Norsk Regnesentral
NORWEGIAN COMPUTING CENTER

Note

Bluetooth Security

Note no	DART/05/05
Authors	Hans Jakob Rivertz
Date	03.03.2005

NR Norsk Regnesentral
NORWEGIAN COMPUTING CENTER

Norsk Regnesentral

Norsk Regnesentral (Norwegian Computing Center, NR) is a private, independent, non-profit foundation established in 1952. NR carries out contract research and development projects in the areas of information and communication technology and applied statistical modeling. The clients are a broad range of industrial, commercial and public service organizations in the national as well as the international market. Our scientific and technical capabilities are further developed in co-operation with The Research Council of Norway and key customers. The results of our projects may take the form of reports, software, prototypes, and short courses. A proof of the confidence and appreciation our clients have for us is given by the fact that most of our new contracts are signed with previous customers.

Norsk Regnesentral
Norwegian Computing Center
Postboks 114, Blindern
NO-0314 Oslo, Norway

Besøksadresse
Office address
Gaustadalléen 23
NO-0371 Oslo, Norway

Telefon · telephone
(+47) 22 85 25 00
Telefaks · telefax
(+47) 22 69 76 60

Internett · internet
www.nr.no
E-post · e-mail
nr@nr.no

Title	**Bluetooth Security**
Authors	**Hans Jakob Rivertz**
Date	03.03.2005
Year	2005
Publication number	DART/05/05

Abstract

We give an overview of Bluetooth and the state of the art of its security.

Keywords	Bluetooth, security, wireless, communication
Target group	Bluetooth users, Bluetooth developers
Availability	Open
Project number	320302
Research field	Security
Number of pages	15
© Copyright	Norsk Regnesentral

Contents

List of figures

1 About Bluetooth

Bluetooth is a short-range wireless communication protocol for personal area networks (PAN). In any communication link there is one master and one or more slave. The master and its slaves form a piconet. Overlapping piconet is called a scatternet. A unit may be a master in one piconet and a slave in another.

It was initially developed by Ericsson but is formalized as an industrial standard by the Bluetooth Special Interest Group (SIG). The SIG was formed by Ericsson, Intel, Toshiba, Nokia, and IBM but is now expanded to include about 1800 members. There are a numerous devices that support the Bluetooth standard now approximately 6 years after its launch. It is used mostly in consumers products like cell phones and personal digital assistants.

1.1 The Bluetooth channel

1.1.1 Physical channel

The communication in Bluetooth is done over a radio communication band at 2.4GHz. This band is divided into several frequencies, (23 or 79 which depends on the region). Each second is divided into 1600 timeslots of length 0.625ms. Each timeslot can contain up to ca 620 bits. The frequency is changed for every timeslot, i.e. 1600 times per second. This has some positive effect on security while it makes it more difficult to follow the communication between Bluetooth units. For a skilled attacker this frequency hopping is not an effective obstacle.

1.1.2 Packets

Packets are entities of information that are sent over the channel. Packets can span one, three, or five timeslots. The units use these packets to send information packets to each other.

LSB 68	58/0	0-2745 MSB
Access Code	Packet header	Payload

Figure 1: Generic Packet format

The first 68 bits in a packet code is a synchronization code. This contains a 4 bits long preamble and a sync word. The sync word is coded with a 1/3 error correcting code, (ECC).

An ID packet consists of the access code only. There can be two ID packets in each time slot.

4	64/54
Preamble	Content

Figure 2: Generic format of the Access code and the Packet header.

The content field of the packet header is encoded with a 1/3 ECC and contains a 3 bits member code (1-7 identifies the slaves in the piconet and 0 is reserved for broadcast messages), packet header type information and 3 bits for other link control. There is also an 8 bits checksum in the packet header that secures the integrity of the header. The access code and the packet header are never encrypted.

There are two different payload headers, one for single slot packets and one for multi slot packets:

2	1	5
L_CH	FLOW	LENGTH

Figure 3: Payload header format for single slot packets

2	1	9	4
L_CH	FLOW	LENGTH	Undefined

Figure 4: Payload header format for multi slot packets

1.1.3 Logical channels

There are five logical channels in a Bluetooth system. Two for link control:

- LC (Link Control channel). The packet header is the carrier the Link control channel
- LM (Link manager channel) The link manager uses mainly packets called DM packets. The payload header starts with L_CH = 11 for packet in the LM-channel

and two for data traffic:

- UA (User channel for asynchronous data)
- UI (User channel for isochronous data)
- US (User channel for synchronous data)

