The Private Database

Ashley Green and Raghava Vellanki

Independent Project – Spring 2004

Joan Feigenbaum
Avi Silberschatz

Abstract

The revolution of the Internet and data storage architectures has increased the ability to collect and maintain large amounts of data at high speeds. While these technologies have facilitated major advancements in industries such as medicine, finance, education and retail, the unforeseen consequences are just beginning to surface. Consequently, data subjects are beginning to worry about the protection of the sensitive information they distribute to companies who claim to safeguard their data. In order to address these concerns, solutions are needed that go beyond the scope of policy. Privacy policies need to be both legally adopted and technologically implemented in order to protect data. The purpose of this paper is to discuss current solutions, highlight their weaknesses and discuss possible alternative approaches to the protection of sensitive information at the database level.

Section 1: Introduction

Before the age of the Internet and database architectures, companies manually collected and stored data. In order to record customer information, companies would have to write out the information by hand and physically store it in a catalogue system. Today, the Internet coupled with database structures facilitate the collection and storage process making it possible to collect and store thousands of records in minutes. An added bonus is the ability to quickly retrieve, sort, extract and delete any number of records in a few short steps. In the days of manual data collection the time it would take to sort or query data (i.e. to find the addresses of all customers who live in a certain zip code) would not be worth the required effort. As a result, the motivation to perform tasks that may compromise customers' privacy was not high. Today, on the other hand, these queries can be executed and the results sent to third parties in seconds. The point here is that, the high speed at which information can be processed and transferred has increased carelessness in how this information should be treated, thereby creating climates in which the privacy of sensitive information can be compromised.

We see evidence of this carelessness in the increasing number of cases in which the privacy of sensitive information is violated. As a result, data subjects are becoming

further concerned over entrusting their sensitive information to companies who claim to protect this data. For example, earlier this year Jet Blue, a major commercial airline violated their privacy policy by releasing customer records for a government study[1]. Jet Blue's privacy policy ensures customers that their information will only be distributed to third parties who assist Jet Blue in serving their customers[2]. With the ability to easily access, query and send large quantities of data, technology has created a climate where actions can be performed so quickly the user (employee or administrator with access to the database) may not even think about the privacy implications. In the Jet Blue incident it is quite possible that the employee who released the customer information was simply not aware of the company privacy policy or did not think about the policy at the time. Of course, it's also possible that the employee acted maliciously, but for the purposes of this paper we are not going to address the notion of malicious intent. Instead, the focus of this paper is on a solution, which removes the notion of data privacy from database users' consciousness and charges the database itself with the responsibility of protecting the data. A database user will not have to make choices regarding which actions violate privacy constraints because the database will only present them with approved actions.

The structure of rest of this paper is as follows. Section 2 discusses the characteristics desirable for a privacy-enhanced database. We introduce the model on which this paper will be based in Section 3 and Hippocratic Databases, a current proposed solution, in Section 4. We will discuss problems with Hippocratic Databases in Section 5 and possible alternative approaches in Section 6. Concluding remarks will follow in Section 7.

Section 2: Requirements

In this paper we will introduce an existing proposal for a database architecture, which protects the privacy of sensitive information. We recognize that technology alone cannot solve the issue of privacy on the grand scale. An alliance of technology, legal

[1] "Jet Blue's Privacy Blues." AlterNet.org 21 September 2003. 23 April 2004 http://www.alternet.org/rights/2003/09/001264.html.

[2] Jet Blue Airlines Home Page. 23 April 2004. <http://www.jetblue.com/privacy.html#p2>

policies, social standards, etc. must work together to accomplish this task[3]. But, for the purposes of this paper, we are going to focus on a technological solution with the hopes that it will minimize the role that policies, social standards, etc. must play.

A database engine, which successfully protects the sensitive information it houses, needs to explicitly state the privacy concerns. This database must include the functionality currently available in traditional databases, while also protecting the privacy of data. Although, additional responsibilities are required of the database, the system should not perform more slowly, less efficiently or provide less storage space. In addition, this engine will need capabilities beyond traditional uses of roles and views in order to securely protect data.

