Security Weaknesses in Bluetooth

Sebastian Hanschke

About me

- 21 years old
- since 10/2008: Information Systems in Münster

Contact

Sebastian Hanschke – Busso-Peus-Str. 14 B23 – 48149 Münster

SebastianHanschke@gmx.de

Bluetooth®

(I) Introduction

(1) What is Bluetooth?

(2) How should Bluetooth work?

(1) What is Bluetooth?

mobile devices (e.g. celluar phones) are everywhere

→ commerce platform of unprecedented importance (mobile commerce)

→ short-range wireless LANs

Bluetooth:

standard for local wireless communications

→ cellular phones, wireless headsets, printers, cars, etc.

→ hands-free communication, effortless synchronization

Examples:

- phones connected to wireless headsets, to emergency systems of cars
- computers connected to printers

Bluetooth®

Bluetooth

provides the user with increased possibility

BUT: can be a **powerful weapon for criminals**

→ need for privacy and secrecy

(e.g. for applications relating to telephony)

(2) How should Bluetooth work?

- **Ensure that the information goes to the appropriate device**

→ address each other

→ identifying information, unique (avoid collisions)

- Intended **recipient should recieve**, ideally **no other device should**
- No other device should be able to **identify the sender or the receiver** of the information (user privacy)

→ need to **generate and exchange one or more keys** every time they set up a communication link

→ **encrypt the information sent**

Bluetooth®

(II) Details of Bluetooth Specification 1.0B

(1) Device modes

(2) Addressing

(3) Key establishment protocol

(a) Establishment of Initialization Key

(b) Link Key Generation

- one device has a shortage of memory
- both devices have sufficient memory

(1) Device modes

discoverable mode: devices respond to queries made by unknown devices (e.g. new piconet)

non-discoverable mode: device only respons to devices with whom it has already set up communication

connectable mode: will respond to messages received from already discovered devices

non-connectable mode

Bluetooth®

(2a) Addressing

each device: **unique identifier** (Bluetooth device address)

→ used to establish all communication

in connectable mode: **device access code (DAC)** used to address the device

for each communication: a **particular channel:**

channel identifier (channel access code, CAC)

CAC and DAC:

- function of the master's Bluetooth device address
- always transmitted in the clear

(2b) Frequency hopping pattern

- Determined by Bluetooth address and clock of the master device
- Pseudo-random ordering of the 79 frequencies

Bluetooth®

(3) Key establishment protocol

two new devices who have not yet been exposed to each other:

→ negotiate a key → later used for encryption

Devices do not share a cryptographic key until end of key exchange protocol

→ information send in cleartext

Re-initiate communication:

- either an **old shared key**
- or **negotiate a new one**

Bluetooth®

(3a) Establishment of Initialization Key:

executed **before the link key generation protocol**

→ temporary initialization key

→ used for **encryption of information in the link key generation protocols**

one device chooses a random number, transmits it to the other device

→ both devices compute an initialization key as a function of:

- a shared PIN,
- the Bluetooth device address of the device that received the random number
- the random number itself

Bluetooth®

(3a) Mutual verification

based on **challenge response scheme:**

- a first unit chooses a random number,
- computes a function of:
  - the other device's Bluetooth address,
  - the random number
  - the newly generated key
- the chosen random number is transmitted to the other device
  - computes the same function → responds to the first device
- first device verifies the received value,
- roles are switched

(3a) PIN

length of PIN: determines the security

→ can be chosen between 8 and 128 bits

typically: 4 decimal digits

can either be **fixed** or be **arbitrarily selected** and entered by the user through a user interface

if no PIN available: zero as default

PIN and random numbers either:

- communicated in the clear
- out of band (entered by the user),
- in an encrypted fashion (encryption in application layer)

Bluetooth®

(3b) Link Key Generation I

one device has **shortage of memory**

1) devices establish an initialization key

2) the device with restricted memory:

encrypts its **unit key using the initialization key**

→ resulting ciphertext transmitted

3) receiving unit decrypts the received message using the initialization key

→ uses the resulting key as a link key

→ **both devices use unit key of the sender**

Bluetooth®

(3b) Link Key Generation II

both devices have **sufficient memory ressources**

1) devices establish an initialization key

2) both devices choose random numbers

→ compute a number LK_K as a function of this random number and the unique device address

