BlueMirror: Reflections on Bluetooth Pairing and Provisioning Protocols

Tristan Claverie, José Lopes Esteves

Agence nationale de la sécurité des systèmes d’information

May 27, 2021
ANSSI, Wireless Security Laboratory

- Electromagnetic Security (TEMPEST, IEMI)
- Wireless protocols
- Signal processing
- Simulations, measures, electromagnetism
- Embedded systems
ANSSI, Wireless Security Laboratory

- Electromagnetic Security (TEMPEST, IEMI)
- Wireless protocols
- Signal processing
- Simulations, measures, electromagnetism
- Embedded systems

Tristan Claverie

- Wireless protocol security
- Internet of Things
- DVB, Bluetooth LE, Classic, Mesh, LoRaWAN
- Software-Defined Radio
Outline of the presentation

1. Introduction to Bluetooth Classic, LE, Mesh
2. Scope of the study
3. Results
4. Conclusion
1. Introduction to Bluetooth Classic, LE, Mesh
Presentation of Bluetooth technologies

Bluetooth Classic (BT)
- Standardised in 1999
- Communication protocol
- 2+ devices communicate together
- Spec: *Bluetooth Core Specification*

Use cases:
- Cars, Smartphones
- Audio devices

Bluetooth Low Energy (BLE)
- Standardised in 2010
- Communication protocol
- 2 devices communicate together
- Spec: *Bluetooth Core Specification*

Use cases:
- Smartphones
- Smart* (watches, bands...)
- Medical devices
Bluetooth Mesh (BM)

- Standardised in 2017
- Uses BLE PHY/LNK layers
- Network of devices communicate together
- Several applications (light, sensors...) in a Network.

Spec : Bluetooth Mesh \{Model, Profile\} Specification

Use cases :
- Connected homes
Security in Bluetooth technologies

BT / BLE security goals

- Confidentiality
- Integrity
- Authenticity (opt.)

Symmetric secrets:
- EncKey - protect communication between two devices (LK, LTK, ...)

BM security goals

- Confidentiality
- Integrity
- Authenticity (opt.)

- Segregation of applications inside a network

Symmetric secrets:
- NetKey - communicate on the network
- AppKey - send/receive applicative data
- DevKey - device configuration

=> A Key agreement protocol is used to exchange those symmetric secrets
Security in Bluetooth technologies

BT / BLE security goals
- Confidentiality
- Integrity
- Authenticity (opt.)

BM security goals
- Confidentiality
- Integrity
- Authenticity (opt.)
- Segregation of applications inside a network

Symmetric secrets:
- EncKey - protect communication between two devices (LK, LTK, ...)
- NetKey - communicate on the network
- AppKey - send/receive applicative data
- DevKey - device configuration

A Key agreement protocol is used to exchange those symmetric secrets.
Security in Bluetooth technologies

BT / BLE security goals
- Confidentiality
- Integrity
- Authenticity (opt.)

Symmetric secrets:
- EncKey - protect communication between two devices (LK, LTK, ...)

BM security goals
- Confidentiality
- Integrity
- Authenticity (opt.)
- Segregation of applications inside a network
Security in Bluetooth technologies

BT / BLE security goals
- Confidentiality
- Integrity
- Authenticity (opt.)

Symmetric secrets:
- EncKey - protect communication between two devices (LK, LTK, ...)

BM security goals
- Confidentiality
- Integrity
- Authenticity (opt.)
- Segregation of applications inside a network

Symmetric secrets:
- NetKey - communicate on the network
- AppKey - send/receive applicative data
- DevKey - device configuration

A Key agreement protocol is used to exchange those symmetric secrets
Security in Bluetooth technologies

BT / BLE security goals
- Confidentiality
- Integrity
- Authenticity (opt.)

Symmetric secrets:
- EncKey - protect communication between two devices (LK, LTK, ...)