Section 3: Our Model

Before introducing a current solution for protecting the privacy of sensitive information at the database level, we first establish a model from which to work. After researching several companies' privacy policies, we selected Gap as our prototype. Gap is one of the world's largest specialty retailers with 4,100 stores and 165,000 employees worldwide. The company joined the Internet community in 1997, when it established it's website: www.gap.com. Recently, the website was enabled with P3P (Platform for Privacy Preferences) in order to allow their customers to compare their personal preferences with Gap's privacy policy. A P3P enabled website is advantageous to both the company and the customer because not only does it allow the company to develop a thorough privacy policy but it also facilitates the customers comparison between their privacy preferences and the policies of the company. Unfortunately, a major drawback with the P3P platform is that it is unable to enforce that companies comply with their privacy policy. For example, if a company states that they will only use a customer's address for the delivery of a product, there is no way the customer can validate that their address is not being exploited for other purposes. P3P is not able to control the use or the flow of data once a company collects it.

[3] Agrawal, Rakesh and Kiernan, Jerry and Srikant, Ramakrishnan and Xu, Yirong. "Hippocratic Databases." IBM Almaden Research Center 2002. page 1.

Gap presents a solid model for our study for two important reasons: the establishment as a worldwide leader in retail thus collecting, storing and maintaining large amounts of consumer sensitive data and the initiative in protecting customer information by employing P3P in addition to developing a thorough privacy policy. They have such a large consumer base it is in their best interest to adhere to their policies so as to not lose their clientele.

3.1 Gap's privacy policy

Due to their worldwide client base, Gap has adopted a comprehensive policy, which addresses two main issues: collection of customer information and third party privileges. Gap's policy discusses the types of information being collected and the general purposes, for their use.

Sensitive information (name, email addresses, mailing address, phone number and credit card number) is stored to process online orders, send customer specific promotions or surveys or enter them into contests. Once a customer registers, he is automatically listed for e-mail promotions and updates. In order to remove himself from e-mail or postal lists he must directly contact Gap's customer service by phone or in writing. The customer's data, along with public demographic information, is used to ameliorate and personalize the customer's shopping experience[4]. The customer has the option of changing or deleting his Personally Identifiable Information (PII) online. While, the policy does mention general purposes for which the data is used, it does not state which specific information will be used for each purpose. In addition, records are maintained and stored for an unspecified amount of time.

The other major aspect of the policy is the role of third parties. Gap contracts out to third parties for assistance in maintaining and managing customer information to fulfill promotions and seamlessly communicate with customers. The policy states that "[Gap does] not authorize any of the third parties to make any other use of [the customer's] information"[5]. This statement may be well intentioned but it's a bold assurance considering Gap does not have the power to regulate or oversee the actions of third

[4] GAP Home Page. 23 April 2004. http://www.gap.com/asp/cs_security.asp
[5] GAP Home Page

parties. Once other approved companies have the ability to cross-reference Gap databases in order to locate common customers, they have the ability to use the information for their own benefit without the customers' approval.

Overall, we see that Gap established a policy to protect the sensitive information their customers' entrust to them. Unfortunately, some aspects of the policy cannot be enforced. As a result, a solution is needed that goes beyond legal policy.

Section 4: Hippocratic Databases

Recently, a technological solution to enforce privacy policies has been presented by the Alden research group at IBM[6]. The proposal is to automate and integrate privacy policies into database architecture. The aim is to remove the responsibility from database users (anyone with access to data contained in the database) and instead make the database accountable for protecting the privacy of the information. It must be noted that a human element (the database administrator) will always be involved, but we are not assuming that this administrator would be maliciously inclined. A solution to developing a system that does not rely on at least one form of human interaction (even if it's just for setup and initialization, as we will see later in this paper) is currently unknown.

The proposal for automating and monitoring privacy policies involves rethinking the current database architecture and instead restructuring them to better protect the data inside. The purpose of this research is to identify the technical challenges in designing privacy-preserving databases with the "hope[s]…that [it] will serve to catalyze a fruitful and exciting direction for future database research"[7]. These databases are founded on the 10 Hippocratic principles (listed below) and are thus named: Hippocratic Databases (HD)[8].

1. purpose specification: purposes for which the information has been collected shall be associated with that information

2. consent: companies shall have consent of the donor of the personal information.

[6] Hippocratic 1
[7] Hippocratic 1
[8] Hippocratic 1

3. limited collection: personal information collected shall be limited to the minimum necessary for accomplishing the task.

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: personal information stored in the database should not be communicated outside the database for purposes other than those for which there is consent.

6. limited retention: personal information shall be retained only as long as necessary for fulfillment of the purposes for which it was collected.

7. accuracy: personal information should be accurate and up to date.

8. safety: personal information is protected by security safeguards against theft and other misappropriations.

9. openness: donor shall not be able to access all of their information stored in the database.