3) encrypt their random numbers using the initialization key

→ resulting ciphertexts are exchanged

Bluetooth®

4) both units decrypt the received ciphertext using the initalization key

→ both know each others unique device identifiers

→ can computer the other party's number LK_K

5) **both units compute the link key** from LK_KA and LK_KB

6) mutual verification to confirm the success

(III) Vulnerabilities in Bluetooth 1.0B

(1) Eavesdropping and Impersonation

(2) Offline PIN crunching

(a) Eavesdropping

(b) Stealing by participation

(c) Middle-person attack

(3) Location and correlation

(4) Hopping along

(5) A combined attack

(6) Cipher vulnerabilities

(7) Other possible attacks

(1) Eavesdropping and Impersonation

example: printing via Bluetooth in a cyber cafe

→ attacker can eavesdrop, listen to the messages exchanged during pairing (no application layer encryption)

→ can perform a midddle-person attack

→ can obtain a copy of the document, alter the data to be printed

example: eavesdrop on the voice data sent between cell phone and wireless headset

→ leverages on the fact that e.g. during key initialization **data is send without encryption**

Bluetooth®

if an attacker can **determine the initialization key**

→ can compute the link key

→ all encryption keys are generated from the link key

→ decrypt all information send between the devices, impersonate them to each other

if an attacker **learns the unit key of a device**

→ able to impersonate this device to any other device at any time

Bluetooth®

basis of both key generation protocols: protocol for establishment of the initialization key

computed as a function of a PIN, a random number, the bluetooth device address

PIN known to the attacker

- if no PIN available → zero by default
- if PIN transmitted in clear

PIN communicated out of band:

- attacker can learn by **exhaustive search over all possible PINs**

(2) Offline PIN crunching

(a) Eavesdropping

(b) Stealing by participation

(c) Middle-person attack

(2a) Eavesdropping:

attacker eavesdrops on two devices, wishes to determine what key they establish

- **exhaustively guesses all PINs up to a certain lenght**
- **verifies the correctness** of each guess by performing the **verification step of the initialization key protocol**

based on his guess and the **random strings communicated in the clear**

if the result is correct → his guess is correct (with an overwhelming probability)

→ **passive adversary**, does not trasmit

Bluetooth®

(2b) Stealing by participation

attacker performs:

- **one PIN guess** and
- **step 1 of the protocol for establishment of the initialization key (compute the initialization key)**

then: „mutual verification“ (step 2) with the victim device

attacker initiates **first round of the challenge-response protocol**,

→ will output correct if a given initialization key is consistent with:

PIN and random strings sent

→ obtains challenge response transcript from the victim,

→ **computes the corresponding initialization key for each PIN guess**

Bluetooth®

→ **runs the verification algorithm** on

- the computed initialization key and
- the obtained challenge-response transcript (locally, without interaction)

until: verification algorithm outputs correct

→ PIN found, continues key establishment protocol as before

Bluetooth®

back-off method employed to avoid PIN guessing:

(for each subsequent authentication failure, the **waiting interval is increased exponentially**)

does not add any security

→ **attack performed off-line** once the attacker obtains a challenge-response pair

→ exponential back-off benefits the attacker: gives him extra time

→ when **initialization key** is obtained:

- **link key** can be obtained as well
- **encryption keys** are computed from the link key

Bluetooth®

(2c) Middle-person attack

Attacker **obtained link key** used by two devices, two devices have completed communication

→ contacts each one of them, **sets up two new link keys**

→ **middle-person attack**

devices still believe, that they talk to each other

→ attacker can make **both of them slaves or both master**

→ victim devices will **follow different hop sequences**

→ will not see messages they transmit for each other, only messages the attacker chooses to send

→ attacker able to **impersonate the two devices**

(3) Location and Correlation

all packets contain **identifying information**

→ **map the physical whereabouts of users** carrying Bluetooth-enabled devices

→ **bluetooth detecting devices at locations of interest**

may be undesirable for users if their whereabouts can be **correlated**

stalking: users would feel uncomfortable with their location being known

Bluetooth®

anybody could **install a large number of listening nodes**

but: tremendous investment (infrastructure)?

