Lecture 8: Usable Security: User-Enabled Device Authentication

CS 436/636/736
Spring 2015

Nitesh Saxena

---

Course Admin

- HW3 being graded
  - Solution will be provided soon
- HW4 posted
  - The conceptual part is due Apr 17 (Friday)
  - Includes programming part related to Buffer Overflow
  - Programming part needs demo – we will do so next week
    - Demo slot sign-up – today
  - You are strongly encouraged to work on the programming part in teams of two each
    - Please form your own team
Course Admin

- Final Exam – **Apr 23 (Thursday)**
  - 7 to 9:30pm
  - Venue – TBA (most likely the lecture room)
- Covers everything (cumulative)
  - 35% -- pre mid-term material
  - 65% -- post mid-term material
- Again, close-book, just like the mid-term
- We will do exam review Apr 16

Today’s Lecture

- The user aspect in security
  - Security often has to rely on user actions or decisions
    - Many examples?
  - User is often considered a weak-link in security
  - Can we build secure systems by relying on the strengths of human users rather than their weaknesses?
- We study this in the context of one specific application:
  - Device-to-Device Authentication
- The field of Usable Security is quite new and immature
  - Research flavor in today’s lecture
The Problem: “Pairing”

Examples (single-user setting)
- Pairing a Bluetooth cell phone with a headset
- Pairing a WiFi laptop with an access point
- Pairing two Bluetooth cell phones

PIN-based Bluetooth Pairing
Authentication

PIN-based Bluetooth Pairing
(In)Security of PIN-based Pairing

- Long believed to be insecure for short PINs
  - Why?
- First to demonstrate this insecurity; *Shaked and Wool* [Mobisys’05]
**Attack Implementation**

- Coded in C on linux platform
  - Given a piece of code for SAFER algorithm, implemented the encryption functions $E_{22}, E_{21}, E_1$
- **Hardware for sniffing**: bluetooth packet analyzer with windows software
- **Log Parser** (in perl): reads the sniffer log, and parses it to grab $IN\_RAND, RAND\_A \oplus K_{init}$, $RAND\_B \oplus K_{init}, AU\_RAND\_A, AU\_RAND\_B, SRES$

**Timing Measurements of Attack**

- Theoretically: $O(10^L)$, with decimal digits
  - Assuming the PINs are chosen uniformly at random
- Empirically, on a PIII 700MHz machine:

<table>
<thead>
<tr>
<th>No. of digits in PIN (L)</th>
<th>CPU time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1.294</td>
</tr>
<tr>
<td>5</td>
<td>12.915</td>
</tr>
<tr>
<td>6</td>
<td>129.657</td>
</tr>
<tr>
<td>7</td>
<td>1315.332</td>
</tr>
</tbody>
</table>
Timing of Attack and User Issues

- **ASCII PINs**: $O(90^L)$, assuming there are 90 ascii characters that can be typed on a mobile phone
  - Assuming random PINs
- However, in practice the actual space will be quite small
  - Users choose weak PINs;
  - Users find it hard to type in ascii characters on mobile devices
- Another problem: shoulder surfing (manual or automated)

The Problem: “Pairing”

**Idea**
- make use of a physical channel between devices
  - Also known as out-of-band (OOB) channel
  - with least involvement from Alice and Bob
**Seeing-is-Believing** (McCune et al. [Oakland’05])

- **Protocol** (Balfanz, et al. [NDSS’02])

  ![Diagram of protocol](image)

  - Insecure Channel
  - Authenticated Channel

- **Rohs, Gfeller** [PervComp’04]

- **Challenges**
  - OOB channels are low-bandwidth!
  - One of the device might not have a keypad/receiver!
  - Neither has a receiver and only one has a good quality transmitter
    - (Non-)Universality!
  - Usability Evaluation
  - Multiple devices – scalability
Challenges

- OOB channels are low-bandwidth!
- One of the device might not have a keypad/receiver!
- Neither has a receiver and only one has a good quality transmitter
  - (Non-)Universality!
- Usability!
- Multiple devices – scalability

