NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

SECURITY OVER BLUETOOTH

Graduation Project

COM- 400

Student: FADI HAMAD (20032652)

Supervisor: Mr. JAMAL FATHI

Nicosia – 2007

I dedicate my project to my father, mother, sister and brother.

ACKNOWLEDGEMENT

First of all, and before saying anything, I want to thank ALLAH who gave me the morale and opened my mind to get this project done.

I would like to thank to my supervisor Mr. JAMAL FATHI for his kindness, humility, patience, and support and for his friendly behavior with me and his words of encouragement kept me doing my project.

After that, I love to send my special thanks to my family, to whom I won't achieve all of this without them. I won't ever forget their encouragement and support as long as I am alive. And especially my father and my mother my praise for them can not be expressed in words. Thank you dear Dad, you have always been my ideal. Dear Mum, I am so glad to thank you at this particular moment of my life, your prayers helped making this day come true. Also special thanks to my brother and my sister.

And I would like to thank all my friends who helped me and encouraged me to do my work, and who were beside my for all these years.

Finally, I want to thank all the educational staff in Near East University who gave me their efforts and knowledge to get my Bachelor.

ABSTRACT

Bluetooth provides a short range wireless communication between devices making it convenient for users and thus eliminating the need for messy cables. According to Bluetooth Special Interest Group (2006), Bluetooth wireless technology is the most widely supported, versatile, and secure wireless standard on the market today. Bluetooth operates in the open 2.4 GHz ISM band and is now found in a vast array of products such as input devices, printers, medical devices, VoIP phones, whiteboards, and surveillance cameras. However, the proliferation of these devices in the workplace exposes organizations to security risks. So having security is something important. Bluetooth has several aspects of the security. Apart from security, several privacy issues linked to the use of Bluetooth protocols are also discussed. The focus is on the lower layer protocols, called the core Bluetooth protocols. Are these protocol secure enough? As such the Bluetooth protocols alone should not be used to ensure authenticity or privacy.

TABLE OF CONTENTS

INTRODUCTION

Bluetooth is a technology that enables all kind of electronic devices to communicate with each other. It is a wireless protocol and is usually used for short distance communications, about 10 to 100 meters. The Bluetooth protocol is being used by numerous mobile phone devices as a cheap connection method with nearby devices, by printers and other home appliances. It can be seen as the wireless equivalent of the USB protocol.

In this project the first chapter is all about explaining what is computer network. Also about the history of computer network and the goals of it.And ofcourse we will explain the kinds of networks which are common used nowadays.

The second chapter is talking about the activite network security and beginning by discussing the activate network mechanisms, limitations of static security,and finishing by explaining honeytrap systems.

The third chapter is all about Bluetooth, identifying what is Bluetooth and how this technology works.And explaining the Bluetooth networking,also it is including the protocols of Bluetooth and how we can establish a connection in Bluetooth.

And the last chapter it is going to talk about security over Bluetooth, starting from the security modes and going through the Bluetooth key generation from the PIN, Bluetooth encryption process, and talking about the problems with the Blutooth security.And finally,
discussing the Bluetooth attacks.

1. COMPUTER NETWORKS

1.1 Overview

Computer networks interconnect sets of autonomous computers, providing the means by which data can be dispatched from one computer for delivery to one or more of the other machines on the network. Exchange of information paves the way for resource sharing. Application programs and data sets stored on file servers can be made available to users of other network-attached computers; likewise, hardware devices, ranging from laser printers to back-up systems to communications gateways, can process data from other network machines.

1.2 What is Computer Network?

In information technology, a network is a series of points or nodes interconnected by communication paths. Networks can interconnect with other networks and contain sub- networks. A network consists of a number points, network elements (nodes), connected together for purpose of the mutual communication. Those interconnected points are network devices interconnecting particular segments or sub-networks, or end–user stations, i.e. PCs, workstations or servers. There are several criterions for networks classification, for example:

The network can be characterized in terms of spatial distance as local area network LANs, metropolitan area networks MANs, and wide area networks WANs.

The network can be characterized by the way of the links between network nodes are arranged for the purpose of communication, i.e. by network topology.

The network can be characterized by a media access control technique, i.e. by the way in which a signal transmission is organized in the case of multipoint media use.

The network can be characterized by the type of data transmission technology (i.e. network architecture) in use (for example, a TCP/IP or SNA network)

The network can be characterized by whether it carries voice, data, or both kinds of signals

The network can be characterized by who can use the network (public or private)

The network can be characterized by the usual nature of its connections (dial-up or switched, dedicated or non-switched, or virtual connections

The network can be characterized by the physical link type (for example, optical fibre, coaxial cable and twisted pair), etc.

1.3 How and Why Network Exists?

The concept of linking a large numbers of users to a single computer via remote terminal is developed at MIT in the late 50s and early 60s. In 1962, Paul Baran develops the idea of distributed, packet-switching networks. The first commercially available WAN of the Advances Research Project Agency APRANET in 1969. Bob Kahn and Vint Cerf develop the basic ideas of the Internet in 1973.

In early 1980s, when desktop computers began to proliferate in the business world, then intent of their designers was to create machines that would operate independently of each other. Desktop computers slowly became powerful when applications like spreadsheets, databases and word processors included. The market for desktop computers exploded, and dozens of hardware and software vendors joined in the fierce competition to exploit the open opportunity for vast profits. The competition spurred intense technological development, which led to increased power on the desktop and lower prices. Businesses soon discovered that information is useful only when it is communicated between human beings. When large information being handled, it was impossible to pass along paper copies of information and ask each user to reenter it into their computer. Copying files onto floppy disks and passing them around was a little better, but still took too long, and was impractical when individuals were separated by great distances. And you could never know for sure that the copy you received on a floppy disk was the most current version of the information-the other person might have updated it on their computer after the floppy was made.

For all the speed and power of the desktop computing environment, it was sadly lacking in the most important element: communication among members of the business team. The obvious solution was to link the desktop computers together, and link the group to shared central repository of information. To solve this problem, Computer manufactures started to create additional components that users could attach to their desktop computers, which would allow them to share data among themselves and access centrally located sources of information. Unfortunately the early designs for these networks were slow and tended to breakdown at critical moments.

Still, the desktop computers continued to evolve. As it became more powerful, capable of accessing larger and larger amounts of information, communications between desktop computers became more and more reliable, and the idea of a Local Area Network (LAN) became practical reality for businesses. Today, computer networks, with all their promise and power, are more complicated and reliable than stand-alone machines. Figure 1.1 shows the network connectivity of the world.

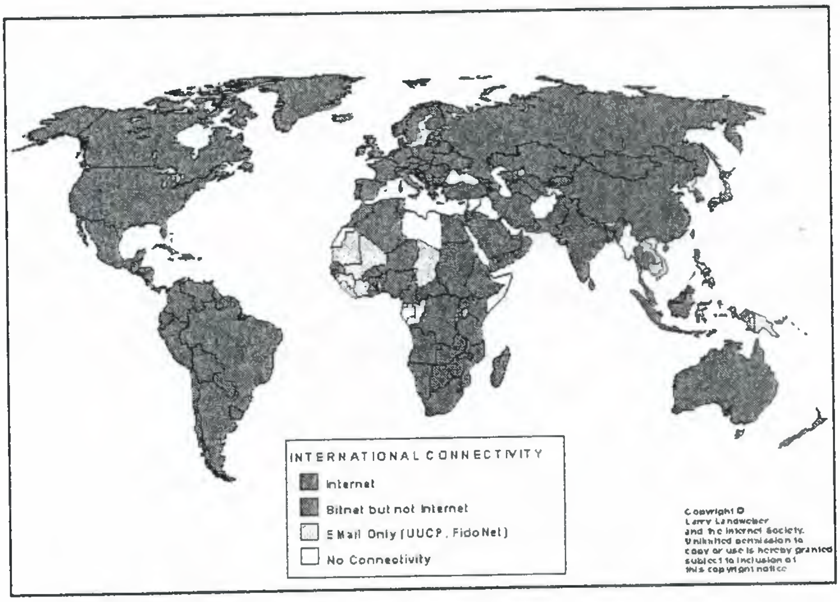

INTERNATIONAL CONNECTIVITY
▨ Internet
▨ Bitnet but not Internet
□ E Mail Only (UUCP, FidoNet)
□ No Connectivity
Copyright ©
Larry Landweber
and the Internet Society.
Unlimited permission to
copy or use is hereby granted
subject to inclusion of
this copyright notice

Figure 1.1 Computer Network Connectivity of the World

1.4 Goals of Computer Networks

1. Resource sharing and accessing them independently of their location.

2. Providing a universal environment for transmission of all kinds of information: data, speech, video, etc.

3. Supporting high reliability of accessing resources.

4. Distribution of loads according to the requirements very fast main frames, minis, PCs, etc.

1.5 The Communication Puzzle

In the near future, fourth-generation (4G) wireless technologies will be able to support Internet-like services. This provision will be achieved through a seamless integration of different types of wireless networks with different transmission speeds and ranges interconnected through a high-speed backbone, as depicted in Figure 1.2. Fourth generation wireless networks include Wireless Personal Area Networks (Wireless PANs or WPANs for short), Wireless Local Area Networks (Wireless LANs or WLANs for short), Wireless Metropolitan Area Networks (Wireless MANs or WMANs for short), Wireless Regional Area Networks (Wireless RANs or WRAN for short) Wireless Local Loops (WLLs), Customer Premise Equipment (CPE), cellular wide area networks and satellite networks (see Figure 1.2). These networks may be organized either with the support of a fixed infrastructure or in the form of an ad hoc network [Cordeiro2003]. Usually, these ad hoc networks are built upon the infrastructures provided by wireless LANs and PANs .The widespread and integrated use of wireless networks will increase the usefulness of new wireless applications, especially multimedia applications deployment such as video-on-demand, audio-on-demand, voice over IP, streaming media, interactive gaming and other applications.

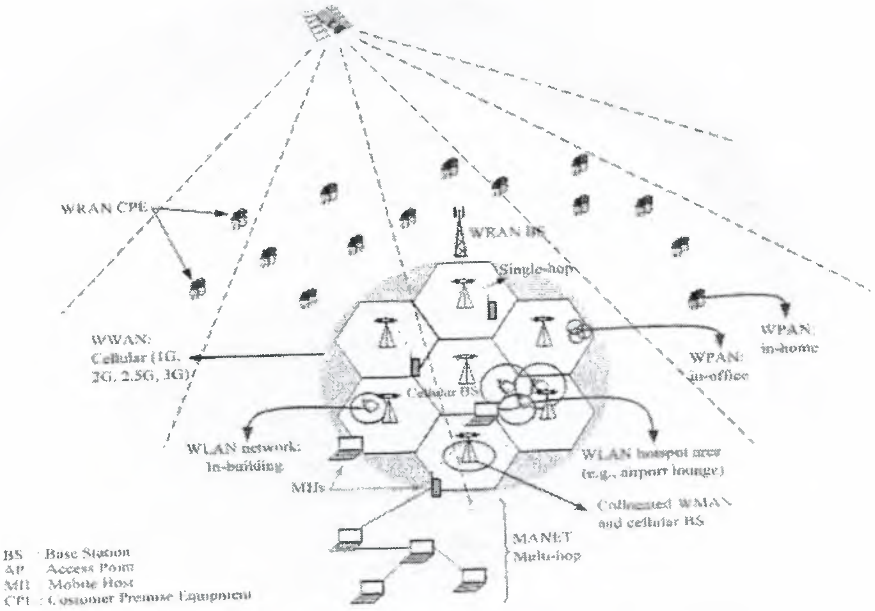


Figure 1.2: The envisioned communication puzzle of 4G and beyond

LANs and Wide Area Network (WANs) are the original flavors of network design. The concept of “area” made good sense in early days, because a key distinction between a LAN and a WAN involves the physical distance that the network spans. A LAN typically connects computers in a single building or campus, whereas a WAN generally covers large distances (states, countries, continents). As technology improved, new types of networks appeared on the scene. A third category, the Metropolitan Area Networks (MANs), also fits into this distance-based scheme as it covers towns and cities. A forth category, the Personal Area Network (PAN) has been designed to interact with personal objects. This category is specially designed for highly mobile device with an idea to share hardware and software resources. Recently, the latest major revolution is the Regional Area Network (RAN) [Cordeiro2005], which promises to provide coverage ranges in the order of tens of kilometers with applications in rural and remote areas. LAN, MAN and WAN were originally started as wired network, and due to increasing demand for wireless connectivity, these networks also gained attention in the wireless domain. PANs and RANs, on the other hand, have been introduced with wireless connectivity in mind. Figure 1.3 compares various wireless networks in terms of the popular standards, speeds, communication ranges and applications.

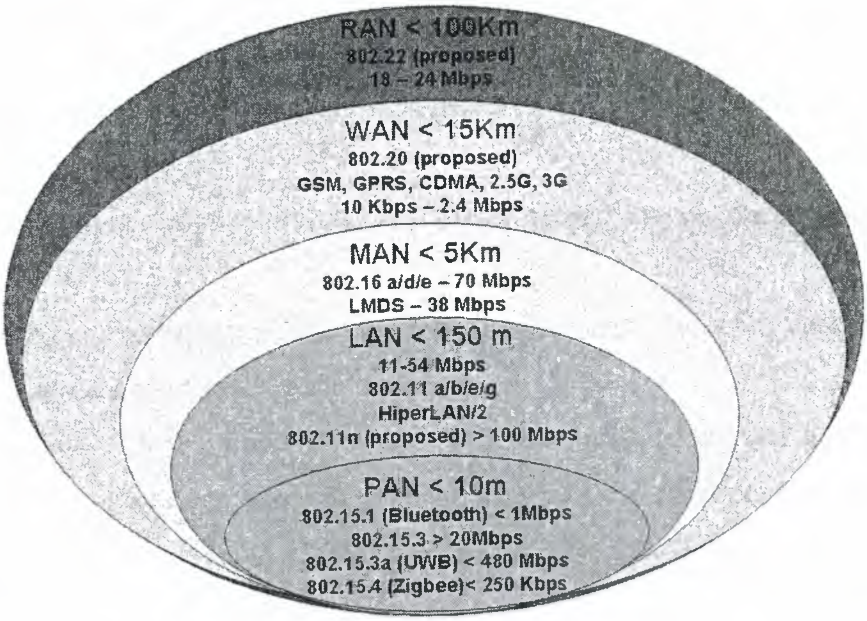


Figure 1.3: The scope of various wireless technologies

Since the infrastructure for building ad hoc networks are mostly within the framework of Wireless LANs and Wireless PANs, their scope given in Figure 1.3 is particularly useful. This is not to say, however, that the infrastructures provided by WMANs, Wireless WANs (WWANs), and WRANs, depicted in Figure 1.3, cannot interoperate with the ad hoc network. As a matter of fact, a lot of movement is currently undergoing as to integrate ad hoc networks with MANs and WWANs, where the infrastructure provided by these networks would serve as a backhaul to, say, connect the ad hoc network with the outside world (e.g., Internet). Furthermore, with the large scale appearance of dual mode and dual band radios where devices are equipped with multiple wireless interfaces or software defined radio [SDRFORUM] capability, heterogeneous networks will become more and more common and the need to integrate them will be of paramount importance.

1.6 Classification of Computer Networks

Network Classification Like snowflakes, no two networks are ever alike. So, it helps to classify them by some general characteristics for discussion. A given network can be characterized by its:

Size: The geographic size of the network

Security and Access: Who can access the network? How is access controlled?

Protocol: The rules of communication in use on it (ex. TCP/IP, NetBEUI, AppleTalk, etc.)

Hardware: The types of physical links and hardware that connect the network

Computer experts generally classify computer network into following categories:

1. Local Area Network (LAN):

LAN is network where various terminals, PCs, workstations, servers and other shared services providing devices (network printers, scanners, etc.) are interconnected within a short distance of one another. Other elements that could be added to the LAN infrastructure are the interconnecting devices:

- Hubs:

In data communication, a hub is a place of convergence where data arrives from one to other or more directions and is forwarded in one or more other directions. A HUB may include a switch of some kind. A product that is called "switch" could usually be considered a HUB as well. The distinction is that the HUB is the place where data comes together and the switch is what determines how and where data is forwarded.