BM security goals
- Confidentiality
- Integrity
- Authenticity (opt.)
- Segregation of applications inside a network

Symmetric secrets:
- NetKey - communicate on the network
- AppKey - send/receive applicative data
- DevKey - device configuration

=> A Key agreement protocol is used to exchange those symmetric secrets
BT / BLE: **Pairing**

- Happens between an Initiator and a Responder
- Used when two devices have no previously shared secret
- At the end of the procedure, both devices share EncKey
- May be authenticated
- Several Pairing protocols exist, not the same between BT/BLE
BT / BLE : **Pairing**
- Happens between an Initiator and a Responder
- Used when two devices have no previously shared secret
- At the end of the procedure, both devices share EncKey
- May be authenticated
- Several Pairing protocols exist, not the same between BT/BLE

BM : **Provisioning**
- Happens between a Provisioner and a Device
- Used when a device wants to join a Network
- At the end of the procedure, the Device receives NetKey and derives DevKey.
- May be authenticated
- Several Provisioning protocols exist
Pairing:

- Feature Exchange
- Key Exchange
- Authentication (opt)
- Devices share EncKey
Pairing:

- Initiator
  - Feature Exchange
  - Key Exchange
  - Authentication (opt)
- Responder
  - Devices share EncKey

Provisioning:

- Provisioner
  - Feature Exchange
  - Key Exchange
  - Authentication (opt)
- Device
  - Key Distribution
  - Device has NetKey, ...
Pairing method depends on: supported version, user interaction.

<table>
<thead>
<tr>
<th>Technology</th>
<th>BT</th>
<th>BLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pairing Mode</td>
<td>Legacy</td>
<td>Secure Simple Pairing</td>
</tr>
<tr>
<td></td>
<td>PIN Pairing</td>
<td>JustWorks</td>
</tr>
<tr>
<td>Pairing Method</td>
<td>Passkey Entry</td>
<td>Passkey Entry</td>
</tr>
<tr>
<td></td>
<td>Numeric Comparison</td>
<td>Out of Band</td>
</tr>
<tr>
<td></td>
<td>Out of Band</td>
<td>Out of Band</td>
</tr>
</tbody>
</table>

- BLE: Legacy/Secure are **different** protocols => Legacy JW ≠ Secure JW
- BLE/BT: SSP and LESP are the **same** protocols => SSP JW ≈ LESP JW
8 kinds of Provisioning

Provisioning depends on:

- How the key exchange is performed (in-band, out of band)
- How authentication data is exchanged (no authentication, input data, output data, static data)
- No specific names for the 8 variants of the Provisioning protocol.

<table>
<thead>
<tr>
<th>In-band; No auth</th>
<th>Out of Band; No auth</th>
</tr>
</thead>
<tbody>
<tr>
<td>In-band; Input</td>
<td>Out of Band; Input</td>
</tr>
<tr>
<td>In-band; Output</td>
<td>Out of Band; Output</td>
</tr>
<tr>
<td>In-band; Static</td>
<td>Out of Band; Static</td>
</tr>
</tbody>
</table>
At a high-level, all Bluetooth key agreement fall into one of three categories:

- **Unauthenticated**: key agreement is not authenticated
- **Authenticated**: key agreement is authenticated
- **Out of Band**: security properties come from an unspecified communication channel

<table>
<thead>
<tr>
<th>Technology</th>
<th>BT</th>
<th>BLE</th>
<th>BM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pairing Mode</strong></td>
<td>Legacy</td>
<td>Secure Simple Pairing</td>
<td>Legacy Pairing</td>
</tr>
<tr>
<td>PIN Pairing</td>
<td>JustWorks</td>
<td>JustWorks</td>
<td>JustWorks</td>
</tr>
<tr>
<td>Passkey Entry</td>
<td>Passkey Entry</td>
<td>Passkey Entry</td>
<td>In-band; Input</td>
</tr>
<tr>
<td>Numeric Comparison</td>
<td>Out of Band</td>
<td>Numeric Comparison</td>
<td>In-band; Output</td>
</tr>
<tr>
<td>Out of Band</td>
<td>Out of Band</td>
<td>Out of Band</td>
<td>In-band; Static</td>
</tr>
</tbody>
</table>
2. Scope of the study
## State of the Art

