Global Information Assurance Certification Paper

Bluetooth Security

Steven Vittitoe
Submitted on 4/13/2004

Table of Contents

1.0 Abstract
2.0 Technical Details
2.1 RF Specifications
2.2 FHSS
2.3 Baseband
2.4 LMP
2.5 L2CAP
2.6 SDP
2.7 RFCOMM
2.8 Modes
3.0 User Perspective
4.0 Security Concerns
4.1 Authentication
4.2 Encryption
4.3 Authorization
5.0 Risk Mitigation
6.0 References

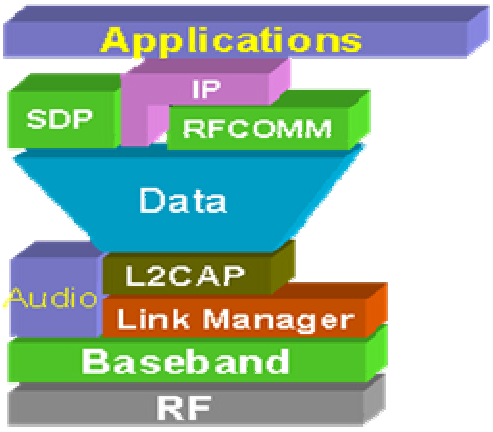
1.0 Abstract

Bluetooth is a wireless protocol that is rapidly gaining popularity. In fact, there are now more Bluetooth than WiFi enabled devices in existence [1]. Bluetooth is becoming an integral part of personal digital assistants (PDAs), cell phones, headsets, and several other frequently used devices. As our culture and data become increasingly reliant on Bluetooth systems, it is essential to recognize and be proactive with their information assurance features and shortcomings.

The aspiration of this work is to familiarize you with: the technical specifications of Bluetooth, how a Bluetooth network is established, security concerns surrounding such networks, and how to mitigate your risk when using the Bluetooth protocol.

2.0 Technical Details

The Bluetooth specifications are very involved. The core documentation alone is over 1000 pages in length and specifies Open System Interchange (OSI) layers one through seven. Obviously, we cannot cover everything in this section. However, a basic knowledge of how Bluetooth communications are setup, managed,



and secured is required before we can understand Bluetooth's security problems.

The diagram to the right [10] is a layered overview of the how Bluetooth divides communication operations. With the exception of IP, Audio, and Application, each module is going to be covered in this section.

2.1 RF Specifications

Bluetooth operates in the S-band ISM. ISM stands for industrial scientific and medical. The ISM bands are unlicensed, meaning anyone can use them free of charge. The only requirement is that all devices using ISM bands operate within specified power ranges.

Bluetooth has three classes of devices: class 1 operating at 1mW, class 2 operating at 2.5mW and class 3 operating with 100mW [2]. Most Bluetooth devices in use today are fall into the class 1 or 2 category [1]. The primary reasons for this are to increase battery life and to minimize the amount of interference generated by the device. Generally, the more power a wireless device uses the farther it can transmit. For example, a class 3 Bluetooth device can be expected to transmit approximately 100m, while class 1 devices will only transmit about 10cm [2]. Under favorable conditions these ranges can be exceeded.

ISM bands are allocated to frequencies that commercial users are not likely to want. For example, the 2.4GHz band is where the microwave oven operates. Interference generated by microwave ovens makes this band unappealing for commercial uses. Additionally, it would not make much sense to require everyone that owned a microwave oven to be government licensed. So, ISM bands are open for anyone to use – and listen to. Because of this, the ISM bands are full of noise making it very difficult for data communication to take place.

Class	Power	Distance
Class 1	1 mW	.1 m
Class 2	2.5 mW	1 m
Class 3	100 mW	100 m

Table 1 [8]

2.2 FHSS

FHSS stands for Frequency-Hopping Spread Spectrum. It is the specification Bluetooth uses at OSI layer one – the physical layer. FHSS, unlike other spread spectrum technologies, uses a narrow band system to communicate. Spread spectrum technologies are very resistant to interference, hence, their popular use in ISM bands and wireless data protocols [8].