- Switches:

A switch is a network device that selects the path or circuit for sending a unit of data to its next destination. A switch may also include the function of the router. In general, a switch is a simpler and faster mechanism than a router, which requires knowledge about the network and how to determine the route.

- LAN routers:

A router is a device or, in some cases, software in computer, that determines the next point to which a data should be forwarded toward its final destination. The router is connected to at least two networks and decides which way to send each data packet.

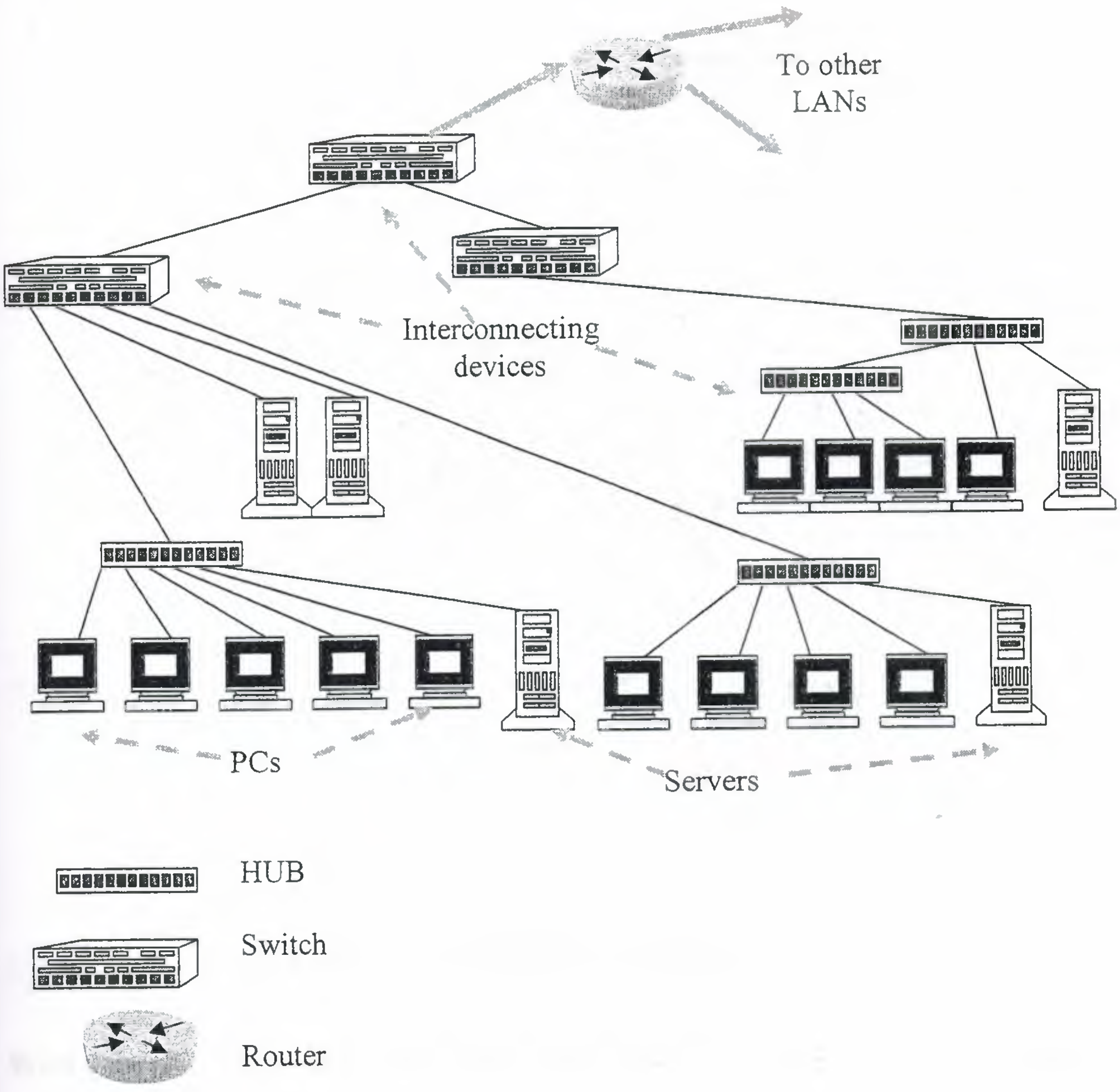


Figure 1.4: The example of Local Area Network

2. Wide Area Network (WAN):

Wide area networks have traditionally been considered to be those that cover a large geographical area. It means, WAN provides inter-city, national or international coverage.

Typically, a WAN consists of a number of interconnected switching nodes (routers), which connected together LANs

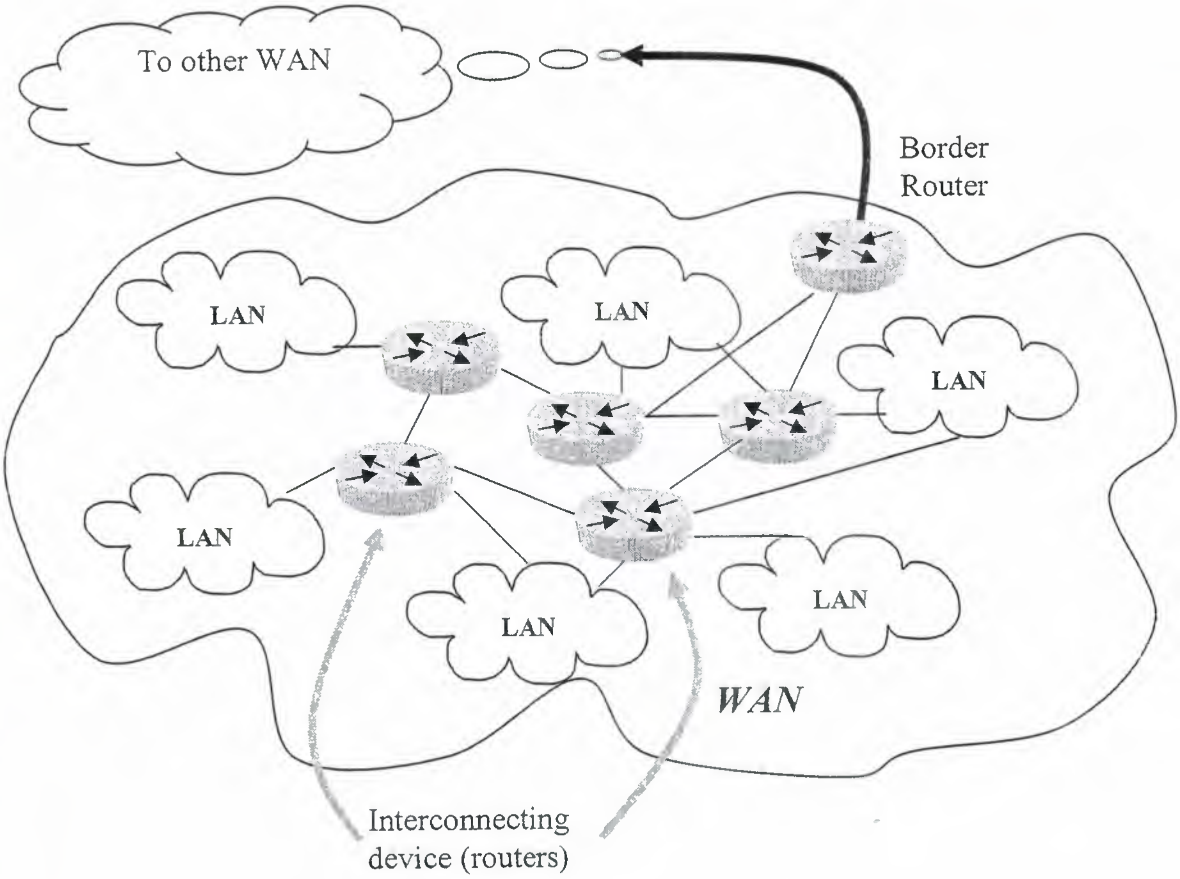


Figure 1.5: The example of Wide Area network

Wide area networks have a long history going back to the 1960s, as they were the first type of computer network in widespread use. The majority of data-carrying WANs have, for many years, used packet switching protocols, either using proprietary protocols e.g. IBM's system network architecture SNA, or based on the international standard packet switching protocol X.25.

A distinguishing feature of WANs is that the circuits are normally owned by a network carrier company and rented or leased to other organizations or residential users.

A WAN is formed from use of one or more of the basic communication options combined with either circuit switching or packet switching protocols:

- Private leased circuit provided by a public telephone operator. These services can offer either analogue or digital circuits.
- Public switched telephone networks, designed for telephone services but used extensively for data transmission also.

- Public data networks, usually based on packet-switching, for exclusive support of data transmissions.

3. Metropolitan Area Network (MAN):

A MAN (metropolitan area network) is a network that interconnects users with computer resources in a geographic area or region larger than that covered by even a large local area network (LAN) but smaller than the area covered by a wide area network (WAN). The term is applied to the interconnection of networks in a city into a single larger network (which may then also offer efficient connection to a wide area network).

4. Personal Area Networking (PAN):

A personal area network (PAN) is a short-range, localized network where nodes are usually associated
with a given person. These nodes could be attached to someone's
cell phone, pulse watch, belt, and so on. In these scenarios, mobility
is only a major consideration when interaction among several PANs
is necessary, illustrating the case where, for instance, people meet in real life. Bluetooth [Haarsten1998] is an example of a technology aimed at, among other things, supporting PANs by eliminating the need of wires between devices such as printers, cell phones, PDAs, laptop computers, headsets, and so on.

5. Campus Area Network (CAN):

The computer network within a limited geographic area is known as campus area network such as campus, military base etc.

6. Home Area Network (HAN):

A network contained within a user's home that connects a person's digital devices. It connects a person's digital devices, from multiple computers and their peripheral devices to telephones, VCRs, televisions, video games, home security systems, fax machines and other digital devices that are wired into the network.

In figure 1.6 the connecttivity of local area networks to metropolitan area networks and typical use of metropolitan area networks to provide shared access to a wide area network is shown.

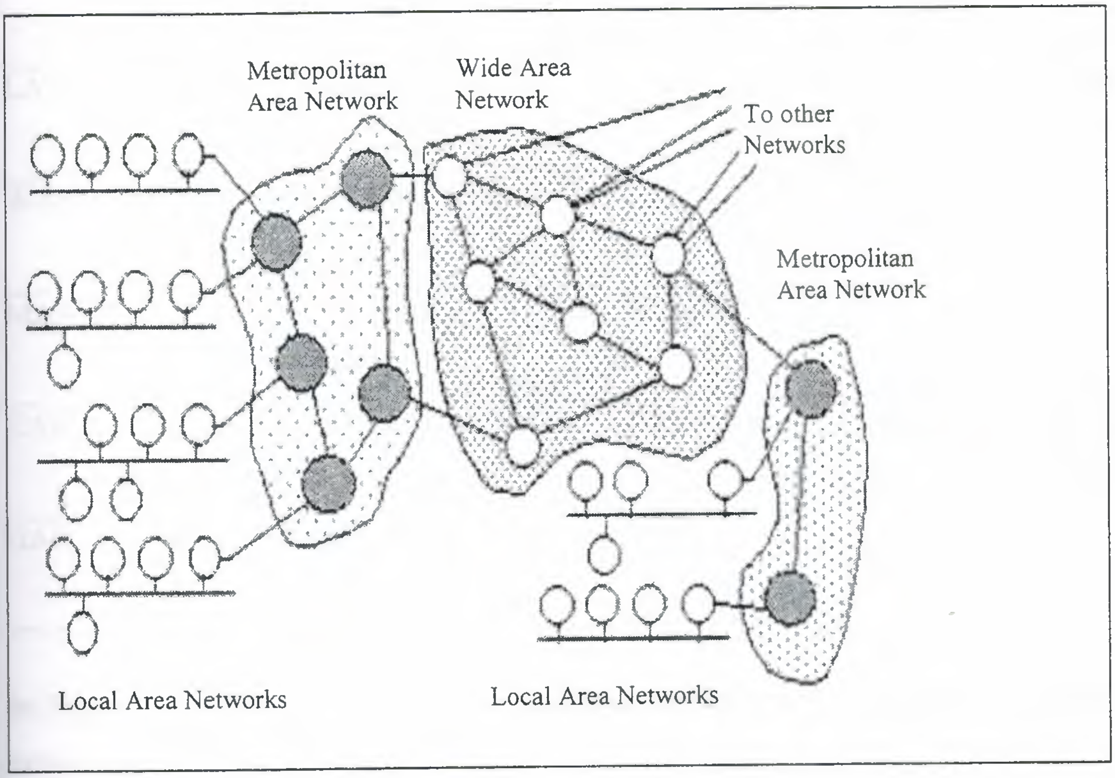


Figure 1.6: A typical use of MANs to provide shared access to a wide area network

Computer networks are used according to specified location and distance. In table 1.1 it is shown that which technology can be applied to the specific location and specific distance.

Table 1.1: Network Techonologies that Fit in Different Communication Spaces

NETWORK TYPE	DEFINITION	DISTANCE RANGE	COMMUNICATION SPACE
LAN	Local Area Network	0.1 to 1 Km	Building, floor, Room
WAN	Wide Area Network	100 to 10000+ Km	Region, Country
MAN	Metropolitan Area Network	10 to 100 Km	City
CAN	Campus Area Network	1 to 10 Km	Campus, Military base, Compnay site
HAN	Home Area Network	0.1 Km	Home

In Figure 1.7 a chart is shown which specifies the distances and speeds of different networks.

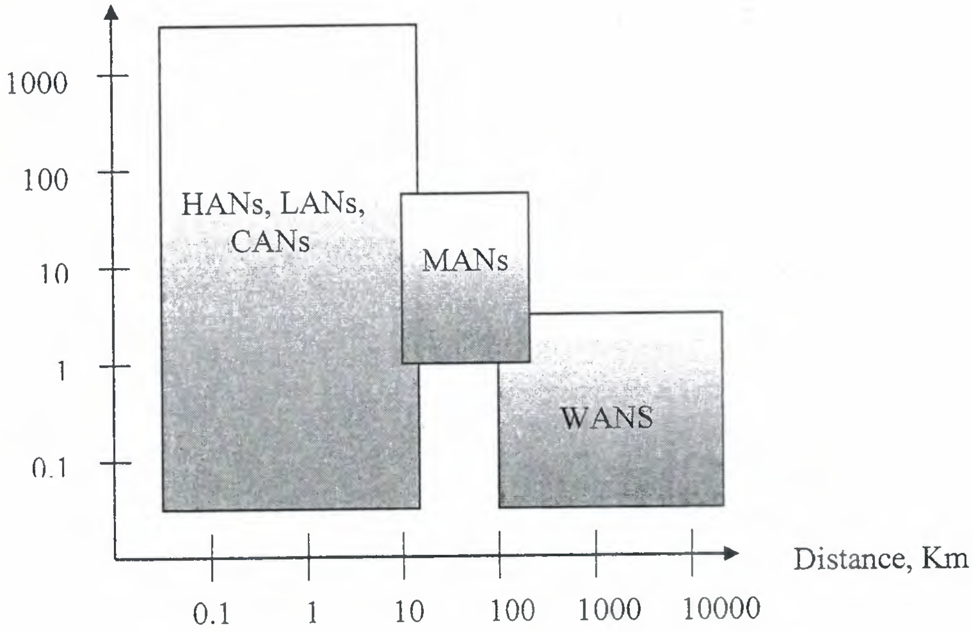


Figure 1.7: Distances and Speeds of the Different Networks

1.7 Summary

In this chapter we discussed what is a computer network and the history of it, we showed the goals of a computer network Also we explained the kind of networks which are used these days like Local Area Networks (LANs) which connect computers in a single building or campus, Wide Area Neteorks (WANs) which covers large distances, Metropolitan Area Networks (MANs) which fit into this distance-based scheme as it covers towns and cities and Personal Area Network (PAN) which interact with personal objects,and ofcourse CANs,HANs,RANs.

2. ACTIVATE NETWORK SECURITY

2.1 Overview

Active Network Security is comprised of a number of techniques that address this shortcoming. The goal is not only to reduce the number of successful abuses of a system, but also to give early warning of abuses in progress. Finally, the objective is to ensure that misuse of the system does not go unnoticed that, should all of the security mechanisms fail, a record exists to allow corrective action.

