Bluetooth Security

2000-05-25

Juha T. Vainio
Department of Computer Science and Engineering
Helsinki University of Technology
firstname.lastname at iki.fi

Abstract

Bluetooth is a way of connecting machines to each other without cables or any other physical medium. It uses radio waves to transfer information, so it is very susceptible to attacks. This paper first gives some background information about Bluetooth system and security issues in ad hoc networks, then it concentrates on specific security measures in Bluetooth, mainly authentication, encryption, key management and ad hoc aspects. Then it points out flaws and possible security holes in the Bluetooth Security Specification.

Contents

1 Introduction

2 Bluetooth

- 2.1 Background
- 2.2 Technical Specifications

3 Distributed Systems Security

4 Security in Ad Hoc Networks

- 4.1 Availability
- 4.2 Authorization and Key Management
- 4.3 Confidentiality and Integrity

5 Bluetooth Security

- 5.1 Key Management

5.2 Encryption

5.3 Authentication

5.4 Ad Hoc Aspects

6 Problems in the Security of Bluetooth

7 Conclusions

Glossary

References

Further Information

---

1 Introduction

From the beginning of the computer era, cables have been used to connect computers to each other and to special accessories. Security measures have been developed to secure these cable connections so that information can travel safely. Now, as the time has passed, cables have become a nuisance. Bluetooth is one of the solutions to form a cable-free environment. This paper concentrates on the security measures of Bluetooth, how they should be different from the traditional security measures of the cable-connected world and are they sufficient enough, so that Bluetooth can be used for everyday communications.

The rest of this paper is organized as follows. First, in section 2, we take a look to Bluetooth as a technology, what possible uses it has and how it is implemented. Next, in section 3, we look into the security of distributed systems and ad hoc networks in general, so we know what is needed from the security specifications of Bluetooth. Then we get to the main part of this paper, Bluetooth security, where we examine the Bluetooth security architecture in detail. We discover if there are any problems and at the end, we sum up the discoveries and make the conclusions that can be made. Specifically, we ponder the facts discovered in the previous chapters and determine if the Bluetooth security measures are any good.

2 Bluetooth

In this section we view the concept and specifications of Bluetooth and discover what it is all about. First we look into the background and different ways to use Bluetooth, then we look into the specifications and architecture in more detail.

2.1 Background

Bluetooth is the new emerging technology for wireless communication. It was developed by a group called Bluetooth Special Interest Group (SIG), formed in May 1998. The founding members were Ericsson, Nokia, Intel, IBM and Toshiba. Since then, almost all of the biggest companies in the telecommunications business (e.g. 3Com, Microsoft, Motorola) have joined the Bluetooth SIG and the number of the participating companies is now over 1,500. The version 1.0 of the Bluetooth specification was approved in the summer of 1999, and the latest version (at the time of writing) 1.0B in December 1999 [3].

Bluetooth can be used to connect almost any device to another device. The traditional example is to link a Personal Digital Assistant (PDA) or a laptop to a mobile phone. That way you can easily take remote connections with your PDA or laptop without getting your mobile phone from your pocket or messing around with cables. Bluetooth can also be used to form ad hoc networks of several (up to eight) devices, called piconets. This can be useful for example in a meeting, where all participants have their own Bluetooth-compatible laptops, and want to share files with each other. [4]

2.2 Technical Specifications

Bluetooth devices are categorized into three different classes by the power they use. A class 3 device has a 1 mW transmission power and a range of 0.1-10 meters. A class 2 device has a transmission power of 1-2.5 mW and a 10-meter range. A class 1 device has a transmission power up to 100 mW and a range up to 100 meters. [7]

The architecture of Bluetooth is formed by the radio, the base frequency part and the Link Manager. Bluetooth uses the radio range of 2.45 GHz. The theoretical maximum bandwidth is 1 Mb/s, which is slowed down a bit by Forward Error Correction (FEC). Bluetooth specification designates the frequency hopping to be implemented with Gaussian Frequency Shift Keying (GFSK).

The base frequency part of the Bluetooth architecture uses a combination of circuit and packet switching technologies. Bluetooth can support either one asynchronous data channel and up to three simultaneous synchronous speech channels, or one channel that transfers asynchronous data and synchronous speech simultaneously.

The Link Manager is an essential part of the Bluetooth architecture. It uses Link Manager Protocol (LMP) to configure, authenticate and handle the connections between Bluetooth devices. It also operates the power management scheme, which is divided into three modes: sniff, hold and park. [10]

As discussed earlier, several Bluetooth devices can form an ad hoc network. In these piconets, one of the Bluetooth devices will act as a master and the others are slaves. The master sets the frequency-hopping behavior of the piconet. It is also possible to connect up to 10 piconets to each other to form so-called scatternets.

