Vision Paper: Enabling Privacy for the Paranoids

G. Aggarwal, M. Bawa, P. Ganesan, H. Garcia-Molina, K. Kenthapadi,
N. Mishra, R. Motwani, U. Srivastava, D. Thomas, J. Widom, Y. Xu
Stanford University
Stanford, CA 94305
{*gagan,bawa,pganesan,hector,kngk,nmishra,rajeev,usriv,dilys,widom,xuying*}@*cs.stanford.edu*

Abstract

P3P [23, 24] is a set of standards that allow corporations to declare their privacy policies. Hippocratic Databases [6] have been proposed to implement such policies within a corporation's datastore. From an end-user individual's point of view, both of these rest on an uncomfortable philosophy of trusting corporations to protect his/her privacy. Recent history chronicles several episodes when such trust has been willingly or accidentally violated by corporations facing bankruptcy courts, civil subpoenas or lucrative mergers. We contend that data management solutions for information privacy must restore controls in the individual's hands. We suggest that enabling such control will require a radical re-think on modeling, release, and management of personal data.

1 Introduction

Information Privacy is concerned with imposing limits on collection and handling of personal information such as credit and medical records by state and private organizations. These are early days for Information Privacy, and norms and laws that impose restrictions on the use of personal information collected by organizations are being worked out as a solution. Technology is being devised to assist the implementation of such laws.

Status The Platform for Privacy Preferences (P3P) [23, 24] is a set of standards that allow organizations to declare their privacy policies. Recently, Hippocratic Databases [6] were envisioned to provide support for an organization's privacy policies within the organization's datastore. In this framework, an organization would post its privacy policies, using agreed-upon language, and an individual would only conduct business with that organization if the published policies were consistent with the individual's expectations.

Example Consider an individual, Alice, who wants to sign up for a DealsRus service on the web. DealsRus requires Alice's email address to inform her of upcoming deals. DealsRus recognizes the privacy concerns of its clients and has placed its P3P policies on their web-site. Alice is privacy-savvy and is using a browser which is P3P enabled. Alice's browser would fetch the P3P policy from the DealsRus web-site. For instance, DealsRus may state that email addresses will only be used for current purpose ("completion and support of the recurring subscription activity") and the recipients of such data will be restricted to ours ("DealsRus and/or entities acting as their agents or entities for whom DealsRus are acting as an agent") but not unrelated third parties. If Alice is happy with this policy, then she can give DealsRus her email address.

Critique With the P3P framework, thus, Alice has to trust that (a) the organization has clearly stated its policies, that (b) the organization will actually adhere to the policies, and that (c) the organization has the means to implement the policies in transit and storage. All three aspects raise troubling issues: Even though DealsRus has used legal language vetted by P3P, the end user may feel inundated with legalese whose exact practicality is open to interpretation [30, 35, 38, 46]. For instance, what exactly does current purpose mean? Perhaps it is within the ambit of current purpose for DealsRus to spam their customer's mailboxes? And what does it mean not to give email addresses to third parties but restrict recipients to ours? Perhaps DealsRus has many wholly-owned subsidiaries which can use the addresses? Does DealsRus provide adequate protection for personal data to prevent easy access to data by intruders? And what would happen if DealsRus declares bankruptcy, or changes management, or changes it policies, or its records are subpoenaed by a court?

Acknowledgements We have bene£ted from the many pertinent and constructive comments of our anonymous VLDB reviewers and Dennis Shasha. This research was supported in part by NSF Grants ITR-0331640 and EIA-0137761, and a grant from SNRC.

**Proceedings of the 30th VLDB Conference,
Toronto, Canada, 2004**

Inherency P3P and Hippocratic Databases put the onus of safeguarding privacy in the hands of organizations that are often themselves guilty of trespass or sloppiness. Indeed, recent history chronicles several episodes where organizations have violated, either deliberately or accidentally, their customer's trust when they faced mergers, bankruptcy, courts or hackers [1, 28, 33, 42]. Even if the underlying datastore at DealsRus follows Hippocratic principles, if the rules the datastore is told to follow by DealsRus management are not "ethical" (as far as Alice is concerned), then the Hippocratic guarantees will be of little use to Alice.

Thesis We contend that there is a better way to approach privacy, and that is to enable individuals to retain "control" over their information. At all times, the individual should be able to "choose freely under what circumstances and to what extent they will expose themselves, their attitudes, and their behavior, to others" [54]. The desired "level of control" may vary: for instance, in some cases, the individual may only want that misuse of her information be auditable. In other cases, she may want to prevent access to information to certain organizations or for particular tasks. In any case, however, it should be the individual who pro-actively decides what the level of control is.

We believe that a case can also be made to organizations to leave control in the hands of individuals. In particular, governments are passing new legislation (e.g., California law SB1386) that forces organizations to inform individuals whenever there has been a privacy breach, and makes organizations liable for improper use of information. Given the high overhead of securing data, and potentially high liability costs, organizations could be persuaded to leave control to owners of the information.

Plan How can an individual retain control at the appropriate level given that organizations must have access to her information to conduct business? We contend that control can be retained if the individual can release information to organizations such that *the released information is unusable for illegitimate tasks*. In this paper, we illustrate through examples a series of scenarios where such control can be enabled (Section 2). We claim that these examples are representative of a small set of "information types", and that for each such type, one can devise a general purpose set of mechanisms to retain control (Section 3). We then propose to gather the set of mechanisms that cover all information types in one framework which we call *P4P: Paranoid Platform for Privacy Preferences* (Section 4).

We call this framework "paranoid" because individuals that use it are less trusting of organizations than individuals who use the P3P framework. We caution that our framework is still in its formative stages, and many of our concepts are still not well-de£ned. Thus, at this point, we are only presenting the *vision* of our framework, with the hopes that others in the community will help us re£ne it, formalize it, and debug it.

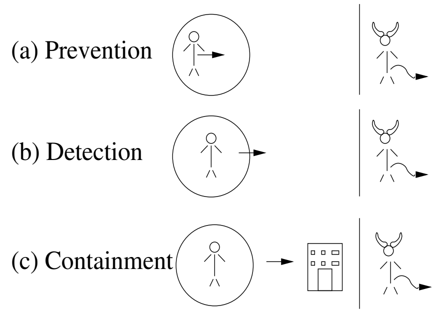

Figure 1: *Privacy has traditionally been assured against adversaries by (a) preventing illegitimate access to personal information or (b) detecting and curtailing release of personal information. In contrast, (c) the P4P framework seeks to contain illegitimate use of personal information that has already been released to an external (possibly adversarial) entity.*

The P4P Framework Figure 1 informally contrasts our P4P framework against prior work on privacy/security of data. We comment on additional related work in Section 5.

Figure 1 depicts an individual, Alice (on the left), who seeks to guard her personal information (the arrow in the £gure) from an adversary, shown on the right. A perimeter around Alice represents her personal space over which she can exercise direct control. Generally speaking, there are three ways in which Alice can protect her information from the adversary, illustrated as Figures 1(a), (b) and (c).

Traditionally, Alice secures her information by placing access-control restrictions to *prevent* the adversary from accessing her information (Figure 1). In addition, Alice can place safeguards at her perimeter (e.g., £rewalls, query auditing, etc.) to *detect and curtail* release of her information to the adversary (Figure 1(b)). Again, notice that such safeguards can be successfully imposed only while Alice's information resides within her perimeter of direct control *and* requests for information are directed to Alice.

