Global Information Assurance Certification Paper

An Overview of Bluetooth Security

Submitted by: Nikhil Anand

Date: February 22, 2001

Objective

This aim of this paper is to provide the reader with an overview of the Bluetooth Security Architecture as it applies to the Bluetooth protocol stack.
Towards the end the paper also briefly examines some of the potential flaws in the Bluetooth Architecure.

Introduction

Bluetooth is a recently proposed standard for local wireless communication that will allow physically disparate devices such as cellular phones, printers, Wireless Headsets, laptops, Personal Digital Assistants (PDA) etc. to communicate and exchange information with each other.

Thus, any two Bluetooth enabled devices would potentially be able to communicate with each other. Some of the highest priority usage models or practical applications of Bluetooth technology as envisaged by the Bluetooth SIG (Special Interest Group[1]) include the ability to transfer files and other objects such as electronic business cards between physically disparate devices such as cell phones and PDAs. Another potentially popular use would be to enable mobile phones as "wireless modems" to connect laptops to the Internet without using any connecting cable between the cell phone and the laptop.

One of the reasons that Bluetooth holds so much potential is that it promises to link up ubiquitous devices such as PDAs and cell phones to different types of hardware platforms thus opening up and linking different networks bringing about "pervasive connectivity".

However, amongst other things, security concerns are slowing the mass adoption of wireless technology and Bluetooth is no exception. Concerns over privacy of wireless communications still ranks high as a deterrent to the adoption of wireless technology.

The next section briefly describes the Bluetooth Architecture and focuses on describing the support for security that has been built into the Bluetooth specifications.

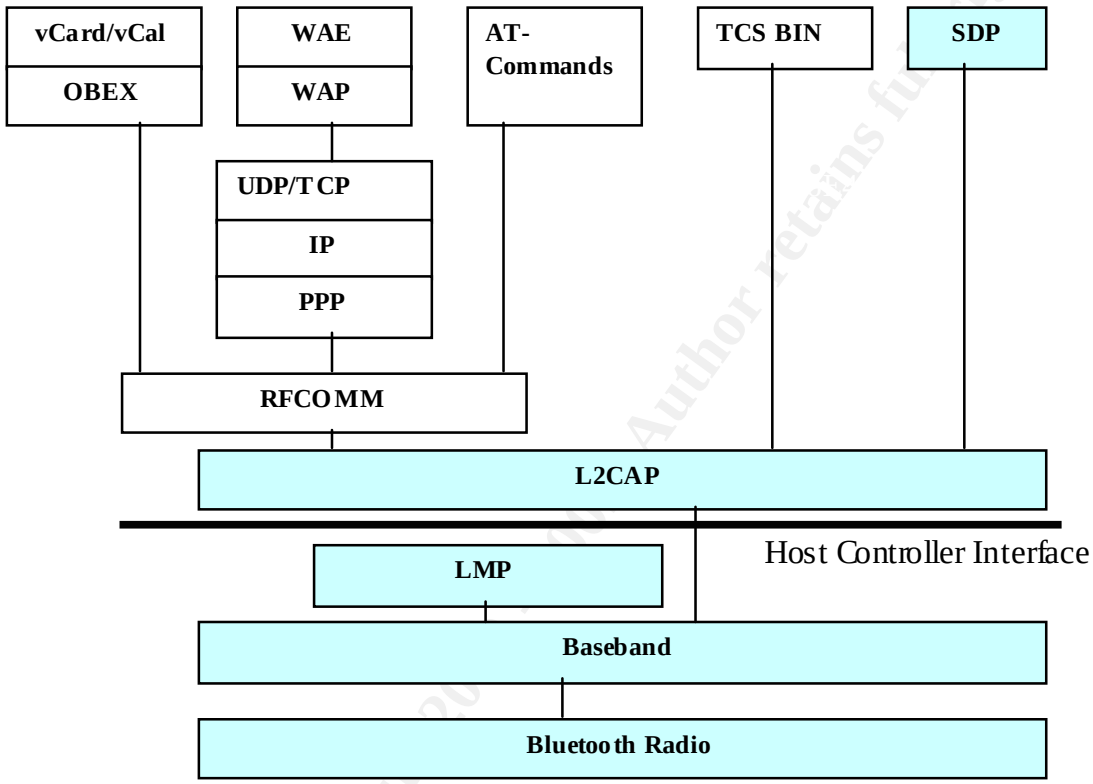
Bluetooth Protocol Architecture

The Bluetooth Protocol Architecture is essentially an amalgamation of Bluetooth specific protocols and other already in use (adopted) protocols such as WAP, WAE, TCP/UDP/IP, PPP, vCard, vCal and IrMC.
It also supports cable replacement protocols such as RFCOMM and telephony adapter protocols such as AT-commands. These protocols have been adopted to suit Bluetooth applications but they are not Bluetooth specific.
One of the reasons for this sort of "mixed" architecture is that it allows for the integration of Bluetooth directly into existing application and transport protocols without having to build up an entirely separate and parallel architecture.