3 Distributed Systems Security

In this section, we take a brief look into the general definitions of threats to computer systems and then move to consider the security of distributed systems. Mainly we concentrate on the aspects that make it different from the traditional centralized systems' computer security.

According to the most widely used categorization of threats to computer systems, the threats are divided into three types: disclosure threats, integrity threats and denial of service threats. This by no means covers all the possible threats, but will suffice for our purposes. The disclosure threat involves the leakage of information from the system to a party that should not have seen the information and it is a threat against the confidentiality of the information. The integrity threat involves an unauthorized change of the information in question. The denial of service threat involves an access to a system resource being blocked by a malicious attacker. It is a threat against the availability of the system. [1]

There are also a couple of other definitions we need in order to proceed. Authentication means the ensuring of the identity of another user, so that he knows whom he is communicating with. Non-repudiation ensures that the user that has sent a certain message cannot deny sending the message later on. [5]

In distributed systems, objects are scattered to different places. This makes the security issues more difficult. There are plenty of additional questions to be answered when compared to centralized systems. For example, in a distributed system, the user authentication is much more difficult. If the authentication is done with passwords, there is the link to the authenticating machine to worry about. If the link is not secure, which it rarely is, you must ensure that no one can sniff your password on the way.

This is just the tip of the iceberg. In a distributed system, there are usually several parties involved. There may not be a clear consensus on the used security policy. If different participants enforce different kinds of security policies, collaboration is impossible.

Another matter altogether is a process called delegation. When a user, using a local access to login into a network, wants to execute a program on a remote machine, some problems arise. The program will need certain rights to use the resources on the remote machine. Then the user typically delegates his access rights to the program, so that it can run on the remote machine. The problem in this is that users have very little control over the remote machine, yet they have to delegate their rights to a program running there. In distributed systems, there is always a possibility that the remote machine is weakly protected and a malicious user can exploit the user's rights.

Another problematic field in distributed systems security is authentication. The decision that should be made is whether the security should be enforced centrally or locally. In centralized security enforcement, there could be some kind of Key Distribution Center (KDC), where the keys of all the devices are stored. The Key Distribution Center acts as

a Trusted Third Party (TTP) that users can use to authenticate themselves and other users, and to get secure connections everywhere in the network. There are several ways this can happen, but you can find detailed descriptions of, for example, Kerberos authentication and key exchange protocol, in [11]. The biggest problem in this is the trustworthiness of the Trusted Third Party. If it is compromised, all the secret keys are available for malicious use and the whole scheme collapses.

On the other hand, if the decision is made that the security enforcement scheme is to be local, other kinds of security measures are needed. Each user enforces his own security policy and trusts the machines he logs in. There could be a trusted Certification Authority (CA), which issues public key certificates and a Certification Distribution Center (CDC), which stores all the certificates issued by the Certification Authority. The users have their own key pairs and can certify their public keys with the Certification Authority. Then, if a user uses his key to sign something, the signature can be checked to correspond with a public key. The public key in turn can be checked with the Certification Distribution Center to certify that the public key in fact does belong to the user that originally did the signing. In this way, we can enforce the security locally and still have working authentication system with Public Key Infrastructure (PKI). [5]

4 Security in Ad Hoc Networks

In this section, we find out how the security questions in ad hoc networks are different from the plain old distributed systems security. First, we examine the characteristics of ad hoc networks in general and then focus on the problems in the security issues.

In ad hoc networks, there is no fixed infrastructure. Networks are formed on-the-fly, as the name implies. All the devices on an ad hoc network connect to each other via wireless links. Individual devices act as routers when relaying messages to other devices, which are too far apart from the sending one to get the message directly. The topology of an ad hoc network is not fixed, either. It changes all the time when these mobile devices move in and out of other devices' transmission range. All this makes the ad hoc networks very vulnerable to attacks and the security issues very complicated.

4.1 Availability

In ad hoc networks, considering the ensuring of the availability is perhaps more important than it is in traditional security. As all the devices in the network are dependent of each other to relay messages, denial of service attacks are easy to perform. And again, as all the information is transmitted on the air, it makes even more denial of service attacks possible. For example, a malicious user could try to jam or otherwise try to interfere the flow of information on the air. Or it could be possible to disrupt the routing protocol used in the network by feeding the network with inaccurate information. [13]

Routing protocols are in fact one of the most vulnerable points in ad hoc networks. Routing protocols should be able to handle both the changing topology of the network and attacks from the malicious users. There are routing protocols that can adjust well to

the changing topology, but according to [13], there are none that can defy the possible attacks.

