Human-Computable Passwords

Jeremiah Blocki
Manuel Blum
Anupam Datta
Santosh Vempala
Previous Work

• Naturally Rehearsing Passwords
  – Presentation on Thursday
Password Management

Competing Goals:

Security  Usability
Password Security Game

1. $p_1$
2. $p_2$
3. $p_3$
4. $p_4$
5. $p_5$

$BCRYPT(p_4)$

$q_{\$1,000,000}$ guesses

PayPaul.com
# Security Results

<table>
<thead>
<tr>
<th>Attacks</th>
<th>k= 1</th>
<th>k= 1</th>
<th>k=2</th>
<th>k=2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reuse</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Strong Random</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Independent</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Shared Cues</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td><img src="image" alt="Phishing Attack" /></td>
<td><img src="image" alt="Offline Attack" /></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Usable + Insecure</td>
<td>Unusable + Secure</td>
<td>Usable + Secure</td>
<td></td>
</tr>
</tbody>
</table>
## Security Results

<table>
<thead>
<tr>
<th>Attacks</th>
<th>k=1</th>
<th>k=1</th>
<th>k=2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reuse</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Strong</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Random</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Independent</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Shared Cues</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### Phishing Attack

- **Usable + Insecure**

### Offline Attack

- **Usable + Secure**
Previous Work

• Naturally Rehearsing Passwords
  – Presentation on Thursday
  – Password Management Scheme: Shared Cues

• Key Question: Can we get better security if we ask the user to perform simple computations to generate his passwords?
Human Computation

• Restricted
  – Simple operations (addition, lookup)
  – Operations performed in memory (limited space)

\[ 9 + 8 = 7 \mod 10 \]

\[ 8945309234 + 2348979234 = ? \]
Human Computation

• Restricted
  – Simple operations (addition, lookup)
  – Operations performed in memory (limited space)

• Improve Security?
  – Simple Computations vs. Pure Recall
  – Security against many breaches?
Candidate Scheme

• Memorize a Random Mapping
  – One time step!
• Password Computed as a Response to Public Challenges
• Required Operations
  – Addition modulo 10
  – Memory lookups
### Random Mapping

<table>
<thead>
<tr>
<th>Image $I$</th>
<th>$\sigma(I)$</th>
<th>9</th>
<th>3</th>
<th>...</th>
<th>6</th>
</tr>
</thead>
</table>

**Initialization:**

User Memorizes Random Mapping

$$\sigma: \{I_1, \ldots, I_m\} \rightarrow \{0, 1, \ldots, 9\}$$

$m$ images
Single-Digit Challenge

Response:

\[ \sigma(\text{lightning}) + \sigma(\text{dog}) = 2 \mod 10 \]
Single-Digit Challenge

Response:

\[ \sigma(\text{lightning}) + \sigma(\text{dog}) = 2 \mod 10 \]
Single-Digit Challenge

Response:

\[ \sigma(\text{Left}) + \sigma(\text{Middle}) + \sigma(\text{Right}) \]

\[ = 7 + 4 + 5 = 6 \mod 10 \]
Username: jblocki
Password:

\[ \sigma(\text{left}) + \sigma(\text{middle}) + \sigma(\text{right}) \]

\[ = 7 + 4 + 5 = 6 \text{ mod } 10 \]
Passwords

Username: jblocki
Password: *
Passwords

Username: jblocki

Password: **
Usability

• Memorization is a one time cost
  – Mapping \( f \) is rehearsed naturally
  – Can Add new Images over Time

• Time
  – 75 seconds for a 10 digit password
  – 7.5 seconds per digit (average)
Open Challenge

• Random mapping $\sigma: \{I_1, \ldots, I_{100}\} \rightarrow \{0,1,\ldots,9\}$

• Examples
  – 1000 single-digit challenge response pairs

• Can you crack the code and guess one of the challenge passwords?

# Open Challenge

<table>
<thead>
<tr>
<th>Function</th>
<th>Secret Length (n)</th>
<th>Challenge Response Pairs</th>
<th>Links</th>
<th>Winner</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f(y_9, \ldots, y_{13}) = y_{13} + y_{12} + y_i \mod 10$</td>
<td>100 digits (Pre-solved Example)</td>
<td>500 (e.g., 50 ten digit passwords)</td>
<td>Examples Password Challenges</td>
<td>Harry Q. Bovik</td>
</tr>
<tr>
<td>Where</td>
<td>$i = y_{11} + y_{12} \mod 10$</td>
<td></td>
<td>Examples Password Challenges</td>
<td></td>
</tr>
<tr>
<td></td>
<td>100 digits</td>
<td>1000</td>
<td>Examples Password Challenges</td>
<td></td>
</tr>
<tr>
<td></td>
<td>50 digits</td>
<td>500</td>
<td>Examples Password Challenges</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30 digits</td>
<td>300</td>
<td>Examples Password Challenges</td>
<td></td>
</tr>
<tr>
<td></td>
<td>150</td>
<td></td>
<td>Examples Password Challenges</td>
<td></td>
</tr>
<tr>
<td>$f(y_9, \ldots, y_{13}) = y_{13} + y_{12} + y_{11} + y_i \mod 10$</td>
<td>100 digits (Pre-solved Example)</td>
<td>500 (e.g., 50 ten digit passwords)</td>
<td>Examples Password Challenges</td>
<td>Harry Q. Bovik</td>
</tr>
<tr>
<td>Where</td>
<td>$i = y_{19} \mod 10$</td>
<td></td>
<td>Examples Password Challenges</td>
<td></td>
</tr>
<tr>
<td></td>
<td>100 digits</td>
<td>500</td>
<td>Examples Password Challenges</td>
<td></td>
</tr>
<tr>
<td></td>
<td>300</td>
<td></td>
<td>Examples Password Challenges</td>
<td></td>
</tr>
<tr>
<td></td>
<td>200</td>
<td></td>
<td>Examples P</td>
<td></td>
</tr>
<tr>
<td></td>
<td>50 digits</td>
<td>300</td>
<td>Examples P</td>
<td></td>
</tr>
<tr>
<td></td>
<td>150</td>
<td></td>
<td>Examples P</td>
<td></td>
</tr>
<tr>
<td></td>
<td>100</td>
<td></td>
<td>Examples P</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30 digits</td>
<td>150</td>
<td>Examples P</td>
<td></td>
</tr>
<tr>
<td></td>
<td>100</td>
<td></td>
<td>Examples P</td>
<td></td>
</tr>
<tr>
<td></td>
<td>50</td>
<td></td>
<td>Examples P</td>
<td></td>
</tr>
</tbody>
</table>