In her daily interactions, Alice has to release her personal information to organizations (e.g., to a bank for obtaining a loan, to an online merchant for a book purchase, etc.). Once her information leaves her perimeter, Alice has no control over its subsequent use. The organization that receives her information may, either deliberately or accidentally, release Alice's information to the adversary, leading to a breach of Alice's privacy. Unfortunately, prevention and/or detection mechanisms discussed above are of little help to Alice now.

The P4P framework (Figure 1(c)) seeks to enable Alice to retain control over her personal information *even after its release to an organization*. Using the P4P framework, Alice can *contain* illegitimate use of remote copies of her information. We note that the P4P framework *complements*, and does not replace, the prevention and detection mechanisms for data privacy. As we investigate in this paper, enabling such control of remote copies requires a radical re-think on modeling and release of information.

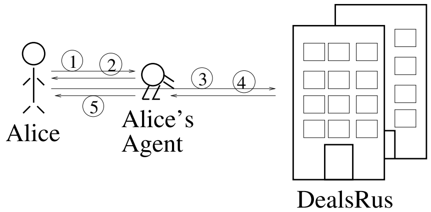

Figure 2: *Alice interacts with DealsRus through her agent. Alice (1) requests and (2) obtains a temporary email address from her agent, which is (3) released to DealsRus. Henceforth, DealsRus can only (4) send messages to the agent at the released email address which are (5) forwarded to Alice after checking for speci£ed restrictions.*

2 Retaining Control – Initial Examples

We claim that it is possible for individuals to retain control over remote copies of their personal information by exploiting the *semantics* of the information. Let us illustrate what we mean using a couple of examples: email addresses and credit card numbers. We note that all of the techniques we will exploit in these two examples are well-known; indeed, solutions based on these techniques have already been deployed today. Our goal here is to illustrate the feasibility of individual control, leading the reader towards Section 3 where general mechanisms are synthesized out of the well-known individual techniques.

2.1 Email Address

To retain control over her email address, Alice constructs a trusted software *agent* that manages her email address. (See Figure 2.) Only the agent is given Alice's true email address, say aly@aliceHost. When Alice wishes to give her email address to DealsRus, the agent generates a new address, say aly1@agentHost, where agentHost is the computer where the agent runs. When DealsRus receives aly1@agentHost (either from Alice or from the agent), it uses that address for communication with Alice. That is, an email to aly1@agentHost is received by the agent, and will be forwarded to aly@aliceHost depending on what restrictions Alice speci£ed when the email was created. These restrictions give Alice some control over how aly1@agentHost can be used. For example:

1 *Timeout:* The email aly1@agentHost is only valid for a period of time. After that time, the agent will refuse to forward messages to Alice.
2 *Limited Use:* The agent will only forward some maximum number of messages.
3 *Restricted Source:* The agent will only forward a message if it comes from a pre-speci£ed source.
4 *Invalidation:* Alice can at any time explicitly instruct the agent to stop forwarding messages.
5 *Isolation:* The email aly1@agentHost is only released to DealsRus. Other organizations will get different email addresses.

How does Alice gain by trusting an agent? Alice now has to trust only *one* entity, as opposed to *every* organization that receives her address under the P3P framework. Furthermore, if the agent code is public and run on a trusted platform [41], Alice (or her auditor) can know precisely what actions the trusted agent will take under different circumstances. This operational description of the agent's "privacy policy" will be much more precise than any legal/natural-language description that an organization provides under P3P. Furthermore, organizations do not have to change their procedures; they handle email just as they did before.

The semantics of email addresses enable Alice to specify restrictions that provide a very useful level of control for Alice. In particular:

A DealsRus can distribute copies of aly1@agentHost to other organizations, but a restricted-source limitation can prevent the other organizations from getting through to Alice.
B If DealsRus gives aly1@agentHost to other organizations, the agent will have proof of this action, since the address aly1@agentHost was only given to DealsRus.
C If the intended use of the address spans a limited time or number of interactions (e.g., perhaps Alice is simply trying to £nd deals that are available this week), then by implementing a timeout or a limited-use restriction Alice can ensure her address is not used for other tasks later in time or for tasks that require more interactions.
D If DealsRus wishes to use Alice's address in some new way, it is likely that it will have to request permission from Alice (or her agent), as aly1@agentHost is likely to be invalid.
E If Alice uses different email addresses for each organization she interacts with, DealsRus will be unable to use Alice's address as an integration key to obtain more information about her from third-party organizations.

Adoption Notice that the agent can be implemented in a variety of ways. The agent could be part of Alice's desktop email software [26], or it could be a trusted third party that provides the temporary email address generation and email forwarding service. Indeed, www.mailshell.com and www.spamgourmet.com provide such a facility today enabling Alice to specify *Timeout* and *Limited Use* restrictions respectively.

2.2 Credit Card Number

The above ideas can be extended to the handling of credit card numbers as well. An agent can ensure control of an individual, say Bob, over the use of his credit card. However, since credit is extended to Bob by his bank, we need to place the agent either at Bob's bank or between the bank and the organizations that use Bob's credit card number. If the agent is not at the bank, it would have to appear to the organizations as a bank that can handle charges, which may be dif£cult to achieve. Thus, let us assume that the bank plays the role of trusted agent for Bob.

The interaction between Bob and an organization, say ShopsRus, is analogous to that between Alice and DealsRus. Neither Bob nor his agent gives out Bob's true credit card number. Instead, ShopsRus receives a unique, temporary credit card number, which we will call a *pseudonum*. The agent manages the mapping between this pseudonum and Bob's credit card number. Bob can instruct his agent to restrict the use of pseudonums in a variety of ways:

1. *Timeout:* The pseudonum is only valid for a £xed period of time, or for a £xed number of charges;
2. *Limited Use:* The pseudonum can only be charged upto a £xed amount;
3. *Restricted Source:* The pseudonum can only be charged from speci£c sites, or for speci£c types of purchases;
4. *Invalidation:* Bob can at any time explicitly instruct his bank to stop honoring charges on the pseudonum.
5. *Isolation:* A unique pseudonum is released to ShopsRus. Other organizations get different pseudonums.

As with his email, Bob retains some level of control as to how his credit card is used.

- A ShopsRus can distribute copies of the pseudonum to other organizations, but a restricted-source limitation prevents other organizations from charging Bob's credit.
- B If ShopsRus gives the pseudonum to other organizations, the agent has proof of this action, since the pseudonum was only given to ShopsRus.
- C If the intended use of the address spans a limited time or credit, then with an appropriate restriction Bob ensures that his credit is not used for other tasks later.
- D If ShopsRus wishes to charge Bob's credit for a new deal, it is likely that it will have to request permission from Bob (or his agent), as the original pseudonum is likely to be invalid.
- E If Bob uses different pseudonums for each organization he interacts with, ShopsRus will be unable to use Bob's credit card number as an integration key to obtain more information about him from third-party organizations.

Adoption Some credit card companies have begun offering a subset of the above functionalities (e.g., one-time use credit card numbers [8, 31]). The technology was hailed as a "landmark event by the industry" and promptly adopted by online merchants who have to bear the brunt of credit-card fraud, unlike of¤ine merchants in which case the liability is assumed by the bank that issued the card. For example, the travel-site www.expedia.com recorded a £scal third-quarter charge of 6 million US dollars in 2000 to cover the cost of fraudulent transactions! The above anecdotes suggest that organizations will indeed be willing to leave control in the hands of individuals if appropriate technology is devised.

