Security Issues for Wearable Computing and Bluetooth Technology

Catharina Candolin
Catharina.Candolin@hut.fi

Telecommunications Software and Multimedia Laboratory
Helsinki University of Technology
P.B. 5400, FIN-02015 HUT, Finland

Abstract

Bluetooth is a short-range wireless cable replacement technology enabling restricted types of ad hoc networks to be formed. All the while, a need for connecting wearable devices, such as PDAs, mobile phones, and mp3-players, is rising. Such networks may be formed using Bluetooth technology, but issues such as security must be taken into consideration. Although an attempt to tackle security is made, the result is too weak to be used for anything else than for personal purposes.

1 Introduction

A current trend in networking is focused on personal computing with wearable devices and means of interconnecting them wirelessly. For example, users may wish to easily connect all their electronical gadgets in order to exchange information between them or to share resources and services on the fly without having to plug them to each other using a lot of wires. Also, the users may wish to carry, or wear, the network as they go along, and keeping such a network connected by cables would be extremely inconvenient. However, wireless communications are more vulnerable to attacks, and as the users are moving along in untrusted environments, their networks are constantly subject to possible malicious attacks.

Numerous wireless technologies exist, and they all have their suitable area of application. Technologies suitable for personal wearable computer networks should be short-ranged, simple in design, and take power consumption into consideration. Possible candidates for such networks are the IEEE 802.11 WLAN, which is simple in design and supports power control, and Bluetooth, which also supports power control. The main difference between these technologies is that WLAN may also be used for real wireless networking, whereas Bluetooth is merely a cable replacement technology. However, this paper will mainly focus on

the suitability of Bluetooth as a technology for connecting wearable computers in a secure fashion.

The paper is structured as follows: Section 2 discusses the concept of ad hoc networking, wearable computing, and Bluetooth technology. Section 3 specifies some security criteria to be met in networks connecting personal wearable devices. Section 4 contains an overview of Bluetooth security, and section 5 analyses the Bluetooth security features in the context of wearable computing with respect to the criteria. Section 6 proposes some solutions for wearable computer networks, and Section 7 concludes the paper.

2 Background

An ad hoc network is a collection of nodes that do not need to rely on a predefined infrastructure to keep the network connected. The main characteristics of such networks are that most or all nodes take part in network operations, such as routing and network management, and that the network topology is dynamic due to node mobility and frequent additions and deletions of nodes. Also, communications is generally peer-to-peer, thus allowing all nodes within range of each other to exchange information and share resources.

Bluetooth [2] is a short-range wireless cable replacement technology, which adopts a master-slave architecture to form restricted types of ad hoc networks called piconets[1]. A Bluetooth piconet can consist of eight devices, of which one is the master and the other are slaves. Each device may take part in three piconets at most, but a device may be master in one piconet only. Several connected piconets[2] form so called scatternets.

Wearable computers, such as PDAs, mobile phones, and mp3-players, have become extremely popular, and a need for interconnecting these devices is rising. The devices should be able to form the network in an ad hoc fashion and to configure themselves accordingly without little or no user intervention. Such personal networks should be restricted to the user carrying them along in order to avoid active attacks or eavesdropping. Also, the devices must not be able to freely interact with strange devices and disclose information that may violate the privacy of the user. These threats are especially relevant should the user ever wish to connect the personal network to a public network such as the Internet.

The main objective of this paper is to evaluate Bluetooth security in the context of wearable computer networks. We restrict our focus to wearable networks consisting of devices belonging to, or controlled by, the same user only. Security concerns of connecting several personal networks belonging to different users is out of scope of this paper.

[1] The nodes in the piconet are relying on an infrastructure, namely the master-slave architecture, and they may not communicate in a peer-to-peer fashion. It is therefore not completely correct to talk about ad hoc networking in the context of Bluetooth.

[2] The maximum number of piconets is ten without performance degradation.