<table>
<thead>
<tr>
<th>Technology</th>
<th>BT</th>
<th>BLE</th>
<th>BM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pairing Mode</td>
<td>Legacy Secure Simple Pairing</td>
<td>Legacy Pairing</td>
<td>LE Secure Pairing</td>
</tr>
<tr>
<td>Pairing/Provisioning Method</td>
<td>PIN Pairing</td>
<td>JustWorks</td>
<td>JustWorks</td>
</tr>
<tr>
<td></td>
<td>Passkey Entry</td>
<td>Passkey Entry</td>
<td>Passkey Entry</td>
</tr>
<tr>
<td></td>
<td>Numeric Comparison</td>
<td>Out of Band</td>
<td>Numeric Comparison</td>
</tr>
<tr>
<td></td>
<td>Out of Band</td>
<td>Out of Band</td>
<td>Out of Band</td>
</tr>
</tbody>
</table>
**Goal**: Study authenticated Bluetooth protocols  
**Means**: Reflection attacks

<table>
<thead>
<tr>
<th>Technology</th>
<th>BT</th>
<th>BLE</th>
<th>BM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pairing Mode</strong></td>
<td>Legacy</td>
<td>Secure Simple Pairing</td>
<td>Legacy Pairing</td>
</tr>
<tr>
<td><strong>Pairing/ Provisioning Method</strong></td>
<td>PIN Pairing</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Passkey Entry</td>
<td>Passkey Entry</td>
<td>Passkey Entry</td>
</tr>
<tr>
<td></td>
<td>Numeric Comparison</td>
<td></td>
<td>Numeric Comparison</td>
</tr>
</tbody>
</table>
Building block in Bluetooth authentication protocols: commitment protocol
Reflection attacks: concept

Building block in Bluetooth authentication protocols: commitment protocol

Example of a reflection attack

<table>
<thead>
<tr>
<th>Initiator</th>
<th>Responder</th>
<th>Attacker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Draw nonce Ni, compute Ci = f(Ni, ...)</td>
<td>Ci</td>
<td>Ci</td>
</tr>
<tr>
<td>Cr</td>
<td>Draw nonce Nr, compute Cr = f(Nr, ...)</td>
<td>Nr</td>
</tr>
<tr>
<td>Ni</td>
<td>Verify Ni matches Ci</td>
<td>Ni</td>
</tr>
<tr>
<td>Nr</td>
<td>Verify Nr matches Cr</td>
<td></td>
</tr>
</tbody>
</table>
Reflection attacks: impact

Goals:
- Complete authentication protocol, do not retrieve encryption key
- Complete authentication protocol, retrieve encryption key

In the literature:
- Reflection in TLS 1.3 PSK mode, no encryption key at the end [DG19]
- Theoretical reflection in a BT security protocol, no encryption key at the end [ATR20a]

=> Easy to patch in implementations, but should be made impossible by good protocols.
3. Results
Used for BT SSP, BLE SP
One device displays a passkey, user inputs in on the other.

Passkey is 20 bits long
Secure Passkey Entry

Used for BT SSP, BLE SP
One device displays a passkey, user inputs in on the other.

Passkey is 20 bits long

1 Feature Exchange

- Feature Exchange
- Key Exchange
- Authentication
  - PKi
  - PKr
  - Ci1
  - Cr1
  - Ni1
  - Nr1
  - Ei
  - Er

20 rounds of commitments
Used for BT SSP, BLE SP
One device displays a passkey, user inputs in on the other.

Passkey is 20 bits long

1. Feature Exchange
2. Diffie-Hellman key exchange
Secure Passkey Entry

Used for BT SSP, BLE SP
One device displays a passkey, user inputs in on the other.