1.2 Overview of the security mechanisms dart in Bluetooth

The security mechanisms are described in section 14 in part B of the Bluetooth specification [1]. The following items maintain security in a Bluetooth system:

- PIN code. The pin code is the only shared security at initialization. The PIN code should be exchanged over an external secure channel. Some Bluetooth units have preprogrammed PIN code.
- Unit Addresses. The unit addresses are not secret but is used in the generation of the different keys.
- 128-bit random number generator. Gives random numbers that are used for the key generation. These random numbers are so called nonces, namely numbers that is used only once. A proper implementation of Bluetooth shall have a random generator where it is a very unlikely that random numbers are drawn twice.
- **Keys**
    - **Link key K**. The link key is used manage the link key and to produce the encryption key. At init the link key is set to the init key, (se under.)

    - **Initialization key K***init*. This key is used only to establish a link between two Bluetooth units. It should not be confused with the authentication. This key is made in each of the units by using following in the algorithm E22:
        1. The device address of one of the two communicating units.
        2. A PIN code and its length, (The PIN may be fixed in some units. that reduces the security offered to that unit.)
        3. A public random number *IN_RAND*.

The device address and the number *IN_RAND* are communicated over the Bluetooth link manager (LM) channel.

- **Unit key**. Each unit has a unit key. This key is almost never changed and should be kept secret. This key may be used as the link key. It is then sent to the opponent by XOR-ing it with the present link key, (the present link key should be discarded after such an exchange.) It is not recommended to use this option. However some units with limited memory must use this as a link key. Then the PIN code could also be fixed for the unit. Such units offer reduced security.

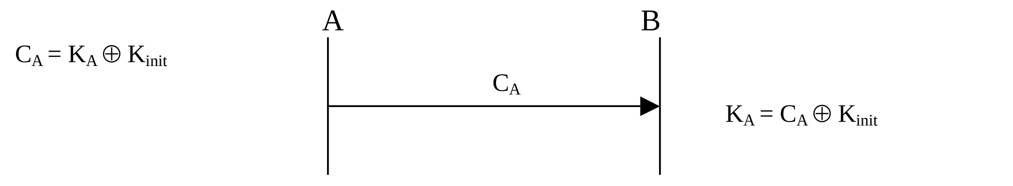

Figure: Protocol for unit key exchange

- **Combination key**. Two units that will communicate and does not want to use the unit key of one of the opponents create a common key, (the Combination key). The opponents exchange this key by a key exchange protocol. Random variables *LK_RAND_A* and *LK_RAND_B* are created. These random variables are exchanged secure by using the present link key as a cipher. The link key is now discharged. The algorithm E21 uses the addresses of the opponents and the two random numbers to produce the combination key, (see figure 14.2 in [1]). One should assume that any attacker knows *XOR (LK_RAND_A, LK_RAND_B)* and the addresses of the units A and B.

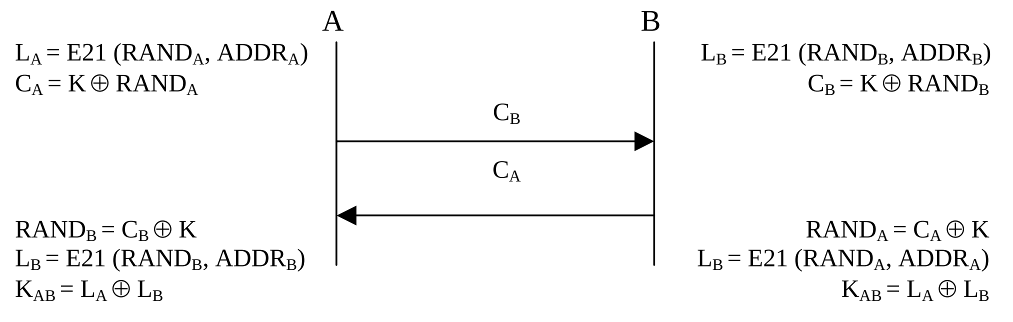
A
B
$L_A = E21\ (RAND_A, ADDR_A)$
$C_A = K \oplus RAND_A$
$L_B = E21\ (RAND_B, ADDR_B)$
$C_B = K \oplus RAND_B$
C_B
C_A
$RAND_B = C_B \oplus K$
$L_B = E21\ (RAND_B, ADDR_B)$
$K_{AB} = L_A \oplus L_B$
$RAND_A = C_A \oplus K$
$L_B = E21\ (RAND_A, ADDR_A)$
$K_{AB} = L_A \oplus L_B$

Figure: Protocol for combination key exchange.

