LoKI: Location-based PKI for Social Networks

Randy Baden
University of Maryland
randofu@cs.umd.edu
http://www.cs.umd.edu/~randofu/loki

Categories and Subject Descriptors

C.2.0 [**Computer Communications Networks**]: General—*Data communications*; C.2.1 [**Computer Communications Networks**]: Network Architecture and Design—*Wireless communication*; C.2.4 [**Computer Communications Networks**]: Distributed Systems—*Client/server, Distributed applications*; C.5.3 [**Computer System Implementation**]: Microcomputers—*Portable devices (e.g., laptops, personal digital assistants)*; E.3 [**Data Encryption**]: [Public key cryptosystems]

General Terms

Security, Design, Performance

Keywords

Public Key Infrastructure, Online Social Networks, Location, Mobility

1. INTRODUCTION

The existence of a public key infrastructure (PKI) is a linchpin of many systems that provide security, privacy, or accountability [2, 3]. For many systems, there are adequate solutions to providing a PKI that, while never perfect, are resilient to attack. Certificate authorities (CA) are a common solution, but they have limitations. Obviously, the CA must be trusted, a reasonable assumption in many systems, especially centralized ones. Less obviously, the CA must be able to independently verify the identities of those principals that the CA certifies.

For this reason, a trusted CA is at best an insufficient solution to the PKI problem for a decentralized online social network (OSN). The problem is not one of trustworthiness; a CA such as Verisign could be just as scrupulous and well-intentioned whether they certify web sites or people. The problem is a practical one: it would require prohibitively many resources for a central authority to independently verify every social identity in the system.

This problem is not new, but the changing landscape of how users interact with social networks — and with each other — offers opportunities for new, complementary solutions. The PGP web-of-trust is a good starting point, effectively making each user her own certificate authority. This matches the notion of a user being in charge of her own privacy and security within her *domain* of the social network, a primary guiding principal in decentralized OSNs. The out-of-band exchange of keys, however, has proven too onerous for typical users [7]. We therefore choose to design techniques that users are able to more readily employ based on how users interact with each other in modern settings.

Our main contribution is a system, LoKI, in which we use the ubiquity of mobile devices to provide users with a new method for verifying identities that does not require immediate user interaction. Concretely, we propose collecting shared secrets from nearby mobile devices over the course of typical mobile activity, then using these shared secrets post-hoc to perform identity verification based on user recollection of when real-world meetings occurred. We estimate the frequency of real-world meetings among social network users with a data set of interactions recorded by Foursquare, Facebook, and Twitter. We evaluate the technical constraints of collecting and storing shared secrets in terms of storage space and power consumption based on the frequency of mobile device encounters. Finally, we provide a solution to allow peer-to-peer bluetooth communication on mobile devices when neither device is able to enter discoverable mode, such as in a background service on an Android phone that has not been rooted. We believe that this problem of peer-to-peer rendezvous on typical mobile devices is an important consideration that has heretofore gone unrecognized in research on mobile peer-to-peer systems.

2. DESIGN OVERVIEW

We design LoKI with one goal in mind: to transform real-world social interactions into secure online public key exchanges. We realize this goal using shared secret data frequently exchanged between proximal devices, where an exchange of secret data represents a real-world meeting. This secret data can later be used to establish a secure channel based on the user's recollection of when real-world meetings occurred [1]. This basic design goal requires proximal devices to be able to communicate privately without user intervention; though this seems like a simple requirement, a number of practical constraints make this non-trivial in practice.

2.1 Assumptions and Constraints

We assume that two users who wish to be friends on an OSN meet in person while in possession of their mobile devices; though not every pair of OSN friends will fall under this assumption, we will show that many do. We also assume that these devices have wifi and bluetooth capabilities along with some means of accessing the Internet. We assume that every user runs our key exchange application at all times, so we restrict the key exchange application to only be able to access features available to non-

SIGCOMM'11, August 15–19, 2011, Toronto, Ontario, Canada.
ACM 978-1-4503-0797-0/11/08.

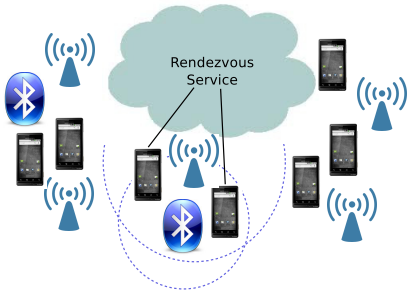

Figure 1: Non-rooted androids cannot observe each other, but they can observe similar nearby wifi access points and discoverable bluetooth devices to detect proximity with the help of a rendezvous service.

rooted devices. Otherwise, a user would have to void the warranty on the device.

Bluetooth's range of about 10 meters is a physical means of verifying locality, making bluetooth the intuitive communication channel for our purposes. If two devices are able to communicate via bluetooth, it most likely means that the device's users are near each other; this may indicate social interaction between the users that is reflected in the social graph in an OSN. Bluetooth is also an appealing choice because it requires less power than alternative communication technologies.