3 Retaining Control – Generalizing to "Information Types"

Can we generalize these concepts, so that an individual can retain control over other "types" of information? It turns out that email addresses and credit cards are the easiest to control as they represent a "service handle" for a work¤ow path that terminates at the individual. An agent can be easily placed in this path to provide limited control on how the service is invoked. Indeed, this is why we already have deployments that exercise control in ways described earlier.

In this section, we consider four types of personal information that are ubiquitous today: (a) Local Identi£ers, (b) Foreign-Key Identi£ers, (c) Value Predicates, and (d) Multi-Source Value Predicates. We will see that it will be harder to retain the same degree of control as with service handles, and organizations may have to dramatically change the way they handle personal information. Nevertheless, we are cautiously optimistic that a collection of techniques can be devised that may lead to the synthesis of a general framework.

3.1 Local Identi£ers

In many cases, organizations demand from its users identi£cation numbers like social security numbers (SSN) in the United States, or national identi£cation numbers in other countries. For instance, the £rst thing that many mail-order or on-line stores ask for is a telephone number, since that is how they locate their customer's records. In many cases, these numbers are *only* used as keys in the local database.

Simple Protocol It is easy for an agent to hide the true identity of an individual, say Carol. The agent generates a unique, private identi£er for Carol, which could be for example, a random 256 bit string. The organization, DealsRus receives this private identi£er, and uses it as a primary key for Carol. The agent of course remembers all of Carol's identi£ers, so whenever Carol needs to contact DealsRus, the proper identi£er can be issued. And as in our previous examples, Carol retains control over her identity: DealsRus does not know Carol's true phone number or SSN, so any abuses of the private identi£er are limited in scope.

Challenges There are a few practical issues that must be considered for such indirect identi£ers to work with today's deployed systems. DealsRus may only be willing to accept identi£ers that look like SSNs or phone numbers. In such a case, Carol's agent must map the random identi£er into one that conforms to what DealsRus expects.

There is a chance that some other individual will generate the same private identi£er as Carol. Organizations may already have procedures in place for duplicate identi£ers. For example, two family members who share a phone may be buying goods at the same mail-order store. Even SSNs are known not to be really unique identi£ers, and con¤icts do happen. The bottom line is that organizations need to be prepared to deal with duplicate user-generated identi£ers, and should have a protocol in place to ask the user for a different one. Such protocols are already common at websites where users select their ID: if the ID is taken, the sites prompts the user for a different ID.

A potential scheme to generate good identi£ers is the following. Carol can provide her agent with a particular (unique and secret) data item, say her SSN. The agent will then generate all identi£ers for Carol based on this data

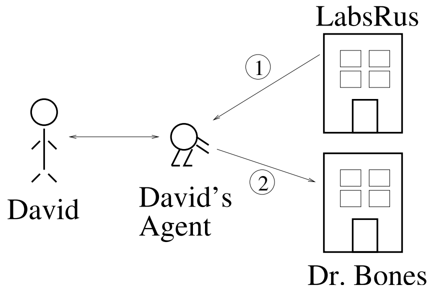

Figure 3: *LabsRus interacts with Dr. Bones through David's agent. (1) LabsRus sends David's reports to his agent, which then (2) forwards the reports to Dr. Bones.*

item and the organization's name, e.g., by using a one-way hash function like SHA-1 [22]. The identi£ers that are generated can be shown to have good cryptographic properties: uniqueness, independence and non-invertibility.

Real identi£ers such as phone numbers have the advantage that they are more readily remembered by users. Thus, when Carol personally phones the DealsRus help line, it will be a lot easier for her to remember her phone number rather than the randomly generated identi£er. Or perhaps Carol has allowed her phone to automatically provide her number to the callee (a feature known as caller-id in the United States), in which case DealsRus will immediately know who is calling. This example illustrates a classic privacy-convenience trade-off which cannot be avoided: If Carol wants privacy, then she is better off giving out agent-generated identi£ers, and always relying on the agent for interactions with outside entities. If Carol wants convenience, then she can give out her personal identi£ers, and hope that organizations do not do anything bad with them.

Summary Local identi£ers can be handled by enabling an individual E to provide organizations $O_1, O_2, \ldots, O_n$ with respective identi£ers $i_1, i_2, \ldots, i_n$ that are:

1 *Unique:* no other individual will have the same respective identi£ers,
2 *Opaque:* an adversary cannot use the identi£er to discover a private attribute of E,
3 *Private:* an adversary cannot discover E's identi£ers,
4 *Independent:* identi£ers i_j and i_k at organizations O_j and O_k have a small probability of belonging to the same individual.

3.2 Foreign-Key Identi£ers

In some cases, an individual's identi£ers are used for purposes other than just as local identi£ers. The organization may need to use them as foreign-keys to allow a legitimate (individual approved) integration or retrieval of records from other organizations.

To illustrate, say David is a patient at Dr. Bones' clinic, and had some tests done at LabsRus. (See Figure 3.) The patient information system that Dr. Bones uses identi£es David by i_1, a local identi£er generated by David's agent. Similarly, LabsRus identi£es David by i_2, a different local identi£er. When David gets a blood test at LabsRus, he requests that the results be sent to Dr. Bones' clinic. Thus, LabsRus needs a way to send records for i_2 that are received as records for i_1 at Dr. Bones' clinic.

Naive Protocol LabsRus asks David's agent for David's identity with Dr. Bones. David's agent provides LabsRus with i_1 after which LabsRus can communicate directly with Dr. Bones to integrate David's records. Although this scheme provides the opt-in property, the privacy and opt-out properties are lost. LabsRus now knows that David is being treated at Dr. Bones' clinic; Dr. Bones now learns that David has his tests done at LabsRus. LabsRus now knows David's identity with Dr. Bones, and David would not know of any future sharing of information between LabsRus and Dr. Bones. David has hence lost control of his personal information.

Simple Protocol LabsRus can route the blood test results to Dr. Bones through David's agent. One way to do this is as follows: David instructs his agent to anticipate blood test results from LabsRus that are to be routed to Dr. Bones. When LabsRus has the results, it sends a message to David's agent which includes:

1 David's local identi£er i_2;
2 David's blood test results;
3 A signature that can be used to prove that only LabsRus could have generated the given test results;

At this point, David's agent removes the i_2 identi£er and the signature of LabsRus from the message. The agent logs the signature as proof of the authenticity of the report, if needed later. The agent then adds the i_1 identi£er and forwards the results to Dr. Bones.

Notice that in this scheme, LabsRus remains unaware of David's doctor who receives the reports. Dr. Bones is unaware of the place where the tests were performed. Thus, David again retains some control over his information. Anytime an organization wants to contact another organization to share David's information, it must go through David's agent.

Challenges In this scenario, it is clear that organizations will have to change the way they operate. That is, they need to be aware that following foreign keys needs to be done though agents, and not directly as they do today. We also observe that the above is still a sketch and a rigorous protocol needs to be de£ned that will allow such foreign-key mappings to be used via a trusted agent without leaking any personal information. For example, how can David's agent be assured that LabsRus is not hiding information within the test reports that reveals David's and its identity? How can Dr. Bones be assured that the test reports are from a valid laboratory?