Passkey is 20 bits long

1. Feature Exchange
2. Diffie-Hellman key exchange
3. Commitment protocol uses 1 bit of the passkey
Secure Passkey Entry

Used for BT SSP, BLE SP
One device displays a passkey, user inputs in on the other.

Passkey is 20 bits long

1. Feature Exchange
2. Diffie-Hellman key exchange
3. Commitment protocol uses 1 bit of the passkey
4. 20 rounds of commitments
Secure Passkey Entry

Used for BT SSP, BLE SP
One device displays a passkey, user inputs in on the other.

Passkey is 20 bits long

1. Feature Exchange
2. Diffie-Hellman key exchange
3. Commitment protocol uses 1 bit of the passkey
4. 20 rounds of commitments
5. Final exchange of messages
1. Reflect Initiator’s public key, then all rounds

\[ \text{Reflect Initiator’s public key, then all rounds} \]
Secure Passkey Entry: Impersonation

1. Reflect Initiator's public key, then all rounds.
Secure Passkey Entry: Impersonation

1. Reflect Initiator’s public key, then all rounds

2. \( \Rightarrow \) Attacker can learn the passkey: retrieve \( p_k \) from \((C_{x_k}, N_{x_k})(\text{Lindell, 2008 \cite{Lin08}})\)
1. Reflect Initiator’s public key, then all rounds

2. => Attacker can learn the passkey: retrieve $p_k$ from $(C_x, N_x)(\text{Lindell, 2008 [Lin08]})$

3. => Use the passkey to authenticate to the legitimate responder
Secure Passkey Entry: Impersonation

1. Reflect Initiator’s public key, then all rounds

2. => Attacker can learn the passkey: retrieve $p_k$ from $(C_{x_k}, N_{x_k})$ (Lindell, 2008 [Lin08])

3. => Use the passkey to authenticate to the legitimate responder

- Attacker ends impersonating Initiator, with EncKey
- Works in BT SSP, BLE SP
- Initiator has failed Pairing

Details and variants in the proceedings
Authenticated Provisioning: Key exchange is performed in-band; one device outputs AuthData and the user inputs it on the other end.

**Provisioning protocol**

**Authenticated Provisioning:** Key exchange is performed in-band; one device outputs AuthData and the user inputs it on the other end.

**Commitment protocol:**

- $CK = f(DHKey, FeatureExchange)$
- $Cp = AES-CMAC(CK, Np||AuthValue)$
- $Cd = AES-CMAC(CK, Nd||AuthValue)$

Trivial reflection attack (cf. proceedings)

Cryptographic misuse!
Authenticated Provisioning: Key exchange is performed in-band; one device outputs AuthData and the user inputs it on the other end.

- AuthData is padded into AuthValue.
- AuthValue, nonces and confirmations are 16 bytes long.

Commitment protocol:

\[
CK = f(DHKey, FeatureExchange)
\]

\[
Cp = AES-CMAC_{CK}(Np||AuthValue)
\]

\[
Cd = AES-CMAC_{CK}(Nd||AuthValue)
\]

- Trivial reflection attack (cf. proceedings)
- Cryptographic misuse!
Provisioning: Cryptographic misuse

Problem: CMAC mode is not pre-image resistant => with known key, one block of plaintext leaks.

AES-CMAC: RFC4493

\[
CK_1 = f(CK) \\
C = \text{AES-CMAC}_{CK}(N || AuthValue) \\
C = \text{AES}_{CK}(\text{AES}_{CK}(N) \oplus CK_1 \oplus AuthValue)
\]

Retrieve AuthValue with \((CK, N, C)\):

\[
\text{AuthValue} = \text{AES}_{CK}^{-1}(C) \oplus \text{AES}_{CK}(N) \oplus CK_1
\]

Retrieve \(N\) with \((CK, \text{AuthValue}, C)\):

\[
N = \text{AES}_{CK}^{-1}(\text{AES}_{CK}^{-1}(C) \oplus CK_1 \oplus AuthValue)
\]
Provisioning: Attack