- **Master key**. If the links are parts of a point to multipoint piconet there may be need for a master key. This key is made from random numbers and the algorithm E22. It is exchanged by using a key made by E22 from the present link key and a random number.

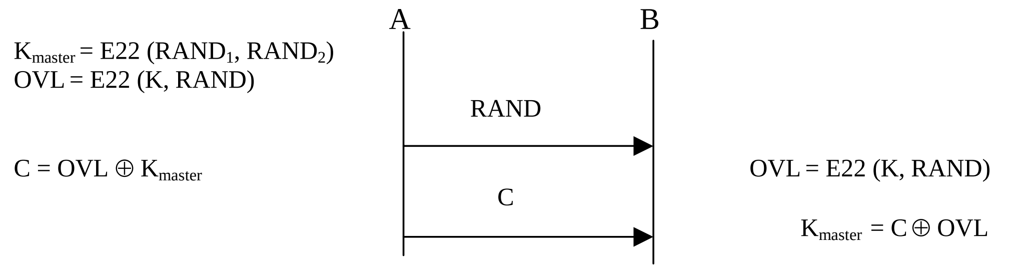

Figure: Protocol for master key exchange.

- **Encryption key**. The encryption key is made by the algorithm E3 from
  1. The current link key **K**
  2. A random number *EN_RAND*
  3. *COF* (128bit): Either a number computed in the authentication procedure or made from the master key. Using the master key for creating COF is obligatory if such is used as the current link key.

  The length of the encryption key can be from 8 of 128 effective bits. This is up to each device and is not user configurable.

- **Payload encryption key**. This key is made from the encryption key, a unit address, the master clock, and a 128 bit publicly known random number *EN_RAND*. This key is used to encrypt up to 2745 bits in one payload. The payload key is unique for each packet. The length of the key is 128 bits.

1.3 Protocols

SDP (Service Discovery Protocol) In this protocol
LMP (Link manager protocol)
L2CAP (Logical link Control and Adaptation Control)
RFCOMM protocol (a cable replacement protocol)
TCS Binary and AT Commands protocol

1.4 Security Modes

There are 3 different security modes (SMs), SM1, SM2, and SM3, with increasing security. For details refer to the Bluetooth standard [1] or some overview articles such as [5]. For security critical systems one should ensure that the highest available level (mode) is chosen.

1.5 Algorithms involved

The following algorithms are used in the Bluetooth standard

- **E0** Stream cipher encryption algorithm. This algorithm is used to encrypt the payload in the Bluetooth packets. It uses the payload encryption key. A description of this algorithm is showed in Figure 5 on page 11. E0 is a linear shift feedback register (LSFR) based crypto scheme. It consist of 4 LSFR combined with a finite state machine. The length of the encryption key is 128 bit and the period of the stream cipher is 2^{125}.
- **E1** Used for authentication. The crypto algorithm E1 is used for authentication.

- **E21** Combination key generation
- **E22** Master key generation and exchange. The algorithm E22 is used twice for each master key. Once for generating the key and once for exchange the key.

- **E3** Encryption key generation

The algorithms E1, E21, E22, and E3 build on the block encryption cipher SAFER+. There are reported that there are weaknesses of SAFER+ [15], but these involves known plaintext - cipher text attacks and an amount of data that is much more than

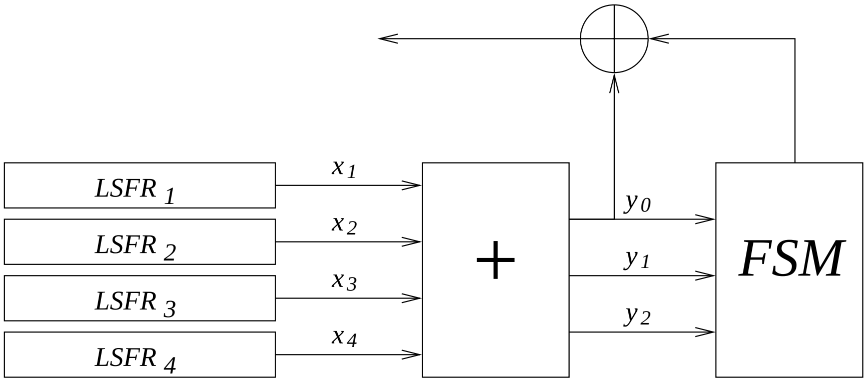

Figure 5: E0, encryption algorithm used in Bluetooth

1.6 Encryption.

Each payload is encrypted by a unique payload encryption key. There are known correlation attacks against the encryption scheme E0, but with the frequent change of payload key, these attacks will not work. There are also some known algebraic attacks.