2.2 Active Security Mechanisms

Active network security, as described in this document, encompasses networking tools and systems that allow system administrators to observe, inspect and improve the security of their networks. Many conventional security mechanisms are effective in enforcing security in a system, but lack the responsiveness necessary to maintain security on an ongoing basis. In recent years, a number of security tools have been developed that may best be classified under this heading: while these tools often have no direct effect in preventing misuse, they allow administrators to improve the overall security of their systems. Examples include:

- Intrusion Detection Systems (IDS) Intrusion Detection Systems monitor the state of a system, attempting to recognize and report improper behavior. These systems protect a network in much the same way as security cameras protect buildings: by letting security personnel keep an eye on what is going on.
- Network Security Scanners Security scanning systems inspect a network or host system, looking for known weaknesses and possible misconfigurations. The best known example is probably the Satan system it scans hosts and connected networks for a specific series of weaknesses, reporting any found, and suggesting solutions.
- System Integrity Checkers Many of the ways in which systems are attacked involve changes to the host's software and data. Integrity checkers compare the contents of a system to a known safe state allowing administrators to know exactly what has been changed.
- Honeytrap systems If an IDS is a security camera, this is a burglar alarm; systems whose sole purpose is to be attacked. By closely monitoring these

systems, network administrators can observe attackers in action – allowing them to repair, learn and strengthen security against future attacks.

- Special purpose tools Specific tools have been developed to address security weaknesses present in systems. While not as generally applicable as those listed above, still deserve a place in every security administrator's toolkit. In Section 8, we will touch on two examples: password cracking systems and sniffer detector software. In a world where security mechanisms were infallible, none of these systems would be necessary. In fact, none of these systems can, in itself, prevent an attack from succeeding. The function of these tools is to minimize the effect of an attack, mitigate resulting damage, enhance the effectiveness of other mechanisms, and ensure that future similar attacks do not succeed.

2.3 The Limitations of Static Security

2.3.1 Authentication

The core of many current security mechanisms, authentication encompasses the technologies used to identify and verify the authenticity of users, network components and processes. This ranges from simple password based schemes through to biometric and cryptographic mechanisms. The ultimate goal is to uniquely associate an entity external to a system with an identity stored inside the system. In most systems, this is done by requesting some identifying information from a client, for example a password, biometric reading or response to some challenge. This information is then verified against information held inside the system. Should the identifier and stored information match, the user is authenticated; otherwise the user is denied. Extensions of this scheme include the addition of timing or locality information in the identification data, and encrypting the dialogue all aimed at making the synthesis of a counterfeit identification token more difficult.

2.3.2 Cryptography

With the recent increase in dependence on shared resources, especially public networks, the security of information in storage and transit has become a concern.

Strong authentication may prevent active use of restricted resources, but passive interception of information can be as great a risk. In addition, where information is held in an untrusted system, ensuring that data remains unchanged in transit is also a concern. Cryptographic techniques are becoming increasingly prevalent in resolving these issues: ensuring that only authorized users can interpret sensitive information (encryption); and ensuring that vulnerable information is communicated intact (authentication). Encryption is the process of applying a transformation to data that can only be reversed using secret information. Depending on the application, one of two forms of encryption may be used: secret-key

cryptography, where the transform and its reverse make use of the same secret, and public-key cryptography, where the encrypting transform does not require the use of secret information. Public key cryptography bears a close resemblance to the authentication problem: a user may be defined as anyone capable of reversing a given transform, thereby authenticating a communication partner. Cryptographic authentication involves the derivation of a message signature from a message, based on the use of secure hashing techniques. Should the message be modified in transit, the signature and resulting message will no longer match. In order to ensure that the message signature is not modified, encryption techniques are used (restricting the set of users capable of generating a message to those sharing a specific secret). In the case of a modified message, it is infeasible to generate a new encrypted signature that would decrypt to validate that modification. Therefore, if the signature matches the message, it is unlikely that the message was changed or counterfeited.

2.3.3 Access Control

Authentication verifies the internal identity of external parties. Access controls define which resources those parties have access to – limiting the capabilities of those users. These controls are no stronger than the authentication mechanism underlying them, and have potential weaknesses independently of authentication failure.

2.3.4 Firewalls

While firewalls could be considered a specific application of the mechanisms

described

above, they form one of the main pillars of current network security, and merit separate consideration. The function of a firewall is to separate networks with different security needs and policies in the most general case, to separate the internal, controlled network and any external public networks. Effectively, a firewall acts as a filter on network traffic controlling what goes into or comes out of a network.

2.4 What Do Static Methods Offer

The static methods described here, perfectly applied, are effective in ensuring the security of any network. Even in realistic environments, static security mechanisms are capable of significantly improving the security of networked resources.

- Static mechanisms can increase the security of networks in the context where they apply.
- These mechanisms can increase the technical expertise and resources required to compromise the security of a network.
- Static methods can reduce the range of attacks that Active Security mechanisms must deal with.
- Static methods can combine with Active methods to provide a synergetic improvement in security.
- Static methods can prevent attacks from succeeding.

2.5 The limitations of Static Security

In spite of the wide variety of security mechanisms available, intrusions continue to occur. Based on this fact, a number of limitations in static security mechanisms can be identified:

- The protection offered by these mechanisms is limited in scope. While these mechanisms may be effective in the context in which they are applied, they do not offer universal protection. For example, firewalls, while being effective against external attack, offer no protection against internal abuse which, as shown in a previous section, is a significant risk factor. The same type of argument applies to other mechanisms: authentication is vulnerable to trust networks, where the authentication mechanisms are bypassed. Encryption only protects information while in an encrypted form. All of the current static mechanisms can be bypassed, negating their effect.

- The security mechanisms themselves are sensitive to technical and implementation problems. Such systems can become vulnerable due to theoretical advances (such as the DES encryption standard, which can no longer be considered completely secure), or poor implementation (for example Microsoft PPTP).
- Even if theoretically sound and correctly implemented, security mechanisms must be correctly applied in order to be effective. Describes an organization that had its web server defaced while their firewall was hidden deep inside their network, acting as a log server. Many of the security mechanisms available are very complex (both in structure and in application), and a single mistake may be enough to nullify the efficacy of the system. An example of this is the use of dial in lines allowing direct access to a trusted network. No matter how good the firewall blocking official connections to that network is, it is still vulnerable.
- Static security mechanisms, by their very nature, are prone to silent failure. Often, the first sign that your security has failed comes when it is far too late (such as when an entire server is wiped clean an effective method for an intruder to erase a history of his actions). Even when a system's security has not yet been penetrated, that may lead to a mistaken sense of security. In general, these mechanisms also cannot recognize when they are under attack – at best, an attack is logged as a series of failed transactions.
- Associated with the previous point is the issue of remedial information. Once a failure is identified, it may be difficult or impossible to trace the cause of that failure. Information on the identity and methods of an intruder may allow the effects of an intrusion to be mitigated but none of the mechanisms described offer any such capabilities inherently. The audit information collected by some tools, while being useable, does not have sufficient detail.
- Finally, the security mechanisms can themselves be subject to attack. Authentication servers can be corrupted, firewalls crashed or circumvented, and cryptographic distribution channels can be compromised. In many cases it is a simple exercise to disable system by attacking its underlying infrastructure. A

good illustration of this is the number of tools that are freely available, aimed at allowing users to circumvent the restrictions applied by security mechanisms anonymous proxies, network tunneling applications and the like.

The essential problem with many of the mechanisms listed above is that they are essentially passive. While this may be sufficient for a degree of security, it does not hold up in the imperfect world of modern networks, where network administrators are often over-worked, do not have the necessary specialized skills, and where the attacks on networks are ever-escalating in complexity and intensity.

2.5.1 Sources of Attack

(a) Script Kiddies

This is the name given to the masses of relatively unskilled hackers that use the tools written by others, without necessarily having any real skill. They are typified by having endless time to spend probing networks for victims to their latest exploit tool9 – it is on these that the common perception of hackers is based. This is not to say that they do not pose a risk, however – far from it. These hackers often have an array of tools available, and keep up to date with the latest new exploit software that becomes available. In addition, since they often have no specific aims in mind (beyond the trophy of having hacked a system), they will not necessarily target the most visible or valuable machines – obscurity is no deface.

(b) Employees

Possibly the most dangerous group of potential attackers are the very people who use the

networks every day the staff. They know what in a network is of value, what defences are in place, and have a ready foothold from which to escalate their control. It is a telling statistic that, in the CSI/FBI survey, 86% of respondents consider disgruntled employees as a likely source of attack (compared with 74% for independent hackers). Also, recall that 55% of respondents reported inside abuse of their networks.

(c) Mistakes

Not all anomalies in your network have hostile intent. Many "attacks" might be result from a lack of user expertise or from simple user error. This is does not imply that such errors are not dangerous: the case of the 1980 ARPAnet collapse is a clear example of how devastating a simple mistake can be.

(d) Automated Agents

This category includes such things as worms (such as the infamous 1988 Internet Worm), automated hacking tools, viruses, and Trojan software. There does not need to be a human active in order to attack systems a good example of this is the recent Melissa

macro virus. With minimal modification, the Melissa virus would be capable of sending whatever document is being worked on to email address effectively leaking information.

(e) Expert Hackers

A number of expert hacker groups have been in the media over the past few years as government witnesses, software developers, and network security experts . These groups do not merely use exploits written by others; they produce tools of their own12. They constitute the highest skill level that network security will be faced with; an administrator can expect to see completely new attacks, if any signs remain at all. The reason behind a given attack may differ wildly: recreation, industrial espionage, fraud, and attempts by foreign governments to destabilize national infrastructure have all been proposed as causes for intrusions. To place this discussion into context, consider some specific reports:

- However, the hackers of the cases on which this paper is based are known. All of them were male, and computer science students doing their master's. They all had access to the Internet, and were reasonable well acquainted with UNIX. All of the hackers, except one, had the level of an ordinary UNIX programmer with a little bit more understanding of network software.
- A sixteen year-old from the U.K. entered a plea bargain and paid a $1900 fine while another twenty-two year old pled not guilty and was acquitted on all charges in February 1998. The 16 year old was operating on a home computer in his parents' house and had a "C" grade average in his high-school computer class.

- The attackers were two teenagers from California and one teenager from Israel. Their motivations were ego, power, and the challenge of hacking into U.S. DoD computer systems.

It would appear as if the common preconception of hackers being young, male and bored holds. However, real information is scarce though a question would be whether experienced hackers get caught.

2.5.2 Outline of an Attack

The process involved in gaining control of a system generally follows a number of discrete stages, outlined below. One of the aspects that make internal abuse so dangerous is that the attacker can often bypass the early (and from an intruder's point of view, dangerous) stages, and proceed directly to escalating their control over a system13.

2.5.2.1 Exploring the Target

The first step in any intrusion is generally to build up an image of what potential targets a network contains. A number of different techniques are available to hackers, including:

- **Network Scanners**

These tools send specially constructed packets to addresses in the range being scanned. Based on the nature of the reply, it can be deduced which addresses correspond to active machines, and often even more information can be extracted: the operating system running on such systems, open ports, and the presence of intermediary network filters (such as firewalls). Detecting such sweeps has, in the past, been relatively simple: they generate a large number of similar events in system logs, within a short period of time. Increasingly, however, more complex tools are becoming effective in obscuring the details of such scans. Tools exist that allow scans to be conducted slowly, using only a few packets per hour or day or conduct a scan co-operatively from different source addresses. One common tool, allows the source of a scan to be masked by generating a number of fake scans (from spoofed addresses), and has a number of stealth scan mechanisms. One of these, a TCP ACK scan, has been found to be effective in penetrating our testbed firewall.

- **DNS Zone Transfer**

  By retrieving all information available for a network from the DNS hierarchy, an attacker can retrieve a list of all externally accessible points for that network. In addition, if the internal DNS servers are accessible externally, an attacker has access to a wealth of information: a map of the host names and addresses of all machines on the network, and possibly even account details for the system maintainer.

- **Tracing the system neighborhood**

  Using the DNS and addressing information and tools such as trace route, an attacker can determine what machines are in a network neighborhood. Compromising a machine on the external path of a target network, a number of attack forms become available – ranging from simple traffic snooping to TCP session hijacking. Compromising a machine that the target network depends on, such as a DNS cache server, similarly opens the door for attacks on the target network – and that machine may be significantly less secure than the protected network.

- **Public Information**

  The information on an organisation's external presence can offer a significant amount of information. From the services and formats offered, an attacker can deduce which operating system may be in use, and identify possible weaknesses. From URLs and email addresses, an attacker can deduce machine names, accounts that may have administrative privileges, and naming schemes used. Based on the header information on emails and HTTP requests from a site, an attacker can extract the operating systems used, and a wealth of information on the SMTP structure of a network. In addition, some sites offer details on the systems they run on their web sites greatly simplifying this step for an attacker.

- **Predictable names**

  Host and service names are often chosen to maximize their convenience: using sequenced host names, naming themes, NIS domain names that correspond to

Internet domain names, predictable account names and details (e.g. root), and IP allocations based on the service hosted. Any such features allow attackers to make intelligent guesses as to network structures. Once an attacker has a map of a target network, an attack may not be immediately forthcoming: such network maps are often stored, distributed, and used at a later stage.

2.5.2.2 Vulnerability Identification

The second step in preparing for an attack consists of determining which of the machines located in the initial exploration may have exploitable vulnerabilities. These often take the form of wide sweeps, looking for machines vulnerable to a given attack often using an exploit script just released15. An alternative mechanism is to match the network information from Step 1 against the set of available exploits – picking viable attacks for a specific network. Favorite targets for these sweeps are the external and support services offered by a network: FTP, DNS, SMTP and HTTP servers. Recognizing these sweeps can be simple, using local knowledge of a network: repeated probes on port 143 (IMAP) (for example), on machines not running mail software is reason for suspicion16.

2.5.2.3 Penetration

The goal of this step is to gain an executing process on the target system. A vast number of exploits are known (with more being discovered every month) allowing an unauthorized user to gain a foothold on the victim host. Examples include server buffer overflows, system backdoors and weak authentication or access control mechanisms discusses some specific examples of well known attack techniques. It is this phase that IDS attempts to recognize – therefore it is also at this point that monitoring systems are likely to be attacked. Using denial of service (DoS) attack, or customized exploits, an attacker may attempt to disable the security mechanisms in a network. Alternatively, an attacker would use his knowledge of the organisation's traffic patterns to hide the attacking traffic in normal traffic streams – making filtering and detection more difficult. For example, a CGI exploit disguised as a normal HTTP request is likely to bypass any filtering mechanisms in place (as demonstrated in the firewall experiments).

2.5.2.4 Escalation

Once an attacker has a foothold on a system, the next step is to escalate to control over

the system. In this step, the goal is to gain sufficient administrative privileges to allow the next step, Embedding, to proceed – or to do damage. This often takes the form of a bootstrapping process: initially, the attacker starts with minimal privileges. Then, using a succession of exploits and attacks, an attacker gains successively greater privileges until he has complete control over the system. Alternatively, this could be bound to the Penetration step: many services run with extensive privileges, and grant an attacker those privileges when compromised (effectively allowing an attacker to bypass this step which is why most services run with as few privileges as possible).

2.5.2.5 Embedding

Having gained control of a system, an attacker will cement his control over a system, so that later intrusions do not require the dangerous Penetration and Escalation steps to be repeated. This step involves removing all records of the initial intrusion, bypassing or disabling the reporting mechanisms, and building access routes that will allow the attacker to resume control of the compromised system at a later time. This ensures that the attack and access routes are not detected ensuring that backdoors remain accessible.

Examples of embedding techniques include: modifying access control files to allow the attacker access (e.g. adding accounts to a system); modifying access control mechanisms so that they do not apply to the attacker (e.g. adding a master password to the login program). Another mechanism is to place tools that allow rapid escalation into low-privilege accounts (and ensuring that those remain accessible) these may be harder to detect. An example of this method is the placement of SUID-root command shells (under UNIX) – allowing the user to instantly gain complete control over a system. A final mechanism is placing a server process on the machine that will accept commands from the attacker.

2.5.2.6 Extraction

At this point, the attacker has effectively gained complete control over the system. In many cases it is at this point that an attacker would extract information from the system, or attack the information held on the system (such as vandalizing a web site hosted from a compromised server). Security systems such as firewalls may no longer hinder an attacker many techniques exist for communicating invisibly through filtering systems.