3 Criteria

The following criteria are specified for the personal network considered in this paper:

Criterion 1 *Privacy: the wearer of the device should be able to protect his private information from disclosure to unauthorized parties. The devices must not reveal any information that could be of use for purposes of identifying their users (for example, in order to track them).*

Criterion 2 *Data confidentiality: information transmitted over the communication channel must be protected from disclosure to unauthorized parties.*

Criterion 3 *Data integrity: information transmitted over the communication channel must not be altered by an unauthorized party.*

Criterion 4 *Access control: unauthorized nodes should be prevented from accessing services in the network.*

Criterion 5 *Availability: services in the network must be available to authorized users within a given time interval.*

Criterion 6 *Source authentication: it should be possible to verify the claimed source of a message.*

4 Bluetooth security

Wireless communications is extremely vulnerable to security attacks and eavesdropping. In order to provide security resembling the physical security of the wires in a fixed network, security can be enforced on the link layer, although securing the link is hardly enough to provide overall security of the network. However, there is nothing that prevents security mechanisms from being enforced on higher levels as well, for example, to provide end-to-end security. [3]

Bluetooth security was designed to meet the requirements of usage protection and information confidentiality, and is enforced on the link layer. The security services provided are entity authentication and data encryption. The entities referred to in this context are the devices rather than the users.

Bluetooth specifies four entities that are involved in providing security at the link layer: a unique 48-bit IEEE address (BD_ADDR), a 128-bit secret authentication key, an 8–128-bit secret encryption key, and a random number (RAND). Key management and random number generation will be further discussed below.

Also, different security levels are specified both for devices and services. Devices may be either trusted or untrusted: a trusted device has unrestricted access to all services, whereas an untrusted device has restricted access. For services, three levels are specified: services requiring authorization and authentication (except for trusted devices), services requiring authentication only, and services open to anyone.[1]

4.1 Random number generation

A pseudorandom number generator is a deterministic algorithm which, given a truly random binary sequence of length k, outputs a binary sequence of length $l \gg k$ which *appears* to be random. The input of the pseudorandom number generator is called the *seed*, while the output is called a *pseudorandom bit sequence*.[12]

The Bluetooth specification sets two requirements for the pseudorandom number generator:

1. The random numbers must be **non-repeating**: it shall be highly unlikely that the value should repeat itself within the lifetime of the authentication key.

2. The random numbers must be **randomly generated**: it shall not be possible to predict the value with a chance that is significantly larger than 0, for example, greater than $1/2^L$ for a key length of L bits.

In [7], it is argued that pseudorandom number generators are their own type of unique cryptographic primitive. They form a single point of failure for many real-world cryptosystems, and an attack could make even a careful selection of good algorithms and protocols irrelevant. Many systems also tend to use badly designed pseudorandom number generators, or use them in ways that make attacks easier than they need be.

In order to gain confidence that the pseudorandom number generator is secure, it should, for example, be submitted to various statistical tests designed to detect the specific characteristics expected of random sequences. Passing these tests is a necessary condition, but by no means sufficient. Pseudorandom number generators and statistical tests are discussed further in e.g. [12]. Statistical test suites, evaluation frameworks, and the interpretation of the results obtained are discussed in [13].

4.2 Key management

Key management involves creating, storing, and distributing keys for the purpose of encryption, authentication, and authorization, and therefore forms the basis for the security of the whole network. Although public key cryptography would ease the burdain of key management significantly, it is not considered in the Bluetooth specification. One reason may be that public key cryptography requires more computational power than symmetric key cryptography, but elliptic curve cryptosystems may present a solution to this problem in the near future. However, key management might not become a problem if Bluetooth is used for the purposes it was designed for.

Bluetooth specifies five different types of keys; four link keys[3] and one encryption key. The link key is a 128-bit random number which is shared between two or more parties and forms the base for all security transactions between these

[3]Traditionally, the term *link key* has been used to denote the session key used for encrypting a communication link.

parties. It is also used for deriving the encryption key. The link keys may either be semi-permament or temporary. The term *current link key* is used to denote the link key in use at the current moment.

The following keys are defined:

1. The comination key K_{AB}: The combination key K_{AB} is generated by the units A and B.
2. The unit key K_A: The unit key is generated in unit A only, and is hardly ever changed. Bluetooth units that have very limited memory capacity or units that need to be accessible to a large group of users are recommended (by the specification) to use the unit key.
3. The temporary key K_{master}: The master key is to be used only during one ongoing session. It replaces the original link key temporarily when the master wishes to reach more than one units simultaneously using the same encryption key.
4. The initialization key K_{init}: The initialization key is used during the initialization process before the combination or unit keys have been defined and exchanged, or when the link key is lost. The purpose of the initialization key is to protect the transfer of the initialization parameters, and it should not be used after that. The key is derived from a random number, a PIN-code, and the BD_ADDR of the claimant unit.
5. The encryption key K_c: The encryption key is an 8–128-bit key, which is derived from the current link key and a random number, which is issued by the master before encryption.