Another vulnerable point, which has no equivalence in traditional networks, is the battery of the device in the ad hoc network. Normally, these devices try to save energy with all kinds of battery saving schemes, so that when the device is not in active use, energy is not consumed. With battery exhaustion attacks, a malicious user can consume more energy from the battery of a device, so that eventually the power will go out prematurely. [12]

4.2 Authorization and Key Management

Authorization is another difficult matter in ad hoc networks. As there is very little or no infrastructure, identifying users (e.g. participants in an ad hoc network in a meeting room) is not easy. There are problems with trusted third party -schemes and identity-based mechanisms for key agreement, as is described in [2].

However, it is possible to construct very good authentication mechanisms for ad hoc networks. Again in [2], a generic protocol for password authenticated key exchange is described. It has several drawbacks and does not suite well for ad hoc networking devices (with perhaps smaller CPUs than normal desktop computers). So there is presented a password authenticated multi-party Diffie-Hellman key exchange, which seems to avoid the problems of the generic protocol mentioned above. [2]

4.3 Confidentiality and Integrity

Confidentiality, too, is a very vulnerable point in ad hoc networks. With wireless communication, anyone can sniff the messages on the air and without proper encryption, all the information is available to anyone. On the other hand, without proper authentication, there is no point even to talk about confidentiality. If you cannot be certain who you are talking to, the confidentiality is poor anyway. And if the proper authenticity has been established, the securing of the connection with available keys is no problem.

For integrity, the same things apply. In addition to malicious attacks, integrity may be compromised because of radio interference, etc., so some kind of integrity protection is definitely needed.

5 Bluetooth Security

In this section, we look into Bluetooth's security measures in more detail. First, we examine the Bluetooth security in general and how have different things been taken into account. Then we keep on going and get a closer look to some specific details in key management, encryption and authentication. And finally, we take a look in the ad hoc aspects of Bluetooth security.

In every Bluetooth device, there are four entities used for maintaining the security at the link level. The Bluetooth device address (BD_ADDR), which is a 48-bit address that is unique for each Bluetooth device and defined by the Institute of Electrical and Electronics Engineers (IEEE). Private authentication key, which is a 128-bit random number used for authentication purposes. Private encryption key, 8-128 bits in length that is used for encryption. And a random number (RAND), which is a frequently changing 128-bit random or pseudo-random number that is made by the Bluetooth device itself. [3]

In Bluetooth Generic Access Profile, the Bluetooth security is divided into three modes:

- Security Mode 1: non-secure
- Security Mode 2: service level enforced security
- Security Mode 3: link level enforced security

The difference between Security Mode 2 and Security Mode 3 is that in Security Mode 3 the Bluetooth device initiates security procedures before the channel is established. [8]

There are also different security levels for devices and services. For devices, there are 2 levels, "trusted device" and "untrusted device". The trusted device obviously has unrestricted access to all services. For services, 3 security levels are defined: services that require authorization and authentication, services that require authentication only and services that are open to all devices. [8]

5.1 Key Management [3]

All security transactions between two or more parties are handled by the link key. The link key is a 128-bit random number. It is used in the authentication process and as a parameter when deriving the encryption key. The lifetime of a link key depends on whether it is a semi-permanent or a temporary key. A semi-permanent key can be used after the current session is over to authenticate Bluetooth units that share it. A temporary key lasts only until the current session is terminated and it cannot be reused. Temporary keys are commonly used in point-to-multipoint connections, where the same information is transmitted to several recipients.