2.5.2.7 Relay

Once an attacker has completed modifying or extracting information from a system, he will often retain that system for use as a springboard for further attacks. Tracing an attacker backward through the complex interconnected networks available is a very difficult attacker makes use of multiple systems to obscure the true source of attack. In addition, tools are emerging that allow distributed attack and scanning of systems – not only obscuring the attacker, but making the attacks harder to detect and counter. An emerging trend is for attackers to target home machines permanently connected to the Internet. Such machines often have very low security, and are ideal as staging areas for further attacks. Who would be liable for damage done from such a compromised machine is unclear what is clear is that systems need protection, whether or not they contain critical resources.

2.6 Typical Attack Techniques

- **Scanning a network**. The first step in an attack is reconnaissance – finding out as much as possible about the target. Many tools are available for investigating a network – ranging from simple scripts to commercial network mapping tools, to dedicated scanning applications19. In essence, these tools send a packets to a potential host, and deduce information about that host from any reply. Mapping a network consists of checking every possible address for that host. In particular, a number of scan types can be distinguished.

- **Ping scan**: The simplest form of scan, an attacker sends an ICMP echo request packet to every candidate machine (which is the same way the ping tool works). Any addresses that respond are noted as active.
- **TCP Connect() scan**: Another simple scan, an attacker attempts to open a standard TCP connection to a typical port on the candidate machine (such as the HTTP port 80). Any machine where such a connection succeeds is noted as active. Since many systems log any connection attempts, this type of scan is relatively easy to recognize from standard audit data.
- **TCP SYN (Stealth) scan**: This scan sends a connect request to every candidate machine (similar to the Connect () scan), but does not complete the connection by sending a final SYN/ACK packet. In this way, the connection fails and does not generally show up in the system logs – hence a "stealth" scan. Since this scan has a similar signature to a SYN flood attack, many security systems now log such occurrences.
- **Stealth FIN, Xmas, ACK and NULL scans**: These scans all form part of the same family of variations on the SYN scan techniques. Each sends a special packet to a candidate address, deducing whether a port is open or not from RST reply packets (which indicate a closed port). If not reply is received the port is open – or the request lost in transit, such as being discarded by a firewall. FIN scans consists of packets with the FIN flag set, Xmas scans of packets with the FIN, URG and PUSH flags set, and NULL scans of packets with no set flags. The ACK scan consists of packets with the ACK flag set (generally denoting replies), and so are often capable of penetrating firewalls.
- **UDP scans**: This scan consists of sending UDP packets to likely ports on candidate machines at worst, scanning for any open UDP ports. Since UDP is connectionless, such attempts are harder to control using filtering firewalls, and may be capable of finding unprotected services and hosts. Many variations on these scanning techniques exists – including scans using fragmented packets, and scans spread across a long period or a number of source machines. In practice, completely blocking scans is probably infeasible – but may give an administrator early warning of an impending attack.

- **Buffer Overflows**. This is actually rich category of specific attacks, all using similar weaknesses in software. The core of the attack is to pass an unusually structured (often very long) value as a parameter to a system, when it is expecting something else – for example, requesting an FTP server to change the working directory to an extremely long filename. What happens, in general, is that the parameter overflows its storage buffer, overwriting commands that would later be executed – allowing an attacker to have arbitrary commands executed by the remote server. These commands can then be used to do any number of things – typically, creating an interactive shell, modifying access restrictions, or retrieving sensitive information, such as a password list. for details on this technique.
- **Open doors and abused trust**. In order to simplify authentication and access control, many systems accept assertions made by trusted systems. For example, the rsh series of commands accepts the remote machine's claims to user identity, if the remote machine is authorized to make such claims. This allows a number of attack techniques, based around abusing the assumptions made in such systems. One technique involves an attacker assuming the identity of a trusted machine, allowing it access to the trusting system. Another is based on the fact that under some systems (such as some Unix variants), users can control which other machines are trusted (using the .rhosts file). A common escalation step in attacking such a host is to modify this file, to allow the attacker free access. For an example of the process involved.
- **Social Engineering**. This type of attack is one of the oldest and most effective way of bypassing security mechanisms: fool somebody with the ability to do it for you. Variations range from guessing information based on the attacker's knowledge of the target involved, to impersonating personnel, and more. The only way to protect an organization is to ensure that it has a sufficiently clear security policy, and that its users are educated – no technical measures can prevent this type of attack. For a good example of how effective this can be.
- **Application Attacks.** These attacks depend on convincing an application to do something it was not expected to – overwrite files, execute commands it should not, or give away information that should be hidden. In addition, these attacks are notable since they can often penetrate even the best developed security

mechanisms – the only defence is to keep the applications themselves secure. Examples include requesting password files via FTP or HTTP, attempting to overwrite sensitive files via the same, or passing unexpected information to server applications – such as any of the range of CGI exploits available. For a good example of how this type of attack proceeds.

- **Trojan software**. The problem of computer viruses is well-known; but the techniques used for propagating these programs can also be used to compromise security. A good example is the Back Orifice system – once an infected application is run on a system, it installs a backdoor on the system, allowing the attacker free access. Preventing this type of attack is difficult – it requires user education, and security to be deeply embedded into systems.

2.7 Policy Issues for Active Security

2.7.1 What is Security Policy?

An organization's Security Policy defines and outlines the measures present to ensure that the confidentiality, integrity and availability of systems remain intact20. This includes such items as:

- **System review**: What systems are in place and in need of protection.
- **Risk assessment**: What the risk factors affecting such systems are, and how vulnerable the organization is to harm should one of these risks be realized.
- **General intent**: How the policy is to be interpreted, and how to resolve issues not directly covered in the policy.
- **Measure selection**: A listing of what measures are in place, describing their placement, configuration, and operational parameters.
- **Operational protocols**: What steps are to be taken under specific circumstances, such as system update protocols and change management, intrusion response and general operations.
- **Responsibility allocation and authority**: Who is responsible for specific actions or parts of the systems, and what authority they bear.

- **Security policy information**: When and how the policy is reviewed, where it is kept, and what authority underwrites it.

In effect, the security policy of an organization circumscribes the measures taken by an organization to ensure that computing systems are protected under operational and adverse circumstances. Two main techniques are generally used to ensure that resources are adequately protected: baseline protection and customized protection.

Baseline protection implies the application of security mechanisms across the entirety of a system or subsystem, without regard for the specific needs of components. This requires minimal risk assessment, and may offer acceptable security in low-risk environments, but generally will not offer the most cost-effective protection or adequately protect sensitive systems. In addition, certain safeguards may actually reduce the security of a system (in terms of the critical factors mentioned above). For example, encryption improves the confidentiality of systems, but decreases availability. Therefore, for systems where high availability supersedes confidentiality (e.g. internal email systems), the use of this mechanism reduces overall security. Customized protection is the application of security mechanisms based on a detailed risk assessment, in order to address the particular needs of a system. This ensures the most efficient allocation of resources, and avoids the problem of inappropriate security measures, but requires a more complex assessment of the needs of an organization. In addition, an incomplete assessment would result in a mismatch between the actual and estimated needs of a system, creating gaps in the security present. A method that is often used is to combine the techniques described above: using baseline security to increase overall protection, and protecting critical or sensitive systems with custom measures. This offers many of the advantages of both worlds: a common base of protection system-wide, sufficient protection for vulnerable systems, protection against changes in risk patterns, and simplified administration. Intrusion Detection and Active Security mechanisms lend themselves to both baseline and customized security. Applying these measures system-wide allows the system to be protected against general misuse, but may require significant resources. By optimizing the placement and configuration of these tools, it is possible to offer both increased protection for sensitive systems, and more context-sensitive detection, at the cost of general protection. For example, IDS deployment often concentrates monitors in high-risk areas, such as network ingress points (e.g.

adjacent to firewalls), or in the presence of valuable resources (such as network server farms).

2.7.2 The Relationship between Active Security and Security Policy

The Active Security tools discussed in this document are capable of being used as part of a baseline security strategy. This is also effectively what an organization defaults to, when no formal Security Policy is set out. In order to be used to greatest effect, however, these tools need to be deployed and configured with knowledge of the needs and behavior of the specific systems involved. As an illustration, IDS can function on any network or host system, attempting to recognize generally known abusive behavior (such as invalid network traffic). Such a system will not be capable of recognizing misuse, where such misuse does not correspond to anomalous or illegal activity. For example, such an IDS would offer no protection against users attempting to access resources in an inappropriate manner: for example, Joe from Sales attempting to read the personnel database (using a syntactically legal query). Embedding information from the security policy into such tools can greatly improve their efficacy. To extend the example, if it is known that certain actions are precluded by the security policy, the IDS and other tools could be configured to include this information. Knowing that nobody outside the personnel department can access that database, an IDS could easily detect Joe's attempt. The IDS can report the problem to security personnel – whether this is a case of internal abuse, or Joe's identity has been compromised and abused. In addition, Active Security tools can only function correctly if they are constantly maintained and monitored. As such, they depend on a security policy that defines how, and by whom, they are to be cared for these tools rapidly lose their function if they are ignored. As described more fully in the next section, the reporting capabilities of these tools also imply the need for policies to be set out, in order to handle the changing system. The security policy may also develop from the results gained from Active Security measures. These tools offer rich detail on the security state of a system: which areas are weak, which areas are being

attacked, and the general behavior of a system. This allows the system administration to extract system-specific information on the real security needs of the system, and modify the security policy accordingly. The information gained from these tools can show not only security problems – but also performance, management and configuration problems, and may give early warning of system failures.

2.8 Tools Supporting Active Security

2.8.1 Network Mappers

A variety of commercial and free network discovery tools are currently available examples, These tools use many of the same techniques described in section 3 to explore the content of networks: DNS zone transfers, scanning the address and port space, requesting information from hosts found, and promiscuous monitoring of a network. In fact, many of these tools are now used by attackers nmap, for example, was an invaluable aid in inspecting the exact coverage of the firewall policy during our experiments.

As an example of how a typical network mapper works, consider the nmap tool. It is a powerful aid in exploring networks not only because it offers a wide variety of scanning options but also due to its unique ability to identify a wide variety of hosts systems, down to the operating system, and sometimes version. Nmap works by sending packets with a wide variety of special characteristics to hosts being investigated: packets with specific (often illegal) flags set, ICMP echo packets, fragmented packets (again, sometimes with illegal fragmentation), etc. Every host has a particular style of responding to such packets – by combining these response characteristics, it is possible to narrow down exactly what system is present on the interrogated host. In fact, nmap uses a signature analysis system which bears some similarity to that used by IDS systems to recognise specific attacks – allowing the tools to easily extend its library of recognized systems. For example, it is possible to recognize Linux systems with older kernels than version 2.0.35 by the fact that, presented with a packet with the SYN flag and an illegal flag set, these systems retain the illegal flag in their response. Scanning a network generates a mass of highly anomalous packets alerting any good IDS tools present – and may have unwanted side effects. Because of the use of unusual traffic patterns, these tools are capable of damaging a network system certain types of fragmentation patterns.

2.8.2 Network Security Scanners

Configuring networks and network hosts to be secure is a difficult task: validating that such a system is secure may be even more difficult. A single security weakness in a configuration is all an attacker needs: a single weak password, a single outdated server, or a single vulnerable port. Network mapping tools go some way towards allowing an administrator to verify systems. Network security scanners (also known as vulnerability assessment tools) take this a step further – they actively test the security of a system against a number of attack scenarios, reporting on the location, severity, and solution to weaknesses found. These tools have had a contentious history – from the early COPS system, to the controversial Satan tool, to the current range of freely available toolkits, such as Nessus, Internet Security Scanner and Cybercop Scanner. Because these tools are capable of automating the vulnerability identification phase of an attack, it was felt by some that releasing such tools encourage script kiddies to attack systems. In practice, similar tools are available in the hacker community scan being a good example. Like IDS systems, these tools come in two varieties: host-based and network-based systems. Hostbased systems (such as COPS) analyse the security mechanisms in place on a system, looking for possible misconfigurations or dangerous settings. Examples include accounts with weak passwords, excessively trusting systems, and applications with unusual privileges (which may simply be a misconfiguration, or may be indicative of a past intrusion). This review is generally extremely system specific, but allows a wide range of issues to be checked across many user accounts a potentially significant saving for overworked administrators.

The second class, that of network-based systems, check hosts for secure networking policies. Tests include weak passwords for well-known accounts, the presence of services known to be dangerous (e.g. NFS available from outside a firewall), and unnecessary services (e.g. NFS without shared file systems). In addition, these tools include libraries of exploits, which are tested against subject systems checking whether such systems are susceptible to the specific weaknesses. In effect, the tool attempts to break into the subject system – if it succeeds, there is clearly a security flaw.

Finally, network-based systems are presently developing mechanisms for reviewing other security systems, such as IDS and firewalls. In particular, these systems can simulate the techniques used by attackers, allowing an administrator to verify that these are blocked or detected by the firewall or IDS, as appropriate. One issue with such systems that is sometimes overlooked is that these systems must be kept up to date constantly ensuring that a network is secure against last year's attacks does not offer any benefit against current risks. As the attack techniques used against systems evolve, these systems should be updated, and the systems re-inspected.

2.8.3 System Integrity Checkers

Once a system is compromised, one of the first actions taken by an intruder involves changing system files: to disguise the intrusion, facilitate future penetrations, or support escalation in control over the system. In addition, there is a variety of events that will result in unauthorized changes to system files ranging from viral infection, unauthorized changes by administrative personnel, or failing hardware. A tool developed to address this problem is the well-known Tripwire package. It has since become a standard component in many system administrators' toolkits. In essence, the Tripwire system stores a hashed snapshot of file system features and content, compares this to the current system state, and reports any discrepancies.

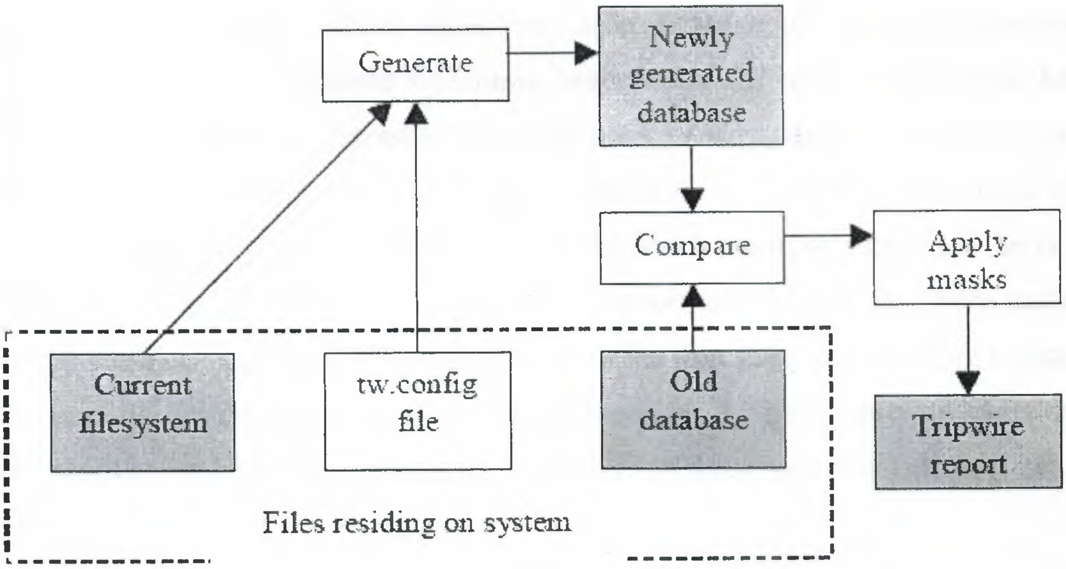


Figure 2.1: Structure of the Tripwire System

As can be seen from the above diagram, a Tripwire configuration consists of two main components: the Tripwire configuration files, and a previously generated reference database for that system. The configuration files consist of a series of file or directory paths and attribute masks (defining which attributes of a file may safely be ignored), or of M4-style preprocessing commands (similar to those used by the cpp C-language preprocessor). Using these features, it is possible to create fine grained configurations with support for host-specific variations.