The master key is distributed among several nodes. Since all nodes now share a common secret, there is nothing that would prevent a node from successfully impersonating another node. Sharing the same secret between several nodes is dangerous; not only must the nodes be trusted when the secret is distributed, they must remain trusted as long as the keys are used. To be more specific, the node must be trusted to behave properly, not to share the secret with outsiders, and not to get stolen. This assumption is worth questioning even in traditional networking, but even more so in ad hoc networking due to, for example, the weak physical security of the nodes.

4.3 Authentication

Authentication in Bluetooth is performed using a challenge-response scheme. The verifying unit A sends a random number to the claimant unit B. B computes a signed response (SRES) consisting of the key K that A and B share, its BD_ADDR, and the received random number. A performs the same computation. B sends the SRES to A, which compares the received SRES to the one it computed itself. If they equal, authentication succeeds, otherwise authentication fails. The authentication protocol is described in Figure 1.

In order to prevent a node from repeating the authentication attempts, the verifier node keeps a list of nodes (defined by their BD_ADDR) that have tried

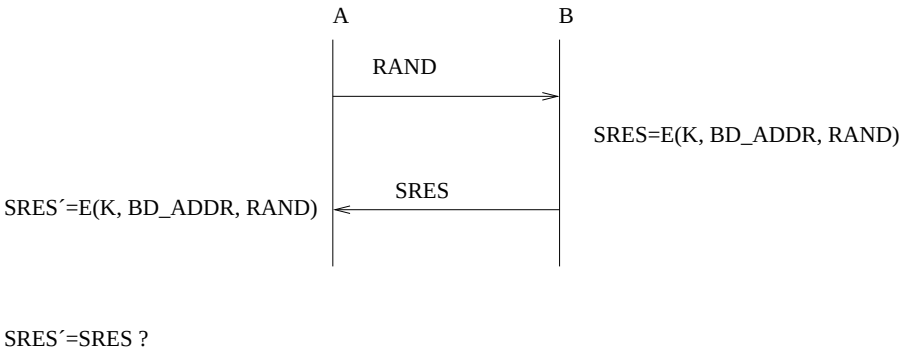

Figure 1: The authentication protocol

to authenticate themselves but failed. These nodes are not allowed to make any new attempts within a given time interval. If authentication fails again, the time interval is increased exponentially.

The Bluetooth authentication process is vulnerable to a variety of denial-of-service attacks. In fact, the means of handling repeated failed authentication attempts allows an attacker to use the BD_ADDR identifiers of other units in order to cause failed authentications. The verifier will place these devices on its black list, thus disabling any authentication attempts from legitimate devices. The attacker will naturally iterate through the addresses several times, thus causing a situation where the legitimate nodes are never even allowed to attempt to authenticate themselves. At the same time, assuming that the attacker also presents the verifier with bogus addresses, the memory of the verifier is filled with entries in the black list. If the verifier node has not taken such events into consideration by restricting the number of entries in the list, the attacker may as good as well fill up all available memory. On the other hand, if the number of entries is restricted, but the number of attempting addresses is larger than the maximum number of entries, then there will be a number of nodes that are not blacklisted at all. At the same time, the attacker could try different keys while he is at it, but this is not the main purpose of this attack.

Another possible denial-of-service attack would be if the verifier unit A sent a random number to the claimant unit B, and the malicious unit C would return a garbage value to A while B is busy computing the SRES, and claim that it came from B. Since A is working on a FIFO basis, it will now neglect the response coming in somewhat later from B, and deny B to try again within a given time limit. The malicious node may continue this process as long as it finds it amusing, and B will thus be prevented from authenticating itself to A.

4.4 Encryption

Encryption in Bluetooth is performed using the encryption algorithm E_0, which is resynchronized for every payload. The process is divided into three parts: payload key generation, key stream bit generation, and encryption/decryption. The payload key generator combines the input bits consisting of the encryption key K_c, a 128-bit random number, the BD_ADDR, and the 26 master clock bits

CLK_{26-1} in an appropriate order and shifts them to the Linear Feedback Shift Registers of the key stream generator. The output is then used for encryption and decryption.

