CHAPTER 1 *Introduction*

1.1 The fundamental premise

Technology does not exist in a social vacuum. The design and patterns of use of any particular technological artifact have implications both for the direct users of the technology, and for society at large. Decisions made by technology designers and implementors thus have political implications that are often ignored. If these implications are not made a part of the design process, the resulting effects on society can be quite undesirable.

The research advanced here therefore begins with a political decision: It is almost always a greater social good to protect personal information against unauthorized disclosure than it is to allow such disclosure. This decision is expressly in conflict with those of many businesses and government entities. Starting from this premise, a multi-agent architecture was designed that uses both strong cryptography and decentralization to enable a broad class of Internet-based software applications to handle personal information in a way that is highly resistant to disclosure. Further, the design is robust in ways that can enable users to trust it more easily: They can trust it to keep private information private, and they can trust that no single entity can take the system away from them. Thus, by starting with the explicit political goal of encouraging well-placed user trust, the research described here not only makes its social choices clear, it also demonstrates certain technical advantages over more traditional approaches.

We discuss the political and technical background of this research, and explain what sorts of applications are enabled by the multi-agent architecture proposed. We then describe a representative example of this architecture---the Yenta matchmaking system. Yenta uses the coordinated interaction of large numbers of agents to form coalitions of users across the Internet who share common interests, and then enables both one-to-one and group conversations among them. It does so with a high degree of privacy, security, and robustness, without requiring its users to place unwarranted trust in any single point in the system.

The research advanced here attempts to break a false dichotomy, in which systems designers force their users to sacrifice some part of a fundamental right—their privacy—in order to gain some utility—the use of the application. We demonstrate that, for a broad class of applications, which we carefully describe, this dichotomy is

indeed *false*—that there is no reason for users to have to make such a decision, and no reason for systems designers to force it upon them.

If systems architects understand that there is not necessarily a dichotomy between privacy and functionality, then they will no longer state a *policy* decision—whether to ask users to give up a right—as a *technical* decision—one required by the nature of the technology. Casting decisions of corporate or government policy as technical decisions has confused public debate about a number of technologies. This work attempts to undo some of this confusion.

The research presented here is thus intended to serve as an exemplar. The techniques presented here, and the sample application which demonstrates them, are intended to serve as examples for other systems architects who design systems that must manipulate large quantities of personal information.

1.2 What's ahead?

In this chapter, we shall:

Section 1.3
- Describe which type of privacy we are most interested in protecting

Section 1.4
- Discuss the concept of privacy as a right, not a privilege

Section 1.5
- Show some of the technical, social, and political problems with centralized manipulation of personal information

Section 1.6
- Show some of the advantages of a decentralized solution

Section 1.7
- Discuss the components of the work presented here, specifically its *architecture*, the *sample application* of that architecture, the *implementation* of that application, and issues of *deployment and evaluation*

Section 1.7
- Briefly summarize the remaining chapters of this dissertation

Later chapters will:

Chapter 2
- Discuss the system architecture for the general case

Chapter 3
- Analyze user privacy and system security

Chapter 4
- Detail the sample application—the matchmaking system Yenta

Chapter 5
- Discuss the evaluation of the architecture and of Yenta

Chapter 6
- Examine some related work

Chapter 7
- Draw some general conclusions

1.3 What are we protecting?

Privacy means different things to different people, and can be invoked in many contexts. We define privacy here as *the protection of identifiable, personal information about a particular person from disclosure to third parties who are not the intended recipients of this information.* This sentence deserves explanation, and we shall explain it below. We shall also touch upon some related concepts, such as *trust* and *anonymity,* which are required in this explanation.

Protection

Protecting a piece of information means keeping it from being transmitted to certain parties. Which parties are not supposed to have the information is dependent upon the wishes of the information's owner. This process is transitive—if party A willingly transmits some information about itself to party B, but party B then transmits this information to some party C, which A did not wish to know it, then the information has not been protected. Such issues of transitivity thus lead to issues of *trust* (see below) and issues of *assignment of blame*—whether the fault is in A (who *trusted* B not to disclose the information, and had this trust violated) or in B (who *disclosed the information without authorization* to C), or in both, depends on our goal in asking the question.