Bluetooth's implementation of FHSS uses 83.5MHz of total bandwidth. Including frequency padding, this gives Bluetooth 79 1MHz frequency channels to hop between every 625 microseconds [9]. The hopping rate and pattern determines the communication channel in FHSS – not just the frequency alone. Some countries, such as France, use a smaller band with only use 23 RF channels defined. In either case, FHSS is used.

For two Bluetooth devices to communicate they must align on the same hopping pattern. This can take a long time to do. Sync times range anywhere from 2.5 to 10

seconds [1]. From a security prospective, this feature might appear to add an additional layer of protection. In fact, it is little more than a minor annoyance to any cracker (a malicious and unskilled hacker) with the right software and a little bit of time. More will be discussed on this topic later in this paper.

2.3 Baseband

The baseband is primarily responsible for managing channels and links between Bluetooth devices. In addition, it handles several other functions including: error correction, hop selection, data whitening, and the one we are most interested in security [9].

When two or more Bluetooth devices communicate on the same channel a piconet is formed. These are usually temporary and are to be used in an ad-hoc fashion. Piconets use a master/slave structure. Meaning, each piconet consists of one master and at least one slave. The maximum number of slaves in a piconet is 7 [10]. However, several piconets can become linked to form one scatternet. So, an internet is to a network as a scatternet is to a piconet.

One of the major limitations of Bluetooth is that the data transfer rate of a piconet is only 1Mbps. If the protocol is to fail this will be one of the primary reasons why. On the other hand, given Bluetooth's extensive power and size constraints 1Mbps is pretty good.

The master sets the hopping sequence and phase within that sequence. The hopping sequence is calculated from the masters Bluetooth device address or BD_ADDR. The BD_ADDR is a 48bit MAC address just like the one used on Ethernet cards. On an important security note, the first three octets are vendor defined, meaning that they are always the same for all Bluetooth devices made by the same vendor. The table below outlines the common vendors and their associated MAC identifier.

Bluetooth's baseband handles two types of links: synchronous connection-orientated (SCO) and asynchronous connection-less (ACL). SCO is a point-to-point link and ACL is point-to-multipoint. SCO links generally carry voice information at a speed of 64 kB/s. Masters can only handle three simultaneous SCO links. [9]

MAC Prefix	Company name
000BAC	3Com Europe Ltd.
0001EC	Ericsson Group (pre Sony-Ericsson)
008037	Ericsson Group (Sony-Ericsson)
000AD9	Sony Ericsson Mobile Communications Ab
006057	Murata Manufacturing Co. Ltd. (Nokia)
0002EE	Nokia Danmark A/s
008098	TDK Corporation
0080C8	D-link Systems Inc. (CSR Chipset)

0050CD	Digianswer A/s
0003C9	Tecom Co. Ltd.
000393	Apple Computer Inc.
00033A	Silicon Wave Inc.
00025B	Cambridge Silicon Radio
000361	Widcomm Inc.
000A1E	Red-M (Communications) Limited
001060	Billionton Systems Inc.
00E003	Nokia Wireless Business Communications
0002C7	Alps Electric Co. Ltd. (Ipaq 38xx)
00D0B7	Intel Corporation (Bluetooth)
000476	3 Com Corporation (Bluetooth)
00308E	Cross Match Technologies Inc. (Axis)
00081B	Windigo Systems
00037A	Taiyo Yuden Co. Ltd.
00E098	AboCom Systems Inc. (Palladio USB CSR Chipset)
004005	Ani Communications Inc.
0007E0	Palm Inc.

