Location Services with Built-In Privacy

Arvind Narayanan
Stanford University

Joint work with Narendran Thiagarajan, Mugdha Lakhani, Dan Boneh
Location-based social networking

Finally taking off?
EFF, tech companies lobbying for ECPA revision

Why do service providers care? **Positive externality**
What can we do privately

Proximity testing: detect when friends are nearby

When not nearby, friends don’t see your location

Server never sees location

Building block for more complex functionality
Proximity testing: some applications

Granularity must be user-configurable
<table>
<thead>
<tr>
<th></th>
<th>All-pairs</th>
<th>Friends-only</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client-server</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Peer-to-peer</td>
<td>✓</td>
<td>✗</td>
</tr>
</tbody>
</table>

Only client-server model supports configurable granularity

Poor/nonexistent infrastructure for complex peer-to-peer protocols
Mathematical formulation: not obvious

“Pairs of friends get notified whenever they are within 100ft of each other”
Reducing proximity testing to equality testing
Equality testing

Space of possible locations is small!

ElGamal-like cryptographic protocol based on Decisional Diffie Hellman (DDH) problem (Lipmaa)

Improved constant factor

Requires shared secret keys between pairs of friends
Server participation

Server can pretty much learn everyone’s location

\[
ax + b \quad \rightarrow \quad x
\]

\[
ay + b \quad \rightarrow \quad y
\]
Server participation done right

Server can cause users to compute wrong answer but cannot cause privacy breach

Avoids need for big integer arithmetic
Information-theoretic security

\[ s(x-y) \]

\[ s(x-y) \]
Problem: online brute-force attack

If only there were a way to verify that a user really is where they claim to be...
Location tags

Shared entropy pool
Properties of location tags

Location tag = vector + matching function
i.e., space-time fingerprint

Unpredictability
cannot produce matching tag unless nearby

Reproducibility
two devices at same place & time produce matching tags (not necessarily identical)
Location tags using WiFi packets

Discard packets like TCP that may originate outside local network
  – DHCP, ARP, Samba etc. are local

15 packets/sec on CS/EE VLAN

Two different devices see about 90% of packets in common

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Device 1</th>
<th>Device 2</th>
<th>Common</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARP</td>
<td>1088</td>
<td>1071</td>
<td>832</td>
</tr>
<tr>
<td>BROWSER</td>
<td>262</td>
<td>286</td>
<td>255</td>
</tr>
<tr>
<td>DHCP</td>
<td>249</td>
<td>237</td>
<td>208</td>
</tr>
<tr>
<td>MDNS</td>
<td>600</td>
<td>551</td>
<td>541</td>
</tr>
<tr>
<td>NBNS</td>
<td>1134</td>
<td>1190</td>
<td>1117</td>
</tr>
<tr>
<td>All</td>
<td>3333</td>
<td>3335</td>
<td>2953</td>
</tr>
</tbody>
</table>
Location features

Each packet is a “location feature”

Timing, source/destination and other packet contents

At least around 10 bits of entropy

Tag with 15 location features gives > 80-bit security level
Comparing location tags

Need to compare two vectors that match approximately: **fuzzy set intersection**

Basic concept:
Alice encodes vector as polynomial
Sends random points on polynomial to Bob

Intersection size is large ➞ few enough “errors” ➞ Bob can decode using Berlekamp-Massey algorithm
Shared secret keys

Traditional solution: PKI

PGP (un)usability study

Better solution: Identity-based encryption

Our solution: bind public keys to social identities
My Facebook profile

Website:

http://www.cs.utexas.edu/~arvindn/
http://33bits.org/
http://arvindn.livejournal.com
http://twitter.com/random_walker
http://randomwalker.info

https://socialkeys.org/pubkey?alg=DH&keylen=1024&p=oakley&g=2&key=LIII+IKCAIEHmjbAwTLSSj6EnbXG1w9NYp5msV7DbuPsteg2t3PJ1tSPYwjlqLPxjrbxZJe/FJwttbUf9Wf8Re7eZg4NVf
Android implementation

Preferences

Proximity Threshold
How far to check

Update Frequency
How often to check
Android implementation
Remember the password and the below words to recreate key in future

fidgets, particle, self-protection

Generated Public Key is https://socialkeys.org/pubkey?alg=DH&k...

This key is copied to clipboard. Click the profile button and add the string to your website info

Goto Facebook Profile

Verify Public Key
Android implementation
Other location privacy questions

Location based advertising

Location based search

Location statistics
Summary

Proximity testing: useful primitive, tricky to define!

Improve constant factor in crypto protocols for Private Equality Testing

Location tags to enhance location privacy

SocialKeys: transparent crypto via key sharing over social networks
Thank you