Identifiability

Unlinkability

In many cases, *disclosure* of information is acceptable if the information cannot be traced to the individual about whom the information refers—we refer to this as *unlinkability.* This is obvious in, for example, the United States Census, which, ideally, asks a number of questions about every citizen in the country. These answers to these questions are often considered by those who answer them to be private information, but they are willing to answer them for two reasons: The collection of the information is deemed to have *utility* for the country as a whole, and the collectors of the information make assurances that the information will not be *identifiable*, meaning that it will not be possible to know which individual answered any given question in any particular way—the respondents are *anonymous*. Because the Census data is gathered in a *centralized* fashion, it leads to a *concentration of value* which makes trust an important issue: central concentrations of data are more subject to institutional abuse, and make more tempting targets for outsiders to compromise.

Particular person

Whether or not the information is about a *particular* person—someone how is identifiable and is linkable to the information—or is instead about an aggregate can make a large difference in its sensitivity to disclosure. Aggregate information is usually considered less sensitive—although cross-correlation between separate databases which talk about the same individuals can often be extremely effective at revealing individuals again in the data, and represent a serious threat to systems which depend for their security solely on aggregation of data [169].

Personal information

When we use the term *personal information*, we mean information that is known *by* some particular individual *about* himself, *or* which is known to some set of parties who that individual considers to be authorized to know it. If no one else knows this information yet, the individual is said to *control* this information, since its disclosure to anyone else is presumably, at this moment, completely up to the individual himself. We are *not* referring to the situation whereby party A knows something about party B that B does not know about himself. Such situations might arise, for example, in the context of medical data which is known to a physician but has not yet (or, perhaps is not ever) revealed to the patent. In this case, B cannot possibly protect this information from disclosure, for two reasons: B does not have it, and because the information is known by someone who may or may not be under A's control.

Disclosure

If personal information about someone is not *disclosed,* then it is known only to the originator of that information. In this case, the information is still private. One of the central problems addressed by this dissertation is how to *disclose* certain information so that it may be used in an application, while still giving the subject control over it.

Third parties

Many existing applications which handle personal information do so by surrendering it, in one way or another, to a third party. This work attempts to demonstrate that this is not always required. In many instances, there is no *need to know*—knowledge of this information by the third party will not benefit the person whom this information is about. We usually use the term *third party* to mean some other entity which does not have a compelling need to know.

Intended recipients

The *intended recipient* of some information is the party which the subject desires to have some piece of personal information. If the set of intended recipients is empty, then the information is *totally private*, and, barring involuntary disclosures such as search and seizure, the information will stay private. The work presented here concerns cases where, for whatever reason, the set of intended recipients is nonempty.

Trust

Whenever private information is surrendered to an intended recipient, the subject *trusts* the recipient, to one degree or another, not to disclose this information to third parties. (If the subject has no trust in the recipient at all, but discloses anyway, either the subject is acting against his own best interests, or the information was not actually private to begin with—in other words, if the information is *public* and it does not mat-

ter who knows it, then there is no issue of trust.) Trust can be misplaced. A robust solution in any system, social or technological, that handles private information generally specifies that trust be extended to as few entities, in as minimal a way as possible to each one. This minimizes the probability of disclosure and the degree of damage that can be done by disclosure due to a violation of the trust extended by the subject.

Anonymity and pseudonymity

In discussing *unlinkability of information*, such as that expected by respondents to the US Census, we mentioned that the respondents trust that they are *anonymous*. To be fully anonymous is to know that information about oneself cannot be associated with one's physical extension—the actual individual's body—or with any other anonymous individual—all anonymous individuals, to a first approximation, might as well be the same person. This also means that the individual's real-world personal reputation, and any identities in the virtual world (such as electronic mail identification), are similarly dissociated from the information. Full anonymity is not always possible, or desired, in all applications—for example, most participants in a MUD are pseudonymous [20][33][49][59][60][116]. This means that they possess one or more identities, which may be distinguished from other identities in the MUD (hence are not fully anonymous), but which may not be associated with the individual's true physical extension. The remailer operated at penet.fi.net [77], for example, also used pseudonyms. There are even works of fiction whose primary focus is the mapping between pseudonyms and so-called *true names* in a virtual environment [176].