Protocol: **Short Authenticated Strings (SAS)**

<table>
<thead>
<tr>
<th>Insecure Channel</th>
<th>Authenticated Channel</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_A \in {0,1}^k$</td>
<td>$c_A, d_A \leftarrow \text{comm}(pk_A, R_A)$</td>
</tr>
<tr>
<td>$c_B, d_B \leftarrow \text{comm}(pk_B, R_B)$</td>
<td>$R_B \in {0,1}^k$</td>
</tr>
<tr>
<td>$pk_A, c_A$</td>
<td>$pk_B, c_B$</td>
</tr>
<tr>
<td>$d_A$</td>
<td>$R_A \leftarrow \text{open}(pk_A, c_A, d_A)$</td>
</tr>
<tr>
<td>$R_B \leftarrow \text{open}(pk_B, c_B, d_B)$</td>
<td>$d_B$</td>
</tr>
<tr>
<td>$SAS_A = R_A \oplus R_B$</td>
<td>$SAS_B = R_B$</td>
</tr>
</tbody>
</table>

Accept $(pk_B, A)$ if $SAS_A = R_B \oplus R_B$

Accept $(pk_B, B)$ if $SAS_B = R_B$

Laur et al. [eprint’05]

Pasini-Vaudenay [PKC’06]

Vaudenay [Crypto’05]

Secure (with prob. $1-2^{-k}$)
Challenges

- OOB channels are low-bandwidth!
- One of the devices might not have a keypad/receiver
  - e.g., keyboard-desktop; AP-phone
- Neither has a receiver and only one has a good quality transmitter
  - (Non-)Universality!
- Usability
- Multiple devices – scalability

Unidirectional SAS (Saxena et al. [S&P'06])

- Blinking-Lights

\[ pk_A, H(R_A) \]
\[ pk_B, R_B \]
\[ R_A \]
\[ hs(R_A, R_B; pk_A, pk_B) \]

Success/Failure

Secure (with prob. \(1 - 2^{-15}\)) if
- 15-bit AU \( hs() \)

Multiple Blinking LEDs (Saxena-Uddin [MWNS'08])
Challenges

- OOB channels are low-bandwidth!
- One of the device might not have a receiver!
- Neither has a receiver and only one has a display
  - e.g., AP-laptop/PDA
- Usability!
- Multiple devices – scalability

A Universal Pairing Method

- Prasad-Saxena [ACNS’08]
- Use existing SAS protocols
- The strings transmitted by both devices over physical channel should be
  - the same, if everything is fine
  - different, if there is an attack/fault
- Both devices encode these strings using a pattern of
  - Synchronized beeping/blinking
  - The user acts as a reader and verifies if the two patterns are same or not
Is This Usable?

- Our test results are promising
  - Users can verify both good test cases and bad ones
- **Blink-Blink** the easiest
  - Very low errors (less than 5%)
  - Execution time ~22s
- Then, **Beep-Blink**
  - Very low errors with a learning instance (less than 5%)
  - Execution time ~15s
- **Beep-Beep** turns out error-prone

Further Improvement: Auxiliary Device

- Saxena et al. [SOUPS’08]
- Auxiliary device needs a camera and/or microphone – a smart phone
- Does not need to be trusted with cryptographic data
- Does not need to communicate with the devices
Further Improvement: Auxiliary Device

- **Blink-Blink**
  - ~14s (compared to 22s of manual scheme)

- **Beep-Blink**
  - Approximately takes as long as the same as manual scheme
  - No learning needed

- In both cases,
  - Fatal errors (non-matching instances decided as matching) are **eliminated**
  - Safe errors (matching instances decided as non-matching) are reduced

- It was preferred by most users

Challenges

- OOB channels are low-bandwidth!
- One of the device might not have a receiver!
- Neither has a receiver and only one has a good quality transmitter
  - (Non-)Universality!
- Comparative Usability!
- Multiple devices – scalability
Many Mechanisms Exist