Summary Foreign-key identi£ers can be handled by enabling an individual E to allow organizations O_1 and O_2 to integrate E's records such that:

1 *Opt-in:* the integration cannot occur without the explicit approval of E,

2 *Opaque:* after integration, an adversarial organization (O_1, O_2 or both) cannot discover a private attribute of E that is not in the records at O_1 and O_2, and
3 *Opt-out:* when the explicit approval of E is withdrawn, the integration cannot occur unless the integrated record has enough information to identify E.

3.3 Value Predicates

In our next example, say Ellen is purchasing a cruise from ShipsRus, and the site asks her for her age. Let us assume that ShipsRus has a legitimate need for Ellen's age y. Perhaps ShipsRus offers a senior citizen discount to individuals with an age over 60 years who take the cruise. We can model this situation by saying that ShipsRus needs to compute a predicate $p := (y > 60)?\ \mathsf{true} : \mathsf{false}$. Thus, ShipsRus does not need to know y itself but the value $p(y)$. How can ShipsRus obtain $p(y)$ while Ellen retains control over her age attribute y?

Naive Protocol We can proceed as follows. ShipsRus sends the predicate p to Ellen's agent. (The query will run in a sandbox-ed environment so that it does not have undesired side effects.) Ellen gives y to her agent, which then computes $p(y)$ and sends the result to ShipsRus. In this way, Ellen retains control over her age information, and only gives ShipsRus the *minimal legitimate* information ($p(y)$) it needs to have to provide service to Ellen.

There are, of course, two shortcomings in the naive protocol as stated. First, the organization can "cheat" by using predicates that are easy to invert. For example, ShipsRus may give the trusted agent a series of predicates that serve to identify Ellen's age y uniquely (e.g., $p_1 : (y == 58)?\ \mathsf{true} : \mathsf{false}$, $p_2 : (y == 59)?\ \mathsf{true} : \mathsf{false}$, etc.) The only way to avoid this problem is to have DealsRus disclose the nature of the predicates by making the source SQL code visible to scrutiny. The source SQL code may be checked for privacy breaches in one of two ways. (1) Ellen and her agent can understand the nature of the information that is being given to ShipsRus. For example, if p is the predicate given earlier that checks if age is greater than 60, Ellen will know that she is disclosing the fact that she is or is not a senior citizen to ShipsRus. (2) Query restriction algorithms may be used to prevent privacy breaches by auditing trails of predicates evaluated.

Second, Ellen may cheat and not give her true age to her agent. Of course, cheating may have later repercussions for Ellen. For example, she may run into trouble when she shows up for the cruise with a senior discount ticket and looking like a teenager! We note that Ellen could cheat by giving a false age even if ShipsRus were to ask Ellen for her age directly.

Notary Protocol Is there anything that could be done to prevent cheating by Ellen? For instance, say ShipsRus does not trust Ellen, but does trust some other organization (e.g., Dr. Bones) that can act as a *notary* and vouch for Ellen's age. Can we enable Ellen to compute a ShipsRus predicate, whose result is vouched for by Dr. Bones?

We present a weak version of the notary protocol that requires Ellen to trust Dr. Bones not to divulge her information. With this protocol, the P3P guarantees are the best we can hope for. Let us say that Ellen discloses her age to Dr. Bones, e.g., by having a medical examination or by showing her birth certi£cate. There is no way Bones will vouch for Ellen's age without knowing the age, so we cannot avoid disclosing the information.

Ellen's agent must also disclose the mapping between the identi£er Ellen used at DealsRus, i_1 and the identi£er Ellen uses with Dr. Bones, i_2. Without the $i_1 : i_2$ mapping, Dr. Bones cannot really say whose age he is vouching for. Ellen's agent also provides a signature for the $i_1 : i_2$ mapping, so that if a dispute arises in the future, Dr. Bones can prove that Ellen's agent claimed that i_1 and i_2 were the same person. In summary, the request for Ellen's age proceeds as follows:

1 DealsRus asks Dr. Bones to evaluate $p(y)$ for the person DealsRus calls i_1 and who uses Ellen's agent.
2 Dr. Bones asks Ellen's agent for Ellen's id at DealsRus. If the agent approves, it sends the $i_1 : i_2$ mapping, appropriately signed.
3 Bones then looks up i_2's age, y and computes $p(y)$. The result is returned to DealsRus.

Challenges The weaker protocol we presented requires Ellen to reveal her DealsRus id to Dr. Bones. The notarizing organization (Dr. Bones) could thus gradually get to know Ellen's identity at various organizations.

Of concern to DealsRus is the fact that it has to reveal its predicate p to Dr. Bones. If the predicate p is proprietary, DealsRus may insist on keeping p private. Cryptographers have studied secure multi-party [55] (e.g., 3-party) computation that computes a function $f(a, b, c)$ with inputs a, b and c at three different parties, such that the three parties learn only $f(a, b, c)$ and nothing else.

Ellen, DealsRus and Dr. Bones can engage in a secure multi-party computation with copies of y at Ellen and Dr. Bones, and an encoding of p at DealsRus as the respective secret inputs. However, such schemes incur an excessive communication overhead. Is it possible to devise an *ef£cient* protocol for a restricted set of predicates? On the other hand, if DealsRus presents the predicate p as a secret input to the above computation, Ellen will be unable to audit p and curtail its computation. Is it possible to achieve the privacy of both Ellen and DealsRus simultaneously?

Trusted Third Party In the above protocol, Ellen's agent could cheat and provide a false mapping since only it knows the relationship between Ellen's two personas. An improvement would be to run agents at sites that could be trusted by both Ellen and the organizations she deals with. In such a scenario, a trusted organization, which we can call the *agency* can run privacy agents for a variety of individuals. The agency somehow gathers evidence that a particular individual is who they say they are (perhaps they have to show up in person and identify themselves with a photo identi£cation), and then runs an agent on their behalf. The code used for the agent can be public so that customers gain trust in the provided services. Even if the agency is trusted, individuals can still cheat in various ways, but at least organizations are able to go to the agency for help in resolving con¤icts that may arise.

As far as Ellen is concerned, her agent is a part of the agency. Thus the mappings of its personas are known only to the agency, and need not be revealed to Dr. Bones. DealsRus still has to reveal its predicate p to the agency. All parties now have to trust only *one* organization.

Summary Value Predicates can be handled by enabling an individual E to evaluate a predicate p speci£ed by an organization O_1 such that:

1 *Opaque:* O_1 learns only the result of evaluation of p; the evaluation must not reveal to O_1 information that cannot be computed using the output of the evaluation.
2 *Veri£able:* O_1 can ascertain that the predicate was computed over the correct value of E's attribute.

3.4 Multi-Source Value Predicates

The need for parties trusted by individuals and organizations becomes more evident if we consider more complex scenarios where predicates need values from different sources. To illustrate, consider a bank, EasyLoan, that needs Fred's age and salary in order to determine if it can give him a loan. All that EasyLoan needs is the output of a predicate $p(y, s)$, where y is Fred's age, and s is his salary. Fred does not want to disclose his attributes to EasyLoan; EasyLoan does not trust Fred to compute $p(y, s)$. Fortunately, EasyLoan does trust Fred's employer, Acme, to provide the salary, and Fred's doctor, Dr. Bones, to provide the age needed by the computation. How can EasyLoan obtain $p(y, s)$ from values provided by Acme and Dr. Bones while Fred retains control over his age and salary attributes?