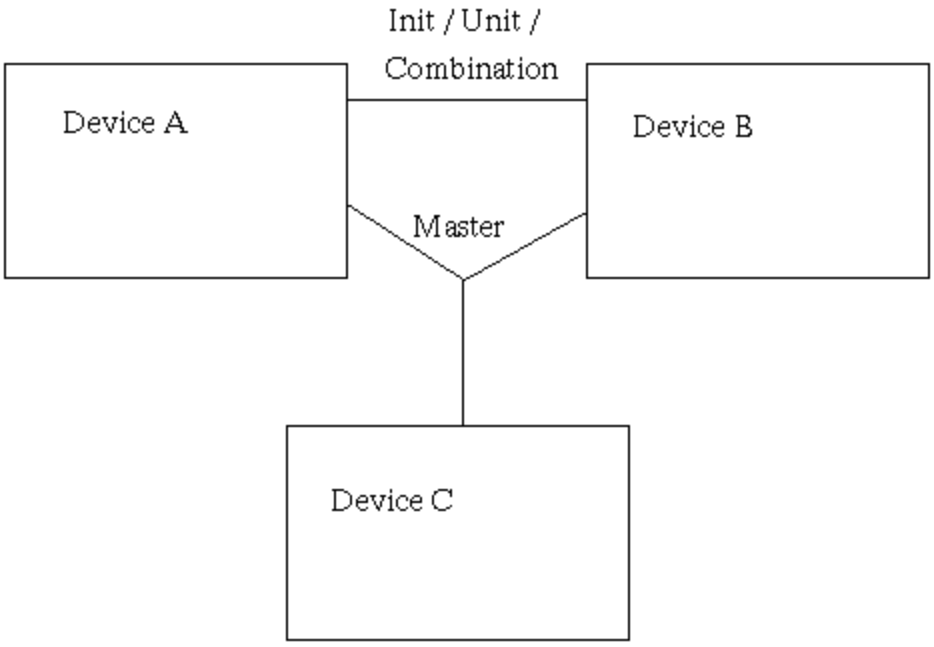

Figure 1: Different link keys between devices

There are several different types of keys defined in Bluetooth. Link keys can be combination keys, unit keys, master keys or initialization keys, depending on the type of application. In addition to link keys, there is the encryption key.

The unit key is generated in a single device when it is installed. The combination key is derived from information from two devices and it is generated for each new pair of Bluetooth devices. The master key is a temporary key, which replaces the current link key. It can be used when the master unit wants to transmit information to more than one recipient. The initialization key is used as link key during the initialization process when there are not yet any unit or combination keys. It is used only during the installation.

The length of the Personal Identification Number (PIN) code used in Bluetooth devices can vary between 1 and 16 octets. The regular 4-digit code is sufficient for some applications, but higher security applications may need longer codes. The PIN code of the device can be fixed, so that it needs to be entered only to the device wishing to connect. Another possibility is that the PIN code must be entered to the both devices during the initialization.

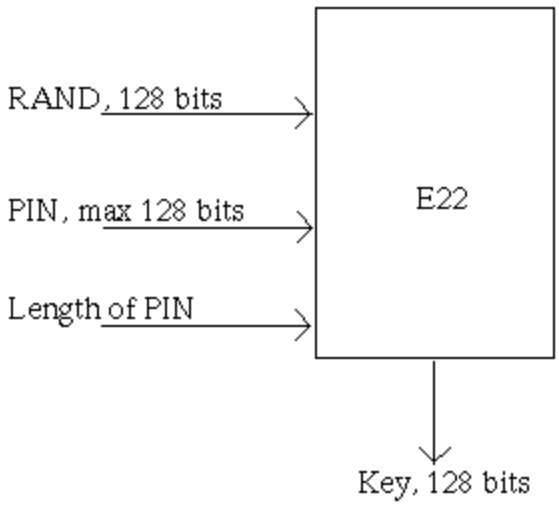

Figure 2: Key generating algorithm E22 for master and initialization keys

The initialization key is needed when two devices with no prior engagements need to communicate. During the initialization process, the PIN code is entered to both devices. The initialization key itself is generated by the E22 algorithm, which uses the PIN code, the Bluetooth Device Address of the claimant device and a 128-bit random number generated by the verifier device as inputs. The resulting 128-bit initialization key is used for key exchange during the generation of a link key. After the key exchange the initialization key is discarded.

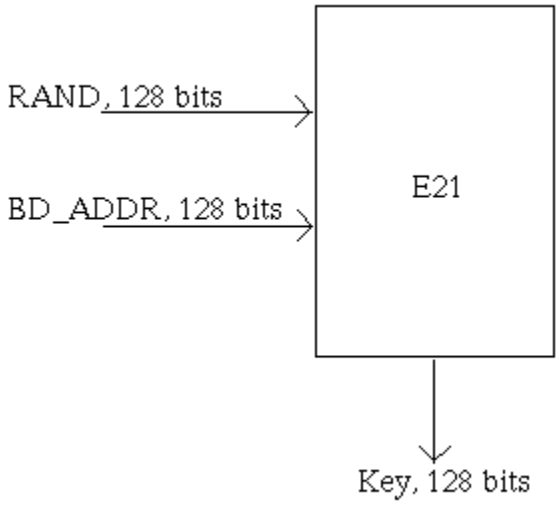

Figure 3: Key generating algorithm E21 for unit and combination keys

The unit key is generated with the key generating algorithm E21 when the Bluetooth device is in operation for the first time. After it has been created, it will be stored in the

non-volatile memory of the device and is rarely changed. Another device can use the other device's unit key as a link key between these devices. During the initialization process, the application decides which party should provide its unit key as the link key. If one of the devices is of restricted memory capabilities (i.e. cannot remember any extra keys), its link key is to be used.

