Usable Encryption

Class Presentation for CMSC 818D
Wei Bai
Application

- Hardware Encryption
- Web Encryption
  - Email Encryption
    - OpenPGP
    - S/MIME
  - Online Social Network
Public Key Encryption

- Encryption/Decryption
- Signing/Verifying
Prototype for Public Key Encryption

Bob

Hello Alice!
Encrypt
6EB69570 08E03CE4
Alice's public key

Alice

Decrypt
Hello Alice!
Alice's private key
Prototype for Signing/Verifying

Bob

Signing

Data

Hash function

Hash

101100110101

Encrypt hash using signer's private key

Certificate

Signature

Attach to data

Digitally signed data

Verification

Alice

Digitally signed data

Data

Hash function

Hash

101100110101

Signature

Decrypt using signer's public key

Data

Hash function

Hash

101100110101

If the hashes are equal, the signature is valid.
Usable Encryption Design Aspects

- Key Management
- Encryption UI
- Integration
- (Automatic) Encryption and Decryption

Design Aspects
A. Whitten et al. “Why Johnny can’t encrypt: a usability evaluation of PGP 5.0”

- One of the pioneer works for encryption usability
- Objective: Investigate usability in standard UI design vs security
- A case study of PGP 5.0 through
  - Cognitive walkthrough analysis
  - Lab Study
Definitions of Usability for Security

1. Reliably made aware of the security tasks they need to perform
2. Able to figure out how to successfully perform those tasks
3. Don’t make dangerous errors
4. Comfortable with the interface to continue using it.
Properties

1. The unmotivated user property
2. The abstraction property
3. The lack of feedback property
4. The barn door property
5. The weakest link property
Usability Standard for PGP

- Encrypt/decrypt
- Sign/verify
- Key generation
- Own public key publication
- Public key acquiring
- Avoid dangerous errors
- Reasonable time
Usability Analysis

- Cognitive walkthrough Analysis
  - Wide considerations for more factors
  - Subjective

- Lab Study
  - Limited scope of factor testing
  - Objective
Cognitive analysis: Flaws in Design

- Key management Issue
  - Visual: sign/verify
  - Different key types
    - RSA for PGP,
    - Hellman/DSS for PGP 5.0
  - Key server
  - Errors playing with keys. Irreversible!
    - Delete the private key, publicize the private key
Lab study

- Integrate Eudora with PGP
Lab study

- Confirmed some points:
- What keys to use? How to use them?
  - Confused about private/public keys
  - Use own/counterpart’s keys?
1. Dangerous errors and the barn door property:
   - If reversible? Regret allowed?

2. Whether tutorials about encryption tasks, such as generating keys should be included?
   - Learnability: Learn by themselves, or taught by others?

3. Is signing and verification necessary?
   - Closed circle
   - Phishing exists

4. How about separating encryption and decryption tasks, to make study shorter?
Objective

Investigate whether it makes more usable if hiding as many security details as possible

Method:

Lab study of Pwm (private webmail) system
Pwm Highlights

- Automatic key management and automatic encryption
- Integrate tightly with existing webmail services
- Key management by a key escrow
  - Advantage:
    - Automatic key management
    - Users never lose their keys
    - Keys ported to new devices automatically
  - Disadvantage
    - Escrow has access to users’ keys
Comparative Usability Study

- Task scenario:
  - Decrypt an email first
  - Send an encrypted email
  - Open a new Gmail session (with Pwm ended)

- Perform well compared to existing webmail tools (w.r.t. SUS score)
Key Findings

- Performance of Message Protector is, on par with, slightly higher than Pwm.
- Too transparent design loses trust to some extent
- Reconsider manual encryption
  - The idea also comes from “Johnny for Facebook” paper.
Discussions

- Automatic key management by using a third party service?
  - Chicken and egg problem!
  - Tradeoff between usability/security
Helping Johnny 2.0 to encrypt His Facebook conversations

- **Objective:**
  - Encryption usability for online social networks (OSNs)

- **Methods:**
  - Two lab studies
## Table 2: Properties of the tasks in the lab study.

<table>
<thead>
<tr>
<th>Task</th>
<th>Interface</th>
<th>Encryption</th>
<th>Key Management</th>
</tr>
</thead>
<tbody>
<tr>
<td>T01</td>
<td>Facebook</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>T02</td>
<td>Mockup 2</td>
<td>Manual</td>
<td>Manual</td>
</tr>
<tr>
<td>T03</td>
<td>Mockup 3</td>
<td>Automatic</td>
<td>Manual</td>
</tr>
<tr>
<td>T04</td>
<td>Mockup 4</td>
<td>Manual</td>
<td>Automatic</td>
</tr>
<tr>
<td>T05</td>
<td>Mockup 5</td>
<td>Automatic</td>
<td>Automatic</td>
</tr>
</tbody>
</table>
Mockup Lab Study

- Encryption schemes:
  - Auto/not auto: encryption button

- Key management:
  - Manual: send keys over webmail
  - Auto: Passwords created at the first time, and then web browser caches it for further use.
Key findings

- Auto encryption and auto key management is preferable.
- Manual encryption / manual decryption have higher security feeling, but lower acceptance
- Key (password) recovery capability
Discussions

- **Do/show something makes users assured?**
  - Auto/not auto: encryption button

- **Key management:**
  - Manual: send keys over webmail
    - (chicken and egg problem again?)
  - Auto: Passwords created at the first time, and then web browser caches it for further use.
    - Password protection? Guessibility for password is much easier than PKI keys.