Man-in-the-Middle Attack and its Countermeasure in Bluetooth Secure Simple Pairing

Thesis submitted in partial fulfillment of the requirements for the degree of

Master of Technology

in

Computer Science and Engineering

(Specialization: Information Security)

by

M Thrinatha Reddy

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela, Odisha, 769 008, India
May 2011

Man-in-the-Middle Attack and its Countermeasure in Bluetooth Secure Simple Pairing

Thesis submitted in partial fulfillment of the requirements for the degree of

Master of Technology

in

Computer Science and Engineering

(Specialization: Information Security)

by

M Thrinatha Reddy

(Roll- 209CS2091)

Supervisor

Prof. Sanjay Kumar Jena

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela, Odisha, 769 008, India
May 2011

Dedicated to My Parents

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India.

Certificate

This is to certify that the work in the thesis entitled ***Man-in-the-Middle Attack and its Countermeasure in Bluetooth Secure Simple Pairing*** by ***M Thrinatha Reddy*** is a record of an original research work carried out by him under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of Master of Technology with the specialization of Information Security in the department of Computer Science and Engineering, National Institute of Technology Rourkela. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Place: NIT Rourkela
Date: 20 May 2011

Dr. Sanjay Kumar Jena
Professor, CSE Department
NIT Rourkela, Odisha

Acknowledgment

I am grateful to numerous local and global peers who have contributed towards shaping this thesis. At the outset, I would like to express my sincere thanks to Prof. Sanjay Kumar Jena for his advice during my thesis work. As my supervisor, he has constantly encouraged me to remain focused on achieving my goal. His observations and comments helped me to establish the overall direction of the research and to move forward with investigation in depth. He has helped me greatly and been a source of knowledge.

I am very much indebted to Prof. Ashok Kumar Turuk, Head-CSE, for his continuous encouragement and support. He is always ready to help with a smile. I am also thankful to all the professors of the department for their support.

I express my gratitude to Saroj Kumar Panigrahy for generously sharing his time and knowledge and for making life fun while working, just like a friend.

I am really thankful to my all friends. My sincere thanks to everyone who has provided me with kind words, a welcome ear, new ideas, useful criticism, or their invaluable time, I am truly indebted.

I must acknowledge the academic resources that I have got from NIT Rourkela. I would like to thank administrative and technical staff members of the Department who have been kind enough to advise and help in their respective roles.

Last, but not the least, I would like to dedicate this thesis to my family, for their love, patience, and understanding.

M Thrinahta Reddy
m.thrinathareddy@gmail.com

Abstract

With the development of more types of devices which have bluetooth as a primary option to communicate, the importance of secure communication is growing. Bluetooth provides a short range wireless communication between devices making convenient for users and thus eliminating the need for messy cables.

The proliferation of the Bluetooth devices in the workplace exposes organizations to security risks. Bluetooth technology and associated devices are susceptible to general wireless networking threats, such as denial of service attack,eavesdropping, man-in-the-middle attacks, message modification, and resource misappropriation. Preventing unauthorized users from secure communication is a challenge to the pairing process.

The Man-in-the-Middle attack is based on sending random signals to jam the physical layer of legitimate user and then by falsification of information sent during the input/output capabilities exchange; also the fact that the security of the protocol is likely to be limited by the capabilities of the least powerful or the least secure device type. In addition, proposed a countermeasure that render the attack impractical. We have shown that, the proposed method can withstand the MITM attack and achieving all the security needs like authenticity, confidentiality, integrity and availability as well as it is an improvement to the existing Bluetooth secure simple pairing in order to make it more secure.

Contents

List of Figures

List of Tables

List of Abbreviations

Aboott	A Bluetooth Threat Taxonomy
AP	Access Point
BD ADDR	Bluetooth Device Address
DoS	Denial of Service
ECDH	Elliptic Curve Diffie Hellman
EDR	Enhanced Data Rate
FCC	Federal Communication Commission
FEC	Forward Error Correction
HS	High Speed
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection Schemes
IO	Input/Output
IPS	Intrusion Prevention Schemes
ISM	Industrial, Scientific, and Medical
J2SE	Java 2 Platform, Standard Edition
JW	Just Works
L2CAP	Logical Link Control and Adaptation Protocol
LDPC	Low Density Parity Codes
ME	Mobile Equipment
MITM	Man-in-the-Middle
NC	Numerical Comparison
NFC	Near Field Communication
OBEX	OBject EXchange
OOB	Out-Of-Band
PDR	Packet Delivery Ratio
PE	Passkey Entry
PHY	Physical Layer
PIN	Personal Identification Number

PSMs	Protocol Service Multiplexers
RF	Radio Frequency
SDP	Service Discovery Protocol
SMS	Short Message Service
SSP	Secure Simple Pairing
UDDA	Unauthorized Direct Data Access
USRP2	Universal Software Radio Peripheral
VERA-NG	Very Eccentric Radio frequency Antenna - Nerf Gun