Reputations

The reason why the distinction between anonymity, pseudonymity, and true names matters has to do with *reputations*. In a loose sense, one's reputation is some collection of personally-identifiable information that is associated, across long timespans, with one's identity, and is known to a possibly-large number of others. In the absence of any sort of pseudonymous or anonymous identities, such reputations are directly associated with one's physical extension. This provides some degree of *accountability for one's behavior*, and can be either an advantage or a disadvantage, depending on that behavior—those with good reputations in their community are generally afforded greater access to resources, be they social or physical capital, than those with poor reputations. Pseudonymous and anonymous identities provide a degree of decoupling between the actions of their owners and the public identity. Such decoupling can be invaluable in cases where one wishes to take an action that might land the physical extension in trouble. This decoupling has a cost: because a pseudonym, and, particularly, an anonym, is easier to throw away than one's real name or one's body, they are often afforded a lower degree of trust by others.

A legal definition

Another way to look at the question of what we are protecting is to examine legal definitions. For a US-centric perspective, consider this definition from Black's Law Dictionary [14]:

> Privacy, Right of:
>
> The right to be let alone; the right of a person to be free from unwanted publicity; and right to live without unwarranted interference by the public in matters with which the public is not necessarily concerned. Term "right of privacy" is generic term encompassing various rights recognized to be inherent in concept of ordered liberty, and such right prevents governmental interference in intimate personal relationships or activities, freedoms of individual to make fundamental choices involving himself, his family, and his relationship with others. *Industrial Foundation of the South v. Texas*

Indus. Acc. Bd., Tex., 540 S.W.2d 668, 679. The right of an individual (or corporation) to withhold himself and his property from public scrutiny, if he so chooses.

It is said to exist only so far as its assertion is consistent with law or public policy and in a proper case equity will interfere, if there is no remedy at law, to prevent an injury threatened by the invasion of, or infringement upon, this right from motives of curiosity, gain, or malice. *Federal Trade Commission v. American Tobacco Co.,* 264 U.S. 298, 44 S.Ct. 336, 68 L.Ed. 696. While there is no right of privacy found in any specific guarantees of the Constitution, the Supreme Court has recognized that zones of privacy may be created by more specific constitutional guarantees and thereby impose limits on governmental power. *Paul v. Davis* 424 U.S. 693, 712, 96 S.Ct. 1155, 1166, 47 L.Ed.2d 405; *Whalen v. Roe*, 429 U.S. 589, 97 S.Ct. 869, 51 L.Ed.2d 64. See also Warren and Brandeis, *The Right to Privacy*, 4 Harv.L.Rev. 193.

Tort actions for invasion of privacy fall into four general classes: *Appropriation*, consisting of appropriation, for the defendant's benefit or advantage, of the plaintiff's name or likeness. *Carlisle v. Fawcett Publications*, 201 Cal. App2d 733, 20 Cal. Rptr 405. *Intrusion* [. . .] *Public disclosure* of private facts, consisting of a cause of action in publicity, of a highly objectionable kind, given to private information about the plaintiff, even though it is true and no action would lie for defamation. *Melvin v. Reid* 112 Cal. App. 285, 297 P. 91. [. . .] *False light in the public eye* [. . .]

1.4 The right to privacy

Constitutional arguments

Why is personal privacy worth protecting? Is it a right, which cannot be taken away, or a privilege, to be granted or rescinded based on governmental authority?

In the United States, there is substantial legal basis that personal privacy is considered a right, not a privilege. Consider the Fourth Amendment to the US Constitution, which reads:

> The right of the people to be secure in their persons, houses, papers, and effects, against unreasonable searches and seizures, shall not be violated, and no Warrants shall issue, but upon probable cause, supported by Oath or affirmation, and particularly describing the place to be searched, and the persons or things to be seized.

While this passage is the most obvious such instance in the Bill of Rights, it does not explicitly proclaim that privacy itself is a right.

There are ample other examples from Constitutional law, however, which have extended the rights granted implicitly by passages such as the Fourth Amendment above. Supreme Court Justice Brandeis, for example, writing in the 1890's and later, virtually created the concept of a Constitutional right to privacy [180]. For example, consider this quote, from *Olmstead v. United States* [130], writing about the then-new technology of telephone wiretapping:

> The evil incident to invasion of the privacy of the telephone is far greater than that involved in tampering with the mails. Whenever a telephone line is tapped, the privacy of the persons at both ends of the line is invaded, and all conversations between them upon any subject, and although proper, confidential, and privileged, may be overheard. Moreover, the tapping of one man's telephone line involves the tapping of the telephone of every other

> person whom he may call, or who may call him. As a means of espionage, writs of assistance and general warrants are but puny instruments of tyranny and oppression when compared with wire tapping.

