Building a CCN Trust Model

Karthikeyan Natarajan
Vahab Pournaghshband

June 3, 2010
Overview

- Problem Formulation
- Current Approaches
- Our Approach
- Security Analysis
## Problem Formulation

**Interest Packet**

<table>
<thead>
<tr>
<th>Content Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selector</td>
</tr>
<tr>
<td>(order preference, publisher filter, scope, …)</td>
</tr>
<tr>
<td>Nonce</td>
</tr>
</tbody>
</table>

**Data Packet**

<table>
<thead>
<tr>
<th>Content Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signature</td>
</tr>
<tr>
<td>(digest algorithm, witness, …)</td>
</tr>
<tr>
<td>Signed Info</td>
</tr>
<tr>
<td>(publisher ID, key locator, stale time, …)</td>
</tr>
<tr>
<td>Data</td>
</tr>
</tbody>
</table>

- Digital Signatures
  - Private key signs, public key verifies
  - But, are we using the “right” public key?
  - Key verification problem
Current Approaches

- Leap-of-Faith
- Pretty Good Privacy (PGP)
- Certificate Authority (CA)
- DNSSEC
Our Approach

Hello, Bob

Bob’s Key?

Client Policy

Consistent
Inconsistent

Offered Key
Observations

Accept Key, Continue
Reject Key, Abort Connection

Consistent
Inconsistent

Bob’s Key?

Bob’s Key?

Bob

Alice
Key Interest Message

- **Name Convention**
  - `/myfriend/key_service/pubkeyID--keyLocator--issuer’s nonce`

- **Requires Signature by Issuer**

- **Maintaining a Table**
  - Pairs of `<issuer’s nonce, pubkeyID>`

**Interest Packet**

<table>
<thead>
<tr>
<th>Content Name</th>
<th>Selector</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(order preference, publisher filter, scope, ...)</td>
</tr>
<tr>
<td></td>
<td>Nonce</td>
</tr>
</tbody>
</table>
Key Response Message

- Two Possible Response Messages
  - Signed(publisher’s name || publisher’s key)
  - Signed(publisher’s name || I don’t have it)

- Why publisher’s name?
- Why signed?
- Why “I don’t have it” response?
Trust Bootstrapping

Trust Community

- Friends
  - Who are my friends?
  - Out-of-band mechanism
    (e.g. Facebook, visit cards)

- Notaries
  - What are notaries?
  - How to obtain keys?
    (e.g. Security through Publicity)
**Notion of Master Key**

- **What is a master key?**
  - Public Key
  - Longer Lifespan
  - Used Only to Sign Keys in Sub-domain

- **Role of Master Key?**
  - Plays as the Certifying Authority
  - Signs (certifies) Sub-domain Keys

- **Why is it useful?**
  - Less Network Overhead:
    - Less Key Verification by Flooding Trust Zone
    - Less Frequent (master) Key Changes
  - Flexibility

- **Why better than current CA?**
  - Does not Involve Third Party
Key Revocation

- A Key consists of:
  - Public Key
  - Key Identifier
  - Expiration Date

- How to Revoke?
  - Key Expiration Date
  - Immediate Rollover by *Notifying Notaries*

- How to Learn?
  - When Verification Fails
Trust Policy

**Quorum**: minimum agreement needed to consider a key valid

<table>
<thead>
<tr>
<th>Notary #1</th>
<th>Friend #1</th>
<th>Notary #2</th>
<th>Friend #2</th>
<th>Friend #3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_A$</td>
<td>$K_A$</td>
<td>$K_A$</td>
<td>$K_B$</td>
<td>$K_A$</td>
</tr>
</tbody>
</table>

If offered key is $K_A$:
- if $Q \geq 80\%$ then Accept
- else then Reject
Security Analysis

- Man-in-the-Middle Attack
  - Accepting Fake Key
  - Key Change Deception

- Denial-of-Service Attack
  - Dropping Key Response Messages

- Replay Attack
Security Analysis

- Compromised/Malicious Notary
  - Incorrect Responses

- Compromised/Malicious Friend
  - Incorrect Responses
  - Issuing frequent bogus key interest messages
Factors to Consider:

- Correctness
- Cooperativeness
- Personal Trust
- Responsiveness?
Summary

- Basics Implemented
- Avoids One-fits-all Model
- Higher degree of client control over trust decision
  - Chooses who to trust
  - Trust community
  - Defines security
- No need to trust/pay third party
- Robust against attacks
- Privacy issues