Trusted Third Party As for value predicates, a secure multi-party computation can be used to evaluate multi-source value predicates. A simpler solution is to use an agency that is trusted by all parties. The protocol is then as follows: EasyLoan asks the agency to compute $p(y, s)$ for the person it knows as i_1 and who uses a particular agent. By this point, Fred has already disclosed to EasyLoan who may provide the age and salary, so the request from EasyLoan also includes the identities of Acme and Dr. Bones. The agency then asks Fred's agent for Fred's identities at Acme and Bones, and asks these organizations for the required data. Finally, the agency computes $p(y, s)$ and returns the value to EasyLoan. The agency keeps a record of the computation in case of future disputes.

Summary Multi-source value predicates can be handled by enabling an individual E to evaluate a predicate p speci£ed by organizations $O_1, O_2, \ldots, O_n$ such that:

1 *Opaque:* O_i learns only the result of evaluation of p; the evaluation must not reveal to O_i information that cannot be computed using the output of the evaluation.
2 *Veri£able:* the organizations can ascertain that p was computed over the correct value of E's attributes.

4 P4P: Paranoid Platform for Privacy Preferences

We have illustrated through examples a set of information types where an individual can retain control over his information. We claim that for each such information type, one can devise a general-purpose set of mechanisms to retain control. We propose to gather these set of mechanisms into one framework, which we call *P4P: Paranoid Platform for Privacy Preferences*.

We believe that private information can be classi£ed along three dimensions: (a) ownership, (b) type, and (c) desired level of control. In this section, we illustrate the dimensions and glean principles that can underly the framework. We caution that our framework is still in its formative stages, and many concepts are still not well de£ned.

For our illustration, we need to refer to a data model, and here we chose a simple entity-attributes model, although of course other models are possible.

- *Attributes* represent the basic building blocks of information, and let us say they are (label, value) pairs. For example, (name:"Alice") and (address:"123 Main St.") are attributes. For ease of notation, we will remove the quotation marks from string values.
- *Entity* has a set of related attributes. We are especially interested in entities that represent individuals or organizations. For ease of notation, we will use the term "entity" to refer to both (a) an individual/organization E in the real world, and (b) the representation of E in our framework as a set of related attributes. For instance, Alice is an entity that may be represented by the following set of attributes: [Name:Alice, Address: 123 Main St., Phone: 555-1234].

Note that in our world, attributes by themselves are usually not sensitive, e.g., nobody will care if someone knows the attribute phone: 555-1234. It is only the *association* of phone: 555-1234 and name: Alice with the entity Alice that is sensitive information. Thus, associations of attributes with an entity will be considered sensitive.

Each entity has a datastore whose contents can be classi£ed by the information's ownership, type and desired level of control. Each entity also associates a trust level for other entities it deals with, and uses those trust levels to determine how to interact with them. Each interaction involves exchange of data between the participant entities that reveals attributes of one entity to others.

Example [*Entity Interaction*] Suppose Alice buys *bread* and *butter* at AllMart using her credit card. Figure 4 shows the interaction between Alice and AllMart as seen in our framework. Alice is an entity with attributes address, credit card information and a grocery list. Similarly, AllMart is an entity with attributes address, inventory and prices of goods. Alice's interaction with AllMart reveals information about one entity to the other. Thus, Alice must

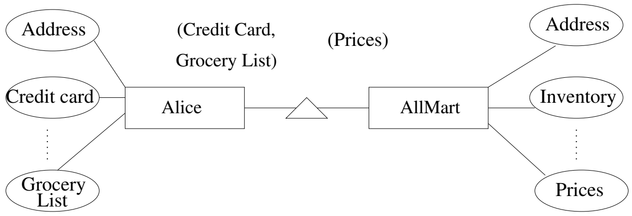

Figure 4: *Rectangles represent entities. Ovals are attributes of, and Triangles are interactions between, entities.*

share her credit card information and grocery list with AllMart. AllMart must reveal its prices of goods to Alice.

Ownership Consider the interaction between two entities, an individual Alice and an organization AllMart. The participant entities share data with each other. The data that is shared can be owned by Alice or by AllMart.

- *By Individual:* This covers data generated by Alice, i.e., her personal data (e.g., address) and preferences (e.g., grocery list). The fact that Alice owns this data means that Alice should retain full control over it, deciding when and how it can be revealed to others.
- *By Organization:* This covers data generated by AllMart. For instance, say Alice buys a particular book there. The fact that someone purchased a book may be represented by a tuple <invoiceNo:123, bookTitle: War and Peace>. This tuple is owned by AllMart, not Alice. Even though it was a purchase by Alice, Alice cannot control the information. Similarly, the tuple: <invoiceNo: 123, identi£er: 89> that links this purchase to Alice's local identi£er is owned by AllMart.

Information Type In order to control information, one needs to know what semantics it has in interactions between entities. Information can be of the following types, as illustrated by our examples in Section 3. Note that the same information can be of multiple types, depending on how it is used in an interaction. For instance, an email address could be a service handle, and it could also be an identi£er.

- *Identi£er:* An identi£er is an attribute (or a set of attributes) that is used to identify an entity in a datastore. For example, if Alice gives her phone number to AllMart, then AllMart may use that number to refer to Alice in its interactions with Alice (or her agent).
- *Service Handle:* A handle is an attribute that provides a path to a service that some other entity provides on behalf of Alice. Email addresses and credit card numbers are examples of service handles.
- *Input to Predicate:* An attribute is of this type if it can be used as an input to predicates other entities wish to evaluate. In our examples, age and salary were attributes of this type.
- *Copy:* Attributes can be copied to another entity's datastore. In such cases, it is critical to track which of the copies is the *primary* copy (at the site that owns the information) and which are the *secondary* copies.

Desired Level of Control This level speci£es how the owner wants the information managed. The level may vary by entity, that is, one level may be desired for one entity, and a different level for another, probably depending on the trust level placed on the entities (see below). We remark that a level of control (except *Sharable*) provides an *operational* description of privacy to individuals which is much more precise than the P3P policies that are descriptions of *intent*.

- *Complete Privacy:* The information should not be revealed at all.
- *Limited Time/Use:* The information can be used for a limited time, or a limited number of times, or for a particular task, as in our email example.
- *No Predicate Input:* The information cannot be used as input to predicates. We may disallow either single-source or multi-source value predicates.
- *No Integration:* The information is given to an entity, but that entity should not be able to integrate that information to other subsets we have given other entities. To enforce this restriction, we need to control the foreign keys we give out.
- *Accountable:* The information is given to an entity, but in a watermarked form such that the entity can be held responsible for a misuse of its copy of the information.
- *Anonymous:* The information is given to an entity, but in a distorted form such that the entity is unable to deduce the true value with a high degree of certainty.
- *Sharable:* As we have argued, there are cases where we must give information to an entity and hope the entity will protect our data. For this type of sharable data, we may specify what guarantees we may want from the entity, as in the P3P framework. For instance, we may only share information with an entity if it promises not to divulge it to third parties, or if it promises to only use it for computing statistics.