Later examples supporting this view include *Griswald v. Connecticut* [71], in which the Supreme Court struck down a Connecticut statue making it a crime to use or counsel anyone in the use of contraceptives; and *Roe v. Wade* [147], which specified that there is a Constitutionally-guaranteed right to a personal sphere of privacy, which may not be breached by government intervention.

Moral and functional arguments

But the laws of the United States are not the only basis upon which one may justify a right to privacy—for one thing, they are only valid in regions in which the United States government is sovereign. It is the author's contention that there is a *moral* right to privacy, even in the absence of law to that effect, and furthermore that, even in the absence of such a right, it is a *social good* that personal privacy exists and is protected—in other words, that personal privacy has a *functional benefit*. In other words, even if one were to state that there is no legal or moral reason to be supportive of personal privacy, society functions in a more productive manner if its members are assured that personal privacy can exist. For example, there are spheres of privacy surrounding doctor/patient and attorney/client information which are viewed as so important that they are codified into the legal system of many countries. Without such assurances of confidentiality, certain information might not be exchanged, which would lead to an impairment of the utility of the consultation.

One might also argue that the *fear of surveillanc*e is itself destructive, and that privacy is a requirement for many sorts of social relations. For example, consider Fried [64]:

> Privacy is not just one possible means among others to insure some other value, but . . . it is necessarily related to ends and relations of the most fundamental sort: respect, love, friendship and trust. Privacy is not merely a good technique for furthering these fundamental relations; rather without privacy they are simply inconceivable.

For the purposes of this work, we shall take such moral and social-good assertions as *axioms*, e.g., not requiring further justification.

Implications for systems architects

Those who design systems which handle personal information therefore have a special duty: They must not design systems which unnecessarily require, induce, persuade, or coerce individuals into giving up personal privacy in order to avail themselves of the benefits of the system being designed. In other words, system architects have a moral, ethical, and perhaps even—in certain European countries, which have stronger data privacy laws than the US—legal obligations to design such systems from a standpoint that is protective of individual privacy when it is possible to do so.

There may be strong motives *not* to design systems in such a fashion that they are protective of personal privacy. We shall investigate some of the motives, with examples, in the next section, but overall themes include:

See Section 1.5.

- It is often conceptually far simpler to design a system which centralizes information, yet such systems are often easily compromiseable, either through accident, malice, or subpoena.
- The architects of many systems often have an incentive to violate users' privacy, often on a large scale. The business models of many commercial entities, especially in the United States, depend on the collection of personal information in order to obtain marketing or demographic data, and many entities, such as credit bureaus, exist solely to disseminate this information to third parties. The European Union

has data-protection laws forbidding this [47].

- Government intervention may dictate that users' privacy be compromised on a large scale. CALEA [21] is a single, well-known example; it requires that US telephone switch manufacturers make their switches so-called *tap-ready.*

Hiding policy decisions under a veil of technological necessity

In many instances, the underlying motives which lead to a system design that is likely to compromise users' privacy are hidden from view. Instead of being clearly articulated as decisions of policy, they are presented as requirements of the particular technological implementation of the system. For example, consider most Intelligent Transportation Systems [18], such as automated tollbooths which collect fees for use of roads. These systems mount a transponder in the car, and a similar unit in the tollbooth. It is possible, using essentially the same hardware on both the cars and in the tollbooths, to either have a *cash-based* system or a *credit-based* system. A cash-based system works like Metrocards in many subways—users fill up the card with cash (in this case, cryptographically-based electronic cash in the memory of the car's transponder), and tollbooths instruct the card to debit itself, possibly using a cryptographic protocol to ensure that neither the tollbooth nor the car can easily cheat. A transaction-based system, on the other hand, assigns a unique identifier to each car, linked to a driver's name and address, and the car's transponder then sends this ID to the tollbooth. Bills are sent to the user's home at the end of the month.

