Two-Factor Authentication Resilient to Server Compromise Using Mix-Bandwidth Devices

<table>
<thead>
<tr>
<th>Name</th>
<th>University</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maliheh Shirvanian</td>
<td>University of Alabama at Birmingham</td>
<td><a href="mailto:maliheh@uab.edu">maliheh@uab.edu</a></td>
</tr>
<tr>
<td>Stanislaw Jarecki</td>
<td>University of California, Irvine</td>
<td><a href="mailto:stasio@ics.uci.edu">stasio@ics.uci.edu</a></td>
</tr>
<tr>
<td>Nitesh Saxena</td>
<td>University of Alabama at Birmingham</td>
<td><a href="mailto:saxena@cis.uab.edu">saxena@cis.uab.edu</a></td>
</tr>
<tr>
<td>Naveen Nathan</td>
<td>University of California, Irvine</td>
<td><a href="mailto:nnathan@uci.edu">nnathan@uci.edu</a></td>
</tr>
</tbody>
</table>
Outline

• Current State
• Desirable Properties
• Our Contributions
• Protocols and Security Analysis
• System Implementation
• Discussion
Introduction

- Password only systems
- Two Factor Authentication TFA
- Online guessing attack
- Offline dictionary attack
  - Many real-world instances
  - Password re-use

More than 200,000 of these passwords have reportedly been cracked so far.
Current State

| D | = 2^d = Size of a password dictionary
| t = |z| = bandwidth of Device to Client channel
| x = time
## Desirable Goals

<table>
<thead>
<tr>
<th>In case of:</th>
<th>Desired:</th>
</tr>
</thead>
<tbody>
<tr>
<td>On-line guessing</td>
<td>Probability of ((1/</td>
</tr>
<tr>
<td>Offline Dictionary attack</td>
<td>Complexity of (O(</td>
</tr>
<tr>
<td>Lunch time attack/ C-D communication</td>
<td>Shouldn’t affect above</td>
</tr>
<tr>
<td>Adversary breaks into the user's device</td>
<td>security degrades to password-only</td>
</tr>
<tr>
<td>Adversary learns the user's password</td>
<td>security degrades to the device-only</td>
</tr>
</tbody>
</table>
Our Contributions

- Novel TFA Protocols to achieve desired TFA properties and improve security of TFA Schemes.

- Mix-Bandwidth Device TFA Mechanisms to improve ODA resistance by increasing bandwidth $t$. 
The Main Idea

- Server stores a hash of the password and a secret $s$, $h=H(p,s)$
- Device stores the secret $s$
- Authentication decision based on whether user provides the correct password and owns the device which stores $s$
Protocols

- **Time-based TFA protocol**
  - Applicable to all device types (Low, Mid, High Bandwidth)
  - Rely on a clock synchronized with the server

- **Challenge-Response TFA Protocols**
  - Symmetric-key and public-key TFA protocols
  - Applicable for devices that receive a challenge and show PIN
Time-Based TFA Protocol

1. \( z = s \oplus F_k(T_d) \)
2. \( (UN, p, z) \)
3. Accept if: \( H(p, z \oplus F_k(T_s)) = h \)
Symmetric-Key TFA Protocol

1. \( x \)
2. \( s, K \)
3. \( z = s \text{ xor } F_k(x) \)
4. \( UN, p, z \)
5. Accept if: \( H(p, z \text{ xor } F_k(x)) = h \)
Public-Key TFA Protocol

1. \( s, K, Sk \)

2. \( c = Enc_{pk}(r) \)

3. \( z = s \oplus Dec_{sk}(a) \)

4. \( (p, z) \)

5. Accept if: \( H(p, z \oplus r) = h \)
## Security of the Protocols

<table>
<thead>
<tr>
<th>In case of:</th>
<th>Desired:</th>
</tr>
</thead>
<tbody>
<tr>
<td>On-line guessing</td>
<td>Probability of $(1/</td>
</tr>
<tr>
<td>Offline Dictionary attack</td>
<td>Complexity of $O(</td>
</tr>
<tr>
<td>Lunch time attack/ C-D communication</td>
<td>Shouldn’t affect above</td>
</tr>
<tr>
<td>Adversary breaks into the user's device</td>
<td>security degrades to password-only</td>
</tr>
<tr>
<td>Adversary learns the user's password</td>
<td>security degrades to the device-only</td>
</tr>
</tbody>
</table>
Notes on System Design and Implementation

- Total 13 TFA mechanisms categorized based on:
  - The underlying protocol
  - The underlying device type
  - The underlying Device - Client channel – PIN, QR, BT, WiFi
    - PIN: 6 digits, manual entry
    - QR: The QR code encoding and decoding ZXing library, HTML5 Server codes and a plain browser on the Client
    - BT: Android application listening on a RFCOMM socket, Client runs a browser extension (Bluetooth API)
    - WF: Virtual WiFi between Client and Device, Client runs a browser extension (chrome.socket API)
LBD Authentication Phase

Username:  
Password:  
Verification Code: 497173

Account: example.com
PIN: 497173
MBD Authentication Phase

\[ z = s \ xor F_k(x) \]
FBD Authentication Phase

\[ z = s \oplus \text{Dec}_{sk}(a) \]

Device

s, K, sk

Client

User

- Wi-Fi
- Bluetooth

Communication to android device, keep this app open during login to Auth Server!
Discussion and Conclusion

• **Security:**
  - All mechanism provide improved resilience to offline dictionary attacks and online attacks.
  - Challenge-Response protocols are secure against a lunch-time attacker.
  - FBD mechanisms are more secure against online attacks.

• **Usability:**
  - There is no time synchronization requirement in Challenge Response mechanisms.
  - In high bandwidth channels user does not need to manually transfer the PIN.

• **Deployability:**
  - Traditional and LBD work with a plain browser and no special hardware.
Thank you!

Questions?