The reference database is generated by Tripwire, based on some initial trusted file system. It is important to ensure that this initial generation is done on an uncompromised system ideally, this should be created for a system after the initial configuration, but before that system is taken into use. Tripwire cannot detect preexisting problems – only changes that occur after its installation. The security of the Tripwire system is based on a number of factors: the integrity of the Tripwire software itself, the integrity of the reference database, and the strength of the hashing algorithms used to identify files. Therefore, it is suggested that the reference database be stored in a secure location: on a different, secure system, or on read-only media.

To minimize the chance of an attacker making undetectable modifications to files, Tripwire supports the use of up to 10 different, simultaneous hashing algorithms (by default: MD5, MD4, MD2, Snefru, SHA, POSIX 1003.2 CRC-32 and CCITT CRC-16 signatures are available). These algorithms offer a range of security/performance features and the use of multiple signatures increase the difficulty of generating hash collisions greatly. From an Intrusion Detection point of view, this type of tool is most useful as a last line of defence, and for recovering from an intrusion. These tools will only report changes already present in a system at which point the attack may be in an advanced stage. In addition, these tools will only report that changes have been made – not what those changes were. For example, one of the first steps in controlling a system is to purge the system logs of evidence of the intrusion. While integrity checkers may detect that the logs have been modified, the nature of those modifications may not be evident.

System integrity checkers offer a strong deterrent, and can be of inestimable value in mitigating the effects of an intrusion, but they are best suited as a last line of defence. Once an intrusion has progressed to the point where system files are compromised, much of the potential damage could already have occurred – particularly where a loss of confidentiality is concerned.

2.8.4 Password Crackers

Many modern security systems have moved away from the user password authentication scheme, using biometric identities, cryptographic schemes, one-time passwords, and the like. For the large group remaining, however, weak passwords remains a significant problem. Password crackers are tools that attempt, through a combination of social engineering and brute force, to guess the password associated with a resource [Muffett92]. These tools are well-known as a major risk in the Unix world [Farmer93], but have recently found their way into many other systems in fact, a password cracker is now available for virtually every system using key-phrase based protection, such system authentication and file encryption.

In addition, the computing power available to crackers is increasing the level of complexity needed for secure passwords current recommendations in length, and changed every 3-6 months. However, even with these recommendations in place, human nature tends to use the simplest solutions – hence the problem of weak passwords. From a security point of view, password crackers allow an administrator to identify and address weak passwords before they become a problem. On the counter side, attackers also find such tools invaluable in gaining access to systems. Running periodic checks on passwords used, especially for sensitive accounts, can make a system more secure – but is not a replacement for user education, and stronger authentication mechanisms.

2.8.5 Sniffer Detection

Many of the current network protocols were designed to function in a trusted environment. Protocols such as Telnet, HTTP, FTP, and many others carry sensitive information in clear format any person observing the network traffic can extract such information, a typical example being username password pairs on a network login. Attackers are well aware of this fact, and often place network monitoring tools, or sniffers, on compromised hosts. The traffic captured on such hosts can then be used to compromise better protected hosts, or gather sensitive information. Since there is often no need for special equipment for this monitoring, it can be very difficult to identify which hosts may be observing confidential exchanges. In response to this problem, a series of tools have been developed so called sniffer. These tools use a number of techniques to attempt to isolate eavesdroppers:

- **MAC / Protocol addressing mismatches**: Many network protocol stacks do not verify that messages received were actually sent to their addresses – they rely on lower levels in the stack for that. A machine in promiscuous mode may therefore respond to requests sent out to its correct protocol address – even if the lower level MAC address was incorrect. A machine will then only respond to such requests if the MAC address filtering is not active – or it is in promiscuous mode. Examples of such requests include ICMP Ping, UDP or TCP echo (or other ports that always respond), or requests that generate error replies. The core of the method is attempting to fool a host into replying to a request it should not have been capable of seeing.
- **DNS Test**: Many machines automatically do reverse DNS lookups on IP addresses not yet mapped. Therefore, by sending messages to fictitious hosts, and monitoring reverse lookups, a sniffer detector can recognise machines monitoring traffic.
- **Decoy method**: Attackers will often sniff networks looking for such items as remote login sessions in the setup phase of protocols such as FTP, Telnet, or POP. By generating false login transactions, and monitoring for attempts to make use of that information, it is possible not only to identify the presence of monitors, but also to verify that these are being used to attack a network.
- **Latency tests**: In most modern networks, the resolution of MAC addresses is handled by hardware on the network interface. Therefore, the workload of a machine on a heavily loaded network segment will depend only on the traffic destined for that machine – unless it is observing all traffic. By comparing the response patterns of machines on lightly and heavily loaded links, sniffer detector tools can determine whether a machine appears to be in promiscuous mode. A machine monitoring the segment will have to interpret every message on a heavily loaded link, placing a high processing overhead on that machine. Therefore, the response pattern for a monitor will differ greatly between light and heavy loads, while for normal configurations the patterns should be near identical.
- **Direct inspection**: Directly checking the state of a network adapter on host machines is possible and may be the only way to detect which machines are in promiscuous mode under certain circumstances. This method may not be

feasible on large networks, and on an ongoing basis, however. Of course, tools have been developed to attempt to avoid detection – but the presence of an unauthorized monitor on a network is a strong indication that there is a security problem.

2.8.6 Honeytrap Systems

As pointed, current IDS methodologies have a number of shortcomings, including problems recognizing novel attacks, the occurrence of false positives, and reporting of attacks that are of no interest (because the system is known to be invulnerable to these attacks). A tool which attempts to bridge these gaps is that of honeytrap systems simulated or real systems that exist for the sole purpose of being attacked. In essence, the goal of these systems is to act as bait encouraging attackers to attack these in preference to more valuable parts of a network. Once such a system is attacked, an administrator knows that the network is under attack, and can closely monitor the attacker. Since the system is not generally used, the problem of false positives does not occur any activity on that system is hostile. Since the system does not depend on recognizing specific attacks, and the limited activity levels allow thorough

review of all activity on that system, novel attacks can be observed and studied. Finally, the fact that an attacker has penetrated the honeytrap system implies that other systems on the network are vulnerable. One of the most important aspects of a honeytrap system is that it should not be recognizable as such to an attacker, it should look and behave as a real system would. In addition, in order to learn from such a system, it should be configured similarly to the real systems on a network, allowing lessons learned there to be applied directly in improving the security of more valuable machines. While software tools are available that simulate networks and hosts, in this section we shall focus on honeytraps built out of dedicated systems.

The first step issue in setting up a honeytrap is to ensure that it does not reduce the security of other systems on the network. Should a honeytrap system be compromised, it must be ensured that this system cannot be used to attack other systems (on the same network, or any other). Many of the mechanisms used in firewalls apply here aimed at keeping the intruder in, rather than out. Secondly, an administrator should ensure that the honeytrap system gathers as much information as possible. In addition, this information should be kept in a safe area since it is assumed that the honeypot will become compromised. In addition, this increased logging should be hidden from an attacker, to avoid them focusing on "less protected" and more valuable systems. For an eminently readable description of how a honeytrap works, and what its value in a system under attack. The legality of using such systems has been a subject of some discussion the conclusion of which was that a honeytrap is no more illegal than a burglar alarm.

2.9 Summary

In this chapter we discussed activate security mechanisms, limitations of static security and we identified the limitation, what static methods offer, sources of attack, typical attack techniques, policy issues for active security, tools supporting active security and in the last we discussed honeytrap systems.

3. BLUETOOTH

3.1 Overview

In the past, the only way to connect computers together for the purpose of sharing information and/or resources was to connect them via cables. This can be not only cumbersome to set up, but it can get messy real quick. Bluetooth provides a solution to this problem by providing a cable-free environment.

3.2 What is Bluetooth?

Bluetooth wireless technology is a short-range communications technology intended to replace the cables connecting portable and/ore fixed devices while maintaining high levels of security. The key features of Bluetooth technology are robustness, low power, and low cost. The Bluetooth specification defines a uniform structure for a wide range of devices to connect and communicate with each other.

The idea behind Bluetooth technology was born in 1994, when a team of researchers at Ericsson Mobile Communications initiated a feasibility study of universal short-range, low-power wireless connectivity as a way of eliminating cables between mobile phones and computers, headsets and other devices. (2005, Bialoglowy) In 1998, this group evolved to the Bluetooth Special Interest Group (SIG). Along with Ericsson, other founding members included Nokia, Intel, IBM and Toshiba.

Many people wonder where the name Bluetooth came from. According to Bluetooth SIG (Bluetooth SIG, 2006),The name "Bluetooth" is taken from the 10th century Danish King Harald Blatand- or Harold Bluetooth in English. During the formative stage of the trade association a code name was needed to name the effort. King Blatand was instrumental in uniting warring factions in parts of what is now Norway, Sweden, and Denmark - just as Bluetooth technology is designed to allow collaboration between differing industries such as the computing, mobile phone, and automotive markets. The code name stuck.

3.3 How Bluetooth Works?

3.3.1 Frequency Range

Bluetooth is a wireless protocol that operates on the unlicensed 2.4GHz band. Since version 1.2 it uses an adaptive frequency hopping algorithm to avoid ser- vice interruption due to other equipment using the same frequencies, and also to avoid causing interference to other equipment as well.

Table 3.1: Bluetooth frequency range and channels

Frequency range	Channels
2.400 – 2.4835 GHz	f = 2402 + k, k = 0, . . . , 78 MHz

As we can see in Table 3.1 there are 79 frequencies in the 2.4GHz band that Bluetooth may use for its hopping algorithm. This hopping algorithm does not add any security on the link, since the hopping sequence is broadcasted in clear at the initiation of a connection.

3.3.2 Distance Covered

Not all Bluetooth devices have the same signal strength nor can cover the same distance. Most of the devices have a freedom in selecting their output power level. The Bluetooth specification sorts devices based on their power class which is summarized in Table 3.2.

Table 3.2: Bluetooth power classes

Power class	Minimum output power	Maximum output power	Distance covered
Class 1	1 mW	100 mW	up to 100 meters
Class 2	0.25 mW	2.5 mW	to 10 meters
Class 3	1 mW	1mW	up to 1 meter

3.4 Bluetooth Network

All Bluetooth devices hold a unique address called the Bluetooth Device Address. This is a 48 bit number assigned at production time to the device by the manufacturer and cannot be altered. This is very similar to the ethernet unique MAC addresses. These device addresses are usually represented in hexadecimal colon separated format such as 00:0f:fa:ad:ea:f0. The importance of these addresses in networking is substantial since they are used for device identification in a network.

3.4.1 Layers

The lowest Bluetooth core protocol layers are shown in Figure 3.1.

L2CAP layer	responsible for managing the ordering of submission of PDU fragments to the baseband and scheduling
Link Manager Protocol layer	responsible for all aspects of a Bluetooth connection, such as power con- trol, roles, encryption etc.

Link Controller layer	responsible for the encoding and decoding of Bluetooth packets from the data payload and parameters related to the physical channel, logical transport and logical link
Radio layer	responsible for the actual transmitting and receiving of packets of information on the physical channel.

Figure 3.1: Bluetooth Core Protocol Stack

The most important transport layers available in the Bluetooth link controller are summarized below:

SCO: This is the Synchronous Connection Oriented transport, which is a point to point channel between a master and a slave. It has a constant data rate whilst no retransmission is available. It is typically used for voice connections.

eSCO: The extended Synchronous Connection Oriented transport, which is a point to point channel between a master and a slave. It offers some extensions over the SCO that include a limited retransmission.

ASB: The Active Slave Broadcast transport carries L2CAP traffic to the devices in the piconet that are connected to the channel of ASB. This is uni-directional communication between the master and the slaved and there is no acknowledgment of packet receipt.

ACL: The Asynchronous Connection Oriented transport carries the Link Manager and the L2CAP control and data. It is a bidirectional point to point protocol.

3.4.2 Network Structure

Since in wireless networks devices do not really share a physical link, such as a common cable, some other way of joining a network has to be used. In Bluetooth networked devices are in a common piconet channel. That means that they share a common clock and a common frequency hopping sequence. The common clock is the clock of a device called the master, which is the same device that provides the hopping sequence. All the other devices are called slaves. Devices can be members in several piconets and in that case they are called as being part of a scatternet as shown in figure Figure 3.2. Routing between such piconets is not part of the Bluetooth core protocols and thus left for upper layer protocols.

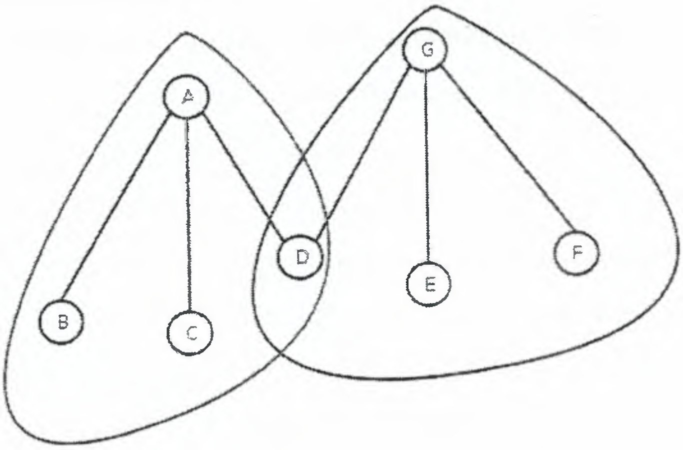


Figure 3.2: A scatternet consisting of two piconets

3.4.3 Networking Functions

Some low level networking functions of Bluetooth are Inquiry and Paging. Inquiry is the procedure of discovering for nearby devices. This is done by sending inquiry requests. Bluetooth devices that are available to be found listen for these inquiries and send responces. The physical channel used for these requests and responces is a separate one.

Paging is the procedure of a device connecting to another one. This is a targeted procedure which means that only one device will respond to this request. For this request also a separate physical channel is used to listen for the requests.

3.5 Bluetooth Protocol

Figure 3.3 illustrates the Bluetooth protocol stack, which can be divided into four layers according to their purpose, in the following way:

1. Bluetooth Core Protocols, including Baseband, LMP, L2CAP, and SDP, comprise exclusively Bluetooth-specific protocols developed by the Bluetooth SIG that are required by most of the Bluetooth devices.

2. Cable Replacement Protocol, i.e. RFCOMM protocol, is based on the ETSI TS 07.10 that emulate serial line control and data signals over Bluetooth Baseband to provide transport capabilities for upper level services.

3. Telephony Control Protocols, including TCS Binary and AT-commands, are used to define the call control signalling, mobility management procedures, and multiple usage models for the Bluetooth devices to establish the speech and data calls and provide FAX and modem services.

4. Adopted Protocols, including PPP, UDP/TCP/IP, WAP, WAE, etc. Due to the open nature of the Bluetooth specification, additional protocols (e.g., HTTP, FTP, etc.) can be accommodated in an interoperable fashion.

5. Host Controller Interface (HCI), i.e. the boundary between hardware and software, provides a uniform command interface to access capabilities of hardware, e.g. Baseband controller, link manager, control and event registers.

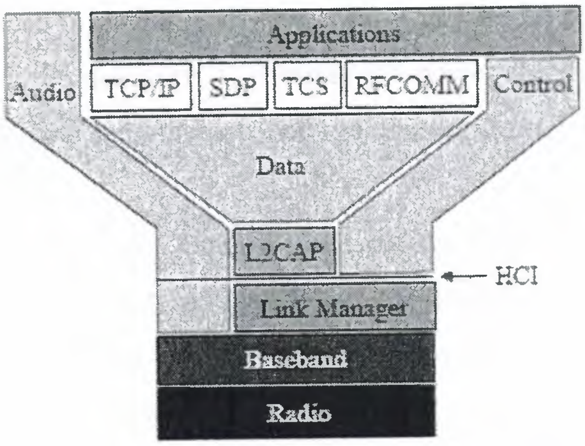


Figure 3.3: Bluetooth protocol stack

The layers of Cable Replacement, Telephony Control, and Adopted Protocols form the application-oriented protocols that enable applications to run over the Bluetooth core protocols. Not all applications make use of all the protocols shown in Figure 3.3. Instead, applications run over one or more vertical slices of this protocol stack. In other words, applications may run over different protocol stacks. Nevertheless, each one of these different protocol stacks uses a common Bluetooth data link and physical layer, i.e. Bluetooth core protocols, including:

- Baseband. Based on the physical radio link, the Baseband can form the piconet between Bluetooth units and decide the roles of master and slave in the piconet. The Baseband provides physical links of both Synchronous Connection- Oriented (SCO) and Asynchronous Connectionless (ACL) to support the transmission of data and/or audio with corresponding packets. Other functions include error correction, link management and control, audio transmission, etc.