- See survey: [Kumar, et al. @ Percom’09]
  - Manual Comparison or Transfer:
    - Numbers [Uzun, et al. @ USEC’06]
    - Spoken/Displayed Phrases: Loud & Clear [Goodrich, et al. @ ICDCS’06]
    - Images: [Goldberg’96][Perrig-Song’99][Ellison-Dohrman @ TISSEC’03]
    - Button-enabled data transfer (BEDA) [Soriente, et al. @ IWSSI’07]
    - Synchronized Patterns [Saxena et al. @ ACNS’08 & SOUPS’08]
  - Automated:
    - Seeing-is-Believing (SiB) [McCune, et al. @ S&P’05]
    - Blinking Lights [Saxena, et al. @ S&P’06]
    - Audio Transfer [Soriente, et al. @ ISC’08]

A Comparative Usability Study

- How do these mechanisms compare with one another in terms of usability?
  - Timing; error rates; user preferences
- Needed a formal usability study

- Automated testing framework
- 40 participants; over a 2 month long period

- Surprise:
  - Users don’t like automated methods: handling cameras not easy
Tested methods (1/5)

- Number Comparison
  “65473” =? “75853”
- Phrase Comparison
  “Alice buys jackets” =? “John likes elephants”
- Image Comparison

Tested Methods (2/5)

- Audiovisual synchronization methods
  - Beep-Blink
    ![Beep-Blink Diagram]
  - Blink-Blink
    ![Blink-Blink Diagram]
Tested methods (3/5)

- Button enabled (BEDA) methods
  - LED-Button
  - Vibrate-Button
  - Button-Button

Tested methods (4/5)

- Loud and Clear (L&C) variants
  - Speaker-Speaker
  - Display-Speaker
Tested Methods (5/5)

- Seeing is Believing (SiB)
- Blinking Lights
- HAPADEP Variant

Comparative Usability Study: Time
Comparative Usability Study: Ease-of-Use

Answers given as "Very Hard", "Hard", "Easy" or "Very Easy" on a 4 point scale
**Cluster Analysis: Big Picture!**

![Cluster Analysis Diagram]

**Conclusions from the study**

- Both devices have a display
  - Numeric Comparison

- One device does not have a display but an audio interface
  - L&C Display-Speaker
    - if one has a display and the other has a speaker
  - Audio Transfer
    - if microphone is available on one, and speaker on the other

- Interface constraint device(s)
  - BEDA Vibrate-Button, if possible
  - BEDA LED-Button otherwise
Challenges

- OOB channels are low-bandwidth!
- One of the device might not have a receiver!
- Neither has a receiver and only one has a good quality transmitter
  - (Non-)Universality!
- Usability!
- Multiple devices – scalability

Secure Group Association

- Small groups (> 2) of users + devices
  - phones, PDAs, laptops
- Common use-cases:
  - Share content as part of an ad hoc meeting
  - Multiplayer games
  - Multimedia streaming
- Two user tasks:
  - Comparison of SAS strings
  - Verification of group size
Usability Evaluation of Group Methods

- Usability evaluation of FIVE simple methods geared for small groups (4-6 members)
  - Three leader-based
  - Two peer-based

Study Goals

- How well do users perform the two tasks when multiple devices and users are involved:
  - Comparison/Transfer of SAS strings?
  - Counting number of group members?
Leader-Based Methods (1/3)

- **Leader-VerifySize-VerifySAS (L-VS-VS):** Leader announces 5-digit SAS, group members verify the displayed SAS and the group size.

  1. Enter the group size: 4
  2. Announce the verification code: 39715
  3. Accept or Reject

  The Verification Code is: 39715

Leader-Based Methods (2/3)

- **Leader-VerifySize-CopySAS (L-VS-CS):** Leader announces SAS, members enter it to their devices and verify group size.

  1. Enter verification code: 39715
  2. Group Size is: 4
  3. Accept or Reject

  The Verification Code is: 39715
Leader-Based Methods (3/3)

- Leader-VerifySize-AudioSAS (L-VS-AS): Leader’s device broadcasts SAS (over audio), other devices record & verify. Users only verify group size.

Peer-Based Methods (1/2)

- Peer-VerifySize-VerifySAS (P-VS-VS): Each peer verifies group size and compares SAS with left-hand peer.
Peer-Based Methods (2/2)

- **Peer-InputSize-VerifySAS (P-IS-VS):** Each peer enters group size and compares SAS with left-hand peer.