An example from the Intelligent Transportation System infrastructure

In other words, a cash-based system works like real, physical cash, and can be easily anonymous—users simply go somewhere to fill up their transponders, and do not need to identify themselves if they hand over physical cash as their part of the transaction. Even if they use a telephone link and a credit card to refill their transponders at home, a particular user is not *necessarily* linked to a particular transponder if the cryptography is done right. And even if there is such a linkage between users and transponders, there is no need for the system as a whole to know *where* any particular transponder has been—once the tollbooth decides to clear the car, there is no reason for any part of the system to remember that fact. On the other hand, a credit-based system works like a credit card—each tollbooth must report back to some central authority that a particular transponder went through it, and it is extremely likely that *which* tollbooth made this report will be recorded as well.

Cash vs credit

Both cash- and credit-based systems can use the same hardware at both the car and the tollbooth; the difference is simply one of software. In fact, the cash-based system is simpler, because each tollbooth need not communicate in real-time with a central database somewhere. (Tollbooths in either system must have a way of either detaining cars with empty or missing transponders, or logging license plates for later enforcement, but the latter need not require a real-time connection for the tollbooth to function.) Furthermore, a cash-based system obviates the need for printing and mailing bills, processing collections, and so forth.

Same hardware either way; cash is actually simpler

Yet it is almost invariably the case that requests for proposals, issued when such systems are in the preliminary planning stages, simply *assume* a credit-based system, and often *disallow* proposals which can enable a cash-based system. This means that such systems, from the very beginning, are implicitly designed to enable *tracking the movements of all drivers who use them,* since, after all, each tollbooth must remember this information for billing purposes. Furthermore, drivers are likely to demand itemized bills, so they can verify the accuracy of the data. (After all, it is no longer the case that they need worry only about the contents of their local transponder—they must worry about the central database, too.) Yet such a system can easily be used, either by someone with access to the bill mailed to an individual, or via subpoena or compromise at the central database, to stalk someone or to misuse knowledge about where the individual has been, and when. Large-scale data mining of such systems can infringe on people's freedom of assembly, by making particular driving patterns

ITS RFP's implicitly assume that drivers should be tracked

inherently suspicious—imagine the case whereby anyone taking an uncommon exit on a particular day and time is implicitly assumed to have been going to the nearby political rally. And even the *lack* of a record of a particular transit has already been used in court proceedings [18].

ITS RFP's are setting policy, not responding to technological necessity

The aim of the work presented in this dissertation is the demonstration that many, if not most, of these systems can be technically realized in forms that are as protective of users' individual privacy as one might wish. Therefore, designers of systems who fail to ensure their users' privacy are making a policy decision, not a technical one: they have decided that their users are not entitled to as much personal privacy as is possible to provide, and are implementing this decision by virtue of the architecture of the system.

Unnecessary polarization of the terms of the debate

While it is the author's contention that most such decisions are, at best, misguided, and at worst unethical, the fact that they are often disguised as purely technical issues polarizes the debate unnecessarily and is not a social good. If some system, whose capabilities would improve the lives of its users, is falsely presented as necessarily requiring them to give up some part of a fundamental right in order to be used, then debate about whether or not to implement or use the system is likewise directed into a false dichotomy. By allowing debate to be thus polarized, and by requiring users to trade off capabilities against rights, it is the author's contention that the designers and implementors of such a system are engaging in unethical behavior.

Legitimate reasons against absolute personal privacy

There may be many legitimate reasons why absolute privacy a system's users is undesirable. It is not the aim of this work to assert that there are no circumstances under which personal privacy may be violated; indeed, the moral and legal framework of the vast majority of countries presupposes that there must be a balancing between the interests of the individual in complete personal privacy, and those of the state or sovereign state in revealing certain information about an individual to third parties.

This work aims to decouple technical necessity from decisions of policy

However, we should be clear about the nature of this balancing. It should be dictated by a decision-making process which is one of policy. In other words, *what is the desired outcome?* It should not instead be falsely driven by assertions about *what the technology forces us to do.* The aim of this research is to decouple these two issues, for a broad class of potential applications, and to demonstrate by example that technological issues need not force our hand when it comes to policy issues. Such a demonstration by example, it is hoped, will also make clearer the ethical implications of designing a system which is insufficiently protective of the personal privacy of its users.