The combination key is generated during the initialization process if the devices have decided to use one. It is generated by both devices at the same time. First, both of the units generate a random number. With the key generating algorithm E21, both devices generate a key, combining the random number and their Bluetooth device addresses. After that, the devices exchange securely their random numbers and calculate the combination key to be used between them.

The master key is the only temporary key of the link keys described above. It is generated by the master device by using the key generating algorithm E22 with two 128-bit random numbers. As all the link keys are 128 bits in length, the output of the E22 algorithm is 128 bits, too. The reason for using the key generating algorithm in the first place is just to make sure the resulting random number is random enough. A third random number is then transmitted to the slave and with the key generating algorithm and the current link key an overlay is computed by both the master and the slave. The new link key (the master key) is then sent to the slave, bitwise XORed with the overlay. With this, the slave can calculate the master key. This procedure must be performed with each slave the master wants to use the master key with.

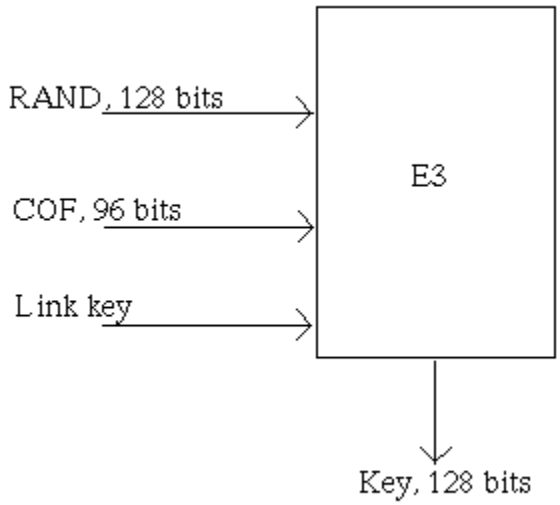

Figure 4: Key generating algorithm E3 for the encryption key

The encryption key is generated from the current link key, a 96-bit Ciphering Offset Number (COF) and a 128-bit random number. The COF is based on the Authenticated Ciphering Offset (ACO), which is generated during the authentication process. When the

Link Manager (LM) activates the encryption, the encryption key is generated. It is automatically changed every time the Bluetooth device enters the encryption mode.

5.2 Encryption

The Bluetooth encryption system encrypts the payloads of the packets. This is done with a stream cipher E0, which is re-synchronized for every payload. The E0 stream cipher consists of the payload key generator, the key stream generator and the encryption/decryption part. You can find a detailed C implementation of the E0 stream cipher in [9].

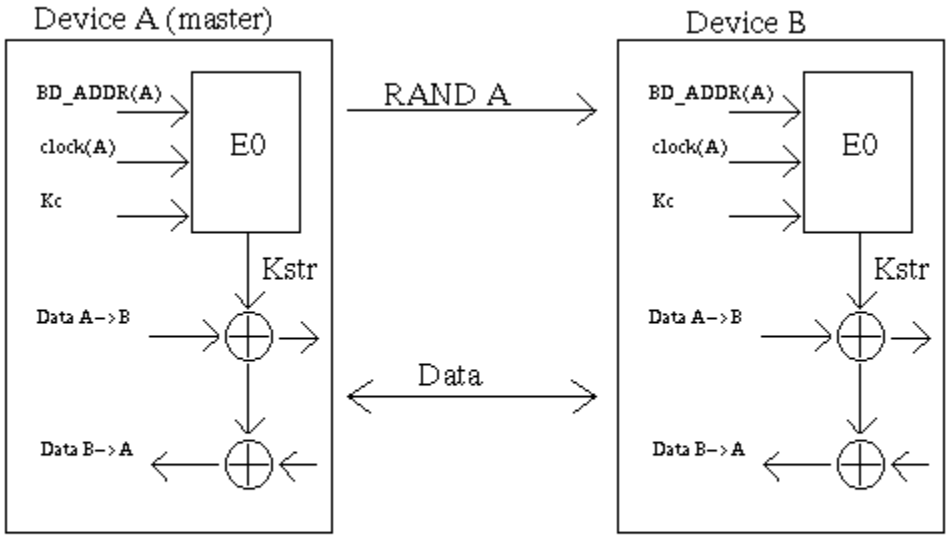

Figure 5: Description of the encryption process

The payload key generator combines the input bits in an appropriate order and shifts them to the four Linear Feedback Shift Registers (LSFR) of the key stream generator. The key stream bits are generated by a method derived from the summation stream cipher generator by Massey and Rueppel [11].