→ not true: place devices at well chosen locations (e.g. airport gates)

already **existing infrastructure**: legally built for another (acceptable) purpose (e.g. entertainment)

information can be **correlated to user identitites** by:

- side information (credit card transaction)
- manual effort (walking around outside congress)

Bluetooth®

attack has Bluetooth **devices distributed over the city** (own: 10$ each or gain control over devices owned by others)

Several versions:

- devices in discoverable mode
- malware (virus, corrupt website)
- index victims by CAC (special hardware needed)

(4) Hopping Along:

to follow a conversation: needs to **listen to all the bands** or: **follow on the frequencies on which they communicate**

device listening to all bands (e.g. U.S. 79, Spain, France: 23) in parallel can easily be built

in order to follow: **pseudo-random hopping sequence** → **can easily be found out**

Bluetooth®

(5) A combined Attack

attacker first **obtains unit or link keys**, later can **pinpoint its position**, it can also eavesdrop on its communication effectively

attacker would:

- determine **device identifier and clock** of his targeted victim (a master device)
- **obtain the hopping sequence**
- **Intercept traffic** on the corresponding bands

→ obtain large portions of the communication

if victim device **moves out of reach of one attacker device** → nearby attacker devices would search for its appearance

Bluetooth®

(6) Cipher vulnerabilities

At first: 128 bit security, but: techniques to attack the cipher:

a) break the secuirty of the cipher requiring 2^100 bit operations

b) time and memory complexity of 2^66

neither constitute a practical threat

but: expose a **weakness in the cipher** which uses 128bit keys

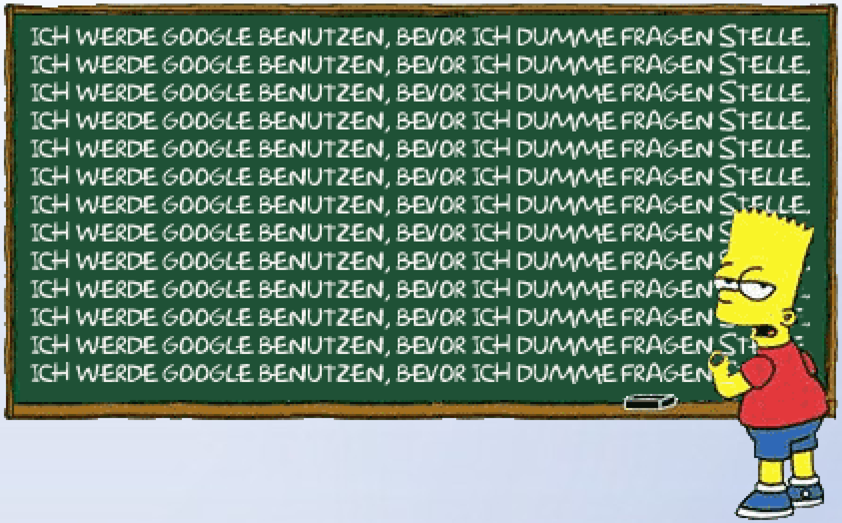
Techniques have improved:

→ Cipher vulnerable (on top of everything else)

Bluetooth®

(7) Other possible attacks

Bluebugging, -printing, -jacking, -snarfing, -casting



(IV) Counter-Measures to our Attacks

(1) PIN length

(2) Protecting unit keys

(3) Application layer security

(4) Policies protecting against middle-person attacks

(5) Physical protection

(6) Pseudonyms against CAC location attacks

(7) Cipher: replacing the cipher, e.g. with AES

(8) Examples

(9) Conclusion

Bluetooth®

(1) PIN length: sufficiently long and sufficiently random, e.g. 64 bit

(attacker will then choose to **attack a different vulnerability** of the system)

(2) Protecting unit keys: device with low memory capabilities may use **large-enough set of keys**, one for each device it communicates with

or: generate such keys by using its **unit key as the input to a pseudo-random generator**

(3) Application layer security: use of **application layer key exchange/encryption methods** to secure communication on top of the existing Bluetooth security measures

e.g. Standard certificate-based methods to defend against middle-person attacks

(4) Policies protecting against middle-person attacks:

middle-person attacks relie on convincing both devices to become masters or slaves

→ policies governing **what devices may take the role of master vs. slave** under what cirumstances