1.5 The problems with centralized solutions

It is often the case that applications which must handle information from many sources choose a centralized system architecture to accomplish the computation. Using a single, central accumulation point for information can have a number of advantages for the developer:

Why centralized solutions are handy

- It is easy to know where the information is
- Many algorithms are easy to express when one may trivially map over all the data in a single operation
- There is no problem of coordination of resources—all clients simply know where the central server is, and go there

Unfortunately, such a centralized organization has two important limitations, namely *reliability* and *trust*. Reliability is an issue in almost any system, regardless of the kind of information it handles, whereas trust is more of a serious concern in systems which must handle confidential information.

Reliability

A single, central point also implies a single point of failure. If the central point goes down, so does the entire system. Further, central points can suffer *overload*, which means that all clients experience slowdown at best, or failure at worst. And in systems where, for example, answering any query involves mapping over all or most of the database in a linear fashion, increasing the number of clients tends to cause load on the server to grow as $O(n^2)$.

Because of issues like this, actual large systems, be they software, business models, or political organizations, are often divided into a hierarchical arrangement, where substantial processing is done at nodes far from any center—if there even *is* a center to the entire system. For example, while typical banks are highly centralized, single entities—there is one master database of the value of each account-holder's assets—there is not a single central bank for the entire world. Similarly, the Internet gets a great deal of its robustness from its lack of centralization—for example, there is not a single, central packet router somewhere that routes all packets in the entire network.

Trust

Of greater importance for this work, however, is the issue of trust. We use the definition of *trust* advanced in Section 1.3, namely, trust that private information will not be disclosed.

It is here that centralized systems are at their most vulnerable. By definition, they require that the subject of the information surrender it to an entity not under the subject's direct control. The recipient of this information often makes a *promise* not to disclose this information to unauthorized parties, but this promise is rarely completely trustworthy. A simple taxonomy of ways in which the subject's trust in the recipient might be misplaced includes:

How might trust be violated?

- *Deception by the recipient*. It is often the case that the recipient of the information is simply dishonest about the uses to which the information will be put.
- *Mission creep*. Information is often collected for one purpose, but then used later for another, unforeseen purpose. In many instances, there is no notification to the original subjects that such repurposing has taken place, nor methods for the subjects to refuse such repurposing. For example, the US Postal Service sells address information to direct marketers and other junk-mailers—it gets this information when people file change-of-address forms, and it neither mentions this on the form, nor provides any mechanism for users to opt out. Often, the organization itself fails to realize the extent of such creep, since it may take place slowly, or only in combination with other, seemingly-separate data-collection efforts that do not lead to creep except when combined. Indeed, the US Federal Privacy Act of 1974 [175] recognizes that such mission creep can and does take place, and explicitly forbids the US government from using information collected for one purpose from being used for a different purpose—how the USPO is allowed to sell change-of-address orders to advertisers is thus an interesting question. Note, of course, that this Act only forbids the government from doing this—private corporations and individuals are not so enjoined.
- *Accidental disclosure*. Accidents happen all the time. Paper that should have been shredded is thrown away unshredded, where it is then extracted from the trash and read. Laptops are sold at auction with private information still on their disks. Computers get stolen. In one famous case in March 1998, it was revealed that GTE had inadvertently disclosed at least 50,000 unlisted telephone numbers in the southern California area—an area in which half of all subscribers pay to have unlisted numbers. The disclosure occurred in over 9000 phonebooks leased to telemarketing firms, and GTE then attempted to conceal the mistake from its customers while it attempted to retrieve the books. The California Public Utilities Commission had the authority to fine GTE $20,000 per name disclosed, an enormous, $1B penalty that was not actually imposed [9]. In March of 1999, AT&T accidentally disclosed 1800 email addresses to each other as part of an unsolicited electronic commercial mailing; Nissan did likewise with 24,000 [26].