  1. Enter Group Size: 4
  2. Your code is: 39715
  3. Does it match with the code of the person next to you?
  4. Accept or Reject

Usability Testing Details

- 64 Participants
  - 7 four-person groups
  - 6 six-person groups
- Nokia smart phones communicating over Wi-Fi
- Test Cases:
  - Normal
  - Simulated Insertion Attack
  - Simulated Evil-Twin Attack
- User Feedback
  - SUS questionnaire (system usability scale)
  - Additional perception of security question
### Test Results (Normal Case)

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Group Size</th>
<th>Successful Completion Rate</th>
<th>Avg. Completion Time in secs</th>
<th>Avg. SUS Score</th>
<th>Perception of Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>L-VS-AS</td>
<td>4</td>
<td>85.7%</td>
<td>36.71 (5.46)</td>
<td>62.86 (3.86)</td>
<td>2.96 (0.10)</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>100.0%</td>
<td>42.33 (9.71)</td>
<td>67.85 (5.20)</td>
<td>3.25 (0.21)</td>
</tr>
<tr>
<td>L-VS-CS</td>
<td>4</td>
<td>85.7%</td>
<td>51.57 (9.34)</td>
<td>64.91 (4.07)</td>
<td>3.07 (0.32)</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>83.3%</td>
<td>49.67 (6.24)</td>
<td>65.97 (4.38)</td>
<td>3.31 (0.23)</td>
</tr>
<tr>
<td>L-VS-VS</td>
<td>4</td>
<td>100.0%</td>
<td>41.29 (7.77)</td>
<td>68.75 (4.01)</td>
<td>3.36 (0.33)</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>100.0%</td>
<td>31.33 (4.64)</td>
<td>65.76 (2.39)</td>
<td>3.25 (0.23)</td>
</tr>
<tr>
<td>P-IS-VS</td>
<td>4</td>
<td>85.7%</td>
<td>27.57 (3.66)</td>
<td>68.57 (3.87)</td>
<td>3.57 (0.09)</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>100.0%</td>
<td>38.83 (3.00)</td>
<td>77.36 (5.16)</td>
<td>4.14 (0.24)</td>
</tr>
<tr>
<td>P-VS-VS</td>
<td>4</td>
<td>100.0%</td>
<td>40.43 (8.64)</td>
<td>63.57 (3.31)</td>
<td>3.14 (0.27)</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>100.0%</td>
<td>43.00 (8.27)</td>
<td>74.03 (5.96)</td>
<td>3.75 (0.35)</td>
</tr>
</tbody>
</table>

Values in parenthesis show the standard error of the mean.

### Test Results (Attack Cases)

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Method Name</th>
<th>Group Size</th>
<th>Avg. Completion Time in secs</th>
<th>Secure Completion Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insertion Attack</td>
<td>L-VS-AS</td>
<td>4</td>
<td>23.57 (4.67)</td>
<td>100.0%</td>
</tr>
<tr>
<td></td>
<td>L-VS-CS</td>
<td>4</td>
<td>47.14 (14.1)</td>
<td>100.0%</td>
</tr>
<tr>
<td></td>
<td>L-VS-VS</td>
<td>4</td>
<td>52.14 (18.4)</td>
<td>100.0%</td>
</tr>
<tr>
<td></td>
<td>P-IS-VS</td>
<td>4</td>
<td>44.57 (16.4)</td>
<td>100.0%</td>
</tr>
<tr>
<td></td>
<td>P-VS-VS</td>
<td>4</td>
<td>37.86 (15.1)</td>
<td>100.0%</td>
</tr>
<tr>
<td></td>
<td>L-VS-AS</td>
<td>5</td>
<td>33.86 (9.67)</td>
<td>100.0%</td>
</tr>
<tr>
<td></td>
<td>L-VS-CS</td>
<td>5</td>
<td>31.83 (6.72)</td>
<td>100.0%</td>
</tr>
<tr>
<td></td>
<td>L-VS-VS</td>
<td>5</td>
<td>48.50 (5.93)</td>
<td>100.0%</td>
</tr>
<tr>
<td></td>
<td>P-IS-VS</td>
<td>5</td>
<td>44.57 (16.4)</td>
<td>100.0%</td>
</tr>
<tr>
<td></td>
<td>P-VS-VS</td>
<td>5</td>
<td>33.00 (6.87)</td>
<td>83.3%</td>
</tr>
<tr>
<td>Evil Twin Attack</td>
<td>L-VS-AS</td>
<td>4</td>
<td>33.86 (9.67)</td>
<td>100.0%</td>
</tr>
<tr>
<td></td>
<td>L-VS-CS</td>
<td>4</td>
<td>42.00 (7.03)</td>
<td>66.7%</td>
</tr>
<tr>
<td></td>
<td>L-VS-VS</td>
<td>4</td>
<td>29.00 (2.54)</td>
<td>50.0%</td>
</tr>
<tr>
<td></td>
<td>P-IS-VS</td>
<td>4</td>
<td>28.00 (6.20)</td>
<td>100.0%</td>
</tr>
<tr>
<td></td>
<td>P-VS-VS</td>
<td>4</td>
<td>32.43 (6.23)</td>
<td>100.0%</td>
</tr>
</tbody>
</table>

