Bluetooth Security

Tarek Sheha
Media Informatics - WS 2010/2011
Seminar: Mobile Security @ B-IT
February 9th, 2010
Prof. Joachim von zur Gathen
Daniel Loebenberger and Yona Raekow
Outline

- Introduction
- Threats in Computer Networks
- BT Security Architecture
- BT Security Philosophy
- Key Types: Link Keys
- Algorithms
- Key Management Protocol
- BT Security Weaknesses
- Recommendations
Introduction

- Bluetooth is developed by SIG in 1998.
- Always-on, low power, low cost, short range, wireless technology for devices.
- Wireless Personal Area Network (WPAN) is established.
- Cable Replacement and Ease of Sharing.
Connectivity

- Ad-Hoc Network by forming a star shaped cluster called **Piconet** (Master and Slaves).
- Using a Bluetooth Access Point (BT-AP).
- Up to a maximum of Seven active slave and 255 inactive ones in a connection.
- **Scatternet** can be formed by sharing of common slaves or different roles of one device in two piconets.
Connectivity

Bluetooth Piconet 3
User C’s PDA
Slave of Piconet 3

User C’s Laptop
Master of Piconet 3
Slave of Piconet 1

Bluetooth Piconet 2
User B’s PDA
Slave of Piconet 2

User B’s Mobile Phone
Slave of Piconet 2

Bluetooth Piconet 1
User A’s Laptop
Master of Piconet 1

User B’s Laptop
Master of Piconet 2
Slave of Piconet 1

Piconet 2 and 3: Users B and C Share Contact Information with Their Own Personal Devices.

Piconet 1: Laptops of Separate Users in a Meeting
Sharing Files and Contact Information (e.g., Meeting Attendee List).
Technical Specifications

- Devices can exchange data up to 723 Kbps.
- Operates in the unlicensed radio range of 2.45 GHz.
- The range of a BT device is divided into one of three classes according to the power level:
  - **Class 1**: High Power of 100mW and a range of 100 meters.
  - **Class 2**: Medium Power of 2.5mW and a range of 10 meters.
  - **Class 3**: Low Power of 1mW and a range of 0.1 – 10 meters.
Link Manager Protocol (LMP)

- Handling the connection, authentication, authorization, encryption, and key management between devices.

- Used by each device to keep track of connected devices.

- Communicate together via PDU.

- New Connections: (1) Inquiry (2) Page

- Pairing start afterwards, opposite device is (trusted).
Threats in Computer Networks

- **Disclosure**: a threat against the confidentiality of the information.

- **Integrity**: a threat that involves an unauthorized change of the information.

- **Denial of Service (DoS)**: a threat against the availability of the system.
Bluetooth Security Architecture
Generic Access Profile (GAP)

- Supports 3 Security Services:
  - Authentication
  - Authorization
  - Confidentiality
- Supports 3 Security Modes
- Supports 2 Security Levels for Devices and 3 Security Levels for Services
- Keys Generation, Exchange, Random Numbers, and etc...
Security Services

- **Authentication**
  - Verifying the identity of communicating devices. It is commonly done through the use of PIN.

- **Authorization**
  - Allowing the control of resources, checking whether a device is allowed to use a service or not.

- **Confidentiality (Encryption)**
  - Protecting the information exchange in such a way that no one can understand it except the designated recipient.
Security Modes

- **Mode 1: Non-secure**
  No security measures. No authentication, authorization, or encryption. Both devices and connection are vulnerable to attack.

- **Mode 2: Service-level enforced security**
  Flexibility in security measures. ACL link can be established in a non secure manner. Security procedures are enforced when there is a service request.

- **Mode 3: Link-level enforced security**
  Strict in security measures. Security measures are initiated when the ACL link is being established.
Security Modes

- The Difference between Security Modes 2 and 3.

- In Mode 3, three phases are defined:
  - **Initialization phase**: Construction of trust and exchange of keys.
  - **Meeting phase**: Proof of authenticity.
  - **Communication phase**: Secure data exchange.
Security Levels

- **Devices**
  - Trusted
  - Untrusted

- **Services**
  - Accessible to all devices. No need for a PIN or a Password.
  - Authentication only. Need for a Password.
  - Authentication and authorization. Need for a Password followed by an authorization procedure.
Philosophy of Bluetooth Security

Chain of events: (Briefly)

1) Each device calculate its own Unit Key (\(K_A\) and \(K_B\)).
2) When they meet, devices begin with PIN.
3) From PIN, derive Kinit.
4) After \(K_{\text{init}}\), Pairing Process occurs.
5) Link Key Exchange using \(K_{\text{init}}: K_A\) or \(K_B\), or a more secured one, \(K_{AB}\)
6) \(K_A\) or \(K_{AB}\) are used in Authetication and Encryption.
Bluetooth Security Overview

- Unit A First Startup
- Unit B First Startup
- Unit - Unit First Handshake
- Unit - Unit following Handshakes
## Security Entities