- *Disclosure by malicious intent*. Information can be stolen from those authorized to have it by those intent on disseminating it elsewhere. Examples from popular media reports include, for example, IRS employees poking through the files of famous people, and occasionally making the information public outside of the IRS [173]. *Crackers,* who break into others' computer systems, may also reveal information that the recipient tried to keep private. There is often significant commercial value in the deliberate disclosure of other companies' data; industrial espionage and related activities can involve determined, well-funded, skilled adversaries whose intent is to compromise corporate secrets—perhaps to do some stock manipulation or trading based on this—or to reveal information about executives which may be deemed damaging enough to be used for blackmail or to force a resignation. Intelligence agencies may extract information in a variety of means, and entities which fail to exercise due diligence in strongly encrypting information—or which are prevented from using strong-enough encryption by rule of law—may have information disclosed while it is being transmitted or stored.
- *Subpoenas*. Even though an entity may take extravagant care to protect information in its possession, it may still be legally required to surrender this information via a subpoena. For example, Federal Express receives several hundred subpoenas *a day* for its shipping records [178]—an unfortunate situation which is not generally advertised to their customers. This leads to a very powerful general principle: *If you don't want to be subpoenaed for something, don't collect it in the first place.* Many corporations have growing concerns about the archiving of electronic mail, for example, and are increasingly adopting policies dictating its deletion after a certain interval. The Microsoft antitrust action conducted by the US Department of Justice, for example, entered a great many electronic mail messages into evidence in late 1998, and these are serving as excellent examples of when too much institutional memory can be a danger to the institution.

This is hardly a complete list, and many more citations could be provided to demonstrate that these sorts of things happen all the time. The point here is not a complete itemization of all possible privacy violations—such a list would be immense, and far beyond the scope of this work—but simply to demonstrate that the issue of trusting third parties with private information can be fraught with peril.

Is this software, or a business model?

Note that the discussion above is not limited to *software* systems. Replace *algorithm* with *business practice*, *client* with *customer*, and *central server* with *vendor*, and you have the system architecture of most customer/vendor arrangements. However, we shall not further investigate these structural similarities, except to point out that business models themselves often have a profound impact on the architecture of an application.

1.6 Advantages of a decentralized solution

Decentralized solutions can assist with both reliability and trust. Let us briefly examine reliability, and consider a system which does *not* contain a single, central, physical point whose destruction results in the destruction of the system. By definition, therefore, a single, physical point of failure cannot destroy this system. This says nothing about the system's ability to survive either multiple points of failure, nor its ability to survive a single *architectural* failure (which may have been replicated into every part of the resulting system), but it does tend to imply that particular, common failure modes of single physical objects—theft, fire, breakdown, accidents—are much less likely to lead to failure of the system as a whole. This is nothing new; it is simply good engineering common sense.

The issue of trust takes more examination. If we can build a system in which personal data is distributed, and in which, therefore, no single point in the system possesses *all* of the personal data being handled, then we limit the amount of damage—disclosure—that can be accomplished by any single entity, which presumably cannot con-

trol all elements of the system simultaneously. Systems which are physically distributed, for example, multiply the work factor required to accomplish a physical compromise of their security by the number of distinct locations involved. Similarly, systems which distribute their data across multiple administrative boundaries multiply the work factor required by an adversary to compromise all of the data stored. In the extreme case, for example, a system which distributes data across multiple sovereigns (e.g., governments) can help ensure that no single subpoena, no matter how broad, can compromise all data—instead, multiple governments must collude to gain lawful access to the data.

Cypherpunk remailer chains

Cypherpunks remailer chains [10][23][66] are example of using multiple sovereigns. A remailer chain operates by encrypting a message to its *final* recipient, but then handing it off to a series of intermediate nodes, ideally requiring transmission across multiple country boundaries. In one common implementation, each hop's address is only decodeable by the hop immediately before it, so it is not possible to determine, either before or after the fact, the chain of hops that the message went through. Properly implemented, no single government could thereby compromise the privacy of even a single message in the system, because not all hops would be within the zone of authority of any single government.

Costs of a decentralized solution

Of course, as applied to the applications we examine in this dissertation, the advantages of a decentralized solution do not come for free. They require *pushing intelligence to the leaves*—in other words, that the users whose information we are trying to protect have access to their own computers, under their own control. Decentralized systems are also somewhat more technically complicated than centralized solutions, particularly when it comes to *coordination* of multiple entities—for example, how are the entities supposed to *find each other* in the first place? And such solutions may not work for *all* applications formerly handled by centralized solutions, but only for those that share particular characteristics. We will investigate each of these issues in later chapters.