- Link Manager Protocol (LMP). The Bluetooth protocol LMP is responsible for link set-up between Bluetooth devices. This includes security aspects and the control and negotiation of Baseband packet sizes. Furthermore, it controls the power modes and duty cycles of the Bluetooth radio device, and the connection states of a Bluetooth unit in a piconet.
- Logical Link Control and Adaptation Protocol (L2CAP). The protocol of L2CAP provides connection-oriented and connectionless data services to the upper layer protocols over the Baseband, with protocol multiplexing capability, segmentation and reassembly operation, and group abstractions, which permits higher level protocols and applications to transmit and receive L2CAP data packets. L2CAP is defined only for ACL links.
- Service Discovery Protocol (SDP). Using SDP to discover services is a crucial part of the Bluetooth framework and provides the basis for all the usage models. SDP query device information, services information, and the characteristics of the services, according to which a suitable connection between two or more Bluetooth devices can be established.

3.6 Establishing a Connection in Bluetooth

This section describes the basic procedures to be followed by two or more Bluetooth devices to start a connection between them. Consider the following scenario: A person walks in to a hotel lobby and wants to access her email over her Bluetooth enabled device, which could be a laptop or a Personal Digital Assistant. What would she have to do? Depending on the implementation., she would be clicking on a menu or an email application icon. The device would automatically carry out the following steps, (except perhaps for the authentication step if the device has come to the environment for the first time):

1. **Inquiry**: The device on reaching a new environment would automatically initiated an inquiry to find out what access points are within its range. (If not, it'll do so when the email application asks for a link.) This will result in the following events:
- all nearby access points respond with their addresses.
- the device picks one out the responding devices.

2. **Paging:** The device will invoke a baseband procedure called paging. This results in synchronization of the device with the access point, in terms of its clock offset and phase in the frequency hop, among other required initializations.

3. **Link establishment**: The LMP will now establish a link with the access point. As the application in this case is email, an ACL link will be used. Various setup steps will be carried out as described below.

4. **Service Discovery**: The LMP will use the SDP(Service Discovery Protocol) to discover what services are available from the access point, in particular whether email access or access to the relevant host is possible from this access point or not. Let us assume that the service is available, otherwise, the application cannot proceed further. The information regarding the other services offered at the access point may be presented to the user.

5. **L2CAP channel:** With information obtained from SDP, the device will create an L2CAP channel to the access point. This may be directly used by the application or another protocol like RFCOMM may be run over it.

6. **FCOMM channel:** Depending on the need of the email application an RFCOMM or other channel(in case of other applications) will be created over the L2CAP channel. This feature allows existing applications developed for serial ports to run without modification over Bluetooth platforms.

7. **Security:** If the access point restricts its access to a particular set of users or otherwise offers secure mode communications to people having some prior registration with it, then at this stage, the access point will send a security request for "pairing". This will be successful if the user knows the correct PIN code to access the service. Note that the PIN is not transmitted over the wireless channel but another key generated from it is used, so that the PIN is difficult to compromise. Encryption will be invoked if secure mode is used.

8. **PPP:** Assuming that a PPP link is used over serial modem as in dial up networking, the same application will now be able to run PPP over RFCOMM(which emulates the serial port). This link will allow the user to login to his email account.

9. **Network Protocols:** The network protocols like TCP/IP, IPX, Appletalk can now send and receive data over the link.

In the above procedure, user interaction is required only at the usual login for his email and additionally for the security to be implemented. The remaining steps are automatic. The above procedures now be described in detail to demonstrate the connection establishment process. The explanation of the above procedures requires a brief description the device clocks in Bluetooth.

3.6.1 Clock

Every Bluetooth unit has an internal system clock that determines the timing and hopping of the transceiver. The Bluetooth clock is derived from a free running native clock that is never adjusted and is never turned off. For synchronization with other units, only offsets are used that, added to the native clock, provide temporary Bluetooth clocks which are mutually synchronized. The Bluetooth clock has no relation to the time of day. The Bluetooth clock is very important for the Bluetooth transceiver as it is involved in timing a number of important events without which communication is not possible. Its resolution is at least half the TX or RX slot length, or 312.5 microseconds. The clock has a cycle of about a day. If the clock is implemented with a counter, a 28-bit counter is required that wraps around at 228 -1. The LSB ticks in units of 312.5 microseconds, giving a clock rate of 3.2 kHz.

The timing and the frequency hopping on the channel of a piconet is determined by the Bluetooth clock of the master. When the piconet is established, the master clock is communicated to the slaves. The slaves store the required offset to be used while communicating with the particular master and use it to synchronize to the channel. As the local clock itself is not modified, different offsets can be used to participate in various piconets.

The minimum clock accuracy required is +/- 20ppm in active mode and +/-250ppm in low power states like Hold, Sniff, Standby and Park.

3.6.2 Inquiry and Paging

These are the initial steps in starting a connection. The device before, during and after these procedures can be viewed to be in different states shown in Fig. 3.4.

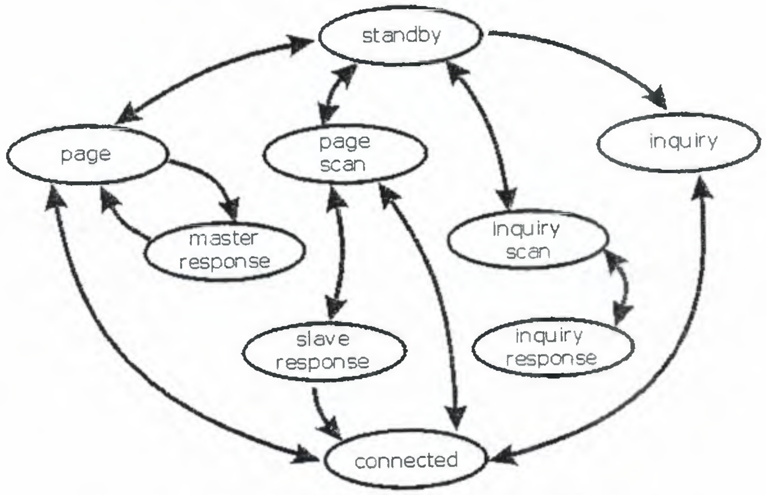


Figure 3.4: State diagram of the link controller.

The device is in Standby state by default. In this state only the native clock is running and power consumption is very low. It may leave this state to go to Inquiry, Inquiry Scan, Page or Page Scan states. These states are described below:

1. **Inquiry:**

In this state, the device sends an Inquiry packet addressed to either the General Inquiry Access Code(GIAC) or Dedicated Inquiry Access code(DIAC) which refers to a particular class of devices, say printers. The transmission is repeated at 16 frequencies which form the inquiry hop sequence, called a train. A device which is allowing itself to be inquired will be listening at one of these frequencies. The transmission is carried out in every alternate slot and the intermediate slots are used for listening to responses if any. There are two trains of hop frequencies- A and B. Each train must be repeated 256 times to collect all responses in an error free environment. The total time required for doing this is 10.24 seconds. However, if enough responses are collected in a smaller interval, inquiry may be aborted in between.

If the inquiry procedure is automatically initiated, say once every minute, then the interval between successive inquiries must be random to avoid synchronization and hence a collision with another device involved in inquiry.

2. Inquiry Scan:

A device which allows itself to be discovered will periodically enter the inquiry scan substate and listen for inquiry packets at a single frequency, which it will chose out of the 16 frequencies in the inquiry hop sequence depending on its device address. It will stay in that state long enough for an enquiring device to cover 16 different frequencies. A device may be entering inquiry scan state from standby or connected states. If it is entering from the connection state, the SCO links in operation will be maintained while the ACLlinks will be suspended. The presence of SCO inks may prolong the inquiry procedures.

3. Inquiry Response

When an inquiry message is received in the inquiry scan state, a response packet containing the responding device address must be sent. However it is not sent in the immediately following slot after the slot in which inquiry is received as that might cause many devices listening at a given frequency to respond simultaneously, resulting in a collision. So the responding device waits for a random number of slots and then sends its FHS packet to the inquirer. The FHS packet contains the device address; its clock and information about when the device enters its page scan states. After responding to an inquiry, the device continues its inquiry scan at another frequency, without waiting for an acknowledgement.

The inquiring device on receiving an inquiry does not acknowledge the response but continues its inquiry procedure as long as it wishes to. Only when the inquiring device wants to page the device that responded, say at a later time when a connection is required, it will use the response information to page.

After the inquiry has been successfully carried out, or the device address of the device to which a connection has to be made has been determined by some other means like information from previous connections, the device will start a paging procedure if a connection is desired. Paging requires only the address of the device to be paged but the clock information, from the FHS response packet, may be used to speed up the procedure. The device starting the paging procedure is called the master, and it will be the master of the piconet consisting of itself and the paged device if the paged device accepts the connection. Before starting data communications however, the devices may exchange their roles.

This procedure will usually occur whenever the Bluetooth device enters a new environment or some older links become unavailable. Now, when the application is invoked, the device will start paging procedures.

4. **Page:**

This state requires the master to do the following:

The master uses the clock information, if any, about the slave to be paged, to determine where in the hop sequence, the slave might be listening in the page scan mode. This estimate may be totally wrong.

The master calculates the Device Access Code (DAC) of the slave from the device address of the slave using a well-defined procedure.

The master sends a page message. The master transmits this page message at a number of frequencies in the page hop sequence, starting with the frequency at which it had estimated the slave to be listening. The page hop sequence consists of 32 frequencies divided into two trains of 16 each. The train A includes the 16 frequencies surround the predicted frequency and train B the remaining ones. If the clock estimate is wit 8x1.28 to +7x1.28 seconds then the slave will be able to respond within the train itself. The master does not know when the slave enters the page scan mode, so the pa train is repeated Npage times, unless a response is received earlier.

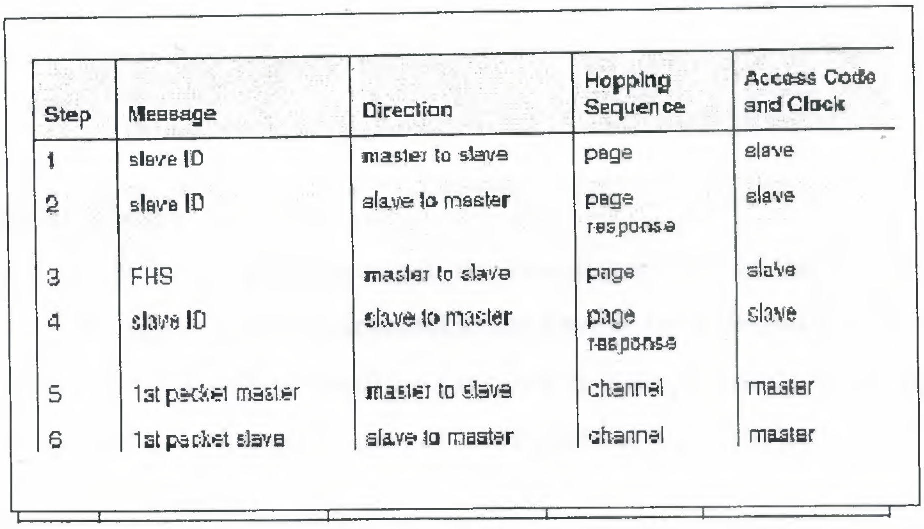
Npage is determined such that slaves using any of the allowed scanning intervals m be covered. If train A fails, train B is tried, again Npage times. If train A is successf the paging procedure will be over in 1.28 seconds, else it will take 2.56 seconds.

5. Page Scan:

The page scan substate can be entered from the standby state or the connecti state. In this state, the slave listens to page packets addressed to its DAC for an interv Tw-page-scan at a frequency it chooses out of the page scan sequence. This window long enough to cover 16 frequency hops of a paging device. These listening periods a separated by time interval of Tpage- scan. This interval may be zero (continuous scan Three different scan modes, that is values of Tpage-scan are fixed. Other values may b used by a slave after informing the master. Thus one of the standard values is used fo the first link.

6. Page Response

On receiving the page message, the slave enters the slave page response substate. I sends back a page response consisting of its ID packet that contains its DAC, at th frequency for the next slot from the one in which page message was received. Th master on receiving this packet enters the master page response substate. At this point, i knows which frequency the slave had been listening. The master sends its FHS packe to the slave informing the slave of the master clock, still using the slave DAC, at th slave's listening frequency. The FHS packet also assigns a three-bit active membe address to the slave. The slave acknowledges this packet again sending its ID packet at its slave response frequency. The slave now uses the FHS packet received from the master to determine the channel access code for the piconet newly formed, or to which this slave has newly entered. It also calculates the clock offset to be used while communicating over this piconet. The next packet from the master to slave, which is the POLL packet addressed to the active member address of the slave, is at the master clock dependent frequency hop and uses the channel access code. The slave may respond to this packet with any packet, say a NULL packet (containing only channel header), but it must respond. If the response procedure is successful, the paging is over, and the slave is in connected state. Otherwise, paging is considered to have failed and the error procedures are followed.



Step	Message	Direction	Hopping Sequence	Access Code and Clock
1	slave ID	master to slave	page	slave
2	slave ID	slave to master	page response	slave
3	FHS	master to slave	page	slave
4	slave ID	slave to master	page response	slave
5	1st packet master	master to slave	channel	master
6	1st packet slave	slave to master	channel	master

Figure 3.5: Initial message exchanges during start-up.

After the page procedure that is in the connected state the devices are in a position to establish a link.

The Link Managers of the devices in connection state now exchange vital information for starting up the link, which will be described below. They may later detach the link, in which case the address and clock information will stay valid after detachment, or the link may get snapped due to other reasons in which case all information related to the link is reset.

The Bluetooth units can be in several modes of operation during the connection state: active mode, sniff mode, hold mode, and park mode. These modes are now described.

1. Active mode:

In this mode, the Bluetooth unit actively participates on the channel. Master and slaves transmit in alternate slots. The master transmits in all even numbered slots and the addressed slave transmits in the subsequent slot. Regular transmissions are made by the master to keep the slaves synchronized to the channel. Various optimizations are provided to save power. For instance if the master informs the slave when it will be addressed, the slave may sleep until then. The active slaves are polled by the master for transmissions.

2. Sniff Mode:

This is a low power mode in which the listening activity of the slave is reduced. The LMP in the master issues a command to the slave to enter the Sniff mode giving it a sniff interval Tsniff, an offset Dsniff, and number of attempts Nsniff. In the sniff mode, the slave listens for transmissions only at fixed intervals Tsniff, at the offset slot Dsniff for Nsniff times.

3. Hold Mode:

In this mode the ACL link to a slave is put on hold. This means that the slave temporarily does not support ACL packets on the channel any more (possible SCO links will still be supported). With the hold mode, capacity can be made free to do other things like scanning, paging, inquiring, or attending another piconet. The unit in hold mode can also enter a low-power sleep mode. During the hold mode, the slave unit keeps its active member address (AM_ADDR). The master and slave agree upon a duration for the hold interval, after which the slave comes out of hold mode.

4. **Park Mode:**

This is a very low power mode. The slave has very little activity in this mode. It gives up its active member address and is given an eight bit parked member address and an eight bit access request address. The parked member address can be used by the master to unpark a slave while the access request address is used by the slave to ask the master to unpark it. The slave however, stays synchronized to the channel. Any messages to be sent to a parked member are sent over the broadcast channel, that is the active member address of all zeros. The parked slave has to be informed about this transmission in a beacon channel that is supported by the master to keep parked saves in synchronization and send them any other information. The parked slaves regularly listen for beacon signals at intervals decided by the beacon structure communicated to the slave during the start of parking. Apart from saving power, the park mode helps the master to have more than seven slaves(limited by the three bit active member address space) in the piconet.

3.7 Summary

In this chapter we argued Bluetooth is a wireless technology is a short-range communications technology and how it works , how we can have a network using it and the protocols of Bluetooth also how to make a connection in Bluetooth.