1. Send public key
Provisioning: Attack

1. Send public key
2. Send random confirmation
Provisioning : Attack

1. Send public key
2. Send random confirmation
3. Retrieve AuthValue
1. Send public key
2. Send random confirmation
3. Retrieve AuthValue
4. Craft nonce
Provisioning : Attack

Impersonation :
- Gains NetKey, may get AppKey(s)
- Legitimate Device couldn’t join the Network
Impersonation:
- Gains NetKey, may get AppKey(s)
- Legitimate Device couldn’t join the Network

MitM:
- Gain DevKey of the legitimate device
- Legitimate device appears to have joined the network
- Not patchable at the implementation level => specification update
Secure Passkey Entry
Before:
- If passkey is perfectly random, no problem \cite{Lin08}
This work:
- If passkey is perfectly random, problems remain
Secure Passkey Entry
Before:
- If passkey is perfectly random, no problem [Lin08]
This work:
- If passkey is perfectly random, problems remain

Mesh
Before:
- No analysis of Provisioning protocol
Related:
- Malleable commitment in BLE Legacy
  Passkey Entry $\Rightarrow$ Authentication is broken [Ros13]
This work:
- Malleable commitment in BM
  Provisioning $\Rightarrow$ Authentication is broken
- In total, 7 attacks discovered
- Results were validated experimentally on real-world implementations
- Responsible disclosure to Bluetooth SIG in September, 2020 => 6 CVEs allocated

<table>
<thead>
<tr>
<th>Attack</th>
<th>Technology</th>
<th>Security</th>
<th>Attacker position</th>
<th>Key recovered</th>
<th>Impact</th>
<th>Target</th>
<th>Test</th>
<th>Weakness</th>
<th>CVE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLE-A</td>
<td>BT</td>
<td>X</td>
<td>Legacy</td>
<td>Spoofer</td>
<td>No</td>
<td>Impersonation</td>
<td>Initiator</td>
<td>Complete</td>
<td>Reflection</td>
</tr>
<tr>
<td></td>
<td>BLE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BT-A</td>
<td>BT</td>
<td>X</td>
<td>Legacy</td>
<td>Spoofer</td>
<td>Yes</td>
<td>Impersonation</td>
<td>Initiator</td>
<td>Partial</td>
<td>Reflection</td>
</tr>
<tr>
<td></td>
<td>BLE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PE-A2</td>
<td>BT</td>
<td>X</td>
<td>Secure</td>
<td>MitM</td>
<td>Yes</td>
<td>Impersonation</td>
<td>Responder</td>
<td>Complete</td>
<td>Reflection</td>
</tr>
<tr>
<td></td>
<td>BLE</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PE-A1</td>
<td>BT</td>
<td>X</td>
<td>Secure</td>
<td>Spoofer</td>
<td>No</td>
<td>Impersonation</td>
<td>Initiator</td>
<td>Partial</td>
<td>Reflection</td>
</tr>
<tr>
<td></td>
<td>BLE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M-A1</td>
<td>BT</td>
<td>X</td>
<td>Secure</td>
<td>Spoofer</td>
<td>Yes</td>
<td>Impersonation</td>
<td>Provisioner</td>
<td>Complete</td>
<td>Reflection</td>
</tr>
<tr>
<td></td>
<td>BLE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M-A2</td>
<td>BT</td>
<td>X</td>
<td>Secure</td>
<td>Spoofer</td>
<td>Yes</td>
<td>Impersonation</td>
<td>Provisioner</td>
<td>Complete</td>
<td>Crypto</td>
</tr>
<tr>
<td></td>
<td>BLE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BM</td>
<td></td>
<td></td>
<td>MitM</td>
<td></td>
<td>MitM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M-A3</td>
<td>BT</td>
<td>X</td>
<td>Secure</td>
<td>Spoofer</td>
<td>Yes</td>
<td>Impersonation</td>
<td>Provisioner</td>
<td>Complete</td>
<td>Crypto</td>
</tr>
<tr>
<td></td>
<td>BLE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BM</td>
<td></td>
<td></td>
<td>MitM</td>
<td></td>
<td>MitM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Authenticated key agreements