<table>
<thead>
<tr>
<th>Entity</th>
<th>Description</th>
<th>Length (Bits)</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIN</td>
<td>Personal identification number</td>
<td>8, 16, ..., 128</td>
<td>Private</td>
</tr>
<tr>
<td>BD_ADDR</td>
<td>Bluetooth device address</td>
<td>48</td>
<td>Public</td>
</tr>
<tr>
<td>$K_{\text{init}}$</td>
<td>Initialization key</td>
<td>128</td>
<td>Private</td>
</tr>
<tr>
<td>$K_A$</td>
<td>Unit key</td>
<td>128</td>
<td>Private</td>
</tr>
<tr>
<td>$K_{AB}$</td>
<td>Combination key</td>
<td>128</td>
<td>Private</td>
</tr>
<tr>
<td>$K_{\text{master}}$</td>
<td>Master key</td>
<td>128</td>
<td>Private</td>
</tr>
<tr>
<td>$K_C$</td>
<td>Encryption key</td>
<td>8, 16, ..., 128</td>
<td>Private</td>
</tr>
<tr>
<td>IN_RAND</td>
<td>Random number for generating $K_{\text{init}}$</td>
<td>128</td>
<td>Public</td>
</tr>
<tr>
<td>LK_RAND</td>
<td>Random number for generating $K_{AB}$</td>
<td>128</td>
<td>Private</td>
</tr>
<tr>
<td>AU_RAND</td>
<td>Random number for authentication</td>
<td>128</td>
<td>Public</td>
</tr>
<tr>
<td>EN_RAND</td>
<td>Random number for generating $K_C$</td>
<td>128</td>
<td>Public</td>
</tr>
<tr>
<td>SRES</td>
<td>Authentication result</td>
<td>32</td>
<td>Public</td>
</tr>
<tr>
<td>ACO</td>
<td>Authenticated ciphering offset</td>
<td>96</td>
<td>Private</td>
</tr>
</tbody>
</table>
Security Entities

- **Personal Identification Number (PIN)** – 8, 16, ... ,128 bits – **Private.** Fixed, so entered to the device wishing to connect or entered to both devices at the beginning.

- **Bluetooth Device Address (BD_ADDR)** – 48 bits – **Public.** Unique for each device and identified by IEEE.

- **Random Number (RAND)** – 128 bits – **Public.** Every Bluetooth device is equipped with a random number generator that can create as 128-bit random binary number on demand.

- **Encryption Key (Kc)** – 8, 16, ... ,128 bits – **Private.** Used to change plain text into cipher and vice versa.
Key Types: Link Keys

- All security transactions between two or more parties are handled by the link key.
- Regardless of its type, a link key is always 128 bits long.
- They can be either initialization ($K_{\text{init}}$), semi-permanent ($K_A$ or $K_{AB}$), temporary ($K_{\text{master}}$) keys.
Key Types: Link Keys

- **Initialization Key** \( (K_{\text{init}}) \): created once from PIN when two devices with no prior agreement or previous communication meet. Discarded afterwards.
- **Unit Key** \( (K_A) \): created once for a device that has low memory resources.
- **Combination Key** \( (K_{AB}) \): created from the combination of inputs provided by Devices A and B.
- **Master Key** \( (K_{\text{master}}) \): created for the purpose of broadcasting packets to multiple slaves.
Algorithms

- They are all based on Secure and Fast Encryption Routine (SAFER+). A Symmetric Block Cipher operating on a fixed length groups of bits (blocks).

- Keys Generation and Authentication
  - E22 for deriving $K_{\text{init}}$ as initialization key.
  - E21 for deriving $K_A$ and $K_{AB}$ as link keys.
  - E1 for applying authentication procedures.
  - E3 for deriving $K_c$ as encryption key.

- Encryption
  - E0 for Cipher Stream generation.
Key Management Protocol

- **IN_RAND**: For Initialization Key
- **RAND_A**: For Unit Key
- **LK_RAND_A** and **LK_RAND_B**: For Combination Key

<table>
<thead>
<tr>
<th>Link Key Name</th>
<th>Symbol</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialization key</td>
<td>$K_{init}$</td>
<td>$E_{22}(PIN', L', IN_RAND)$</td>
</tr>
<tr>
<td>Unit key</td>
<td>$K_A$</td>
<td>$E_{21}(BD_ADDR_A, RAND_A)$</td>
</tr>
<tr>
<td>Combination key</td>
<td>$K_{AB}$</td>
<td>$E_{21}(BD_ADDR_A, LK_RAND_A) \oplus E_{21}(BD_ADDR_B, LK_RAND_B)$</td>
</tr>
<tr>
<td>Master key</td>
<td>$K_{master}$</td>
<td>$E_{22}(RAND1, RAND2, 16)$</td>
</tr>
</tbody>
</table>
(1) Unit Key

- For Device A or B, $K_A$ or $K_B$ is created:
  - $BD\_ADDR_A$ or $BD\_ADDR_B$
  - $RAND_A$ or $RAND_B$
(2a) Initialization Key
First Handshake