4. SECURITY OVER BLUETOOTH

4.1 Overview

Bluetooth has three different modes of security. Each Bluetooth device can operate in one mode only at a particular time. The three modes are the following:

Security Mode 1: Nonsecure mode

Security Mode 2: Service-level enforced security mode

Security Mode 3: Link-level enforced security mode

4.1.1 Security Mode 1: Nonsecure Mode

A device will not initiate any security procedures. In this nonsecure mode, the security functionality (authentication and encryption) is completely bypassed. In effect, the Bluetooth device in Mode 1 is in a promiscuous mode that allows other Bluetooth devices to connect to it. This mode is provided for applications for which security is not required, such as exchanging business cards.

4.1.2 Security Mode 2: Service-level Enforced Security Mode

In the service-level security mode, security procedures are initiated after channel estab- lishment at the Logical Link Control and Adaptation Protocol (L2CAP) level. L2CAP resides in the data link layer and provides connection-oriented and connectionless data services to upper layers. For this security mode, a security manager (as specified in the Bluetooth architecture) controls access to services and to devices. The centralized security manager maintains polices for access control and interfaces with other proto- cols and device users. Varying security polices and trust levels to restrict access may be defined for applications with different security requirements operating in parallel. Therefore, it is possible to grant access to some services without providing access to other services. Obviously, in this mode, the notion of authorization

– that is the process of deciding if device A is allowed to have access to service X – is introduced.

4.1.3 Security Mode 3: Link-level Enforced Security Mode

In the link-level security mode, a Bluetooth device initiates security procedures before the channel is established. This is a built-in security mechanism, and it is not aware of any application layer security that may exist. This mode 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.

4.2 Bluetooth Key Generation from PIN

The link key is generated during an initialization phase, while two Bluetooth devices that are communicating are ”associated” or ”bonded.” Per the Bluetooth specification, two associated devices simultaneously derive link keys during the initialization phase when a user enters an identical PIN into both devices. The PIN entry, device associ- ation, and key derivation are depicted conceptually in Figure 4.1. After initialization is complete, devices automatically and transparently authenticate and perform encryption of the link. It is possible to create a link key using higher layer key exchange meth- ods and then import the link key into the Bluetooth modules. The PIN code used in Bluetooth devices can vary between 1 and 16 bytes. The typical 4-digit PIN may be sufficient for some applications; however, longer codes may be necessary.

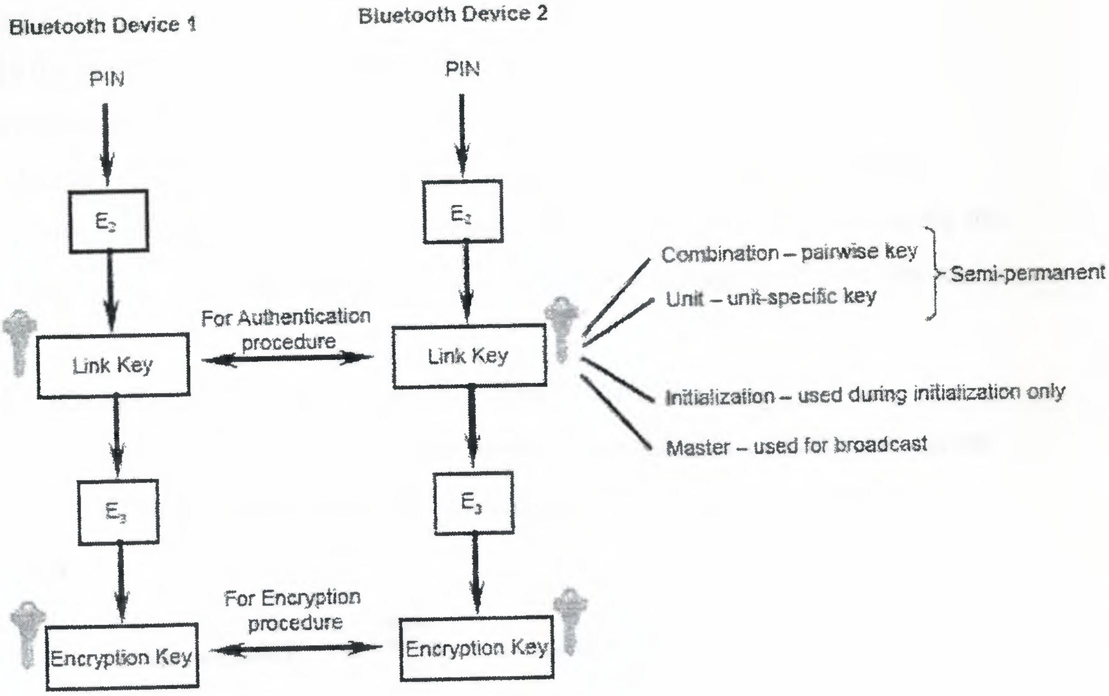


Figure 4.1: Bluetooth Key Generation from PIN

Karygiannis T., and Owens L., Wireless Network Security 802.11, Bluetooth and Handheld Devices, NIST National Institute of Standards and Technology, November 2002

4.3 Bluetooth Authentication

The Bluetooth authentication procedure is in the form of a ”challenge-response” scheme. Two devices interacting in an authentication procedure are referred to as the claimant and the verifier. The verifier is the Bluetooth device validating the identity of another device. The claimant is the device attempting to prove its identity. The challenge- response protocol validates devices by verifying the knowledge of a secret key (a Blue- tooth link key). The challenge-response verification scheme is depicted conceptually in Figure 4.2. As shown, one of the Bluetooth devices (the claimant) attempts to reach and connect to the other (the verifier).

The steps in the authentication process are the following:

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 the E1 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 com- putes.
6. If the two 32-bit SRES values are equal, the verifier will continue connection establishment.

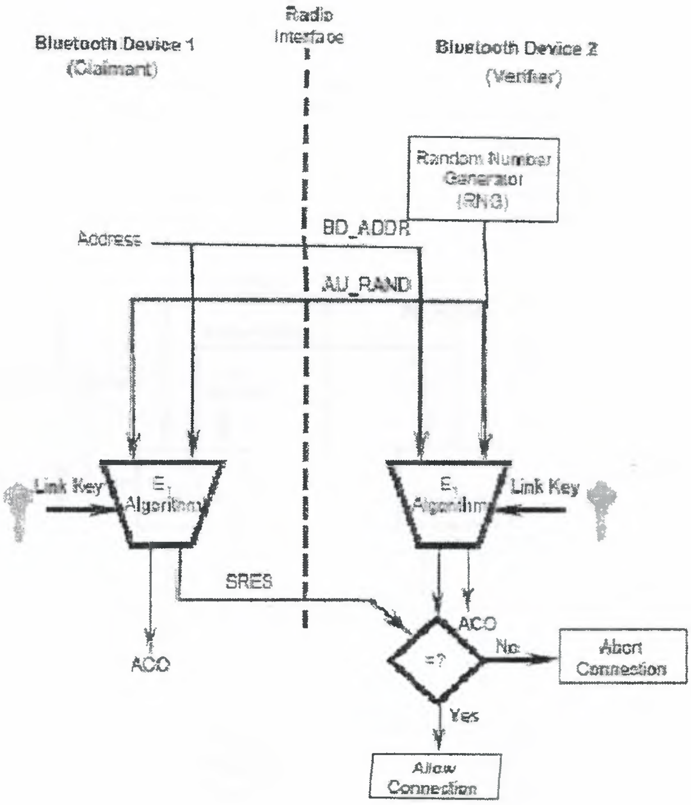


Figure 4.2: Bluetooth Authentication

Karygiannis T., and Owens L., Wireless Network Security 802.11, Bluetooth and Handheld Devices, NIST National Institute of Standards and Technology, November 2002

4.4 Bluetooth Encryption Process

The Bluetooth specification also allows three different encryption modes to support the confidentiality service:

- Encryption Mode 1:

No encryption is performed on any traffic.

- Encryption Mode 2:

Broadcast traffic goes unprotected (not encrypted), but individually addressed traffic is encrypted according to the individual link keys.

- Encryption Mode 3:

All traffic is encrypted according to the master link key.

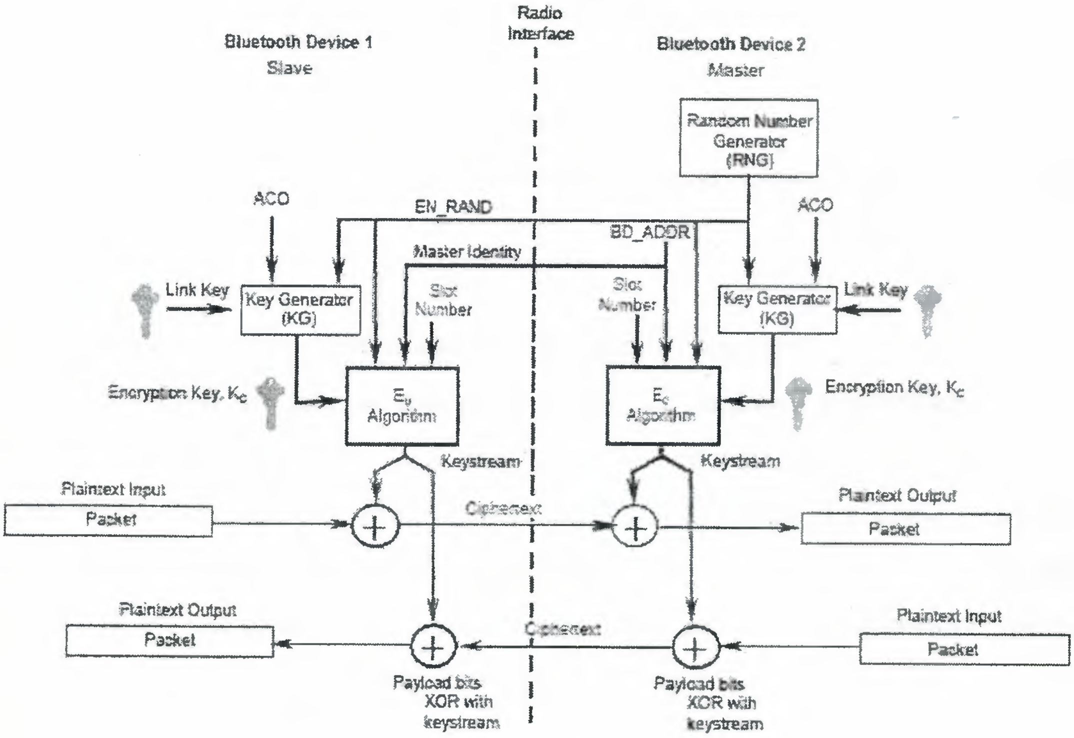


Figure 4.3: Bluetooth Encryption Process

Karygiannis T., and Owens L., Wireless Network Security 802.11, Bluetooth and Handheld Devices, NIST National Institute of Standards and Technology, November 2002

4.5 Problems with the Bluetooth Standard Security

- Strength of the challenge-response pseudorandom generator is not known:

The Random Number Generator (RNG) may produce static number or periodic numbers that may reduce the effectiveness of the authentication scheme.

- Short PINS are allowed:

Weak PINs, which are used for the generation of link and encryption keys, can be easily guessed. Increasing the PIN length in general increases the security. People have a tendency to select short PINs.

An elegant way to generate and distribute PINs does not exist:

- Establishing PINs in large Bluetooth networks with many users may be difficult. Scalability problems frequently yield security problems.

- Encryption key length is negotiable:

The Bluetooth SIG needs to develop a more robust initialization key generation procedure.

- Unit key is reusable and becomes public once used:

A unit key is a link key that one unit generates by itself and uses as a link key with any other device. Unit keys can only be safely used when there is full trust among the devices that are paired with the same unit key. This is because every paired device can impersonate any other device holding the same unit key. Since Bluetooth version 1.2, the use of unit keys is not recommended. But, for legacy reasons, unit keys have not been completely removed from the specification.

- The master key is shared:

The Bluetooth SIG needs to develop a better broadcast keying scheme.

- No user authentication exists:

Device authentication only is provided. Application level security and user authentication can be employed.

- Attempts for authentication are repeated:

The Bluetooth SIG needs to develop a limit feature to prevent unlimited requests. The Bluetooth specification requires a time-out period between repeated attempts that will increase exponentially.

- E0 stream cipher algorithm is weak:

The stream cipher E0 has its roots in the so-called summation combiner stream cipher. This was a stream cipher that was proposed by Massey and Rueppel in the mid-1980s. The most powerful attacks on this type of stream ciphers are the correlation attacks in combination with exhaustive search over a limited key space (this is sometimes also referred to as initial guessing). Recent cryptanalysis shows that the E0 cipher is weaker than this.

- Key length is negotiable:

A global agreement must be established on minimum key length.

- Unit key sharing can lead to eavesdropping:

A corrupt user may be able to compromise the security between (gain unauthorized access to) two other users if that corrupt user has communicated with either of the other two users. This is because the link key (unit key), derived from shared information, is disclosed.

- Privacy may be compromised if the Bluetooth device address (BD ADDR) is captured and associated with a particular user:

Once the BD ADDR is associated with a particular user, that user's activities could be logged, resulting in a loss of privacy.

- Device authentication is simple shared-key challenge-response:

One-way-only challenge-response authentication is subject to man-in-the-middle attacks. Mutual authentication is required to provide verification that users and the network are legitimate.

- End-to-end security is not performed:

Only individual links are encrypted and authenticated. Data is decrypted at intermediate points. Applications software above the Bluetooth software can be developed.

- Security services are limited:

Audit, nonrepudiation, and other services do not exist. If needed, these can be developed at particular points in a Bluetooth network.

4.6 Bluetooth Security Attacks

4.6.1 Impersonation Attack by Inserting/Replacing Data

When no encryption is activated, this can easily be achieved by correctly setting the CRC check data in the payload after the data in the payload has been changed.
When ciphering is activated, the attacker can compute how to modify the CRC to make it agree with modifications in the encrypted data bits.

In a practical system were encryption is activated, it is not at all easy to make some- thing useful of this attack beyond the point of just disrupting the communication. The attacker must somehow know the context of the payload data to conduct changes that are meaningful or effective.

4.6.2 Bluejacking

Although known to the technical community and early adopters for some time, the process now known as "Bluejacking" has recently come to the fore in the consumer arena, and is becoming a popular mechanism for exchanging anonymous messages in public places. The technique involves abusing the Bluetooth "pairing" protocol, the system by which Bluetooth devices authenticate each other, to pass a message during the initial "handshake" phase. This is possible because the "name" of the initiating Bluetooth device is displayed on the target device as part of the handshake exchange, and, as the protocol allows a large user defined name field - up to 248 characters - the field itself can be used to pass the message. This is all well and good, and, on the face of it, fairly harmless, but, unfortunately, there is a down side. There is a potential security problem with this, and the more the practice grows and is accepted by the user community, and leveraged as a marketing tool by the vendors, the worse it will get. The problem lies in the fact that the protocol being abused is designed for information exchange. The abil- ity to interface with other devices and exchange, update and synchronize data, is the reason of existence of Bluetooth. The Bluejacking technique is using the first part of a process that allows that exchange to take place, and is therefore open to further abuse if the handshake completes and the "bluejacker" successfully pairs with the target device.

If such an event occurs, then all data on the target device becomes available to the initiator, including such things as phone books, calendars, pictures and text messages. As the current wave of PDA and telephony integration progresses, the volume and quality of such data will increase with the devices' capabilities, leading to far more serious potential compromise. Given the furore that erupted when a second-hand Blackberry PDA was sold without the previous owner's data having been wiped, it is alarming to think of the consequences of a single bluejacker gathering an entire corporate staff 's contact details by simply attending a conference or camping outside their building or in their foyer with a Bluetooth capable device and evil intent. Of course, corporates are not the only potential targets - a Bluejacking expedition to, say, The House of Com- mons, or The US Senate, could provide some interesting, valuable and, who's to say, potentially damaging or compromising data.

This is also called and OBEX Push Attack: OBEX allows you to PUSH items anonymously in some cases between devices impact:

- Annoying, no real security impact
- Possible extensions to this idea is around sending vCard's with common names such as 'Home' or 'Work' in an attempt to overwrite an existing phone book entry in the recipients cell/smart phone

4.6.3 Bluetooth Wardriving

Map the physical whereabouts of users carrying Bluetooth-enabled devices. Since each Bluetooth device is freely broadcasts its unique 48-bit address, it is possible to track the user movements.