<table>
<thead>
<tr>
<th>Technology</th>
<th>BT</th>
<th>BLE</th>
<th>BM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pairing Mode</strong></td>
<td><strong>Legacy</strong></td>
<td><strong>Legacy Pairing</strong></td>
<td><strong>LE Secure Pairing</strong></td>
</tr>
<tr>
<td>PIN Pairing</td>
<td>JustWorks</td>
<td>JustWorks</td>
<td>JustWorks</td>
</tr>
<tr>
<td>Passkey Entry</td>
<td>Passkey Entry</td>
<td>Passkey Entry</td>
<td>In-band; Input</td>
</tr>
<tr>
<td>Numeric Comparison</td>
<td>Out of Band</td>
<td>Numeric Comparison</td>
<td>In-band; Output</td>
</tr>
<tr>
<td>Out of Band</td>
<td></td>
<td></td>
<td>In-band; Static</td>
</tr>
</tbody>
</table>
Authenticated key agreements

- Secure key agreements according to the specification [Blu19a, Blu19b]

<table>
<thead>
<tr>
<th>Technology</th>
<th>BT</th>
<th>BLE</th>
<th>BM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pairing Mode</td>
<td>Legacy</td>
<td>Secure Simple Pairing</td>
<td>Legacy Pairing</td>
</tr>
<tr>
<td>PIN Pairing</td>
<td>JustWorks</td>
<td>JustWorks</td>
<td>JustWorks</td>
</tr>
<tr>
<td>Passkey Entry</td>
<td>Passkey Entry</td>
<td>Passkey</td>
<td>Passkey</td>
</tr>
<tr>
<td>Numeric Comparison</td>
<td>Out of Band</td>
<td>Numeric Comparison</td>
<td>In-band; Output</td>
</tr>
<tr>
<td>Out of Band</td>
<td>Out of Band</td>
<td>In-band; Static</td>
<td>Out of Band; Static</td>
</tr>
</tbody>
</table>
Authenticated key agreements

- Secure key agreements according to the specification [Blu19a, Blu19b]
- Successfully attacked key agreements in this study

<table>
<thead>
<tr>
<th>Technology</th>
<th>BT</th>
<th>BLE</th>
<th>BM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pairing Mode</td>
<td>Legacy</td>
<td>Legacy Pairing</td>
<td>LE Secure Pairing</td>
</tr>
<tr>
<td></td>
<td>Secure Simple</td>
<td>Legacy Pairing</td>
<td></td>
</tr>
<tr>
<td>Pairing/Provisioning Method</td>
<td>PIN Pairing</td>
<td>JustWorks</td>
<td>JustWorks</td>
</tr>
<tr>
<td></td>
<td>Passkey Entry</td>
<td>Passkey Entry</td>
<td>Passkey Entry</td>
</tr>
<tr>
<td></td>
<td>Numeric Comparison</td>
<td>Out of Band</td>
<td>Numeric Comparison</td>
</tr>
<tr>
<td></td>
<td>Out of Band</td>
<td>Out of Band</td>
<td>In-band; Static</td>
</tr>
</tbody>
</table>

4. Conclusion
Conclusion

- Very informative cases of real-world reflection attacks, with key retrieval
- Numeric Comparison appears (again) to be the most resistant Pairing method
- Most of the problems we found (reflection attacks) can be patched in implementations; some will require a redesign
- Three out of three Bluetooth technologies required complete redesign of initial key agreements protocols

- Bluetooth retrocompatibility may pose new problems in BM
- Don’t rely on Bluetooth built-in security
- If you have to, pair/provision devices in controlled environments (e.g. Faraday cage)
Questions?

Contact

- tristan.claverie@ssi.gouv.fr


