T2Pair: Secure and Usable Pairing for Heterogeneous IoT Devices

Xiaopeng Li, Qiang Zeng, Lannan Luo, Tongbo Luo

CCS 2020
IoT Pairing

• Pairing is supposed to establish a secure communication channel

• IoT pairing is important for
  – adding a new IoT device to a network
  – data transmission between two devices (e.g., a blood-pressure meter in Walmart and your phone)
Design Requirements

• **Secure**: resilient to co-located malicious devices

• **Usable** for heterogeneous IoT devices
  – No conventional UIs like keyboards
  – Not special sensors (e.g., inertial)
Existing Approaches

- **Proximity-based**
  - Move2Auth [InfoCom’17]: wireless signal features
  - Perceptio [S&P’19]: ambient context

  ![Insecure](image)
  **Insecure**: exploited by co-located attackers

- **Physical contact-based**
  - ShaVe/ShaCK [TMC’09]: shake two devices together
  - H2H [CCS’13]: measure heartbeat data

  ![Insecure](image)
  **More secure but needs special hardware/sensors**
Our Insights

• Most IoT devices (>92%) have a button, knob, and/or small touchscreen
• Given a user wearing a smartwatch, when she presses a button of an IoT device, both the IoT device and the smartwatch can sense the operation
• Both sides have clocks: timestamps as evidence
T2Pair: System Architecture
T2Pair: System Architecture

Pairing operations

Press Release Press

Time

Extract Evidence 1

Extract Evidence 2

Cryptographic Protocol
T2Pair: System Architecture

Agree on a key?
Yes
No
Pairing succeeds
Pairing fails
Pairing Operations

- Pressing the button a few times
- Twisting the knob back and forth
- Zig-zag swiping on the touchscreen
Sensing Physical Operations

- Correlation between button events and IMU data
Threat Model and Countermeasures

• Mimicry attacks: an adversary mimics a user to press a device to pair it with the user’s smartwatch
  – Countermeasure: random pauses (enforced automatically)

• Man-in-the-Middle attacks
  – Countermeasure: faithful fuzzy commitment
  – Why fuzzy commitment?
    • two pieces of evidence are similar but not identical

• Online brute-force attacks
  – Countermeasure: Zero-knowledge password proof

• Offline brute-force attacks
  – Countermeasure: Diffie-Hellman Encrypted Key Exchange
# Pairing Protocol

<table>
<thead>
<tr>
<th>Device $d_1$</th>
<th>Device $d_2$</th>
</tr>
</thead>
</table>
| **Phase 1: Initialization**

*Initiates the pairing*

$E_{d_1} = Time\_Int\_Seq(d_1)$

if self-checking fails, aborts

$E_{d_2} = Time\_Int\_Seq(d_2)$

if self-checking fails, aborts and reminds the user

| **Phase 2: Extracting Evidence** |

$\delta = e(E_{d_1}) \oplus \lambda$

| **Phase 3: Fuzzy Commitment** |

1. picks a random value $P \in \mathbb{F}_{2^k}^m$
2. $\lambda \in \mathbb{F}_{2^k}^n$ encoded $\rightarrow$ RS($2^k$, $m$, $n$, $P$)
3. commits: $\delta = e(E_{d_1}) \oplus \lambda$

$\delta$ $\rightarrow$

4. decommits: $\lambda' = e(E_{d_2}) \oplus \delta$

5. $P' \leftarrow$ decode RS($2^k$, $m$, $n$, $\lambda'$)

| **Phase 4: PAKE** |

6. picks $a; A = g^a \mod p; w = h(P)$

7. picks $b; B = g^b \mod p; w' = h(P')$

8. $K' = A^b \mod p; \text{picks a challenge } C_1$

9. $K = B^a \mod p$

10. picks a challenge $C_2$

11. if $C_1$ is not received, aborts

12. if $C_2$ is not received, aborts
Traditional Encoding Does Not Work Well

\[
\begin{align*}
\text{Ham}(121, 57) &= 1 \\
\text{Ham}(127, 128) &= 8
\end{align*}
\]
Traditional Encoding Does Not Work Well

\[
\begin{align*}
\text{"121": } & \quad 0111 \ 1001 \\
\text{"57": } & \quad 0011 \ 1001 \\
\text{"128": } & \quad 1000 \ 0000 \\
\text{"127": } & \quad 0111 \ 1111
\end{align*}
\]

\[\text{Ham}(121, 57) = 1\]
\[\text{Ham}(127, 128) = 8\]

- **Our solution**: reduce an interval value by dividing a base value and represent it by counting “1”.

\[
n = \lfloor i/B \rfloor \\
e(i) = \underbrace{1, 1, \ldots, 1}_n, 0, \underbrace{0, \ldots, 0}_{L}
\]
Evaluation

- Accuracy
- Resilience to mimicry attacks
- Randomness and entropy
- Parameter studies
  - Operation number, IMU sampling rate, postures, …
- Usability
Both FRR and FAR can be improved by adding random pauses.
- Pauses: 0.00 FAR and low FRR for button, knob and screen.