10. compliance: donor shall be able to verify compliance with the above principles.

These founding principles are an extension of the Fair Information Practices, which limit the collection, use and dissemination of personal information[9]. They present a blueprint for how privacy can be preserved in databases.

4.1 The Hippocratic Database Architecture

The Hippocratic database uses metadata to design an automated model for privacy policies. The central concept of this design relies on the *purpose* of each piece of data (see Figure 1). To facilitate the process the Privacy Metadata Schema defines a purpose for each data item (attribute) collected in every table[10]. The *privacy-policies* table captures the privacy policy of the company by including the external recipients and the data subject specified retention period for each attribute. The *privacy-authorizations* table expresses the controls of the data by storing all of the authorized users for each attribute. By storing a purpose for each attribute and each record, database users cannot access the

[9] Hippocratic 3
[10] Hippocratic 3

attributes, which they are not permitted to access. In addition to the purposes stored for each attribute, a purpose is also assigned to each record to reflect the privacy preferences of each individual customer. For example, some customers may only want their data to be used for purchasing, while others may want purchasing recommendations and promotions. A customer can opt in or out of these actions and they will be reflected in the database. It's important to note that this assumes that attributes are fixed and external recipients are known ahead of time.

Privacy Metadata Schema
privacy-policies (purpose, table, attribute, { external-recipients}, retention)
privacy-authorizations (purpose, table, attribute, { authorized-users })

Figure 1: Privacy Metadata Schema

Figures 2 and 3 show a simplified prototype of Gap's policies expressed in a Hippocratic database. This design only includes the purchase and shipping functionality for Gap and its customers. For simplicity, we chose to ignore the promotions and other extra features Gap may offer, which would be reflected in the database.

Database Schema
personal-info (purpose, customer-id, name, age, gender, e-mail, phone, password)
address (purpose, addr-id, nickname, street-name, city, state, zip)
cc-info (purpose, card-id, cc-type, account-number, exp-date)
item (purpose, item-id, price, item-info)
order-info (purpose, order-id, order-number, tracking-number, status)
order (purpose, customer-id, order-id, data)
payment (purpose, customer-id, card-id)
resides (purpose, customer-id, addr-id, shipping, billing)
order-item (purpose, order-id, item-id, quantity)

Figure 2: Database Schema

The process by which Gap goes about implementing their privacy policies begins with the database administrator. The database administrator is responsible for using the Privacy Metadata Creator to automatically generate the privacy metadata tables that specify both the privacy policies and authorized users. Once the initial setup is complete, the database is ready to begin collecting data. It's important to note that once the database

administrator generates the metadata tables, interaction with the architectural side of the database is minimal from that point on.

Privacy Policies Table:

Purpose	Table	Attributes	External-recipients	retention
Promotions	personal-info	name	third-party	3 months
Promotions	personal-info	age	third-party	3 months
Promotions	personal-info	gender	third-party	3 months
Promotions	personal-info	e-mail	third-party	3 months
Purchase	personal-info	name	delivery, cc-company	1 month
Purchase	Address	street	delivery company	1 month
Purchase	Address	city, state, zip	delivery company	1 month
Purchase	cc-info	account-number	cc-company	1 month
Purchase	cc-info	cc-type,	cc-company	1 month
Purchase	order-info	tracking-number	delivery company	1 month
Purchase	order-info	status	empty	1 month
Registration	personal-info	name	empty	5 years
Registration	Address	street, city, state, zip	empty	5 years
Registration	personal-info	e-mail	empty	5 years
purchase-circles	Item	item-info	all	3 year

Privacy Policies Table:

Purpose	Table	Attribute	Authorized-users
Promotions	personal-info	customer-id	all
Promotions	order-item	item-id	mining
Promotions	order-item	order-id	mining
Purchase	personal-info	customer-id	all
Purchase	personal-info	Name	{shipping, charge, customer-service}
Purchase	personal-info	e-mail	{shipping, customer-service}
Purchase	personal-info	Phone	{shipping, customer-service}
Purchase	Address	All	{shipping}
Purchase	cc-info	type, account#, exp-date	{charge}
Purchase	cc-info	card-id	all
Purchase	order-info	order-id	all
Purchase	order-info	tracking#, status	{shipping, customer-service}
Registration	personal-info	customer-id	all
Registration	personal-info	name, e-mail, phone	{registration, customer-service}
Registration	Address	addr-id	all
Registration	Address	street, state, zip	registration
Registration	cc-info	card-id	all
Registration	cc-info	cc-type, account-num, exp-date	registration
purchase-circles	personal-info	customer-id, age, gender	olap
purchase-circles	Address	city, state, zip	olap
purchase-circles	Item	item-id, item-info	olap