[1] A consortium of companies that include Ericsson, Intel, Toshiba, IBM and Nokia

An advantage of this amalgamated structure is that it can allow application specific security controls to be implemented that would be transparent to the lower later security controls at which Bluetooth operates (Bluetooth operates at the Data Link Layer).

The Bluetooth protocol stack is shown in the diagram shown below:



Legend: The Bluetooth specific protocols are colored blue.

A brief description of the some of the protocols is given below:

SDP (Service Discovery Protocol)- The Service Discovery Protocol is needed so that Bluetooth enabled devices can gather information about device types, services and service specifications so that a connection between devices can be set up

Baseband- Baseband is the layer that enables the physical RF connection between Bluetooth enabled devices.

L2CAP (Logical Link Control and Adaptation Protocol)- This protocol adapts the upper layer protocols over the Baseband.

LMP (Link Manager Protocol)- This protocol works in parallel with L2CAP. While L2CAP is responsible for controlling the upper layer protocols, the LMP is responsible for setting up the link between two Bluetooth devices. It includes deciding and controlling the Baseband packet size, security services such as authentication and encryption using link and encryption keys.
Host Controller Interface (HCI)- The Host Controller Interface is used to provide a command interface to Baseband controller, Link Manager and other hardware controllers.

RFCOMM- RFCOMM is a cable replacement protocol. As can be seen a number of upper layer protocols interface with the RFCOMM protocol layer which in turn interfaces with the core Bluetooth Protocols. Thus no separate standard has to be designed for upper layer protocols to work with Bluetooth.

TCS BINARY and AT Commands- these are telephony control protocols and will allow services such as modems and fax to run over Bluetooth.

Together RFCOMM, TCS BIN and AT Commands and the other adopted protocols such as OBEX, TCP/UDP/IP, PPP and WAE/WAP form the application oriented protocols that run over the Bluetooth specific protocols.

Bluetooth Security Framework

We will first look at the generic security levels and features that have been incorporated in the Bluetooth specifications. We shall then see how the Bluetooth Architecture supports these features.

Bluetooth Security Features

The Bluetooth specification includes security features at the **link level**. It supports authentication (unidirectional or mutual) and encryption. These features are based on a secret link key that is shared by a pair of devices. To generate this key a pairing procedure is used when the two devices communicate for the first time.

Bluetooth devices transmit on the heavily used unlicensed 2.45GHz radio band (the same used by microwaves). To keep transmissions from breaking up, Bluetooth employs frequency hopping, a practice of skipping around the radio band 1600 times each second. This improves clarity and also reduces what Bluetooth proponents call "casual eavesdropping" by allowing only synchronized devices to be able to communicate.
Each Bluetooth device has a unique address, allowing users to have some trust in the person at the other end of the transmission. Once this ID is associated with a person, by tracking the unscrambled address sent with each message, individuals can be traced and their activities easily logged.
For Bluetooth devices to communicate, an initialization process uses a PIN. While some devices allow users to punch in an ID number, the PIN can also be stored in the non-volatile memory of the device.

Bluetooth enabled devices can operate in one of three different security modes as per the Bluetooth specifications:
Security Mode 1- This is the most insecure security mode in which the Bluetooth device does not initiate any security procedure. It is in a "promiscuous" or "discovery" mode, allowing other Bluetooth devices to initiate connections with it.
Security Mode 2- This mode enforces security after establishment of the link between the devices at the L2CAP level. This mode allows the setting up of flexible security policies involving application layer controls running in parallel with the lower protocols.
Security Mode 3- This mode enforces security controls such as authentication and encryption at the Baseband level itself, before the connection is set up. The security manager (explained later in this article) usually enforces this onto the LMP.

Bluetooth allows security levels to be defined for both devices and services.

For devices there are two possible security levels. A remote device could either be a:

1. **Trusted device**-Such a device would have access to all services for which the trust relationship has been set.
2. **Untrusted device**-Such a device would have restricted access to services. Typically such devices would not share a permanent relationship with the other device.

For services, three levels of security have been defined.

1. **Services that require authorization and authentication**. Automatic access is only granted to trusted devices. Other devices need a manual authorization.

2. **Services that require authentication only**. Authorization is not necessary.

3. **Services open to all devices**; authentication is not required, no access approval required before service access is granted.

Note: The Bluetooth Architecture allows for defining security policies that can set trust relationships in such a way that even trusted devices can only get access to specific services and not to others.

Fundamentally, the core Bluetooth protocols can be used to implement the following security controls to restrict access to services:

1. **Access to Services would need Authorization** (Authorization always includes authentication). Only trusted devices would get automatic access.
2. **Access to Services would need only authentication**. I.e. the remote device would need to get authenticated before being able to connect to the application
3. **Access to Services would need encryption**. The link between the two devices must be encrypted before the application can be accessed.

