SMILE: Encounter-Based Trust for Mobile Social Services

Justin Manweiler
Ryan Scudellari
Landon P. Cox
Mobile Social Services on the Rise
Trust in Mobile Social Services

- Lessons from Online Social Networks
  - Inadvertent data leakage
  - Exposure by hackers
  - Abuse by system administrators

- Location privacy is particularly sensitive
  - Why trust service providers?

- Root-of-trust should match the application
  - End users may be more appropriate
  - Must we pre-define all social network relationships? (limits applications, e.g., Social Serendipity)
Outline

◆ Security in “Mobile Social Services”

◆ An example application: “Missed Connections”
  ❖ Opportunities
  ❖ Pitfalls

◆ Our approach, relocate root-of-trust
  ❖ Remove the need for a trusted third party
  ❖ Novel protection mechanisms
  ❖ Rely on peers, but without pre-established relationships

◆ Comparison to current practice
  ❖ Suitability of our techniques to human behavior
What is a Missed Connection?

- How can Alice contact Bob?
- Not just for dating
  - Retrieve left-behind items
  - Discussion after a lecture
  - Lost phone number
  -...

Diagram showing two individuals with speech bubbles and a train.
Is there really a demand?

- Plenty of online missed connections
  - Craigslist, Gumtree, I Saw You, ChancedIt, Qwaro...

- Social networking: a growing industry
  - Projected $7.3B by 2013 [Juniper Research]

- Online dating
  - Third-largest revenue of all online paid content
    (after music, gaming) [Forrester Research]

- Dating as a Mobile Social Service
  - “Mobile dating” $1.4B by 2013 [Juniper]
Indirect Approach: Craigslist

“Hi, I’m Bob!”

“We saw each other on the subway…”

craigslist
Chicago listings

Difficult!
Many security flaws
Mobile “Missed Connections”

Impersonation

Alice @ {Subway, 1:00 PM}

Bob @ {Subway, 1:00 PM}

Service Provider

Internal Abuse

Snooping, De-anonymization

“Remember me?”

Trustworthy?

Leaked/Stolen Data

Bob @ {Subway, 1:00 PM}

Alice

Subway 1:00 PM

Bob

Subway 1:00 PM

…”
Secure Missed Connections

- **User-to-user matching**
  - On the basis of *encounters* (incidents of co-location)
  - Encounter event provides the root-of-trust

- **Preserve participant privacy**
  - Location privacy  =  *Where have I been?*
  - Encounter privacy  =  *Whom have I met?*

- **Establishment of a secure messaging channel**
  - Message anonymity/confidentiality/integrity
  - Verifiable encounter proofs
  - Unobservable to *any absent party*
SMILE

Secure
Missed Connections through Logged Encounters
High-level Approach

- Create “shared knowledge” of encounter
  - Peers establish cryptographically-secure secrets
  - Difficult for absent adversaries to guess
  - Secret never revealed to any third party
Key Techniques

◆ **Passive key exchange**
  - Short-range wireless for co-located devices
  - Enables encrypted communication (shared key)

◆ **Key hashes**
  - Hashed shared key provides logical rendezvous
  - Can be (partially) revealed to the server
  - Server never sees locations (location privacy)

◆ **Collisions in the key-hash domain**
  - Hash collisions at server (induced by participants)
  - End-users resolve ambiguities with original keys
Passive Key Exchange

Encounter Key
$= \text{Key}_A + \text{Key}_B$

Keys create shared knowledge
Periodic, no explicit user interaction
Only known to co-located devices
Keys and Key Hashes

Service Provider

| Alice | $H(K_A+K_B)$ |
| Bob   | $H(K_A+K_B)$ |

Protects location privacy
End-to-end secure channel

What about encounter privacy?

| Alice @ $H((K_A+K_B))$ | Alice | $H(K_A+K_B)$ |
| Bob @ $H((K_A+K_B))$ | Bob   | $H(K_A+K_B)$ |

{“Hi...”}$^{KA+KB}$

| K_A+K_B | Subway, 1:00 PM |
| K_X+K_Y | Lecture, 8:00 AM |

H($K_A+K_B$)

H($K_R+K_S$)

| K_A+K_B | Subway, 1:00 PM |
| K_R+K_S | Store, 7:45 PM |
Key-hash Collisions

- Link between encountered peers clear
  - Shared key-hash identifies Alice/Bob pair

- Encounter privacy guards user interactions
  - Protect whom Alice encounters, messages
  - Idea: induce hash *collisions* at service provider
  - Send a key-hash *prefix*, instead of complete hash
Key-hash Collisions

Only Bob can decrypt

Alice
@Pre(H(K_A+K_B)); {"Hi..."}

Bob
@Pre(H(K_A+K_B)); {"Hi..."}

Chris @Pre(H(K_C+K_D))

Dave @Pre(H(K_C+K_D))

Pre(H(K_A+K_B)); {"Hi..."}_{KA+KB}

Only Bob can decrypt?