Trust As mentioned earlier, in our framework, an entity also needs to specify the trust it has in other entities. There are many ways to specify and manage trust that have been discussed in literature [14, 56]. Given our focus on controlling access to information, one option is to simply specify policies stating the level of control desired on information released to a target entity. For example, if an entity is trusted to enforce no-integration, then we believe it will not attempt to integrate the information we give it with what is available at other entities. If we trust an entity in this fashion, then we do not have to implement precautions with the identi£ers we give it.

Adversary In order to build mechanisms, one needs to know what adversaries must be guarded against during interactions between entities. Adversaries can be of the following types:

- *Passive adversary* is an eavesdropper who observes interactions involving a particular individual, say Alice. Such an adversary may have either a *global* view (observes all interactions by Alice) or a *local* view (ob-

I. Attribute	*II. Information Type*	*III. Level of Control*
Image	Copy	Accountable
Email	Service Handle	Limited Use
Email	Identi£er	No Integration
Age	Input to Predicate	No Input

Table 1: *Alice's P4P policy for interactions with DatesRus.*

serves all interactions by Alice with AllMart).

- *Active adversary* acts with deliberate intent to gather information, apart from observing interactions involving Alice. Such an adversary (e.g., a misbehaving AllMart) can issue speci£c queries, record and correlate answers, or even release information it gathers about Alice to other colluding entities.
- *Open-world adversary* has access to information about Alice gained through channels outside the P4P framework (e.g., a misbehaving AllMart may have access to survey results indicating the buying habits of residents of Alice's zipcode).

Properties of Interaction An unconstrained exchange of information in interactions can reveal an entity's private attribute values to others. We propose that information that is exchanged during an interaction be carefully "trimmed" to ensure information privacy. For example, let Alice participate in successive interactions $I_1, I_2, I_3, \ldots$ with DealsRus. To ensure Alice's privacy, we require the following, which we call the *TRIM* properties, on each interaction:

- *Traceability*: The data that is exchanged during an interaction I_j cannot be used by DealsRus for an interaction with another entity, without Alice having proof of DealsRus' involvement.
- *Revocability*: Alice can "sever" associations with a particular interaction in the future (e.g., on expiry of a subscription); the attribute values that was shared in such an interaction cannot be associated with Alice anymore.
- *Isolation*: Data provided in two interactions I_j and I_k $(j \neq k)$ cannot be associated with the same entity Alice.
- *Minimality*: Alice ensures that the data that is exchanged in an interaction is the minimal to successfully achieve its goals.

To illustrate our proposed taxonomy, let us return to our sample entity for Alice [Name:Alice, Age: 23, Email: alice@public.org, Image: myPic.jpg]. Let us assume that Alice wants to sign up with an online dating service called DatesRus, but does not trust DatesRus with her personal information. Alice may de£ne P4P policies as shown in Table 1 to govern her interactions with DatesRus. Each row in the table illustrates a policy that enables Alice to retain control over her personal information vis-a-vis DatesRus. Column I lists examples of attributes over which Alice desires control, Column II speci£es the attribute's type, for which Column III speci£es the level of control desired. We consider each row in more detail next.

Example [*Image* $\mapsto$ *Copy* $\mapsto$ *Accountable*] Alice wants to upload her image at the DatesRus site. However, she is worried about abuses: e.g., DatesRus may sell her image to advertisers without her permission. Since she needs to make a copy of the image which will be shared with DatesRus, she deems her attribute image: myPic.jpg to have type: copy and level of control: accountable. Therefore, Alice's agent must ensure that the image provided satis£es property: traceable so that Alice will (eventually) obtain proof of DatesRus' misuse. Alice's agent may watermark [34] the image to ensure traceability.

Example [*Email* $\mapsto$ *Service Handle* $\mapsto$ *Limited Use*] Alice wants to provide an email address for DatesRus to inform her of possible interests. However, Alice does not foresee herself using DatesRus for more than an year. She does not want DatesRus to contact her once she ends her membership. So she deems her attribute email: alice@public.org to have type: service handle and level of control: limited use. Therefore, Alice's agent must ensure that the released email address satis£es property: revocable. Alice's agent may provide a temporary email address that can be invalidated by Alice at will.

Example [*Email* $\mapsto$ *Identi£er* $\mapsto$ *No Integration*] Alice wants to interact with various organizations (e.g., DatesRus, DealsRus, ShipsRus) each of which wants her email address. Alice realizes that her email address is unique to her, and could be used as her identi£er by the organizations. She does not want (a subset of) these organizations to get together and integrate their datasets without her knowledge. So she deems her attribute email: alice@public.org to have type: identi£er and level of control: no integration. Therefore, Alice's agent must ensure that the released email address satis£es property: isolation as well. Alice's agent may create distinct temporary email addresses, one each for each organization to ensure isolation.

Example [*Age* $\mapsto$ *Input to Predicate* $\mapsto$ *No Input*] DatesRus has promotional offers from local clubs that provides free entry to DatesRus clients under the age of 25. Alice does not want to reveal her age and has decided to decline any offer that requires her to reveal her age. So she deems her attribute age: 23 to have type: input to predicate and level of control: no input. Therefore, Alice's agent must ensure that optional age-based predicates from DatesRus are not evaluated.

In summary, in our P4P framework, it is important to understand who controls (owns) data, how the data is being used (type), what control is desired, and what agents can be trusted. For each attribute (a point in the ownership, type, control space), our goal is then to provide one or more mechanisms that enforce the desired privacy.

In the P4P framework, trusted agencies play a central role. As illustrated in our examples, they provide agent and predicate evaluator services, so that entities can effectively control and at the same time share their information. Each individual would contract with one or more agencies to provide services, and perhaps to store some of their data too. As the individual interacts with organizations, instead of giving out information directly, it asks its agent to provide appropriate attributes, whether they be private email addresses or private identi£ers. When an organization needs additional information about an individual, it can contact its agent or a trusted agency to obtain the data.

5 Analysis of Current Privacy Technologies

Entities (organizations/individuals) have legitimate access to certain information. Privacy is assured by preventing illegitimate use of this information by adversaries. In this section, we discuss proposed privacy solutions and their limits in assuring information privacy.

Secure Databases Research here has focused on enabling storage and query of sensitive information to detect and prevent unauthorized disclosure, alteration, or destruction of data [16, 43]. The work in this area in the context of databases can be broadly classi£ed into role-based access-control and multi-level security. In role-based access-control, accesses to data are allowed or prohibited based on the role in which an entity is acting [44, 51]. Such access-control is usually offered on a per-table granularity. In multi-level security, both data and entities are assigned security levels drawn from a security hierarchy [32]. A "no read up, no write down" policy [11] is imposed: an entity can only read from a level below (and including) their own in the hierarchy, an entity can only write to a level above (and including) their own in the hierarchy. The P4P framework complements this work for securing local copies of data by building mechanisms that seek to contain illegitimate use of released data.

Trust Management Systems generalize the access-control mechanisms described above by operating in distributed systems. They de£ne languages for expressing authorizations and access-control policies [13, 25] that can work under the assumption that all of the parties may not be known when policies are expressed. Recently, researchers have explored automated trust negotiations between two entities participating in an interaction [56]. When an entity E_r requests data from another entity E_s, trust is negotiated between E_r and E_s on the ¤y by iteratively disclosing their credentials and access policies. The access is permitted only if E_r can establish to E_s that it has the necessary authorization. The trust component in the P4P framework will bene£t from the work in this area.