What is important to understand here is that **Bluetooth core protocols can only authenticate devices and not users**. This is not to say that user based access control is not possible. The Bluetooth Security Architecture (through the Security Manager) allows applications to enforce their own security policies. The link layer, at which Bluetooth specific security controls operate, is transparent to the security controls imposed by the application layers. Thus it is possible to enforce user-based authentication and fine grained access control within the Bluetooth Security Framework.

Security Manager

The Security features and policies that can be supported by Bluetooth as mentioned above are enabled by a component called the security manager. The security manager component is the entity that decides what policies are to be enforced when a connection request is made (both for inbound and outbound connections). Based on the service, device type and whether the device is trusted or untrusted the security manager can enforce application level authentication, encryption of the session and any other specific access policies.

The Security manager needs information regarding devices as well as services before it can take a decision whether or not to allow access and if so, to what services. This information is stored in two databases namely, the Device Database and the Service Database.

The Device database stores information about the device type, the trust level (whether trusted or untrusted) and about the link key (used for encryption) length.

The Service database stores information regarding the authentication, authorization and encryption requirements for the services. It also stores other routing information for the services.

The typical process (steps) followed by the security manager in granting access to a remote device to connect to a particular service is as follows:

1. Remote device requests access
2. Connection request comes to L2CAP
3. L2CAP requests security manager to grant access
4. Security Manager queries both device and service databases
5. If device is trusted, then security manager may or may not (depending on the implementation) ask for authentication or authorization
6. If the device is untrusted, the security manager may either terminate the connection (if so desired) or enforce authorization. Authentication at the core Bluetooth protocol level will happen when link keys are exchanged. Depending on the security policy governing access, the security manager might call upon an application protocol to enforce application level security such as a username/password scheme for authentication. Support is also built in for other authentication schemes through the security manager interface.

7. The Security manager will then decide if the service access requires link encryption. If so, keys will be negotiated and exchanged at the L2CAP protocol level and the connection will continue to be setup.

Alternatively, if the device is in security mode 3, the security manager instructs the LMP to authenticate and encrypt (if desired) the communication before the connection to the service is set up.

Thus we see that the Security manager is the central entity that manages and enforces security policy in the Bluetooth Security Architecture. In a way, the Security Manager acts as a bridge in terms of bringing together application level and core Bluetooth protocol level (Link Layer) security controls and thus helps in providing end to end security across layers.

Potential weaknesses

The Bluetooth Security Architecture, though relatively secure, is not without its share of weaknesses. There are a number of weaknesses in the architecture (both directly and indirectly) that can be potentially exploited.

A simple example, though not so simple to implement in practice is the man-in-the-middle attack for stealing identification and encryption keys before the start of a session and using the same to impersonate and/or eavesdrop on communications**. This problem is however not specific to Bluetooth**. Most key exchange systems are prone to this type of attack. One way to mitigate this would be to build in support for digital certificate based authentication systems. Another way might be to make it very difficult for an attacker to lock onto the frequency used for communication.
Making the frequency hopping intervals and patterns reasonably unpredictable may help to prevent an attacker from locking onto the device signal. These considerations have been factored in to some degree in the Bluetooth specifications. However as the paper by the Bell-Lab researchers [3] point out, this is not so difficult to break into.

The other issue deals with the PIN itself. Most devices have extremely short (usually 4 character) PINs. This is itself is a security weakness, though it is implementation and not specification related. Short PINs can be searched exhaustively by attackers.

Conclusion

Bluetooth today has still not reached a stage where it has been deployed and tested in a practical scenario. Bluetooth devices are still in the development stages. Apart from Toshiba, which has come out with a Bluetooth enabled PC Card, most other Bluetooth technologies are still on the drawing board.

It might therefore be premature to pass judgement on whether Bluetooth security is good enough without there being even a couple of products in the market. Acceptable levels of security might be defined more by the relevant use case scenarios rather than by absolute terms.
The general consensus, however seems to be that the Bluetooth Security Architecture is reasonably robust and granular in its present form and is quite secure even in its default state. However since application developers may or may not choose to incorporate security into application layers, it is possible that the security strength of Bluetooth devices will depend more on robust implementations than on making significant changes to the architecture.

References

1. Mekkala, Ritu. "Bluetooth Protocol Architecture". Version 1.0. August 29, 1999 http://www.bluetooth.com/developer/whitepaper/whitepaper.asp (February 18, 2001)

2. Muller, Thomas. "Bluetooth Security Architecture". Version 1.0. July 15, 1999 http://www.bluetooth.com/developer/whitepaper/whitepaper.asp (February 18, 2001)

3. Jakobsson, Markus and Wetzel, Sussane. "Security Weaknesses in Bluetooth". http://www.bell-labs.com/user/markusj/bluetooth.pdf (February 19, 2001)

4. "Bluetooth Wireless Technology bridging the gap between computing and communication"
http://www.intel.com/mobile/bluetooth/index.htm (February 18, 2001)

5. Sutherland, Ed. "Despite the Hype, Bluetooth has Security Issues that cannot be ignored". November 28, 2000.
http://www.mcommercetimes.com/Technology/41 (February 19 2001)