Depending on whether a device uses a semi-permanent link key or a master key, there are several encryption modes available. If a unit key or a combination key is used, broadcast traffic is not encrypted. Individually addressed traffic can be either encrypted or not. If a master key is used, there are three possible modes. In encryption mode 1, nothing is encrypted. In encryption mode 2, broadcast traffic is not encrypted, but the individually addressed traffic is encrypted with the master key. And in encryption mode 3, all traffic is encrypted with the master key.

As the encryption key size varies from 8 bits to 128 bits, the size of the encryption key used between two devices must be negotiated. In each device, there is a parameter defining the maximum allowed key length. In the key size negotiation, the master sends its suggestion for the encryption key size to the slave. The slave can either accept and acknowledge it, or send another suggestion. This is continued, until a consensus is reached or one of the devices aborts the negotiation. The abortion of the negotiation is done by the used application. In every application, there is defined a minimum acceptable key size, and if the requirement is not met by either of the participants, the application aborts the negotiation and the encryption cannot be used. This is necessary to avoid the situation where a malicious device forces the encryption to be low in order to do some harm.

The encryption algorithm uses four LFSRs of lengths 25, 31, 33 and 39, with the total length of 128. The initial 128-bit value of the four LFSRs is derived from the key stream generator itself using the encryption key, a 128-bit random number, the Bluetooth device address of the device and the 26-bit value of the master clock. The feedback polynomials used by the LFSRs are all primitive, with the Hamming weight of 5. The polynomials used are (25, 20, 12, 8, 0), (31, 24, 16, 12, 0), (33, 28, 24, 4, 0) and (39, 36, 28, 4, 0). Information on the fundamentals of LFSRs is found in [11].

5.3 Authentication [3]

The Bluetooth authentication scheme uses a challenge-response strategy, where a 2-move protocol is used to check whether the other party knows the secret key. The protocol uses symmetric keys, so a successful authentication is based on the fact that both participants share the same key. As a side product, the Authenticated Ciphering Offset (ACO) is computed and stored in both devices and is used for cipher key generation later on.

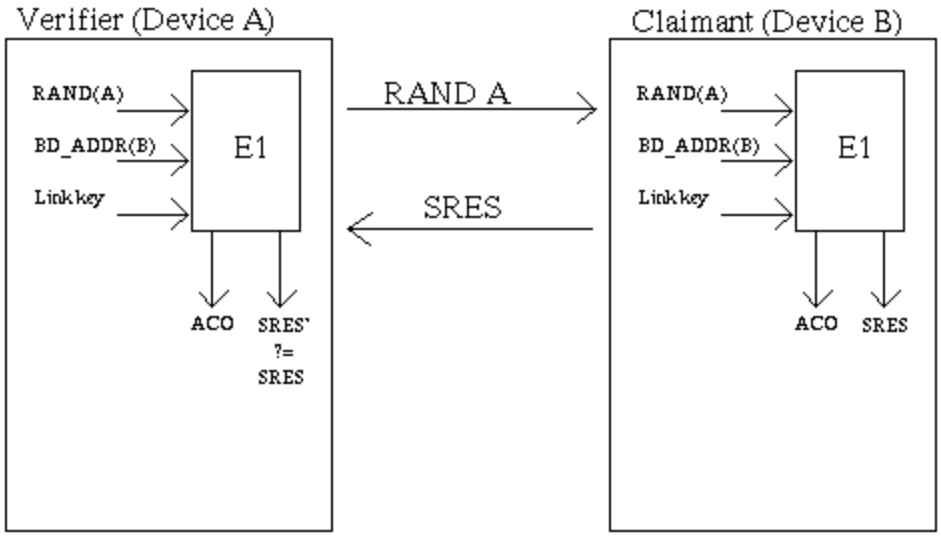

Figure 6: Description of the authentication process

First, the verifier sends the claimant a random number to be authenticated. Then, both participants use the authentication function E1 with the random number, the claimants Bluetooth Device Address and the current link key to get a response. The claimant sends the response to the verifier, who then makes sure the responses match.