Statistical Databases Research here has focused on enabling queries on aggregate information (e.g., sum, count, average, median, etc.) from a database without revealing individual records [2]. The work in this area can be broadly classi£ed into data perturbation and query restriction. Data perturbation involves either altering the input database [40, 52], or altering query results returned [10, 19]. Query restriction includes schemes that check for possible privacy breaches by keeping audit trails [18] and controlling overlap [21] of successive aggregate queries. While we expect some of these techniques to £nd applications in our framework (e.g., data perturbation to enable an anonymous level of control, query restriction to audit value predicates), new mechanisms will be needed to perturb the richer P4P information types and audit the richer P4P queries.

Anonymous Networks Deployed systems have often preferred to achieve privacy for an entity participating in an interaction by making its identity anonymous. Application-speci£c schemes have been designed to enable anonymous messaging (Onion Routing [50]), anonymous emails (Mix Nets [17]), anonymous web browsing (Crowds [45]), anonymous publishing (FreeHaven [20]), and anonymous indexing (Privacy Preserving Indexes [9]). The P4P framework allows an individual to declare the level of control she desires over speci£c information. We expect some of these techniques to £nd applications when an individual opts to share her information anonymously.

Cryptography Primitives are used to protect information in transit and storage from adversaries. There is extensive literature on cryptographic tools [48] that will be of use in the P4P framework: e.g., symmetric and public key encryption for secure transit and storage, signatures to establish ownership of information, certi£cates to establish legitimate access to information, watermarking to trace the origins of data, etc.

Given the new design goals envisioned by our P4P framework, it is not surprising that some of the tools fall short in meeting the challenges posed. For example, the communication overhead of secure multi-party computation [27, 55] makes its application infeasible for "multi-source value predicate": organizations may need to participate in thousands of interactions per minute with their customers. We believe that ef£cient protocols will need to be designed for speci£c functions that arise in the P4P framework, much like the recently invented specialized protocols for £nding medians [4] and intersections [5] of private values at multiple entities.

6 Challenges

We have sketched our vision for an information processing world where individuals can retain control over their information. Organizations can also bene£t by not getting control, and the accompanying liability, of information they do not own. This information processing model will require that organizations and individuals operate with information in different ways. Of course, the challenges to achieve this vision are huge, and in closing we mention a few.

6.1 Interfaces for Entities, Agents and Humans

Adequate programmatic interfaces need to be de£ned for entities, agents, agencies, predicate evaluators and notaries. Agent interfaces for dealing with information types will have generic and application dependent parts. For example, an agent may be asked to create a service handle that is limited for one day (a generic restriction) or a handle that only allows charges of up to 100 dollars (application speci£c for money-related handles). Traceable copies of data may require embedding of application-dependent £ngerprints [34]. It will be important to explore application-speci£c controls and services that would be useful.

Human interfaces must be invented that enable people to describe their privacy goals and select appropriate policies for their agents. The interface must also educate people about risks of their options. The recent work on privacy interfaces for ubiquitous computing will be useful here. Research there has highlighted that individuals tend to release information subjectively while weighing in factors like information function, information sensitivity, and trust in recipient [3, 37] which mirror our *owner - type - level of control* dimensions.

There has recently been an interest in exploring the nature of privacy as a value determined by market forces [36, 53]. Instead of a declarative policy, individuals in this model may be willing to relax their level of control in return for a fair compensation. How can such schemes be incorporated in the interface, and indeed, the framework?

6.2 Reasoning about Information Privacy

While we have presented a few useful points in the *ownership - type - level of control* spectrum, it is important to specify information work¤ows for a variety of interactions and formally reason about privacy guarantees as an aggregate of an entity's interactions.

In our strawman design, we postulated that each entity will log *all* interactions it has participated in with other entities. The agent will use an entity's log to pre-process (or even abort) current interactions to prevent violation of the entity's privacy policies. An entity can query its logs to deduce the personal information that has been released to a particular entity. However, such logs will quickly grow to be quite large. Ef£cient log management, analysis and summarizing algorithms will need to be invented to allow online entity interactions to be fast. Can we design interactions with properties (e.g., TRIM) that reduce the size of logs? Analysis of logs and auditing of P4P queries will require extending statistical databases techniques for audit of aggregate queries in new directions. Furthermore, how would such an audit scheme work against an open-world adversary with its knowledge of auxiliary datasets that may not be currently known to the individual's agent?

6.3 Architecture of a Privacy Agent/Agency

We touched upon challenges in designing privacy preserving protocols in Section 3. Perhaps the recent advances [15] in designing ef£cient group signatures [47] for anonymous authentication can be used to devise a Notary Protocol? A group signature scheme allows a member M of a group G to sign messages on behalf of G such that the resulting signature does not reveal M's identity. Thus, in the examples of Section 3, we could place Acme in a group of employers and Dr. Bones in a group of doctors. Acme and Dr. Bones would vouch for Fred's salary and age using their respective group signatures. EasyLoan can verify the signatures, and still not know the identity of Acme or Dr. Bones.

The examples in Section 3 assumed a cryptographic definition of privacy. Can ef£cient agents be designed when individuals desire an "anonymous" level of control? Such schemes should allow the individual to increase the level of anonymity of interactional data by using various information hiding schemes (e.g., k-anonymity [49], perturbation [7]). The infrastructure should, however, provide statistics to indicate the level of anonymity achieved. How can such statistics be maintained?

6.4 Trust Management

It will be important to understand the interactions between the P3P privacy policies and our privacy control mechanisms. The P3P framework still plays an important role in describing how trusted organizations will manage data they own or have a copy of. Perhaps the agency can play a role in managing trust for the entities it represents. For example, the agency can track privacy breaches (e.g., misuse of limited-use emails or pseudonums) by organizations and assign them "trust ratings". Such trust ratings can be used by individuals to determine policies for their interactions with an organization.

6.5 Secure Society

Individual privacy and societal security are sometimes at logger heads with each other. For example, the "no integration" level of control precludes, among other things, the construction of credit reports and pro£ling of criminals. Such integration of information without the individual's intervention is essential for a smooth functioning of society. The moral dilemma here is akin to the one faced by designers of mechanisms to ensure communication privacy: the technology is of as much use to drug traf£ckers, terrorists and subversive elements as to law abiding citizens. Can the P4P framework be designed with suf£cient "hooks" to allow law-enforcement agencies to monitor interactions that hamper societal security?

7 Final Thoughts

We contend that technology must be devised to allow individuals to retain control over their information. While other commentators have also stressed the need and suggested legal avenues [12, 39], we have sought to devise an information processing framework to enable such control.

For instance, cryptologists have provided primitives (e.g., Public Key Infrastructure [29]) to individuals to achieve communication privacy. We remark that while communication privacy could have been ensured by legislating that organizations (e.g., email servers, ISPs, etc.) respect individual privacy, PKI primitives have put the right to communication privacy £rmly into the individual's hands. Can we, in the database community, build an analogous infrastructure that ensures information privacy?

References

[1] Acxiom opts out of opt-out. http://www.wired.com, 11/17/2003.

[2] N. R. Adam and J. C. Wortman. Security-control methods for statistical databases. *ACM Computing Surveys*, 21(4), 1989.