In [6], a divide-and-conquer attack on the E_0 algorithm is introduced. However, the attack is theoretical in the sense that an attacker is never provided with a sufficient amount of output bits to be able to use this attack against Bluetooth. The cryptographic properties of the combination generator is further discussed in [5].

5 Analysis

Each type approved Bluetooth device is equipped with a 48-bit address that uniquely identifies it. This address is quite freely broadcast, thus making it possible to track user movements. The requirement for privacy is clearly violated by this feature.

In order to protect data from being disclosed, encryption may be enforced on the link. However, the encryption key is derived from the link key, and the generation and transmission of the link key relies on the security of the pseudorandom number generator. Therefore, it is hard to estimate whether the security of the communication channel is met, since it all depends on the manufacturer's interpretation of how to implement the pseudorandom number generator. An attacker could have broken the current link key at some point and would thus be able to derive the encryption key as well.

The integrity of the data transmitted is not checked either. A malicious user may possess the shared secret of two communicating nodes and may now perform man-in-the-middle attacks and modify the data. The communicating parties have no means of knowing whether or not the data has been tampered with, nor will a node have any means of verifying the true source of a message.

Access control in Bluetooth is primarily based on authentication. Authentication is traditionally defined as the process of verifying a claimed identity. The concept of entity authentication in a small personal network may seem feasible, since the owner of the devices tend to be the same person, and we may assume that he can manage the keys himself, thus creating some knowledge or shared secrets between the nodes. When the nodes authenticate each other, they actually verify that the other node knows the secret, and is thus the node it claims to be. However, should the user wish to connect the personal network to another network, then authentication is no longer feasible. It seems like ridiculous nonsense for two devices that have no prior knowledge of each other, such as a laptop and a LAN access point, to try to authenticate each other. How can an identity, of which the other party knows nothing about, be verified? The Bluetooth specification suggests the usage of PIN codes to be used in both devices; however, if the LAN access point is situated 5 meters above the head of the user, it is quite unlikely that the user will find a ladder to climb up the wall in order to enter a PIN into the access point[4].

Bluetooth is also vulnerable to a variety of denial-of-service attacks. It is pos-

[4]Usability aspects are, however, out of scope of this paper.

sible to keep the nodes from accessing any services by making sure that the verifier node always has an entry of them on its blacklist. Also, wireless communications is easy to disturb. Although some Bluetooth manufacturers claim to have tested the coexistence of Bluetooth and WLAN, they have not published any information about how the tests were performed. It is reasonable to believe that Bluetooth is not capable of coexisting with other technologies without disturbance. Hence, the improper design of the technology conveniently becomes a denial-of-service attack on itself.

6 Solutions for wearable computing

Key management is the basis for the whole security of the network. Although schemes based on symmetric keys are extremely impractical, the may still be feasible for small networks intended for personal use only. The key management scheme in Bluetooth may therefore be feasible if the following assumptions hold:

1. The user is in control of all devices and takes part in the process of distributing keys. Keys are distributed in a safe environment.
2. The devices are never compromised, that is, the user is in possession of the devices at all times, and takes proper action if they should get stolen.

A resurrecting duckling security model, as presented in [14], could well be applied to the personal network. The slaves would recognise the first node to send them a secret key as their master, and would thereafter allow only the master to control them. Secret keys could be exchanged by physical contact. Such a scheme leaves no ambiguity about which entities are involved in the key exchange, the shared secret may even be transferred in plain text, and it is still quite easy for the user to perform this operation since the size of the network is small. The user could naturally clear this relationship if necessary, and the slaves are then free to accept new masters.

The resurrecting duckling model does not imply that the slaves are not able to communicate and use services of other nodes, it merely states that the slaves are not controllable by anyone else than the master. However, to ensure user privacy, the nodes should not be allowed to talk to strangers without explicit permission of the user. In other words, the nodes are not allowed to answer to any inquiry messages coming from strange nodes.

Since the user of the wearable computers tends to keep the devices close to each other, especially when carrying them around, there seems to be little need for the high transmission power offered by Bluetooth. A range of 10m is too large for a personal network — a person is hardly even taller than 2m, and even so, the devices are likely to be carried in such a way that they are no further than 1–2m from each other. The user should be able to at least turn down the transmission power of the devices to cut down the range of the network. Although not a solution as such, it would make eavesdropping and some active attacks much harder.

