Strong cryptography for distributed social networks

Geoffrey Thomas geofft@mit.edu

6.857 Project Proposal

One common theme that arises in social networking situations is that of data privacy: if I share my home address, or pictures of myself that I may not want to show to the whole world, how do I guarantee that this information won't later be shared among other people? Multiple incidents — Facebook's Beacon and its attempts at restructuring privacy controls and Google's Buzz and Google Profiles come to mind – have shaken the public's confidence in storing their data on a social network's central server, entrusting access control to just the correct coding and the beneficent decisions of the social network's developers.

One way to avoid trusting the third party is to use the social network only as a network connection, and have end-to-end encryption of content and information between users. For instance, if I only want a photo to be available to certain people, I would upload an encrypted photo (with metadata) to the network, and only share the encryption key with the people who should see it. This ensures that the general public, or unauthorized users, 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 DHT.

There are a couple of issues with such an implementation. First, you'd have to figure out how 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, since a client-side search for all new content, checking if any of that content is decryptable, will quickly become impractical.

Second, you start permitting offline attacks against the data set. Since the network doesn't know if you're authorized to access an item, you can download it and spend arbitrarily long breaking it locally. More interestingly, the network may want to restrict how much public information a user can request at once, to defend against inferring non-public information, but this is harder in a distributed system. Finally, if users are anonymous to the central system, the network needs to protect itself from the "Sybil attack", where a large number of accounts beloing to a single human user break various probabilistic guarantees (and rate limits).