Figure 3: Privacy Policies and Authorizations Tables

4.2 Data Collection

When a customer first visits Gap's website, the *Privacy Constraint Validator* checks whether the customer's privacy preferences match Gap's privacy policy, much like P3P. For example, if a customer wants to opt out of giving his Private Identifying Information (PII) for every purpose except purchase, the validator will check if this preference is compliant with the company's privacy policy. Once the customer's preferences and Gap's privacy policy correspond, the customer may proceed to register by inserting data into the online forms. For each set of information collected (record), the customer selects approved *purposes*, which will correspond to all the attributes contained within that record. The approved *purposes* for each record along with the information in the privacy-authorizations table will restrict access to those attributes. As described earlier, the database stores the *purposes* for each record in every table (peronsal_info, cc_info) according to the customer, but the database has to decide on the *purpose* level of each relation table (i.e. payment). The relation table stores the purpose with a lower level of "privacy". For example, if the personal-info table had a record with a *purpose* "promotions" and the corresponding record had a *purpose* "purchase" in cc_info, the joing table (payment) will have the purpose of "purchase". This would be the case because "purchase" data is only stored until the purchase is completed while promotions data may be kept for years. We would say that promotion data is less "private" compared to purchase data.

purpose	customer-id	name	age	gender	e-mail	password	phone
registration	1	Bob Vellanki	22	M	bob@yale.edu	hello	203 111 2222

Personal-info Table

purpose	addr-id	street	city	state	zip
purchase	1	8 sunset	New Haven	CT	67890

Address Table

purpose	addr-id	customer-id	shipping	billing
purchase	1	1	yes	yes

Resides Table

4.3. Queries

Once the data is stored in the database, approved users via queries may retrieve it. Each query is tagged with its intended purpose. Figure 4 illustrates a simple query initiated by Customer Service that checks the *status* of a purchase with an *order-number* of 12345. Each query goes through three steps in order to successfully execute. Before execution, the database checks the *privacy-authorizations* table to see if the user who issued the query is authorized to access all the attributes within the query. In addition, the *Attribute Access Control* analyzes the query to check if all the accessed attributes are explicitly listed in the privacy-authorizations table with the corresponding query purpose tag.

User: Customer Service
Query Purpose Tag: purchase

Select status
From order-info as oi, personal-info as p, order as o
Where email='bob@yale.edu' and order-number=12345 and p.customer-id=o.customer-id and o.order-id=oi.order-id

Figure 4: Query

Using the query in Figure 4, the database checks to see if customer service is allowed to access the following attributes: personal-info.email, order.order.order-number, customer-id, order-info.order-id and order.order-id. A similar check is done with the query purpose tag. If any of these attributes are not listed in the privacy-authorizations table, the query is rejected.

Next, once the Attribute Access Control validates all the attributes, the query moves into execution. During the execution, the *Record Access Control* "ensures that only records whose purpose attribute includes the query's purpose will be visible to the

query”[11]. This ensures that only the records with the purpose that matches the query tag will be visible to the user. This idea is similar to multilevel relations in multilevel secure databases.[12] In our Gap example query, the *Record Access Control* only shows the records where the *purpose* is “purchase” or a less “private” purpose.

Finally, after the execution of the query, the *Query Intrusion Detector* is run on the query results to analyze the access pattern of the results. If the pattern is different from past access patterns, the query or the results may be tagged with a security check. The intention of this detection is to restrict users from accessing data for malicious purposes. In our example, Customer Service may try to access the e-mail addresses of all customers, but their purpose only allows them to query the ones tied to purchase orders.

Section 5: Hippocratic Database Limitations

Hippocratic databases represent a good first step towards finding a technical solution for protecting data. Unfortunately, this solution has three major limitations: 1) it cannot guarantee all privacy policies 2) it has reduced functionality 3) it has decreased performance. As stated earlier, an effective solution explicitly states the privacy concerns and also includes the functionality of databases currently available.