[3] A. Adams and M. A. Sasse. Taming the wolf in sheep's clothing: Privacy in multimedia communications. In *Proc. of ACM Multimedia*, 1999.

[4] G. Aggarwal, N. Mishra, and B. Pinkas. Secure computation of the k-th ranked element. In *Proc. of the IACR Conf. on Eurocrypt*, 2004.

[5] R. Agrawal, A. Ev£mievski, and R. Srikant. Information sharing across private databases. In *Proc. of the ACM SIGMOD*, 2003.

[6] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Hippocratic databases. In *Proc. of the VLDB*, 2002.

[7] R. Agrawal and R. Srikant. Privacy preserving data mining. In *Proc. of the ACM SIGMOD*, 2000.

[8] American express offers disposable credit card numbers for online shopping. http://www.computerworld.com, 9/7/2000.

[9] M. Bawa, R. J. Bayardo Jr., and R. Agrawal. Privacy-preserving indexing of documents on the network. In *Proc. of the VLDB*, 2003.

[10] I. I. Beck. A security mechanism for statistical databases. *ACM TODS*, 5(3), 1980.

[11] D. E. Bell and L. J. LaPadula. Secure computer systems: mathematical foundations and model. Technical report, MITRE, 1974.

[12] J. Berman and D. Mulligan. Privacy in the digital age: Work in progress. *Nova Law Review*, 23(2), 1999.

[13] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. D. Keromytis. The KeyNote trust-management system (Version 2). *RFC 2704*, 1999.

[14] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management. In *Proc. of the IEEE Symp. on Security and Privacy*, 1996.

[15] D. Boneh. A new life for group signatures. Plenary Talk at RSA Conf., 2004.

[16] S. Castano, M. G. Fugini, G. Martella, and P. Samarati. *Database Security*. Addison-Wesley and ACM Press, 1995.

[17] D. L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *CACM*, 24(2), 1981.

[18] F. Chin and G. Ozsoyoglu. Auditing and inference control in statistical databases. *IEEE TSE*, 8(6), 1982.

[19] D. Denning. Secure statistical databases with random sample queries. *ACM TODS*, 5(3), 1980.

[20] R. Dingledine, M. J. Freedman, and D. Molnar. The Free Haven project: Distributed anonymous storage service. In *Workshop on Design Issues in Anonymity and Unobservability*, 2000.

[21] D. Dobkin, A. Jones, and R. Lipton. Secure databases: Protection against user in¤uence. *ACM TODS*, 4(1), 1979.

[22] D. Eastlake and P. Jones. US Secure Hash Algorithm 1 (SHA1). *Network Working Group*, RFC 3174, 2001.

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

[24] M. Marchiori (Ed.). *The Platform for Privacy Preferences 1.0 (P3P1.0) Speci£cation*. W3C Proposed Recommendation, 2002.

[25] C. M. Ellison, B. Frantz, B. Lampson, R. L. Rivest, B. M. Thomas, and T. Ylonen. SPKI certi£cate theory. *RFC 2693*, 1999.

[26] E. Gabber, P. Gibbons, Y. Matias, and A. Mayer. How to make personalized web browsing simple, secure and anonymous. In *Proc. of the Conf. on Financial Cryptography*, 1997.

[27] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game – a completeness theorem for protocols with a honest majority. In *Proc. of the ACM STOC*, 1987.

[28] Hacker accessed customer information, Acxiom reports (08/07/2003). http://www.siliconvalley.com.

[29] R. Housley, W. Polk, W. Ford, and D. Solo. Internet X.509 Public Key Infrastructure certi£cate and Certi£cate Revocation List (CRL) pro£le. *RFC 3280*, 2002.

[30] C. D. Hunter. Recoding the architecture of cyberspace privacy: Why self-regulation and technology are not enough. http://www.asc.upenn.edu/usr/chunter/ p3p.html, 2000.

[31] iPrivacy's Internet Identity Protection Service. http://www.iPrivacy.com.

[32] S. Jajodia and R. Sandhu. Toward a multilevel secure relational data model. In *Proc. of the ACM SIGMOD*, 1991.

[33] Jetblue shared passenger data. http://www.wired.com, 09/18/2003.

[34] S. Katzenbeisser and F. A. Petitcolas (Eds.). *Information Hiding Techniques for Steganography and Digital Watermarking*. Artech House, 2000.

[35] J. Kaufman, S. Edlund, D. Ford, and C. Powers. The social contract core. In *Proc. of the WWW*, 2002.

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

[37] S. Lederer, J. Mankoff, and A. K. Dey. Towards a deconstruction of the privacy space. In *Proc. Workshop on Ubicomp Communities: Privacy as Boundary Negotiation*, 2003.

[38] K. Lee and G. Speyer. Platform for Privacy Preferences (P3P) and Citibank. http://www.w3.org/P3P/Lee_Speyer.html, 1998.

[39] Lawrence Lessig. Architecture of privacy. http://cyber.law.harvard.edu/works/lessig/architecture_priv.pdf, 1998.

[40] C. K. Liew, U. J. Choi, and C. J. Liew. A data distortion by probability distribution. *ACM TODS*, 10(3), 1985.

[41] Microsoft next-generation secure-computing base. http://www.microsoft.com/Technet/security/news/NGSCG.asp.

[42] Northwest airlines faces privacy suits. http://www.washingtonpost.com, 01/22/2004.

[43] R. Oppliger. Internet security: Firewalls and beyond. *CACM*, 40(5), 1997.

[44] F. Rabitti, E. Bertino, W. Kim, and D. Woelk. A model of authorization for next-generation database systems. *ACM TODS*, 16(1), 1991.

[45] M. K. Reiter and A. D. Rubin. Crowds: Anonymity for Web transactions. *ACM TISSEC*, 1(1), 1998.

[46] M. Rotenberg. What Larry does'nt get: Fair information practices and the architecture of privacy. *Stanford Technology Law Review*, 1, 2001.

[47] D. Schaum and E. van Heyst. Group signatures. In *Proc. of the EuroCrypt Conf.*, 1991.

[48] Douglas R. Stinson. *Cryptography: Theory and Practice*. CRC Press, 2nd edition, 2002.

[49] L. Sweeney. k-anonymity: A model for protecting privacy. *Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5), 2002.

[50] P. Syverson, D. M. Coldsehlag, and M. C. Reed. Anonymous connections and onion routing. In *Proc. of the IEEE Symp. on Security and Privacy*, 1997.

[51] T. Ting, S. Demurjian, and M. Hu. A speci£cation methodology for user-role based security in an object-oriented design model. In *Proc. of IFIP Workshop on Database Security*, 1992.

[52] J. Traub, Y. Yemini, and H. Woznaikowski. The statistical security of a statistical database. *ACM TODS*, 9(4), 1984.

[53] H. R. Varian. Economic aspects of personal privacy. *Privacy and Self-Regulation in the Information Age*, NTIA, 1997.

[54] A. Westin. *Privacy and Freedom*. Atheneum, 1967.

[55] A. C. Yao. How to generate and exchange secrets. In *Proc. of the ACM FOCS*, 1986.

[56] T. Yu, M. Winslett, and K. E. Seamons. Automated trust negotiation over the internet. In *Proc. of Multiconference on Systemics, Cybernetics and Informatics*, 2002.