Pre(H(K_A+K_B)) = Pre(H(K_C+K_D))

<table>
<thead>
<tr>
<th>Alice</th>
<th>Pre(H(K_A+K_B))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob</td>
<td>Pre(H(K_A+K_B))</td>
</tr>
<tr>
<td>Chris</td>
<td>Pre(H(K_C+K_D))</td>
</tr>
<tr>
<td>Dave</td>
<td>Pre(H(K_C+K_D))</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>H(K_A+K_B)</th>
<th>K_A+K_B</th>
<th>Subway, 1:00</th>
</tr>
</thead>
<tbody>
<tr>
<td>H(K_X+K_Y)</td>
<td>K_X+K_Y</td>
<td>Lecture, 8:00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>H(K_A+K_B)</th>
<th>K_A+K_B</th>
<th>Subway, 1:00</th>
</tr>
</thead>
<tbody>
<tr>
<td>H(K_R+K_S)</td>
<td>K_R+K_S</td>
<td>Store, 7:45</td>
</tr>
</tbody>
</table>
Key-hash Collisions

◆ Trading communication overhead for privacy
  ❖ Must select appropriate prefix length (difficult)

◆ Optimal prefix length depends on...
  ❖ Desired privacy level ($k$-anonymity)
  ❖ Number of users and messaging rate (utilization)
  ❖ User mobility (encounter frequency, distribution)
  ❖ Acceptable messaging latency (timing attacks)

◆ In paper: model of privacy/overhead tradeoff
Evaluation

- **Analytical model of privacy/overhead (paper)**

- **Craigslist missed-connections survey**
  - Wanted to characterize typical behavior
  - Looked at 100 posts from 8 US cities
  - Manual content classification

- **SMILE implementation**
  - Can we detect encounters fast enough?
Types of Manual Checks

- **Personal Description**: Often guessable
- **Things Present**
- **Memorable Event**: Observable to co-located adversary
- **Time or Place**
- **Other Observable**
- **Privately Shared**

<table>
<thead>
<tr>
<th>Type of Confirmation Requested</th>
<th>% Classified Encounters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Personal Description</td>
<td>40%</td>
</tr>
<tr>
<td>Things Present</td>
<td>15%</td>
</tr>
<tr>
<td>Memorable Event</td>
<td>10%</td>
</tr>
<tr>
<td>Time or Place</td>
<td>5%</td>
</tr>
<tr>
<td>Other Observable</td>
<td>5%</td>
</tr>
<tr>
<td>Privately Shared</td>
<td>5%</td>
</tr>
</tbody>
</table>
Encounter Distance

- **Intimate**: < 1m
- **Nearby**: 1-5m
- **Around**: 5-10m
- **Distant**: > 10m

% Classified Encounters

**Bluetooth range**
Encounter Duration

- Instantaneous (<15 seconds)
- Short (15 seconds – 1 minute)
- Extended (>1 minute)

% Classified Encounters

<table>
<thead>
<tr>
<th>Location</th>
<th>Instantaneous</th>
<th>Short</th>
<th>Extended</th>
</tr>
</thead>
<tbody>
<tr>
<td>SF Bay Area</td>
<td>10</td>
<td>20</td>
<td>70</td>
</tr>
<tr>
<td>Atlanta</td>
<td>10</td>
<td>20</td>
<td>70</td>
</tr>
<tr>
<td>NYC</td>
<td>10</td>
<td>20</td>
<td>70</td>
</tr>
</tbody>
</table>
Bluetooth Detection Accuracy

% Detected Encounters

Encounter Duration (seconds)

- No Detection (failure)
- Partial Detection (sufficient)
- Full Detection (bidirectional)

Realistic duration

Two devices running SMILE, each pausing 15s between scans
Related Work

◆ **Location proofs**
  - Saroiu and Wolman (HotMobile ’09)
  - Lenders et al (HotMobile ’08)
  - Relies on external infrastructure

◆ **Local key exchange**
  - SPATE (Mobisys ’09)
  - Assumes pre-established user trust

◆ **Location privacy and mobile services**
  - Adeona (USENIX Security ’08)
Limitations

- Difficult to estimate optimal prefix length
  - Requires some difficult-to-acquire parameters

- Incomplete evaluation of system impact
  - Energy, CPU performance on device

- Social impact
  - Risk of co-located adversaries (e.g., stalking)
Conclusion

◆ Efficient missed-connections service
  ◆ Convenient user-to-user messaging service

◆ Non-centralized root-of-trust
  ◆ Rely on encounter co-location event
  ◆ No pre-established trust relationships

◆ Key hashes and key-hash collisions
  ◆ Novel mechanisms for location privacy, anonymity
Thank you

Questions?
http://www.cs.duke.edu/~jgm/