Table 2 [13]

2.4 LMP

LMP stands for link manager protocol. The primary responsibility of this protocol is to manage piconets, configure baseband links (SCO and ACL), and establish security mechanisms. The protocol is nothing more than a set of messages called PDU's or protocol data units. They are as follows:

- General response
- Authentication
- Pairing
- Change of link key
- Change of current link key
- Encryption
- Clock offset request

- Slot offset request
- Timing accuracy request
- LMP version
- Supported features
- Switch Master-Slave role
- Name request
- Detach
- Hold mode
- Sniff mode
- Park mode
- Power control
- Channel quality-drive change
- Quality of service (QoS)
- SCO links
- Control multi-slot packets
- Paging scheme
- Link Supervision
- Connection Establishment
- Test mode
- Error Handling

As you can see, there is a lot happening at this layer. Each one of the above listed operations has its own packet type and format. We do not need to know how every one of these messages works to understand Bluetooth security, but a couple might stand out as interesting: authentication and encryption.

The authentication function uses a challenge and response procedure. That is to say that the verifier sends a challenge message based on a random number to the claimant. Then, a response is calculated and sent back to the verifier based on the BD_ADDR and secret key of the claimant. If this response matches the verifier's expectations the claimant is authenticated. Both master and slave devices can be claimants or verifiers. The only catch is that the two devices must share a secret key. [9]

Encryption functions can be used after at least one authentication has taken place. For encryption to be used, the master device of a piconet sets a temporary key to be used as the link key for all slaves.

All of this sounds very good. Encryption and authentication functions happen at a very low level. The problem is that several of the functions offered at the LMP layer are entirely optional. In fact, many devices manufactured today ship with default settings to disable the majority of Bluetooth's security features. Optional security has become no security at all.

Additionally, several of the algorithms and cryptographic functions used are based on weak keys. For example, the encryption functions only accept a 4 digit PIN that can be easily brute forced. Several security professionals feel that this false sense of privacy is actually more damaging that had the functions been disabled or removed entirely.

2.5 L2CAP

The Logical Link Control and Application Protocol (L2CAP) layer handles both connection and connection less (SCO and ACL) data services. In reference to the OSI layered model L2CAP is at layer 2 – the data link layer. A number of interesting and critical functions are implemented at this layer:

- Protocol Multiplexing
- Segmentation and Reassembly (SAR)
- QoS
- Groups

On the SAR side of things, L2CAP's primary job is to take packets from the baseband layer and convert them to chunks, which are usually composed of several packets. L2CAP is bi-directional in that operation – it will also convert chunks to smaller packets. This conversion frequently occurs because the maximum transmit unit L2CAP sets is larger than the maximum packet size of the baseband layer.

Also interesting to note, is that two L2CAP units can directly communicate through the Bluetooth signaling layer -- channel identifier (CID) 1 (or 0x0001). For the L2CAP layer to accept commands it must be able to determine the BD_ADDR of the sending Bluetooth device.

Do not confuse the QoS at this layer as guaranteeing any QoS level. Rather, L2CAP is merely responsible for the exchange of QoS related information. Guaranteed rates and QoS happen at other layers. [9]

2.6 SDP

SDP stands for service discover protocol. A similar protocol is used by Cisco with Cisco discovery protocol (CDP). Obviously, SDP is designed for Bluetooth networks and CDP is for use with routers.

Some clarification might be needed on exactly what constitutes a service. A service is anything that will "provide information, perform an action, or control a resource" on behalf of a client [9]. When a client uses SDP it will not only find desired services it will also gather information about those services.

For example, if the client wishes to print a document the first thing SDP will do is look for printers in range. SDP does this by sending a service class request with a value specifying printer service. The listening printer (now referred to as the SDP server) receives this request and responds saying it is available. Now, SDP can further interrogate the printer by addressing future (PDU's) directly to the printers BD_ADDR.

Usually, when you start up a Bluetooth device it will be in what is called discoverable mode. Basically, when this is enabled it allows the Bluetooth device to turn into a SDP server. Users can disable this option, but few do. Some of the tools presented will show a flaw in Bluetooth that allows devices to be discoverable even if the user has turned off the discoverable mode. [9].