While Hippocratic databases can implement a significant portion of possible privacy policies, they can’t express them completely. As we’ve seen with Gap’s privacy policy, many companies promise that approved third parties, with which certain customer information is shared, will not pass on this data to any other groups. We see in our Hippocratic implementation of Gap privacy policies that subsets of user data can only be accessed if the user-purpose combination is approved by the system. These policies are effective for many transactions, but they do nothing to prevent the approved third parties who can access approved data from sending the information on to others. One of the major issues of privacy policies is the issue of third party responsibility. How can the use of information be controlled once it has been accessed either intra-company or by third

[11] Hippocratic 7
[12] Hippocratic 7

party vendors? Unfortunately, this represents a significant problem that does not have a solution in Hippocratic databases as they are currently implemented.

The next limitation is the reduction in functionality. The IBM Alden research group points out several functionality limitations. These include the following issues: if data is deleted after a time specified by the user, how do we delete if from the logs without affected recovery? How do we ensure that the data subject providing their private information is, in fact, who they claim to be? How do we ensure that the minimum number of required attributes is actually being collected?[13] All of these issues are important to research and address.

An additional limitation not discussed in the IBM research is arbitrary querying. When privacy policies are explicitly expressed in the database they restrict arbitrary queries, which are the fundamental root of traditional databases. Hippocratic databases restrict users from performing arbitrary queries on the data. For example, if the president of a company wanted to know

(a) how many new customers were registered in 2003

(b) the number of customers who used credit cards that expire in May 2004 for orders placed in May 2004

or

(c) the name and address of the customer who spent the most money at GAP during 2003

these queries could not execute because they cannot be classified in one of the existing purposes (purchase, registration, recommendations or purchase-circles). They are subjective queries that cannot necessarily be predicted ahead of time, the time when the privacy meta-data tables are created for the database. A possible solution to this problem could be the creation of another purpose classification termed *mining*. All customer data would be accessible for this purpose, but select users could only access the data through a higher-level application. Such an application would provide an interface with which users

[13] Hippocratic 9

could derive specific sets of information. Access to this application, would be administered by the database administrator and ideally, the subset of approved users would be relatively small (presidents, CEOs, Vice Presidents, corporate attorneys).

A potential problem with this solution is the scenario in which a database administrator accidentally gives unapproved parties access to the database. Remember, we are not assuming malicious intent in this paper.

The final problem with Hippocratic databases is the various performance and space limitations. Hippocratic databases require added steps to verify purposes and approved users for each transaction. Traditional databases were designed to be quick, efficient and store massive amounts of data[14]. The structure of Hippocratic databases could potentially reduce the speed and efficiency because these checks must be performed for every single transaction[15]. In addition to speed and efficiency, storage space is also a concern. The meta-data tables used to define privacy policies occupy bytes that could be used to store additional user data. Thus, Hippocratic databases offer less data storage capacity than traditional database engines.

Section 6: Future Considerations

Although Hippocratic databases do not provide an ideal solution, they do play an important role by challenging the theoretical boundaries for the design of database architecture. It seems that all of the problems that arise in traditional and Hippocratic database stem from our current general approach with respect to privacy. The challenge is to determine if data privacy and current database architecture can co-exist. If so, then perhaps we need to approach the design of the database engine in a different way. If they cannot seamlessly co-exist, then the solution to data privacy may not reside at the database level. Perhaps the solution lies at the time of data collection, instead of data storage. Much research is needed to determine where the best solution exists, but this research should not be limited to web commerce.

[14] Hippocratic 8
[15] Hippocratic 8

Section 7: Conclusion

Protecting the privacy of personal information is a complicated task. While technology has produced multiple tools to facilitate the collection and storage of information, it has simultaneously created an environment in which sensitive information is becoming increasingly difficult to protect. Current research projects have explored the protection of data at the database level. Unfortunately, these proposals are unable to fully guarantee the data's safety and offer the functionality of a traditional database. It is possible that the architecture of the database must be redesigned in order to accomplish this. On the other hand, the solution to data protection might reside at a different level. Either way, the solution to data protection needs to encompass traditional features, such as arbitrary querying, in addition to those that safeguard data.

Works Cited

Agrawal, Rakesh and Kiernan, Jerry and Srikant, Ramakrishnan and Xu, Yirong. "Hippocratic Databases." IBM Almaden Research Center 2002

GAP Home Page. 23 April 2004. http://www.gap.com/asp/cs_security.asp

Jet Blue Airlines Home Page. 23 April 2004. <http://www.jetblue.com/privacy.html#p2>

"Jet Blue's Privacy Blues." AlterNet.org 21 September 2003. 23 April 2004 http://www.alternet.org/rights/2003/09/001264.html.