The used application indicates who is to be authenticated. So the verifier may not necessarily be the master. Some of the applications require only one way authentication, so that only one party is authenticated. This it not always the case, as there could be a mutual authentication, where both parties are authenticated in turn.

If the authentication fails, there is a period of time that must pass until a new attempt at authentication can be made. The period of time doubles for each subsequent failed attempt from the same address, until the maximum waiting time is reached. The waiting time decreases exponentially to a minimum when no failed authentication attempts are made during a time period.

5.4 Ad Hoc Aspects

There are some aspects of Bluetooth security that should be considered in the light of ad hoc networking. In an ad hoc network formed in a conference room, there are a couple of possibilities for Bluetooth devices to secure the traffic. First of all, they can use the combination keys to encrypt the traffic. This means that the master device forms

combination keys with every slave device in the network. Then the information from a slave is subsequently sent to all other slaves by the master.

Another way of forming a secure ad hoc network is to use the master key concept. Then all the devices in the network can use the same key when encrypting the traffic and no separate relaying of traffic is needed.

This seems to be the limit of the ad hoc aspects of the link level security mechanisms of Bluetooth. If there is to be more complex ad hoc networking, the security must be done on the application level. For example, if any KDCs or distributed secret schemes are to be used, Bluetooth does not support them directly.

6 Problems in the Security of Bluetooth

In the encryption scheme of Bluetooth there seems to be some weaknesses. The E0 stream cipher with 128-bit key length can be broken in O(2^64) in some circumstances. The proof is rather mathematical in nature and therefore out of the scope of this paper, so it will be omitted. However, the detailed version can be read in [6]. In a nutshell, there is a divide-and-conquer type of attack that is possible to perform, if the length of the given keystream is longer than the period of the shortest LFSR user in the key stream generation in E0.

This, however, has been taken into account in the Bluetooth specifications. The above mentioned divide-and-conquer attack needs access to the key stream extending over periods of partial input. This can never happen, as Bluetooth has a very high re-synchronization frequency, i.e. the key stream segment generated to encrypt each frame is independent and so short that no possibilities arise for attacks.

There is a problem in the usability of the Bluetooth devices, too. The use of the PIN code in the initialization process of two Bluetooth devices is tacky. When you have to enter the PIN code twice every time you connect two devices, it gets annoying even with shorter codes. If there is an ad hoc network of Bluetooth devices and every machine is to be initialized separately, it is unbearable. And it does not make upholding the security very easy.

The specification makes a suggestion to use application level key agreement software with the longer (up to 16 octets) PIN codes. So the PIN code need not be entered physically to each device of the connection, but is exchanged with, for example, Diffie-Hellman key agreement.

The generation of the initialization key may also be of concern. The strength of the initialization key is based purely on the used PIN code. The E22 initialization key generation algorithm derives the key from the PIN code, the length of the PIN code and a random number, which is transmitted over the air. The output is highly questionable, as the only secret is the PIN code. When using 4 digit PIN codes there are only 10.000

different possibilities. Adding the fact that 50% of used PINs are "0000", the trustworthiness of the initialization key is quite low.

There is also a problem in the unit key scheme. Authentication and encryption are based on the assumption that the link key is the participants' shared secret. All other information used in the procedures is public. Now, suppose that devices A and B use A's unit key as their link key. At the same time (or later on), device C may communicate with device A and use A's unit key as the link key. This means that device B, having obtained A's unit key earlier, can use the unit key with a faked device address to calculate the encryption key and therefore listen to the traffic. It can also authenticate itself to device A as device C and to device C as device A.

The Bluetooth Device Address, which is unique to each and every Bluetooth device, introduces another problem. When a connection is made that a certain Bluetooth device belongs to a certain person, it is easy to track and monitor the behavior of this person. Logs can be made on all Bluetooth transactions and privacy is violated. Profiling and other questionable ways of categorizing can take place.

Yet another problem with Bluetooth is the battery draining denial of service scheme, against which it has no protection. If this is going to be a big problem, I suspect some countermeasures will be taken by the Bluetooth SIG.

All in all, there are several problems still in the security of Bluetooth. It seems to be adequate for smaller applications, but any sensitive or otherwise problematic data should not be transmitted via Bluetooth.

7 Conclusions

We have now examined Bluetooth in general, some of the security properties of distributed systems and ad hoc networks and the Bluetooth security mechanisms. As was seen, the Bluetooth's security seemed to be adequate only for small ad hoc networks, such as a network of the participants in a meeting. Connecting a PDA to a mobile phone using Bluetooth may also be secure enough, but is Bluetooth secure enough for larger ad hoc networks, money transfers and transferring other sensitive information?