2.7 RFCOMM

RFCOMM is yet another protocol. Its purpose is to provide serial port (RS232) emulation over a wireless link, in this case the L2CAP layer. Though higher layers

determine the actual number of devices that can connect, RFCOMM itself supports up to 60 devices at one time. [9]

RFCOMM does not support loop back addressing. So, if you are running a Bluetooth enabled system you will not be able to test functionality at the RFCOMM layer locally. You will have to use a separate device.

Two types of devices are supported by RFCOMM. The first are communications end points such as printers, headsets, and computers. The second are communication links that join network segments, for example, a modem. RFCOMM takes the tried and true ETSI TS 07.10 standard and adapts it slightly to make it conductive in a wireless environment.

2.8 Modes

Users of Bluetooth have the choice between three security modes. Each level designed for use by different applications. For example, an application that simply exchanges contact information would use security mode 1.

Security Mode	*Description*
Mode 1	No security at all. Authentication and Encryption completely bypassed.
Mode 2	Service-level security only. Authorization takes place at the L2CAP layer.
Mode 3	This is the highest level of security that Bluetooth offers. Security is taken into account at the LM level. Authentication and encryption are enabled at this low level.

Table 3 [3]

3.0 User Perspective

So, what happens on a higher level when you want to share a picture with a friend who is using a Bluetooth device? Hopefully, everyone is using at a class 2 or 3 devices that will allow for communication of distances over 10cm. Assuming all devices were turned on when you walked in range, the first step has already been done.

Bluetooth devices are constantly searching for one another. When one device comes in range of another they will try to talk. So, the user already has a listing of all discoverable devices in range. To send a picture the sender simply: selects the destination device, includes the file or picture, and pushes the send button. If a PIN is needed to establish encrypted communications then the user must input that manually. Usually, this is not the case simply because it is cumbersome to the users. Everyone wants security, but no one wants to have it interfere with operation.

From here, the two devices communicate through paging and signaling channels to establish a connection on the RF, LMP, and L2CAP layers of the Bluetooth protocol stack. Once a channel has been established, the receiving device can either accept or reject the file send request. If the user accepts, the Bluetooth protocol stack has already been established and file transfer control is turned over to the application layer.

If any of the implementations of application layer programs contain vulnerabilities we could very easily start to see a new breed of virus spring fourth. Instead of being spread through e-mail attachments viruses would be spread by walking past an infected Bluetooth device. It would require close physical proximity in much the same way a biological virus spreads. This is all theoretical at this point, but it is very reasonable.

Transfer of this nature raise concerns about proper user settings. If the receiver blindly, or by default, accepts files it is serious cause for concern. Then again, given the number of people that will open executable e-mail attachments seeing such behavior should not come as a big surprise.

Also worth noting is the fact that improper security modes can be used. It is entirely possible for a user to specify security mode 1 on an application transferring privileged information. Unless the users are specifically trained on how they should be using Bluetooth devices securely it is a good bet that the majority of them will not.

4.0 Security Concerns

So far, we have only alluded to potential security flaws in the various layers of the Bluetooth protocol. This section is going to specifically address how security was originally planned for in the Bluetooth stack. Bluetooth's security is based on three things: authentication, encryption, and authorization. None of them are perfect and they all could use some measures of improvement.

4.1 Authentication

Authentication is the act of verifying a user is actually who they say they are. It answers the question "How do I know where this information is coming from?" Bluetooth uses a challenge and response method of authentication [3]. The device asking the question is known as the verifier and the device attempting to authenticate with the verifier is know as the claimant. Below is a listing [3] of steps Bluetooth devices take in authenticating other devices:

1. The claimant transmits its 48-bit address (BD_ADDR) to the verifier.
2. The verifier transmits a 128-bit random challenge (AU_RAND) to the claimant.
3. The verifier uses an algorithm to compute an authentication response using the address, link key, and random challenge as inputs. The claimant performs the same computation.
4. The claimant returns the computed response, SRES, to the verifier.
5. The verifier compares the SRES from the claimant with the SRES that it computes.
6. If the two 32-bit SRES values are equal, the verifier will continue connection establishment.