At the current time there are no known problems with the other algorithms that seems to be a threat to the security of Bluetooth.

1.7 Authentication

A unit A authenticates a unit B by creating nonce which is sent to the unit B. B respond with a number SRES calculated with the algorithm E1 from A's nonce, the present link key, and B's address. A authenticates B if the number is right. A bi-product of the authentication is a number (ACO) that may be used in the creation of the encryption key.

Unsuccessful authentication attempts shall result the unit to wait a time interval that increases exponentially for each unsuccessful attempt until a limit for this interval is reached. This prevents an attacker to perform a brute force attack on the unit. It also protects the device from DoS attacks.

2 Security issues in Bluetooth

2.1 Is short range an effective security feature?

'Short range' is not a security mechanism but is used by the SIG to accept weak encryption. This may be a mistake since 'short range' is only short range for those who follow to the rules.

The range of Bluetooth is limited to 100 meter for Class 1 devices, (e.g. the USB Bluetooth adapter from Socket [2].) The most common devices such as cell phones and PDAs are off Class 1 and Class 2 and work at the maximum range of 10 meter or less. This is under normal circumstances, but if a device is modified with some kind of signal amplification and directional antennae, Bluetooth may work on longer distances. A range of more than 1.0 mile is registered [8]. The conclusion is that the limited range of Bluetooth gives only a partial security. I.e. one is protected against the less malicious and low skill attackers, but for the high skilled and well-equipped attackers low range communication is a more theoretic than a practical barrier.

Even if the range of the Bluetooth is short, the mobility of those devices revokes the positive effect of short range and makes it even more vulnerable then the wider reaching Wireless LAN. This is because we are bringing the devices with us on train, airports and other public areas.

2.2 Weaknesses of the cryptographic protocol

There are many known attacks on the encryption scheme E0 (figure 5) that is used in Bluetooth. There are algebraic attacks [6] and correlation attacks [7,10]. The correlation attacks needs much more than the maximum 2745 bits of data that are encrypted with each payload encryption key to be effective. The best known attack needs up to 500 billion bits to be useful! Therefore the attacks seem not to apply to the usage of E0 in Bluetooth.

2.3 Implementation weaknesses

In some implementations there are security breaches, such as the possibility to overwrite the stack by buffer overflow [4]. In this way one can run arbitrary code on the victim unit. This weakness is in the Bluetooth connectivity software made by WIDCOMM. Newer versions (3+) of this software are not vulnerable. It is not clear if the failure makes the units vulnerable for attacks from arbitrary units, or if the attacking units already must be connected to the victim unit.

2.4 Weakness of the invisibility in version 1.1 and older

If a Bluetooth device is in Non-visible mode, it guaranties not that it is invisible. It is not easy to find such a Bluetooth device. Its identity can be found in less than 11 hours. This attack can be done via so-called Bluetooth ‘war nibbling’ [13], (war nibbling is the version war driving [3] on small units.) Often, when a unit is on the move or when the unit is never switched on for a longer time, the vulnerability of such an attack should be small.

2.5 Bluejacking

Bluejacking is the process of hijacking a Bluetooth session / unit. It can be done in different ways, e.g. through social engineering or by using backdoors in second hand units, (even if the pin-card is changed a unit may still be paired with another unit.) A hacker can also hijack a Bluetooth device by using his own PIN card and then set up a connection to a given Bluetooth device he controls. The hacker needs physical access to the target phone to success with the last attack.

A long user definable name-field in the protocol for requesting a link can be used to send messages to a phone holder in the purpose to trick him to accept a connection request from the attacking party. A good user interface should alert the user and prevent him from being a victim of such an attack.

The short range of Bluetooth will make it harder for intruders, but the nature of the Bluetooth technology one should expect that it would increase the users mobility, and hence we can assume that the users carry the equipment with them.

2.6 Attack on the Link Layer.

The link layer is not encrypted and the integrity protection of the packet headers is weak. That can make the Bluetooth link layer vulnerable for attacks, [13].

2.7 Snarf attack

On some phones it is possible to connect to a cell phone without the knowledge of the owner. It is possible to see some of the stored data in the attacked phone. The entire phonebook, the calendar, the clock, etc is accessible. The IMEI (International Mobile Equipment Identity) is also accessible which makes it possible for an intruder to make a clone of the phone. According to Laurie et all [9] vulnerable phones include: Ericsson T68; Sony Ericsson R520m, T68i, T610 and Z1010; and Nokia 6310, 6310i, 8910 and 8910i. The NOKIA phones mentioned is also vulnerable if it is in invisible mode [9].