In the light of this study, it seems that the security of Bluetooth is still inadequate for any serious, security sensitive work. After the basic problems have been corrected, the more sophisticated security methods may be implemented on the upper levels. The security specification only considers simple issues and the more functional security has to be built above it. This includes the better authorization systems with possible KDCs and distributed secret schemes. The secure routing protocols for larger ad hoc networks must also be implemented separately.

Glossary

ACO	Authenticated Ciphering Offset
BD_ADDR	Bluetooth Device Address
CA	Certification Authority
CDC	Certification Distribution Center
COF	Ciphering Offset Number
FEC	Forward Error Correction
GFSK	Gaussian Frequency Shift Keying
IEEE	Institution of Electrical and Electronics Engineers
KDC	Key Distribution Center
LFSR	Linear Feedback Shift Register
LM	Link Manager
LMP	Link Manager Protocol
PDA	Personal Digital Assistant
PIN	Personal Identification Number
PKI	Public Key Infrastructure
SIG	Special Interest Group (as in Bluetooth SIG)
TTP	Trusted Third Party

References

[1] Amoroso E., Fundamentals of Computer Security Technology, Prentice Hall, 1994, 403p.

[2] Asokan N. & Ginzboorg P., Key Agreement in Ad-Hoc Networks, 2000-02-03

[3] Bluetooth, The Bluetooth Specification, v.1.0B [referred 2000-03-15]
< http://www.bluetooth.com/developer/specification/specification.asp >

[4] Bradbury D., Disable the Cable, Personal Computer World, March 2000

[5] Gollmann D., Computer Security, John Wiley & Sons Inc., 1999, 336p.

[6] Hermelin M. & Nyberg K., Correlation Properties of the Bluetooth Combiner

[7] Joronen J., Bluetooth Tunkee Piireihin, Prosessori, 2000, Nr. 1

[8] Müller T., Bluetooth Security Architecture, 1999-07-15 [referred 2000-03-15]
< http://www.bluetooth.com/developer/download/download.asp?doc=174 >

[9] Saarinen M-J, A Software Implementation of the BlueTooth Encryption Algorithm E0, 2000-03-08 [referred 2000-03-15]
< http://www.jyu.fi/~mjos/e0.c >

[10] Sand K., Bluetooth, 1999-03-04 [referred 2000-03-13]
< http://www.tcm.hut.fi/Opinnot/Tik-111.550/1999/Esitelmat/Bluetooth/bluetooth.html >

[11] Schneier B., Applied Cryptography, 2nd Ed., John Wiley & Sons Inc., 1996, 758p.

[12] Stajano F. & Anderson R., The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks, 1999-04-19 [referred 2000-03-10]
< http://http://www.cl.cam.ac.uk/~fms27/duckling/ >

[13] Zhou L. & Haas Z., Securing Ad Hoc Networks [referred 2000-03-10]
< http://www.ee.cornell.edu/~haas/Publications/network99.ps >

Further Information

Bisdikian C., The Bluetooth End-user Experience, IBM T.J. Watson Research Center
: Comments from an end-user

Bluetooth SIG
: WWW page of the Bluetooth Special Interest Group

Bluetooth.net
: More general information about Bluetooth

Bluetooth SIG, Putting It Simply
: Story about Bluetooth

Cataldo A., Backers look to extend reach of Bluetooth, 2000-02-03
: Contemplation about the future of Bluetooth

Ericsson Bluetooth Core
: Information about Ericssons Bluetooth Products

Gilmore J., Bluetooth Personal Radio LAN Has Bogus Encryption, 1998-07-30
: A bit outdated information about the encryption of Bluetooth

Mettälä R., Nokia Mobile Phones, Bluetooth Protocol Architecture
: Some information about the architecture of the Bluetooth protocol

Mobic.com, Bluetooth-document, 1999-07-26
: Mobic's Bluetooth information page

Motorola's Bluetooth Page
: Information about Motorola's Bluetooth Products

Option International, Technology: Bluetooth, 1998-02-14
: Option International's page about Bluetooth technology

Seybold A., Bluetooth Technology: The Convergence Of Communications And Computing, 1998-05
: Article about the convergence of communications and computing

TechWeb, Intel Demonstrates Bluetooth Suite Protocol, 1999-08-12
: News article about Intel's Bluetooth demonstration