6.857 Project Draft

Jacky Chang Geoffrey Thomas
Christian Ternus Ian Ynda-Hummel

April 8, 2010

Abstract

With the advent of ubiquitous social networking, the problem of controlling one's personal information has become more important than ever. Many users of social networks would like to be confident of who has access to their personal information and data. The most popular social networks — Facebook, Twitter, MySpace, and so forth — rely on a single company running the network that users may not quite consider a *trusted* third party. Switching to a distributed model can fix the problem of a central party having access to a large amount of aggregate data, but it does not necessarily limit who can view a user's data or unintentional information leakage.

Another way to better control the flow of information is to allow users to display different sets of information to different people. If users have the option of giving different sets of people access to different sets of information, they can be more confident that details that they may want to share with one group will not be leaked to another.

The final concern is with creating these friends lists. One of the major sources of data leakage in social networks is the shape of the networks themselves. If an attacker knows which nodes are connected to each other, he can infer data about one node from what he knows about the others. In order to avoid this, users need to have some way of connecting to other users without revealing their own network connections to each other. This concern must be balanced with the fact that some amount of information needs to be exchanged to prove identity ("I am a friend of a friend") in order to ensure that a malicious user is not disguising himself as a legitimate one in order to gain access to privileged information.

We plan to incorporate these ideas into a way of designing a social network that safeguards its users' private information.

Project Plan

We propose to design a method of operation of a social network that satisfies three goals:

- Although the system needs to be distributed to avoid a single third party, it should be sufficiently **hard to data mine** the social network; for instance, requests for an unusual amount of data should be obvious to the people providing this data.
- Users should be able to present **multiple sub-identities** that cannot be identified as related unless the user specifically authorizes this.
- Since we lack a single central authority, we need a trust metric like PGP's web of trust, but the entire goal of this network is to avoid publicizing friend relationships at the outset – we need a **dark web of trust** that requires revealing as little information about the friend graph as possible.

We believe these properties are necessary for creating a social network that is designed to guard its users' privacy, and we believe they are nearly sufficient for designing a good system, although we expect part of the main work of this project to be expanding on and clarifying these constraints.

The first goal, of making data mining hard, is difficult because we would like to avoid any centralized authorities. Therefore, we cannot rely on a single source that meters requests made by a user. One idea towards a solution that our team has discussed involves having the requestee publishing an anonymous but signed note identifying the requestor; future requestees can notice a high number of these notes and deny requests for information. To make this idea workable, we would need to figure out how to publish these notices without too much performance impact and how to authenticate them: it is possible that the answers to the second two goals will address authentication. We would also need to ensure that publication of these notices does not compromise their senders' privacy.

We plan for our system to let users explicity designate different modes of identity. For example, a user might have a personal and work profile, both of which have a distinct set of friends and thus channels of information dispersion. One possible direction is to consider the possibility of a “super user” which delegates to different profiles. The process by which a user creates a friend relationship with another user — discussed below — is largely based around the sharing of some piece of information. This information acts as a gateway to initialize friendship. A user can have multiple such gateways, and share different gateways to different people. Each gateway, then, would map to a distinct node in the graph, connected to the “super user” node. Furthermore, we must create this network so that any other identities connected to this “super user” node are not leaked through the friend relationship with one of them. For example, we must take into consideration the pitfalls of allowing global distribution of information to friends. Consider the case where a personal friend and a business friend each get the same provably real message. This message could leak information about the connection of identities if these friends collude in some way.

We will also need to address the issue of estabilishing a ”trust” metric in the network that is visible to all users, but does not leak information about the social graph. Since there is no way to authenticate users’ claimed names, the standard technique for checking that a friend requestor is who he says he is is to check his profile, which includes a list of the people he currently has a friend relationship with. In fact, Facebook automatically makes more information about you visible to the recipient after you send a friend request or a message. Our system will need to let users specify explicitly what information they want to provide in a friend request and in other actions requiring authentication.

The most useful such information is in fact the identities of *all* of your friends — to be precise, all of your subidentities’ friends — but this is obviously at odds with the entire premise of the network. Ideally, we would find or develop some sort of cryptographic signing mechanism that avoids identifying who did the signing, but only gives hints at who has signed and vouched for the person doing the signing. This voucher itself would also be anonymous, with the recursion terminating only at the verifier. As a concrete example, if Alice and Bob are not friends and wish to verify each other’s identity, and they have five mutual friends whom they trust and who trust them, Alice should be able to know that she has five trust paths to Bob without implicitly learning which five friends Bob also knows.

We expect determining whether this is possible (potentially under a restricted definition) and how best to do so to be a major part of our project. Should this be doable in the general case, it provides a number of useful specific cases: for one, it will extend to providing a way to convert trust for a single subidentity into trust for all subidentities and the super user, because super user nodes are usually not public or connected. It will also nicely turn into a mechanism for providing anonymous but publicly-trustworthy notifications for the data mining protection outlined above.

Our belief is that a system that provides these three abilities will make a large number of privacy desiderata possible. Certainly such a system would satisfy our primary goal of not having a trusted party who has access to all the data. By providing a trust system, it also prevents attackers without access to secret resources — either the user's private key, or the existing friend relationships of that user — from impersonating other uses, and the use of a public key-style infrastructure that does not identify friend relationships should prevent individual nodes from identifying who is who. By allowing the creation of nonce subidentities that retain their trust value, it becomes easy for legitimate users of the network to anonymously publish or anonymously view information. And by requiring users to explicitly designate what personal information they want to publish to socially identify themselves to other users, we address privacy concerns like Facebook's where information is unknowingly made more public than intended.

Background Information

There is a significant problem of identity on existing social networks. In our daily life, we make the distinction between different modes of interaction. The fact that someone goes out drinking with friends on the weekend is often irrelevant with regards to his work life, for example. It is hard, if not impossible, on the current social network model to distinguish different sets of access controlled data.

For instance, college students might prefer it if their parents did not find pictures of last night's party – or, on a more serious note, if their information wasn't available to marketing corporations for use in ads, as Facebook recently discovered might be a problem.

Furthermore, information may be leaked in a social-networking scenario which the users never intended to put online. There was a particularly notable case a while back where Netflix provided supposedly-anonymized data about its userbase – part of a contest to see if programmers could improve the accuracy of Netflix's recommendation algorithm. Soon afterwards, users discovered that they could be uniquely *reidentified* by their information, and that this had been used to "out" the sexual orientation of several Netflix members against their will – simply based on the movies they had watched.

One way to avoid trusting the third-party network maintainer to keep your data private is to use the social network only as a network connection, and have end-to-end encryption of content and information between users. This ensures that protected content cannot leak because of the general public can never see protected content because of programming or design flaws. For increased scalability and to defend against some easy attacks, we can make this a distributed system, e.g., place each content item in a large distributed hash table (DHT).

There are a couple of issues with such an implementation. First, the system should be able to pass metadata around, so that a user's client can see a home screen with all of the information on current social networks. This would require leaking some information on connections to the network, and we would want to limit how much information about friend relationships can be recovered from watching this traffic. The system also needs to have a way to defend against offline attacks and the "Sybil attack", in which an attacker creates and controls a large number of fake profiles and abusing their aggregate trust.

These thoughts and concerns have motivated our proposed design constraints for our system. We are looking at a couple of existing work in data privacy for solutions, for instance, Lysyanskaya et al.'s work on pseudonym systems so that users can construct "nyms", such that two nyms cannot be correlated without the help of the user ownning the nyms, and Sweeney's "k-anonymity" model, which expands on the problem of reidenitification and possible solutions to appropriately anonymizing data sets such that they cannot be reidentified.