claims. (www.consumerreports.org/Special/ConsumerInterest/Reports/0008med0.htm).

References

[1] N. R. Adam and J. C. Wortman. Security-control methods for statistical databases. *ACM Computing Surveys*, 21(4):515–556, Dec. 1989.

[2] D. Banisar. Privacy and human rights. Electronic Privacy Information Center, 2000.

[3] D. Barbara, J. Couto, and S. Jajodia. ADAM: A testbed for exploring the use of data mining in intrusion detection. *SIGMOD Record*, 30(4):15–24, 2001.

[4] L. L. Beck. A security mechanism for statistical databases. *ACM Transactions on Database Systems*, 5(3):316–338, September 1980.

[5] C. J. Bennett. *Regulating Privacy: Data Protection and Public Policy in Europe and the United States*. Cornell Univ Press, 1992.

[6] Business Week. *Privacy on the Net*, March 2000.

[7] S. Castano, M. Fugini, G. Martella, and P. Samarati. *Database Security*. Addison Wesley, 1995.

[8] F. Chin and G. Ozsoyoglu. Auditing and infrence control in statistical databases. *IEEE Trans. on Software Eng.*, SE-8(6):113–139, April 1982.

[9] L. Cox. Suppression methodology and statistical disclosure control. *J. Am. Stat. Assoc.*, 75(370):377–395, April 1980.

[10] L. Cranor, J. Reagle, and M. Ackerman. Beyond concern: Understanding net users' attitudes about online privacy. Technical Report TR 99.4.3, AT&T Labs–Research, April 1999.

[11] D. Denning. Secure statistical databases with random sample queries. *ACM Transactions on Database Systems*, 5(3):291–315, Sept. 1980.

[12] D. Denning. *Cryptography and Data Security*. Addison-Wesley, 1982.

[13] D. Denning, P. Denning, and M. Schwartz. The tracker: A threat to statistical database security. *ACM Transactions on Database Systems*, 4(1):76–96, March 1979.

[14] D. Dobkin, A. Jones, and R. Lipton. Secure databases: Protection against user influence. *ACM Transactions on Database Systems*, 4(1):97–106, March 1979.

[15] The Economist. *The End of Privacy*, May 1999.

[16] R. Elmasri and S. B. Navathe. *Fundamentals of Database Systems*. Benjamin/Cummings, Redwood City, California, 1989.

[17] European Union. *Directive on Privacy Protection*, October 1998.

[18] I. Fellegi. On the question of statistical confidentiality. *J. Am. Stat. Assoc.*, 67(337):7–18, March 1972.

[19] H. Galhardas, D. Florescu, D. Shasha, E. Simon, and C.-A. Saita. Declarative data cleaning: Language, model, and algorithms. In *Proc. of the Int'l Conf. on Very Large Data Bases (VLDB)*, pages 371–380, 2001.

[20] Y. Gertner, Y. Ishai, E. Kushilevitz, and T. Malkin. Protecting data privacy in private information retrieval schemes. In *ACM Symposium on Theory of Computing*, pages 151–160, 1998.

[21] H. Hacigumus, B. R. Iyer, C. Li, and S. Mehrotra. Executing SQL over encrypted data in the database-service-provider model. In *Proc. of the ACM SIGMOD Conf. on Management of Data*, Madison, Wisconsin, June 2002.

[22] H. Hacigumus, B. R. Iyer, and S. Mehrotra. Providing database as a service. In *Proc. of the Int'l Conf. on Data Engineering*, San Jose, California, March 2002.

[23] S. Jajodia, P. Samarati, M. L. Sapino, and V. S. Subrahmanian. Flexible support for multiple access control policies. *ACM Transactions on Database Systems*, 26(2):214–260, 2001.

[24] S. Jajodia and R. Sandhu. Polyinstantiation integrity in multilevel relations. In *IEEE Symp. on Security and Privacy*, 1990.

[25] S. Jajodia and R. S. Sandhu. A novel decomposition of multilevel relations into single-level relations. In *IEEE Symposium on Security and Privacy*, pages 300–315, 1991.

[26] G. Karjoth, M. Schunter, and M. Waidner. The platform for enterprise privacy practices - privacy-enabled management of customer data. In *2nd Workshop on Privacy Enhancing Technologies (PET 2002)*, San Francisco, CA, April 2002.

[27] S. Katzenbeisser and F. A. Petitcolas, editors. *Information Hiding Techniques for Steganography and Digital Watermarking*. Artech House, 2000.

[28] J. Kaufman, S. Edlund, D. Ford, and C. Powers. The social contract core. In *Proc. of the Eleventh Int'l World Wide Web Conference (WWW)*, Honolulu, Hawaii, May 2002.