(5) Physical protection: attacks rely on the attacker being able to detect the signals transmitted by the victim devices

→ use of a **Faraday's cage**

(6) Pseudonyms against (CAC) location attacks:

will not be possible for an attacker to perform the CAC location attack

even better: **change the CACs pseudo-randomly** from packet to packet, much like the hopping sequence is derived

devices may **determine what pseudonym or pseudonym seed to use** at the time of their first key exchange, or at any subsequent initiation of communication

but: cannot be software based, has to be performed on the chip itself → does not require any major modifications

(7) Cipher: replacing the cipher with AES

Bluetooth®

(8) Examples

a) **exhaustively searching through PINs:** prevented by sufficiently long PINs (more than around 64 bits) or

b) **middle-person attack:** prevented by public key mechanisms on the application layer or by means of easily implemented security policies

(9) Conclusion

limit success of the discovered attacks:

easy implementable (application layer or relatively simple hardware modifications)

Bluetooth®

(V) Improvements in other Bluetooth versions

(1) Mutual verification after establishment of the Initialization key was eliminated → Offline PIN crunching more difficult

(2) Secure Single Pairing

(a) Passive eavesdropping protection

(b) Man-in-the-middle protection

(3) Conclusion

(1) Secure Single Pairing

(SSP, since Bluetooth Core Specification 2.0)

- Primary goal: **simplify the pairing procedure** for the user
- Secondary goals: **maintain/improve security** in Bluetooth

But: high levels of security ↔ ease-of-use are often at opposite ends

- Security goals: protection against
  - passive eavesdropping
  - Main-in-the-middle (MITM) attacks (= active eavesdropping)
- **Exceed maximum security level** provided by the use of a 16 alphanumeric PIN with the pairing algorithm
- But: **many devices still use 4-digit PIN** or a **fixed PIN** of commonly known values

considered simple for the following reasons:

in most cases, it does not require a user to generate a passkey.

for use-cases not requiring MITM protection, user interaction has been eliminated.

MITM protection can be achieved with a simple equality comparison by the user.

(1a) Passive eavesdropping protection

- Strength of link key **based on the amount of entropy/randomness** in its generation
- **Legacy pairing:** only source of entropy is the **PIN** (typically four digits either: seleted by the user OR fixed)
  - **exhaustive search** to find the PIN
- **With SSP:** recording attack much harder
  - Protection independent of the length of the passkey or other numeric values
  - uses **Elliptic Curve Diffie Hellman (ECDH)** public key cryptography
    - High degree of strength against passive eavesdropping
    - But: may be subject to MITM attacks (much harder to perform)

- SSP has **95 bits of entropy** using FIPS approved P192 elliptic curve

→ at least as good as the entropy in Bluetooth 2.0 using **16 character alphanumeric, case sensitive PIN**

(1b) MAN-IN-THE-MIDDLE protection

- Devices unknowingly connect to a third attacking device that plays the role of the device they are attempting to pair with
- SSP offers two **user assisted numeric methods:**
  - Numerical comparision
  - Passkey entry
- Strength of SSP: **minimize the user impact**
  - Using a six digit number for numerical comparison and Passkey entry
  - In most cases: users can be alerted to the potential presence of a MITM attacker when the connection process fails

Bluetooth®

Modes of operation:

Just works: no user interaction required, device may prompt the user to confirm the pairing process.

- typically used by headsets with very limited IO capabilities,
- more secure than the fixed PIN mechanism
- provides no man in the middle (MITM) protection.

Numeric comparison: both devices have a display, at least one can accept a binary Yes/No user input

- displays a 6-digit numeric code on each device
- user should **compare the numbers** to ensure they are identical,
- **confirm pairing** on the device(s) that can accept an input
- provides MITM protection, assuming the user confirms on both devices

Passkey Entry: between a device with a display and a device with numeric keypad entry or two devices with numeric keypad entry.

first case: display used to show a 6-digit numeric code

→ enter the code on the keypad

second case: user of each device enters the same 6-digit number.

Both cases provide MITM protection.

Out of band (OOB): uses external means of communication

provides only the level of MITM protection of the OOB mechanism

Bluetooth®

(3) Conclusion

- Backward compatibility
- Not many improvements
- Improvements do not dramatically increase security

Any questions?