We focus on the case of "disengaged" users[1], i.e., users who do not actively establish OSN friendships during a real-world meeting, but later seek to establish the OSN friendship when they have left the presence of the other user. In particular, it is not possible to put an android device into bluetooth discoverable mode without periodically requesting permission from the user. However, bluetooth communication is still possible as long as once of the devices can learn the other device's bluetooth MAC address.

2.2 Rendezvous

We considered many strategies for advertising bluetooth MAC addresses using broadcast or multicast over either bluetooth or wifi. None of these solutions were viable because there is no way to access broadcast packets from non-rooted phones. Our ultimate solution relies on a third-party rendezvous service, analogous to the role of a STUN server in NAT hole-punching.

Given a request from a client containing spatiotemporal data and information about the client's bluetooth MAC address, the rendezvous service matches that data with other requests and return the data necessary to construct nearby users' bluetooth MAC addresses. Ideally, another user should only learn the user's MAC address if that user is sufficiently close, i.e., within bluetooth range.

Narayanan et al. [5] describe a set of possible location tags that can be used to confirm device location. Unlike most of their suggestions, the list of visible wifi access point MAC addresses and the list of discoverable bluetooth device MAC addresses are both available to non-rooted phones. Our setting is depicted in Figure 1. Visible MAC addresses provide a way to match based on location, though these values do not depend on time, so an attacker who visits a location once will be able to forever after check to see which devices are in that location. We believe it may be possible to extend the android API to report information

[1]The case of engaged users requires trivial technical solutions, though it remains part of the complete OSN PKI bootstrapping solution.

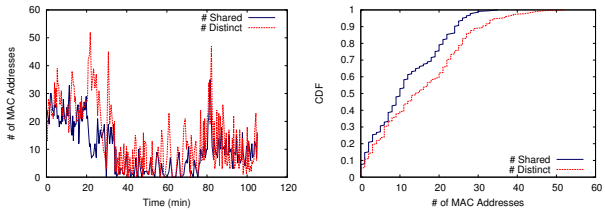

Figure 2: The number of common and distinct MAC addresses (wifi and bluetooth) visible from two colocated Motorola Droids while walking through the University of Maryland campus.

about the timestamps from beacon frames during a wifi scan to incorporate temporal dependence into the rendezvous service.

Complicating matters further, two colocated devices have remarkably different views of visible MAC addresses, as shown in Figure 2. We therefore consider two phones to be likely colocated during epoch t if they agree on some threshold k of visible MAC addresses.

We now describe what the clients communicate to the rendezvous service. First, if the client does not already know the current (publicly known) epoch t, it requests it from the rendezvous service. Let B be the client's bluetooth MAC address. Let $H(\cdot)$ be a one-way hash function, $Enc(\cdot,K)$ be a symmetric encryption function using key K, and let $S(\cdot)$ be a function that computes a secret share according to Shamir's secret sharing [6]. For each visible MAC address M, the client computes and transmits:

- The location tag: $H(M\|t)$
- The secret share: $Enc(S(B,M\|t,k),M\|t)$

The location tags can be used by the rendezvous service to match users who share at least k visible MAC addresses through set intersection. The secret shares can be decrypted and combined to reconstruct B by anyone who knows at least k matching visible MAC addresses, i.e., someone present at the location.

Once one of the pair of devices has B, it can establish an insecure channel and perform Diffie-Hellman key exchange to establish a secure channel over which a secret can be agreed upon. These agreed-upon secrets can thereafter be used to perform SPEKE [4] and exchange public keys with the certainty that the public key belongs to a user who was present at the times that the secrets were collected.

Since an attacker cannot selectively block bluetooth transmissions, the Diffie-Helman exchange is not vulnerable to a classical man-in-the-middle attack, though it can be vulnerable to impersonation; the attacker cannot hide the honest user, but she can create an indistinguishable duplicate. One defense against this attack is to require secrets from multiple meeting times, enough to ensure that no other user could be present at every meeting.

3. REFERENCES

[1] R. Baden, N. Spring, and B. Bhattacharjee. Identifying close friends on the internet. In *HotNets*, 2009.
[2] R. Baden, *et al.* Persona: An online social network with user-defined privacy. In *SIGCOMM*, 2009.
[3] A. Haeberlen, P. Kouznetsov, and P. Druschel. Peerreview: practical accountability for distributed systems. In *SOSP*, 2007.
[4] D. P. Jablon. Strong password-only authenticated key exchange. *SIGCOMM CCR*, 1996.
[5] A. Narayanan, *et al.* Location privacy via private proximity testing. In *NDSS*, 2011.
[6] A. Shamir. How to share a secret. *Commun. ACM*, 1979.
[7] A. Whitten, J. D. Tygar, A. Whitten, and J. D. Tygar. Usability of security: A case study. Tech. rep., CMU, 1998.