** List provided by National Institute of Standards Technology [3]*

If the authentication is valid (SRES values are equal) a connection is passed to the next layer. If they are not equal, the connection is denied and a timer is implemented. A

random and exponentially growing delay must expire before the claimant can attempt to re-authenticate.

In step 3 you will notice the use of a link key. A link key is calculated when a connection is first established. Unlike the BD_ADDR this is a private key and is not sent over the air interface at all. We can think of the link key as a private key and the BD_ADDR as a public key. Though, this method of authentication is not the same thing as Public Key Infrastructure (PKI), it is a challenge-response.

Now that you have a good idea of how Bluetooth authentication works, can you find any potential problems with it? First, only the device is authenticated. Everything is based around the BD_ADDR and not any form of user identification. So, if your Bluetooth device is stolen or otherwise compromised the authentication mechanism is rendered practically useless.

Second, and this one is a little more difficult to pull off, the authentication mechanism is vulnerable to man-in-the-middle attacks [3]. This type of attack takes place when the hacker impersonates the device you wish to transmit to, intercepts your data, and then transfers it on to your original destination. If the attack is executed successfully a user will have no idea that their data was just stolen. This method of authentication is not end to end.

Additionally, the mechanisms for authentication are not based on the most secure algorithms or the strongest keys. The theory is that Bluetooth sessions are intended to be very quick; as such that an attacker will not have enough time to break the keys on a live session. For many situations this is a valid assumption. However, the future use of Bluetooth is unknown, and quick sessions could very easily fall out of popularity.

4.2 Encryption

Standards that do not ensure any means of confidentiality will have much difficulty being adopted. On the other hand, standards that offer perfect confidentiality but are generally overly burdensome to the user will not be adopted either. Certainly, one should never bother users when it is not necessary. A fine line between these criteria is walked when proposing a new protocol. Bluetooth attempts to do this by letting the user choose just how bothered they want to be. Again, we see the concept of modes with encryption:

Encryption Mode	*Description*
Mode 1	No encryption on anything.
Mode 2	Everything but broadcast traffic is encrypted with a master key.
Mode 3	Everything is encrypted.

Table 4 [3]

Without going into excessive technical details, know that each user has to pick a PIN. The PIN's must match for encryption to be enabled. There are two problems with this. First the PIN is usually only 4 digits long and 50% of the time BT users pick 0000. Here the problem is, again, not so much with the protocol as it is with user education.

Additionally, establishing PINS on a large network is not feasible, they will suffer from common password problems – written down, shared, repeated etc [3]. It would be a full time job to constantly change PINS on a 100,000 node network. It would be even more difficult for users to remember 50 different PINS. Bluetooth's encryption was not designed with the enterprise in mind.

4.3 Authorization

Authorization is very similar in concept to authentication. The primary difference is that authorization works between devices and services, not devices and devices. Think of this as an access list. For example, device A is authorized to use the printer service on device B. Device B simply keeps track of device A's BD_ADDR and which services it is authorized to use.

Because the authorization functions are so similar to the authentication functions they are victim to the same problems. For example, the authorization is not end to end. Again, if the device is stolen this layer is next to worthless. The same goes for man-in-the-middle attacks.

5.0 Risk Mitigation

The next question you are likely to ask is "How can I protect myself in this environment?" The paranoid answer is that you can not. Man-in-the-middle attacks have no defense under Bluetooth. Even educated users that do not take the security risks of Bluetooth seriously are likely to be compromised. For example, at a security conference attended by some of the nation's brightest security professionals, over half of people that possessed Bluetooth devices blindly accept a picture that was sent to their PDA, cell phone, or notebook.