2.8 Backdoor attack

This attack is using already establish pairing with a unit. Vulnerable devices are mostly second hand cell phones and PDAs that has not its former pairing erased [9].

2.9 BlueBug

There is a bug in some cell phones that makes them vulnerable for attacks [9]. This attack seems to be serious for those phones that are infected. The attack opens up for sending AT-commands to a cell phone. This attacks opens for reading and sending SMS initiate phone talks, enter the Internet, writing and reading phone book entries. The author does not know which phones that are vulnerable.

2.10 Extra remarks

As more the number of purposes a unit has, as more vulnerable will the unit be. This is not solely related to Bluetooth, but applies to all computer security. If the user for example downloads games from the net and use the unit to play games. The user may be vulnerable to virus, worms, etc.

3 Conclusion

The PIN code should be long in security critical Bluetooth systems. The PIN code is the only secret credential in a Bluetooth network and should be chosen in such a way that it is difficult to find it for an attacker.

The effective protection of the 128 bit cryptographic key is approximately 40 bits due to weaknesses in the E0 encryption scheme. This seems to not be a crucial for the security of Bluetooth since the keys are changed very often.

There are not been possible to find major problems with the algorithms E1, E21, E22, and E3. They are all based on the crypto scheme SAFER+. For the purposes in of SAFER+ in Bluetooth it seems to be safe enough.

Units that have fixed PIN code and limited addresses should be used with care.

The limited range of Bluetooth gives only a partial security due to possible attacks using signal amplifiers. 'Short range' is not a security mechanism but is used by the SIG to accept weak encryption. This may be a mistake since 'short range' is only short range for those who follow to the rules.

4 References:

1 *Bluetooth specification version 1.1*, http://www.bluetooth.org/

2 Socket, http://www.socketcom.com/

3 wardriving.com http://www.wardriving.com/

4 M. Rowe and M. Moore, *WIDCOMM Bluetooth Connectivity Software Multiple Buffer Overflow Vulnerabilities*, http://www.pentest.co.uk/documents/ptl-2004-03.html, (2004)

5 N. Anand, An overview of bluetooth security, (2001)

6 N. T. Courtois, Algebraic attacs on combiners *with memory and several outputs,* to appear in ICISC 2004, just before Asiacrypt, in Korea, LNCS, Springer, (2005).

7 M. Hermelin and K. Nyberg, *Correlation properties of the bluetooth Combiner*, Information Security and Cryptology –ICISC'99, Lecture Notes in Computer Science, **1787**, Springer-Verlag, pp. 17-29, (2000)

8 T. Hurman and M. Rowe, *Bluetooth security issues, threats and consequences*, http://www.pentest.co.uk/documents/wbf_slides.pdf

9 A. Laurie and B. Laurie, *Serious flaws in bluetooth security lead to disclosure of personal data*, The Bunker, http://www.thebunker.net/security/bluetooth.htm, (2004)

10 Y. Lu and S. Vaudenay, *Faster correlation attack on bluetooth key stream generator E0*, Advances in Cryptography – Crypto 2004, Lecture Notes in Computer Science, **3152**, Springer-Verlag, pp 407-425, (2004)

11 M. Träskbäck, *Security of Bluetooth: An overview of Bluetooth Security,* http://www.seecode.com/files/Bluetooth_Security.pdf, (200?)

12 J. T. Vainio, *Bluetooth Security,* http://www.niksula.cs.hut.fi/~jiitv/bluesec.html, (2000)

13 O. Whitehouse, *War Nibbling: Bluetooth insecurity,* http://www.atstake.com/research/reports/acrobat/atstake_war_nibbling.pdf, (2003)

14 T. Yang, *Bluetooth security*, http://www.cs.utk.edu/~tyang/wireless/blue.htm

15 J. Kelsey, B. Schneiery and D. Wagnerz, *Key Schedule Weaknesses in SAFER+*, http://www.schneier.com/paper-safer.pdf, (1999)

16 *Sony Ericsson phones open to 'snarf' attack*, http://news.com.com, (2004)

17 *Nokia: Bluetooth flaw gnaws at phone security*, http://news.com.com, (2004)

18 *Blue Bug*, http://trifinite.org/trifinite_stuff_bluebug.html