1.7 A brief summary of this research

The purpose of the work in this dissertation is to demonstrate that, for a class of similar applications, useful work that requires knowledge of others' private information may nevertheless be accomplished without requiring any trust in a central point, and without requiring very much trust in any single point of the system. In short, such a system is *robust* against violations of trust, unlike most centralized systems.

The work is therefore divided into several aspects, which will be discussed more fully in the chapters that follow, and which are summarized in this section:

- An *architecture* which specifies the general class of applications for which we are proposing a solution—what characteristics are common to those applications which we claim to assist? This architecture also includes our threat model—what types of attacks against user privacy we expect, which of those attacks we propose to address, and how we will address them. — Chapter 2, Chapter 3
- A *sample implementation* of this architecture—the matchmaking system Yenta. — Chapter 4
- *Evaluation* of the sample application as deployed, an analysis of the risks that remain in the design and implementation, and some speculations on how certain other applications could be implemented using the architecture we describe. — Chapter 5
- An examination of *related work*, both with regard to privacy protection via architecture, and the sample application's domain of matchmaking. — Chapter 6

1.7.1 The architecture and its sample application

We present a general architecture for a broad class of applications. The architecture is designed to avoid centralizing information in any particular place, while allowing multiple agents to collaborate using information that each of them possesses. This collaboration is designed to form groups of agents whose users all share some set of characteristics. The architecture we describe is particularly useful for protecting personal information from unauthorized disclosure, but it also has advantages in terms of robustness and avoidance of single points of physical failure. In the description below, the architecture and the sample application described in this dissertation—Yenta—are described together.

Such an architecture assumes several traits shared by applications which make use of it, of which the most important are the existence of a peer application for each user who wishes to participate, running on the user's own workstation; the availability of a network; the availability of good cryptography; and a similarity metric which can be used to compare some characteristic of users to each other and which enables a partial ordering of similarity. The architecture derives much of its strength from its completely decentralized nature—no part of it need reside on a central server. Users are pseudonymous by default, and agents are assumed to be long-lasting, with permanent state that survives crashes and shutdowns. Individual agents participate in a hill-climbing, word-of-mouth exchange, in which they exchange messages between pairs of themselves—with no central server participating in such exchanges. Agents which find themselves to be closely matched form clusters of similar other agents. An agent which is not well-matched to a peer can ask the peer for a referral to some other agent which is a better match, hence using word-of-mouth, based on the above partial ordering of similarities, to aid in the search for a compatible group of other agents.

Once clusters have been formed, agents may send messages into the clusters, communicating either one-to-one or one-to-many. Yenta uses this capability to enable users to have both private and public conversations with each other. Particularly close matches can cause one of the participating agents to suggest that the two users be introduced, even if the users have not previously exchanged messages—this helps those who never send public messages to participate.

We carefully discuss the threat model facing the architecture and the sample application, discussing which attacks are expected and the measures taken to defend against them. We also discuss what sorts of attacks are considered outside the scope of this research and for which we offer no solution. Strong cryptography is used in many places in the design, both to enable confidentiality and authenticity of communications, and as the infrastructure for a system designed to enable persistent personal reputations. Because public evaluation can make systems significantly more robust and more secure, a separate system, named Yvette, was created to make it easier for multiple programmers to publicly evaluate Yenta's implementation; Yvette is not specialized to Yenta and may be used to evaluate any system whose source code is public.

1.7.2 Evaluation

The architecture and the sample application have been evaluated in several ways, including via simulation and via a pilot deployment to real users. The qualitative and quantitative results obtained demonstrate that the system performs well and meets its design goals. In addition, several other applications which might make use of the underlying architecture are possible and speculations on how they might be implemented are briefly described. We also perform a risk analysis of Yenta and describe potential security risks, including some which are explicitly outside of our threat model.

Finally, we describe related work, which includes other types of matchmaking systems, other decentralized systems, and other systems and software that have been designed for explicitly political purposes. We then draw some general conclusions.

1.8 Summary

This chapter has presented the social and political motivations for this work, namely the protection of certain civil liberties, such as privacy, by starting with such motivations and then designing technology that can help. We have described what personal privacy and its protection means, demonstrated some of the social, political, and technical problems with centralized solutions, and touched upon some of the advantages of decentralized solutions. We have then summarized, very briefly, the work that will be presented in later chapters.