Button without pause (FRR: 0.10, FAR: 0.02)

Button with pause (FRR: 0.03, FAR: 0.00)
Resilience to Trained Mimicry Attacks

- The attacker practices well (i.e., training), stands close to the target user, and has a clear view

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>button</td>
<td>0.20</td>
<td>0.27</td>
<td>0.27</td>
<td>0.40</td>
<td>0.20</td>
<td>0.20</td>
<td>0.33</td>
<td>0.27</td>
<td>0.33</td>
<td>0.27</td>
<td>0.274</td>
</tr>
<tr>
<td></td>
<td>knob</td>
<td>0.27</td>
<td>0.20</td>
<td>0.27</td>
<td>0.33</td>
<td>0.20</td>
<td>0.13</td>
<td>0.27</td>
<td>0.20</td>
<td>0.40</td>
<td>0.13</td>
<td>0.240</td>
</tr>
<tr>
<td></td>
<td>screen</td>
<td>0.20</td>
<td>0.07</td>
<td>0.13</td>
<td>0.27</td>
<td>0.33</td>
<td>0.20</td>
<td>0.13</td>
<td>0.20</td>
<td>0.20</td>
<td>0.07</td>
<td>0.180</td>
</tr>
<tr>
<td>Yes</td>
<td>button</td>
<td>0.0</td>
<td>0.07</td>
<td>0.0</td>
<td>0.07</td>
<td>0.07</td>
<td>0.07</td>
<td>0.07</td>
<td>0.0</td>
<td>0.07</td>
<td>0.0</td>
<td>0.040</td>
</tr>
<tr>
<td></td>
<td>knob</td>
<td>0.0</td>
<td>0.0</td>
<td>0.07</td>
<td>0.07</td>
<td>0.0</td>
<td>0.07</td>
<td>0.07</td>
<td>0.0</td>
<td>0.13</td>
<td>0.0</td>
<td>0.040</td>
</tr>
<tr>
<td></td>
<td>screen</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.07</td>
<td>0.07</td>
<td>0.0</td>
<td>0.0</td>
<td>0.13</td>
<td>0.0</td>
<td>0.027</td>
</tr>
</tbody>
</table>
Randomness and Entropy

- Randomness
  - NIST statistical test ($p > 0.01$) confirms randomness.
  - Interval data is abstracted into normal distributions.

- Entropy

\[ E_i = \frac{1}{2} \log_2 (2\pi e \sigma^2) \]
\[ I_E = n_1 \cdot E_1 + n_2 \cdot E_2 + \log_2 \left( \frac{n_1 + n_2}{n_2} \right) \]

<table>
<thead>
<tr>
<th>Device</th>
<th>Entropy (bits)</th>
<th>Bit Rate (bit/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>button</td>
<td>34.3 – 38.5</td>
<td>10.3 – 13.2</td>
</tr>
<tr>
<td>knob</td>
<td>34.3 – 37.9</td>
<td>10.6 – 13.6</td>
</tr>
<tr>
<td>screen</td>
<td>32.3 – 36.6</td>
<td>11.6 – 14.8</td>
</tr>
</tbody>
</table>
Limitations

• If an attacker uses a camera that points at the user performing authentication, T2Pair is vulnerable to online attacks
  – Offline attacks cannot succeed due to DH

• Still a low chance for trained mimicry attacks
  – More random pauses

• Not usable to hold a large phone and twist a small knob
Takeaways

• Prior IoT pairing approaches are insecure or inapplicable to constrained IoT devices
  – We propose the first secure and usable approach

• Simple operations (e.g., pressing a button, twisting a knob) are used for pairing

• Faithful fuzzy commitment: better accuracy

• Zero-knowledge password proof: turn a low-entropy “password” to a high-entropy key
Thank you!

Qiang Zeng
(zeng1@cse.sc.edu)