If the personal network is based on IP, then most of the security concerns of Bluetooth may be forgotten. Security can be enforced on the network layer by

using IPSec [10]. IPSec provides two protocols, the Authentication Header (AH) [8] that provides data integrity and source authentication, and the Encapsulating Security Payload (ESP) [9] that provides data integrity, source authentication, and encryption. Key management could be performed using the ISAKMP [11] framework, and the IKE protocol [4] for key exchange. Access control may conveniently be based on certificates, and authentication could be forgotten completely, since it serves no purpose. Naturally, this means that Bluetooth security could be set to the lowest level possible, that is, no security at all.

The user should also be aware of the risks of ever connecting the piconet to a public network, such as the Internet. While it might seem like a nice idea to offer connectivity to the nodes, it also offers connectivity for malicious users attacking the piconet from outside. The piconet should choose one of the nodes, preferably the master, to function as a gateway to the Internet. The gateway should preferably use a real networking technology for connecting to the LAN access point, for example, WLAN. The other nodes in the piconet may get access to services on the Internet via the master, but the master is also responsible for restricting access to the nodes from outside. In a sense, it could be functioning as a type of firewall between the piconet and the Internet.

7 Conclusions

Bluetooth was originally designed to be a cable replacement technology. For some reason, the whole concept has fallen victim to the same kind of hype that another telecommunications wheel reinvention technology, WAP, has. Bluetooth is supposed to challenge WLAN as a new wireless networking solution. Needless to say, even if Bluetooth was the most secure technology ever invented, it is technically too restricted and complex to be anything else than the cable replacement technology it was designed for.

However, Bluetooth might be quite suitable for small personal networks, such as the wearable computer networks described in this paper. The basic assumption is that the user is in control of the network and participates in the process of distributing the secrets that are to be shared between the nodes. If the user wishes to connect this network to a public network, then a real networking technology, such as WLAN, should be used, and access to the personal network should be restricted from the outside.

If Bluetooth ever hits the market, then it may be used as an underlying network medium with the main purpose of transferring data. Security mechanisms should preferably be enforced by higher layers if the network is to be used for anything else than personal, completely controlled, purposes. For example, for the purposes of m-commerce, Bluetooth security as such is too weak to be relied on.

Acknowledgements

We thank Hannu Kari and Arto Karila for fruitful discussions around wireless networking technologies and security, Johan Wallen for references to and dis-

cussions around random number generation, and Janne Lundberg for comments on the paper as it was written.

References

[1] *Bluetooth Security Architecture*, 1999.

[2] *Specification of the Bluetooth System*, 1999.

[3] A. Aziz and W. Diffie. Privacy and authentication for wireless local area networks. 1993.

[4] D. Harkins and D. Carrel. The Internet Key Exchange (IKE). RFC 2409, November 1998.

[5] M. Hermelin. Cryptographic properties of the Bluetooth combination generator. Master's thesis, Helsinki University of Technology, 2000.

[6] M. Hermelin and K. Nyberg. Correlation Properties of the Bluetooth Combiner Generator. In *ICISC '99*, volume 1787 of *Lecture Notes in Computer Science*. Springer Verlag, 2000.

[7] J. Kelsey, B. Schneier, D. Wagner, and C. Hall. Cryptanalytic Attacks on Pseudorandom Number Generators. In *Fast Software Encryption, Fifth International Workshop*. Springer-Verlag, 1998.

[8] S. Kent and R. Atkinson. IP Authentication Header. RFC 2402, November 1998.

[9] S. Kent and R. Atkinson. IP Encapsulating Security Protocol (ESP). RFC 2406, November 1998.

[10] S. Kent and R. Atkinson. Security Architecture for the Internet Protocol. RFC 2401, November 1998.

[11] D. Maughan, M. Schertler, M. Schneider, and J. Turner. Internet Security Association and Key Management Protocol (ISAKMP). RFC 2408, November 1998.

[12] A. Menezes, P. Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.

[13] J. Soto. Statistical Testing of Random Number Generators. In *Proceedings of the 22nd National Information Systems Security Conference*, 1999.

[14] F. Stajano and R. Anderson. The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks. In *http://www.cl.cam.ac.uk/~fms27/duckling/*, 1999.