Values in parenthesis show the standard error of the mean.
• Interesting result: peer based methods performed better in general

Summary of results

- Peer-based methods generally better than Leader-based ones
- P-IS-VS has the best overall usability
- L-VS-CS has the worst
- L-VS-VS and L-VS-AS are natural choices if peer-based methods are not suitable
  - L-VS-VS > L-VS-AS
- Over-counting unlikely in small groups
- Entering group size is better than verifying it
Other open questions

- Rushing user behavior (Saxena-Uddin [ACNS'09])
- Hawthorne effect
- Security priming
- More usability tests

Extending to VoIP: Crypto Phones

1. Your code is: 39715
   Provide this code to Alice
4. Is Alice’s code equal to your code? Press Yes or No

2. My code is 39715

3. My code is 24641

1. Your code is: 24641
   Provide this code to Bob

4. Is Bob’s code equal to your code? Press Yes or No

Attack is detected since SAS values do not match
Key Weakness: Voice Imitation Attacks

- Shirvanian-Saxena [CCS’14]

1. Your code is: 39715
   Provide this code to Alice
6. Is Alice’s code equal to your code? Press Yes or No

2. My code is 39715
3. My code is 24641
5. My code is 39715

4. My code is 24641

Code: 39715

Code: 24641

Attack is NOT detected since SAS values match

Two Type of Attacks

- Reordering attack

- Morphing attack

<table>
<thead>
<tr>
<th>Training</th>
<th>Morphing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utterances U1-U50 Attacker Voice</td>
<td>Utterances U1-U50 in Victim’s Voice</td>
</tr>
<tr>
<td>SAS in Mallory’s Voice</td>
<td>Morphed SAS in Alice’s Voice</td>
</tr>
</tbody>
</table>
## Results

<table>
<thead>
<tr>
<th>Presented Numeric</th>
<th>Presented 16-Bit PGP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Original</td>
</tr>
<tr>
<td>Detected as: Yes</td>
<td>57.50%</td>
</tr>
<tr>
<td>Detected as: No</td>
<td>42.50%</td>
</tr>
<tr>
<td>Accuracy</td>
<td>71.49%</td>
</tr>
<tr>
<td>False Discovery Rate</td>
<td>20.16%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Presented 16-Bit Madlib</th>
<th>Overall False Positive, False Negative, True Positive and True Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Original</td>
</tr>
<tr>
<td>Detected as: Yes</td>
<td>50.83%</td>
</tr>
<tr>
<td>Detected as: No</td>
<td>49.17%</td>
</tr>
<tr>
<td>Accuracy</td>
<td>64.58%</td>
</tr>
<tr>
<td>False Discovery Rate</td>
<td>29.89%</td>
</tr>
</tbody>
</table>

## Conclusions

- Extending to VoIP setting is more challenging
- Users have to perform speaker verification (in addition to SAS validation)
- Hard to verify the speaker based on short strings
References

- Cracking Bluetooth PINs:
  http://www.eng.tau.ac.il/~yash/shaked-wool-mobisys05/

- OOB channel related references:
  - Many on my publications page:
  - http://spies.cis.uab.edu/research/secure-device-pairing/overview/