- Used to protect the generation and transfer of other keys that are more secure.
- Used in the following steps as temporary link key.
- At the beginning, calculated by the initiator and then by the responder (Pairing Process).
- \( L' \): Length of PIN, PIN', and \( IN\_RAND \): Random Number

\[ E_{22} \]

- \( K_{init} \)
(2b) Pairing Process
First Handshake

- Starts after the creation of $K_{init}$.
- Essential when two devices have never met before via Bluetooth.
- Same Pin is used in both devices.
- $IN\_RAND\_A$ of Device A is transferred to Device B, so same $K_{init}$ can be calculated. Pairing is successful.
(3) Link Key Generation

- **Two Scenarios are possible:**
  - Link Key is a Unit Key (i.e. $K_A$). Obviously, it is already generated in Device A. So, it has to be transferred securely to Device B.
  - Link Key is a Combination Key (i.e. $K_{AB}$). So, a different procedure takes place.
  - Using a Combination Key as a link key is much more secure than a Unit Key (Why?).
(3) Link Key Generation -1-

$K_A = K_{\text{link}}$

- If Device A is with limited memory, then use its Unit Key ($K_A$) as the Link Key.

- How to transfer ($K_A$), securely, to Device B:
  - Encrypt the Unit Key ($K_A$) with the Initialization Key ($K_{\text{init}}$) by XORing them together.
  - In Device B, decrypt the Unit Key by the Initialization Key.
(3) Link Key Generation -2-

\[ K_A = K_{\text{link}} \]

- Therefore, \( K_A = K_{\text{link}} \).
(3) Link Key Generation - 1 -

\[ K_{AB} = K_{\text{link}} \]

- If memory resources is *not* an issue. Then, it is better to generate a more sophisticated link key by *combining* \( K_A \) and \( K_B \) together.

\[ K_{AB} = K_A \oplus K_B \]

- **Fact:** \( K_{AB} \) should be available at both devices. Each device can calculate its own unit key and *should* calculate the unit key of the other device. Each device knows the Bluetooth Device Address (BD_ADDR) of the other device (Public).

- **Needed:** Each device sends its own Random Number (LK_RAND), encrypted (\( \oplus \)) by \( K_{\text{init}} \), to the other device.
Therefore, A sends \((LK\_RAND_A \oplus K_{init})\) to B, and B sends \((LK\_RAND_B \oplus K_{init})\) to A. Each unit can decode the other's LK\_RAND by the following operation:

\[(LK\_RAND \oplus K_{init}) \oplus K_{init} = LK\_RAND\]

allowing the generation of the Unit Key of each device on the other's side.
As explained, $K_{AB} = K_{\text{link}}$. Note that LK_RAND is the only Private Random Number in BT.
(4) Authentication Challenge – Response action

- Two parties are involved, the verifier and the claimer.
- The verifier challenges the claimer and gets a response from it.
- Challenge: 128-bit Random Number (AU\_RAND_{A}) generated by Verifier.
- The following operation, which is based on SAFER+, is performed at the verifier and afterwards at the claimer:

$$E_1(BD\_ADDR_B, AU\_RAND_A, K_{link})$$
Challenge – Response action

- Output from A is SRES’ and Response from B is SRES.
- If SRES’ = SRES, then authentication is successful.
For Device A or B, $K_c$ is created as a first step:

- Public EN_RAND generated by A and then sent to B.
- Current Link Key ($K_A$ or $K_{AB}$).
- 96-bit Authenticated Ciphering Offset (ACO).
(5) Cipher Stream

As a second step, a Cipher Stream is generated at both sides:

- Encryption Key ($K_c$)
- Master Clock (CLK)
- BD_ADDR_A
Bluetooth Security Weaknesses

- Problems with E0 and E1
  - Based on SAFER+ with security weaknesses, in addition to a slow performance.

- Unit Key
  - $K_A$ is sent to B ($K_A \oplus K_{init}$) as $K_{link}$. Now, B has $K_A$. B can claim A’s identity in a communication with C.
Bluetooth Security Weaknesses

- PIN and Initialization Key
  - Recall, Generation of Initialization Key. Everything is Public except PIN. Brute Force attack is possible.
  - By Knowing PIN, $K_{init}$ is no longer a secret.
  - By Knowing $K_{init}$, several processes will be affected.
    - Combination Key Generation
    - Mutual Authentication
Recommendations

- Use long and sufficiently random PINs. People prefer the default value (i.e. 0000). Some of them use weak values (1234, 5555, ....). Brute Force attack is possible.
- Never use a Unit Key ($K_A$) as a Link Key ($K_{link}$). Better to use a Combination Key ($K_{AB}$).
- Use Security Modes 2 and 3. Turn on Authentication, Authorization, and Encryption functionalities.
Wrap Up!

Device A
- PIN
- Initialization key
- Unit key or combination key
- Authentication
- Encryption key
- Encryption

Device B
- PIN
- Initialization key
- Unit key or combination key
- Authentication
- Encryption key
- Encryption

Tarek Sheha
Thank You 😊