Establishing Distributed Hidden Friendship Relations

(Transcript of Discussion)

Sören Preibusch
University of Cambridge

Welcome to the penultimate talk of the workshop. I will be talking today about hidden friendship relations, and I will introduce you to the concept of what hidden friendship relations are, say why we need them, and I will also explain how we can achieve them.

I will do so with the perspective on privacy, and focusing on distributed architectures. What has come up during the past few days is social networking in all of the scenarios that we have seen, and all the applications that can be run on these, and experiments we can do in social networks. Here you can see a screenshot of social network, it’s a profile page for Friendster, and besides the fact that there’s this nice guy, we have his friends over here, and friendship relations are the underlying relations that build a social network, so it can be used as a core to build higher level concepts. It’s simple, but it’s formalised, you can easily crawl it and so forth. There’s also an interesting function property about friendship, that it’s symmetric, so here we have Gwen, and if we browse to Gwen’s profile page and look up her friends we see that Hennesey is again there. So we have the symmetry that is enforced across the network, that’s an important property about friendship relations. Another thing is that they carry the property, to allow privileged actions, so if I have someone, you look over there to the left you can see that you can forward the profile page to one of your own friends but you can’t forward their profile page to other friends. So this privileged action that is based on friendship can also have privileges that are with regard to resources such as hiding some things in the profile. This is what I will call the positive privacy of friendship relations.

Anyway we can see that friendship along with other relations between users is used to enforce some access control. And how this then results on the user requesting a profile page, you can see it on the right that this profile has been restricted, so it’s private, and as long as you’re not a friend to this user you can’t see the details on what he’s doing, his contact information, his activities, and so forth. So that’s good.

However, there are what I will call the negative privacy of friendship relations, that there are friends you want to be friends with, but you don’t want to be seen with these friends in public, that is because friendship relations carry also the understanding that social demographic information propagates along friendship links. There is a good analogy in the offline world that is the so-called red-lining, so if your neighbour doesn’t pay his bills then you have a bad credit rating. We can think of this social networking also as red lining on a social graph, that
if your friends have properties that aren’t socially acceptable, then this will have detrimental consequences for yourself. So if your friends are caught binge drinking then you might not get a job interview. An effect only last week was this, in the news, so that a prison guard was sacked because he was friends on Facebook with some of his inmates. So it would be good to be able to hide some friends, but still maintain the good things about friendship relations, that is, you want to have friends, and you want to show that you have friends, and you maybe want to show off with the number of friends you have. So that’s why we need to be able to selectively hide friends, and purge people we don’t want to be seen with in public from our list of friends.

We have a centralised architecture such as the Friendster network, or any other network, such as Facebook, MySpace and like in western Europe and the Americas, then we see that it’s in principle very easy to hide friendship relations links selectively, because the network operator has strong credentials on who is requesting a list of friends, he has the login information this user is requesting to this provider with regard to the site. So if an unauthorised user is making a request then another list of friends will be served than if an authenticated user is requesting a list of another user’s friends. Even so, implementation isn’t always easy, security in theory doesn’t mean it will work in practice. Because of this symmetry I’ve mentioned earlier there is a trap that unilateral disclosure friendship links may leak information, so if A is friends with B, and B wants to conceal this information but A makes it public, that means because of the symmetry we will know that B is also friends with A, so we would in fact need to hide both ends of the friendship relation. And some networks do not get it right, as we found out in a study on the existing medium size German social network, we’ve been seeing that we can infer lots of the friends.

To quantify this we had the exposure rate you’re seeing: if we have all of the users who hide their friends, how many of these users have at least one friend exposed through incoming friendship links, that means that you have at least one privacy breach, and the answer is two-thirds. So there aren’t many users who hide their friends. This network comprised 120,000 users, some of them more active than the other ones. Further pushing technology on this network, the feature to make a public list of friends was introduced as was an opt-out feature. The most active users opted out more often, but still they’ve been in a bad position because people who they’ve been friends with for a long time didn’t change their settings and there are incoming links towards them. So two thirds of them did not succeed in enforcing their privacy well, and often it was not only one friend that was exposed, but nearly all of them. The in-friends ratio, the degree to which we can infer the complete list of friends, we can see it’s almost all the friends that we can infer by inbound links only. So not paying attention to the symmetry of friendship links turns out to be quite bad if we really want to enforce privacy. Still, let’s keep in mind it works in theory, which will not be the case if we move towards decentralised architectures.

The reason I want to consider decentralised architectures is because mobile is the future paradigm of social networking, and if we do mobile then doing