A more practical answer to the question is to know your risks. Analyze the value of data before you use it with a Bluetooth device. Be aware of the risks you are taking by putting it on a Bluetooth device. If the risk is acceptable fine, if not it is back to hard wire. Bluetooth is not the only solution out there.

As mentioned above, user education is the biggest key to improving the security of Bluetooth. User should be instructed on how to set up encryption with PIN's, change between security modes, manage authorization lists, and secure their device from physical theft.

If Bluetooth is ever to meet its goal of total wire replacement it is going to need a security overhaul that will incorporate end to end authentication and authorization while allowing for auditing capabilities. The best way to avoid the risk from Bluetooth is to not use it at all. Bluetooth is still a very young protocol and new vulnerabilities are being actively release for it. The road ahead is rocky and long; make sure that you have your seat belts fastened.

6.0 References

1. Potter, B. & Caswell, B. (2003, August 1). *Bluesniff -- The Next Wardriving Frontier*. Retrieved March 24, 2004 from, Web site: http://www.shmoo.com/~gdead/dc-11-brucepotter.ppt

2. SIG Forum (2000, March 27). *Bluetooth Range in Relation to Different Power Classes*. Retrieved March 24, 2004 from, Web site: http://www.palowireless.com/infotooth/knowbase/general/10.asp

3. Karygiannis, T. & Owens, L. (n.d.). *Wireless Network Security 802.11, Bluetooth™ and Handheld Devices*. Retrieved March 24, 2004 from, National Institute of Standards Technology Web site: http://csrc.nist.gov/publications/drafts/draft-sp800-48.pdf

4. Niem, T. (2002, November 4). *Bluetooth And Its Inherent Security Issues*. Retrieved March 24, 2004 from, SANS Web site: http://www.sans.org/rr/papers/68/945.pdf

5. Vainio, J. (2000, May 25). *Bluetooth Security*. Retrieved March 24, 2004 from Helsinki University of Technology, Department of Computer Science and Engineering Web site: http://www.niksula.cs.hut.fi/~jiitv/bluesec.html#chap2.1

6. Hall, J. (20003, August 21). *Brush up on Bluetooth*. Retrieved March 24, 2004 from, SANS Web site: http://www.sans.org/rr/papers/68/1222.pdf

7. Gehrmann, C. (2002, April 4). *Bluetooth Security Whitepaper*. Retrieved March 24, 2004 from , Bluetooth SIG Security Expert Group Web site: http://www.bluetooth.com/upload/24Security_Paper.PDF

8. ARC Electronics (n.d.). *DSSS and FHSS - Spread Spectrum modem*. Retrieved March 24, 2004 from, Web site: http://www.arcelect.com/DSSS_FHSS-Spead_spectrum.htm

9. Palo Wireless (n.d.). *Bluetooth Tutorial - Specifications*. Retrieved March 24, 2004 from, Web site: http://www.palowireless.com/infotooth/tutorial.asp

10. Bhagwat, P. (2000, July 31). *Bluetooth Technology Overview.* Retrieved March 24, 2004 from AT&T Labs, Networking Research Group Web site: http://www.ietf.org/proceedings/00jul/SLIDES/ipobt-tech/sld001.htm

11. Leyden, J. (2003, November 17). *Bluetooth is attack vector for mobile phones*. Retrieved March 24, 2004 from, Web site: http://www.securityfocus.com/news/7466

12. Kotadia, M. (2004, February 9). *Bluetooth phones at risk from 'snarfing'*. Retrieved March 24, 2004 from, Web site: http://news.zdnet.co.uk/communications/wireless/0,39020348,39145881,00.htm

13. Whitehouse, 0., Halsall, S., Kapp, S. (2003, October 15). *Redfang - The Bluetooth Hunter*. Retrieved March 24, 2004 from, Web site: http://www.atstake.com/research/tools/info_gathering/