To protect a device against location tracking, an anonymity mode is needed. Devices operating in anonymous mode regularly update their device address by randomly choosing a new one.

Different types of location tracking attacks are possible:

subsubsectionInquiry attack The attack distributes one or more Bluetooth devices throughout a region to locate Bluetooth users. If the potential victim of such an attack has left his device in discoverable mode, attack- ing device can simply interrogate the area using frequent inquiry messages for devices and maintain a log of all the device addresses that are discovered. subsubsectionTraffic monitoring attack This attack succeeds even if the victim device is not in discoverable mode. The attacker simply monitors the communication between two trusted devices belonging to the victim. These devices will communicate using a specific CAC. This CAC is computed from the device address of the master device in the piconet. Furthermore, the whole device address is sent in the FHS packets of the devices, allow-ing an attacker to uniquely determine the identity of a device. But the FHS packets are only used at connection establishment. Subsubsection Pagin attack This attack allows the attacker to determine if a given device with a known BD ADDR or DAC is present within range.

The attack requires that the victims device is connectable.

The attacking device pages the target device, waits for the ID packet to be returned, and then does not respond. If an ID is returned, then the attacker knows that the victim device is present. The target device, waiting for the response, will just time out and the incident will not be reported to the application layer.

Subsubsection Frequency hopping attack The frequency hopping scheme in Bluetooth is determined by a repeating hopping sequence. The hopping scheme is calculated from different input parameters, such as an address and the master clock. In the connection state, the LAP and the four least significant bits in the UAP of the master device are used. In the page state, the LAP/UAP of the paged unit is used. Thus, it is (at least theoretically) possible to get information of the LAP and four bits in the UAP based on the observed hopping scheme. Subsubsection User-friendly name attack A Bluetooth device can request the user-friendly name anytime after a successful baseband paging procedure. The name request com- mand can be used to mount a location tracking attack.

4.6.4 Brute-Force Attack

Brute-force attack on the BD ADDR (MAC address) of a device while not in discoverable mode. Some manufacturer's claim this would take an unreasonable amount of time (eg, 11 hours). However, a multi-threaded version of @stake's RedFang could simultaneously utilize up to 8 USB Bluetooth devices which would reduce the 11hrs to approximately 90 minutes (based on one vendor's range).

Impact:

- Can take a long time before the correct BD ADDR is discovered
- Once the BD ADDR is discovered, a Bluesnarf attack could be set up, while the user thinks he/she is safe because the device is set to hidden mode

4.6.5 Denial-of-Service Attack on the Device

When the Bluetooth authentication fails, a certain amount of time must elapse before the verifier will initiate a new attempt to the same claimant and before the claimant sends a response to an authentication attempt by a unit using the same identity as the unit that notified an authentication failure. For each additional authentication failure, the waiting interval should be exponentially increased until a certain maximum value is obtained. The attacker prevents or prohibits the normal use or management of commu- nications facilities. The resulting system degradation can, for example, be the result of the system being fully occupied by handling bogus connection requests. If the attacker simulates a trustable device during these DoS, making the system decline trustable devices.

4.6.6 Disclosure of Keys

- A Bluetooth device attached to the computer may be exchanged for a false one, whose only purpose is to 'suck' out link keys from the host.

- A rightful USB plug or PCMCIA card may be removed from the owners computer and inserted into a corresponding slot of the adversarys computer. On this computer, one or more keys stored on the Bluetooth controller can be read out. Once the list of keys has been read out, the USB plug (or card) is returned to its proper owner, who may be completely unaware.

- Malicious software

  A Trojan horse disguised as something quite innocent can send the key database to some place where the adversary can access it. If this malicious code is dis- tributed through a virus or worm, the attack can quickly spread to a large number of computers.

Once the link key of a computer and phone (and the BD ADDR of the computer) is known, the adversary can silently connect to the mobile phone, impersonate the computer, and make use of any service the phone offers over Bluetooth.

4.6.7 Unit key Attacks

A unit that uses a unit key is only able to use one key for all its secure connections. Hence, it has to share this key with all other units that it trusts. Consequently, a trusted device (a device that possesses the unit key) that eavesdrops on the initial authentication messages between two other units that utilize the unit key will be able to eavesdrop on any traffic between these two units. The unit will be able to impersonate the unit distributing the unit key.

The potential risks with units keys have also been recognized by the Bluetooth SIG. Originally, the unit key was introduced in order to reduce memory requirements on very limited devices and remains part of the standard for backward compatibility reasons.

4.6.8 Backdoor Attack

The Backdoor attack involves establishing a trust relationship through the "pairing" mechanism, but ensuring that it no longer appears in the target's register of paired devices. In this way, unless the owner is actually observing their device at the precise moment a connection is established, they are unlikely to notice anything untoward, and the attacker may be free to continue to use any resource that a trusted relationship with that device grants access to. This means that not only data can be retrieved from the phone, but other services, such as modems or Internet, WAP and GPRS gateways may be accessed without the owner's knowledge or consent. Once the Backdoor is installed, the Bluesnarf attack will function on devices that previously denied access, and without the restrictions of a plain Bluesnarf attack.

4.6.9 Pairing Attack

The Bluetooth 1.1 specification is sensitive to passive and active attacks on the pairing procedure. The attacks only work if the attacker is present at the pairing occasion, which typically only occurs once between one pair of devices. If pairing is performed in public places during a connection to an access point, point-of-sale machine, or printer, this can be a dangerous threat.

4.6.10 BlueStumbling = BlueSnarfing

It is possible, on some makes of device, to connect to the device without alerting the owner of the target device of the request, and gain access to restricted portions of the stored data therein, including the entire phonebook (and any images or other data asso- ciated with the entries), calendar, realtime clock, business card, properties, change log, IMEI (International Mobile Equipment Identity, which uniquely identifies the phone to the mobile network, and is used in illegal phone 'cloning'). This is normally only pos- sible if the device is in "discoverable" or "visible" mode, but there are tools available on the Internet that allow even this safety net to be bypassed.

They refuse to say how the attacks actually works, but presumably it exploits a flaw where by a default 'pairing' password (probably only four characters) is guessed and the handset owner has left the device with Bluetooth switched on and visibility set to 'all'.

Also called an OBEX Pull Attack: OBEX allows you to PULL items anonymously in some cases between devices.

Impact:

- A number of Nokia, Ericsson & Sony Ericsson handsets are susceptible, so many popular phones are vulnerable for this attack
- Very much dependent on vendor's implementation of OBEX/Bluetooth stack
- Information obtainable can include calendar, real time clock, business card, prop- erties, change log, IMEI

4.6.11 BlueBug Attack

The BlueBug attack creates a serial profile connection to the device, thereby giving full access to the AT command set, which can then be exploited using standard off the shell tools, such as PPP for networking and gnokii for messaging, contact management, diverts and initiating calls. With this facility, it is possible to use the phone to initiate calls to premium rate numbers, send sms messages, read sms messages, connect to data services such as the Internet, and even monitor conversations in the vicinity of the phone. This latter is done via a voice call over the GSM network, so the listening post can be anywhere in the world. Bluetooth access is only required for a few seconds in order to set up the call. Call forwarding diverts can be set up, allowing the owner's incoming calls to be intercepted, either to provide a channel for calls to more expensive destinations, or for identity theft by impersonation of the victim.

Bluesnarf attack does allow the unauthorized downloading of items via the OBEX protocol, while the loophole identified in BlueBug allows to control the device via a plain serial connection.

4.6.12 PSM Scanning

Works on the idea that not all PSM (Protocol/Service Multiplexer) ports are registered with the local SDP (Service Discovery Protocol). So if we bypass the SDP database and try and connect to PSM's sequentially we may locate hidden functionality

Impact:

- No PSM's found to-date that offer other than advertised services
- Idea could be used to create a 'knock' style backdoor for Bluetooth devices

4.6.13 Off-Line PIN (via Kinit) Recovery

Sniff the initial 'RAND' transfer between two devices which occurs in clear-text (effectively the first stage of the bond)
Sniff the XOR'd 'RAND'(s) used for LinkKey generation
Sniff the AUTH RAND and AUTH SRES which both occur in clear-text (the last stage of the bond).
Do some number crunching and have enough data in order to be able to recover the
PIN, LinkKey and all inputs used for both needed to sync the frequency hopping or capture entire 2.4ghz spectrum and do off- line.

4.6.14 On-line PIN Cracking

Attack possible if fixed PIN exists in device (i.e. same PIN is used for every connecting device) need to change the Bluetooth address each time and try different PINs will bypass the ever increasing delay between retries counter measure. The specifications do not provide solution to this problem.

4.6.15 Off-line Encryption Key (via Kc)

Extends on from the Kinit recovery attack Very similar method as 2 of 3 needed seeds are known (i.e.master clock and Kc), simply sniff the EN RAND in addition.

4.6.16 Attack on the Bluetooth Key Stream Generator

Break the security of the cipher. Algebraic attack on the Linear Feedback Shift Register

Work effort circa 2ˆ67,58 operations.

4.6.17 Reflection Attack

A hacker can capture the MIN and ESN and pretend to be someone else stealing the Unit Key highlights weakness of only authenticating the device and not the user.

4.6.18 Replay Attacks

A hacker can record Bluetooth transmissions in all 79 frequencies and then in some way figure out frequency hopping sequence and then replay the whole transmission.

4.6.19 Man-in-the-Middle Attack

Intervention of traffic during pairing Bluetooth authentication does not use public key certificates to authenticate users.

4.6.20 Denial-of-Service Attack on the Bluetooth Network

Not very feasible would require the jamming of the whole ISM band.

4.6.21 A Man-in-the-Middle Attack Using Bluetooth in a WLAN in- Terworking Environment

A man-in-the-middle attack may be possible on the Bluetooth link in a WLAN inter-working environment. The attacker lures the victim to connect to a malicious WLAN access point. The attack does not require to know the Bluetooth link key. The attacker can repeat this attack on the same victim many times in any WLAN network.

4.6.22 Impersonate Original Sending/Receiving Unit

This attack requires the attacker to provide the correct response on the authentication challenge of a correspondent. Currently, no attack on the SAFER+-based E1 authentication function is known that achieves this within any realistic computational effort.

4.6.23 Correlation Attacks

The complexity of the attacks by Courtois is O(2ˆ49). The attack requires 2ˆ23.4 output bits. 2ˆ49 operations can be performed in about 35 hours, however, the result by Courtois shows that the core in E0 is not crypotgraphically strong. There are two possibilities to obtain an actual attack on E0 to recover K'c:

1. make the algebraic attack work with only 2,744 output bits (max number of known plaintext bits encrypted with the same Kp).

2. find a way to utilize that there exists a relation between the consecutive blocks of 2,744 output bits. This is a result of the fact that the output blocks are generated with the same constraint key K'c, BD ADDR and RAND values, but different clock timer values. This is not infeasible, because the relation between the initial state values (only differ in clock timer values) satisfies a linear relation over the finite field GF(2). It is possible to rewrite the relations in terms of a specific initial state, the clock bits and the output bits.

Both possibilities have not been done currently.
No complexity estimates for such attacks are known.

4.7 Summary

In this chapter we discussed security over Bluetooth, security modes and how we a Bluetooth device can operate in a security mode, and how to initiate a secret key,also the Bluetooth security attacks.

CONCLUSION

Although more and more qualified products emerge, at present Bluetooth is still more a laboratory technology to be studied than a widely used supporting technique for multitudinous products. The protocol is still in its research phase partly because of the security problems. Since the Bluetooth security scheme is reasonably robust to applications with less security requirements, the final features may depend more on the implementation than significant changes to the specification.

Based on the original design goal of cable replacement, Bluetooth is more suitable to short-range and small-size wireless personal area networks than for connecting with outside public networks, comparing to e.g. WLAN. To applications such as large ad hoc networks and outside interconnection access, high level security schemes should preferably be enforced for complementation. Examples include e.g. IPSEC for IP, secure routing protocols, distributed secret schemes, etc.

REFERENCES

[1] Tanebaum Andrew S., Computer Networks, 1996

[2] DWDM Network Designs and Engineering SolutionsJeff Apcar MPLS and VPN Architectures, Volume II

[3] Mastering Network Security by Chris Brenton, Cameron Hunt

[4] Bluetooth SIG, Specification of the Bluetooth System: Volume 1, Core, Version 1.1, Feb. 22, 2001.

[5] Mettala R., Bluetooth Protocol Architecture: Version 1.0, Bluetooth White Paper, Document # 1.C.120/1.0, Aug 25, 1999.

[6] Vainio J.T., Bluetooth Security, http://www.niksula.cs.hut.fi/~jiitv/bluesec.html.

[7] O. Levy and A. Wool. A uniform framework for cryptanalysis of the blue- tooth e0 cipher. Cryptology ePrint Archive, Report 2005/107, 2005. Avail- able from http://eprint.iacr.org/2005/107.pdf.

[8] A Security Evaluation of Bluetooth Profiles, Candolin
http://www.tml.hut.fi/˜candolin/Publications/BT/

[9] Attacks on the Bluetooth stack
http://www.esat.kuleuven.ac.be/cosic/thesis/bluetooth_ attacks_en.html

[10] BluejackQ Bluesnarfing :http:// www.bluejackq.com/bluesnarfing.asp

[11] An Algebraic Attack on the Bluetooth Key Stream Generator:
http://th.informatik.uni-mannheim.de/people/Armknecht/ TalkEWSCS.pdf

[12] http://www.bluetooth.com

DEFINITIONS, ACRONYMS and ABBREVIATIONS

ACL: Asynchronous Connection-oriented (logical transport).

BD ADDR: Bluetooth Device Address

CAC: Channel Access Code. A code derived from the master device address in a Bluetooth connection

CAN: Campus Area Network

CPE: Customer Premise Equipment

DAC: Device Access Code. A code derived from a specific slave device in a Bluetooth connection

DES: Data Encryption Standard

E0: Bluetooth ciphering algorithm built around four independent linear feed- back registers and a finite state machine as a combining circuitry. The final state machine is needed to introduce sufficient nonlinearity to make it difficult to re- compute the initial state from observing key stream data.

eSCO: Enhanced Synchronous Connection-Oriented. A logical channel for trans- port of prioritized synchronous user data.

FHS: Frequency Hop Synchronization

GSM: Global System Mobile

HAN: Home Area Network

HCI: Host Controller Interface

HTTP: Hypertext Transfers Protocol.

ID: Identifier

IDS: Intrusion Detection Systems

IEEE: Institute of Electrical and Electronics Engineers. A nonprofit technical professional association for engineers in this area

IP: Internet Protocol.

IPSEC: IP security protocol. An IETF security protocol used to protect IP pack- ets.

ISM: Industrial, Scientific, and Medical. A part of the radio spectrum reserved for these kinds of applications.

L2CAP: Logical link Communication and Adaptation Protocol.

LAP: Lower Address Part. Bits 0 to 23 of the unique 48-bit IEEE device address BD ADDR.

LMP: Link Manager Protocol

LAN: Local Area Network

LSB: Least Significant Bit

MAC: Message Authentication Code

MANs: Metropolitan Area Networks

PAN: Personal Area Network

PDA: Personal digital assistant

PDU: Protocol Data Unit

PIN: Personal Identification Number

PSM: Protocol/Service Multiplexer

RAN: Regional Area Network

RFCOMM: A serial cable emulation protocol based on ETSI TS 07.10

RNG: Random Number Generator

SCO: Synchronous Connection-Oriented. A logical channel for transport of synchronous user data.

SDP: Service Discovery Protocol. A protocol for locating services provided by or available through a Bluetooth device.

SIG: Special Interest Group. The organization owning the Bluetooth trademark, also responsible for the evolution of Bluetooth wireless technology.

TCP: Transmission Control Protocol. An IETF protocol for reliable IP communication.

UAP: Upper address part. Bits 24 to 31 of the unique 48-bit IEEE device address.

USB: Universal Serial Bus

WAN: Wide area network

WLAN: Wireless Local Area Network

WLLs: Wireless Local Loops

WMANs: Wireless Metropolitan Area Networks

WRAN: Wireless Regional Area Networks

WPANs: Wireless Personal Area Networks