Security Management for Mobile Peer-to-Peer Systems

Christian Rohner
Contents

- Introduction
- Security for Networked Devices - Motivation
- Project Outline
- Preliminary Results
- Project Administration: Questions
Mobile Peer-to-Peer Systems

- Variety of specialized devices
- Devices offer their functionality as services
- Spontaneous interaction

Challenges
- many devices, from big to small
- no central services
- individual users
Security for Mobile Peer-to-Peer Systems

- Personal and private information
- Limited resources
- Open and unknown environment

Goal: Control access to information and resources
- Access control
- Encryption if necessary
- Privacy (including meta-data such as location, etc.)

- Security Mechanisms are available, but ...
Background - Building Secure Links

Initially required channels: \( O(n^2) \)
Not all relations are needed

Using trust

Initially required channels: \( O(n) \)
> Trusting an entity to send only authenticated information
> Trusted entity must be available
Background - Authorization

On-line authorization server:
Involved in every interaction

Get credentials in advance
(e.g., Kerberos)
Server must be available
Challenge - Administration

Administrative domain shifts to the user
Security protocols are complex to configure
Project Vision - Self Configuration

Build **security relations** to simplify self configuration of other protocols.

- **Federations**
  - Intuitive bootstrapping
  - Include weak devices
  - Security and trust relations between devices
  - **Sandbox for other protocols**

- **Mobility**
  - Dynamic join/split/leave of federations
1. **Security Bootstrapping**
   - Key generation, authentication, distribution
   - Build security and trust relations

2. **Security Proxy**
   - Support for weak devices
   - Virtual services

3. **Secure Groups**
   - Efficient group notation
   - Recognition of groups
   - Agreement
Activities

1. Security Bootstrapping
   Ownership model
   Security Policy Definition Language
   Delegation
   Novel pairing mechanisms (sense the key, drag&drop)

2. Security Proxy
   Prototype Implementation
   Virtual Services: Concept
   Authentication protocols: Mathematical analysis
   Authentication protocols: Implementation on mobile phone

3. Secure Groups
## Authentication Protocols

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Messages</th>
<th>Claimant</th>
<th>Verifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSH (Diffie-Hellman + pswd)</td>
<td>7</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>Fiat-Shamir (1 round)</td>
<td>3</td>
<td>39</td>
<td>25</td>
</tr>
<tr>
<td>Kerberos</td>
<td>6</td>
<td>36</td>
<td>45</td>
</tr>
<tr>
<td>Modified Needham-Schroeder</td>
<td>3</td>
<td>21</td>
<td>21</td>
</tr>
</tbody>
</table>

number of cycles in 1000
Fiat-Shamir - Implementation

\[
\begin{align*}
A \\
s, v &= s^2 \mod n, n \\
B \\
v, n
\end{align*}
\]

\[
\begin{align*}
r &\in_R [1, n] \\
x &= r^2 \mod n \\
y &= r s^e \mod n
\end{align*}
\]

\[
\begin{align*}
\rightarrow \\
e &\in_R [0, 1] \\
y^2 &= ? x v^e
\end{align*}
\]

Execution on a mobile phone:

<table>
<thead>
<tr>
<th>Rounds</th>
<th>1</th>
<th>5 (1+1+1+1+1)</th>
<th>5 (in one Round)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>14s</td>
<td>50s</td>
<td>18s</td>
</tr>
</tbody>
</table>

n rounds correspond to a security margin of \(2^{-n}\)

Calculations: a few Milliseconds

GPRS req-resp: 4.5s

Is computational complexity an issue ???
Security bootstrapping for networked devices:
- No pre-defined relations
  no secure links, no trust relations, unknown devices
- No central service
  no administrator, no trusted third party

Ownership Model
- Federate devices owned by the same user

Security Policy
- Introduce other devices
- Assign rights to relations
Resurrecting Duckling Policy Model [SA99, Sta00]

- “Secure Transient Association”
  - Exchange of shared secret during physical contact
  - Back into initial state

- Default policy:
  - Master can access all services
  - Policy update is a service

Shortcomings:
- Pair wise associations
- Get authentic credentials to describe peer devices
- Lifecycle (e.g., delegation)
- Exception handling (e.g., loss of device)
Ownership Model

- Random key is the identity of the device
- Pairing: Password-authenticated key-exchange

> Generate certificate
  - results in certificate chains towards one of the user’s devices
  - recognize siblings

> Default policy
Drag&Drop Key Exchange

Visual Codes: Can be used as an authentic channel