[29] J. Kleinberg, C. H. Papadimitriou, and P. Raghavan. On the value of private information. In *Proc. 8th Conf. on Theoretical Aspects of Rationality and Knowledge (TARK)*, 2001.

[30] C. Landwehr. Formal models of computer security. *ACM Computing Surveys*, 13(3):247–278, 1981.

[31] M. Langheinrich, editor. *A P3P Preference Exchange Language 1.0 (APPEL1.0)*. W3C Working Draft, February 2001.

[32] L. Lessig. *Code and Other Laws of Cyberspace*. Basic Books, 1999.

[33] C. K. Liew, U. J. Choi, and C. J. Liew. A data distortion by probability distribution. *ACM Transactions on Database Systems*, 10(3):395–411, 1985.

[34] T. F. Lunt. A Survey of Intrusion Detection Techniques. *Computers & Security*, 12(4):405–418, 1993.

[35] M. Marchiori, editor. *The Platform for Privacy Preferences 1.0 (P3P1.0) Specification*. W3C Proposed Recommendation, January 2002.

[36] S. K. Mishra. On symmetrically private information retrieval. Master's thesis, Indian Statistical Institute, 2000.

[37] Office of the Information and Privacy Commissioner, Ontario. *Data Mining: Staking a Claim on Your Privacy*, January 1998.

[38] R. Oppliger. Internet security: Firewalls and beyond. *Comm. ACM*, 40(5):92–102, May 1997.

[39] Oracle Corporation. *Database Encryption in Oracle8i*, August 2000.

[40] R. Ramakrishnan and J. Gehrke. *Database Management Systems*. McGraw-Hill, 2000.

[41] V. Raman and J. M. Hellerstein. Potter's wheel: An interactive data cleaning system. In *VLDB Journal*, pages 381–390, 2001.

[42] S. P. Reiss. Practical data-swapping: The first steps. *ACM Transactions on Database Systems*, 9(1):20–37, 1984.

[43] M. Rotenberg. *The Privacy Law Sourcebook 2000: United States Law, International Law, and Recent Developments*. Electronic Privacy Information Center, 2000.

[44] M. Rotenberg. Fair information practices and the architecture of privacy. *Stanford Technology Law Review*, 1, 2001.

[45] A. Rubin and D. Greer. A survey of the world wide web security. *IEEE Computer*, 31(9):34–41, Sept. 1998.

[46] B. Schneier. *Applied Cryptography*. John Wiley, second edition, 1996.

[47] A. Shoshani. Statistical databases: Characteristics, problems and some solutions. In *Proc. of the Eighth Int'l Conference on Very Large Databases*, pages 208–213, Mexico City, Mexico, September 1982.

[48] A. Silberschatz, H. F. Korth, and S. Sudarshan. *Database Systems Concepts*. McGraw-Hill, 3rd edition, 1997.

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

[50] P. Stachour and B. Thuraisingham. Design of LDV: A multilevel secure relational database management system. *IEEE Trans. Knowledge and Data Eng.*, 2(2):190–209, 1990.

[51] Time. *The Death of Privacy*, August 1997.

[52] J. Traub, Y. Yemini, and H. Woznaikowski. The statistical security of a statistical database. *ACM Transactions on Database Systems*, 9(4):672–679, Dec. 1984.

[53] J. D. Ullman. *Principles of Database & Knowledge-Base Systems*, volume 1. Computer Science Press, 1988.

[54] J. D. Ullman. *Principles of Database & Knowledge-Base Systems*, volume 2: The New Technologies. Computer Science Press, 1989.

[55] U.S. Department of Health, Education, and Welfare. *Records, computers and the Rights of Citizen: Report of the Secretary's Advisory Committee on Automated Personal Data Systems*, xx-xxiii edition, 1973.

[56] N. R. Wagner. Fingerprinting. In *IEEE Symp. on Security and Privacy*, pages 18–22, Oakland, California, April 1983.

[57] S. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *J. Am. Stat. Assoc.*, 60(309):63–69, March 1965.

[58] A. Westin. E-commerce and privacy: What net users want. Technical report, Louis Harris & Associates, June 1998.

[59] A. Westin. Privacy concerns & consumer choice. Technical report, Louis Harris & Associates, Dec. 1998.

[60] A. Westin. Freebies and privacy: What net users think. Technical report, Opinion Research Corporation, July 1999.

[61] C. Yu and F. Chin. A study on the protection of statistical databases. In *Proc. of the ACM SIGMOD Conf. on Management of Data*, pages 169–181, 1977.