Chapter 1

Introduction

Bluetooth Security
Why Bluetooth Security is Needed?
Bluetooth Security Modes
Motivation
Problem Statement
Thesis Organization

Chapter 1

Introduction

Bluetooth is a technology for short range wireless data and real time two-way audio/video transfer providing data rates up to 24 Mbps. It operates at 2.4 GHz frequency in the free Industrial, Scientific, and Medical (ISM) band. Bluetooth devices that communicate with each other form a piconet. The device that initiates a connection is the piconet master and all other devices within that piconet are slaves. The radio frequency (RF) waves can penetrate obstacles, because of this reason the use of wireless communication systems have grown rapidly in recent years. The wireless devices can communicate with no direct line-of-sight between them. This makes RF communication easier to use than wired or infrared communication, but it also makes eavesdropping easier. Moreover, it is easier to disrupt and jam wireless RF communication than wired communication. Because wireless RF communication can suffer from these threats, additional countermeasures are needed to protect against them.

1.1 Bluetooth Security

The basic Bluetooth security configuration is done by the user who decides how a Bluetooth device will implement its connectability and discoverability options. The different combinations of connectability and discoverability capabilities can be divided into three categories, or security levels [1]:

- **Public**: The device can be both discovered and connected to. It is therefore called a discoverable device.

- **Private**: The device cannot be discovered, i.e., it is a so-called non-discoverable device. Connections will be accepted only if the Bluetooth Device Address (BD ADDR) is known to the prospective master. A 48 bit BD ADDR is normally unique and refers globally to only one individual Bluetooth device.

- **Silent**: The device will never accept any connections. It simply monitors Bluetooth traffic.

In Bluetooth versions up to 2.0+EDR, pairing is based exclusively on the fact that both devices share the same Personal Identification Number (PIN) or passkey. As the PINs often contain only four decimal digits, the strength of the resulting keys is not enough for protection against passive eavesdropping on communication. It has been shown that Man-in-the-Middle attack (MITM) attacks on Bluetooth communications (versions up to 2.0+EDR) can be performed [1–5]. Bluetooth versions 2.1+EDR (Enhanced Data Rate) and 3.0+HS (High Speed) add a new specification for the pairing procedure, namely Secure Simple Pairing (SSP) [1]. Its main goal is to improve the security of pairing by providing protection against passive eavesdropping and MITM attacks. Instead of using (often short) passkeys as the only source of entropy for building the link keys, SSP employs Elliptic Curve Diffie-Hellman public-key cryptography. To construct the link key, devices use public-private key pairs, a number of nonces, and Bluetooth addresses of the devices. Passive eavesdropping is effectively thwarted by SSP, as running an exhaustive search on a private key with approximately 95 bits of entropy is currently considered to be infeasible in short time. In order to provide protection against MITM attacks, SSP either asks for user's help or uses an Out-Of-Band (OOB) channel. The SSP uses four association models named Numerical Comparison (NC), Passkey Entry (PE), Just Works (JW) and OOB.

- **Numerical Comparison:** It was designed for the situation where both Bluetooth devices are capable of displaying a 6-digit number and allowing a user to enter a 'yes' or 'no' response. During pairing, a user is shown a 6-digit number on each display and provides a 'yes' response on each device if the numbers match. Otherwise, the user responds 'no' and pairing will fail.

- **Passkey Entry:** It was designed for the situation where one Bluetooth device has input capability (e.g., Bluetooth-enabled keyboard), while the other device has a display but no input capability. In this model, the device with only a display shows a 6-digit number that the user then enters on the device with input capability.

- **Just Works:** It was designed for the situation where one (or both) of the pairing devices has neither a display nor a keyboard for entering digits (e.g., Bluetooth-enabled headset). The user is required to accept a connection without verifying the calculated value on both devices, so MITM protection is not provided.

- **Out-Of-Band:** It was designed for devices that support a wireless technology other than Bluetooth e.g., Near Field Communication (NFC) for the purposes of device discovery and cryptographic value exchange. It is important to note that the chosen OOB wireless technology should be configured to mitigate eavesdropping and MITM attacks to keep the pairing process as secure as possible.

Figure 1.1 shows the Bluetooth SSP with NC method. The six phases of SSP are explained below:

- **Capabilities Exchange:** The devices that have never met before or want to perform re-pairing for some reason, first exchange their Input/Output (IO) capabilities to determine the proper association model to be used.

- **Public Key Exchange:** The devices generate their public-private key pairs and send the public keys to each other. They also compute the Diffie-Hellman key.

- **Authentication Stage-1:** The protocol that is run at this stage depends on the association model. One of the goals of this stage is to ensure that there

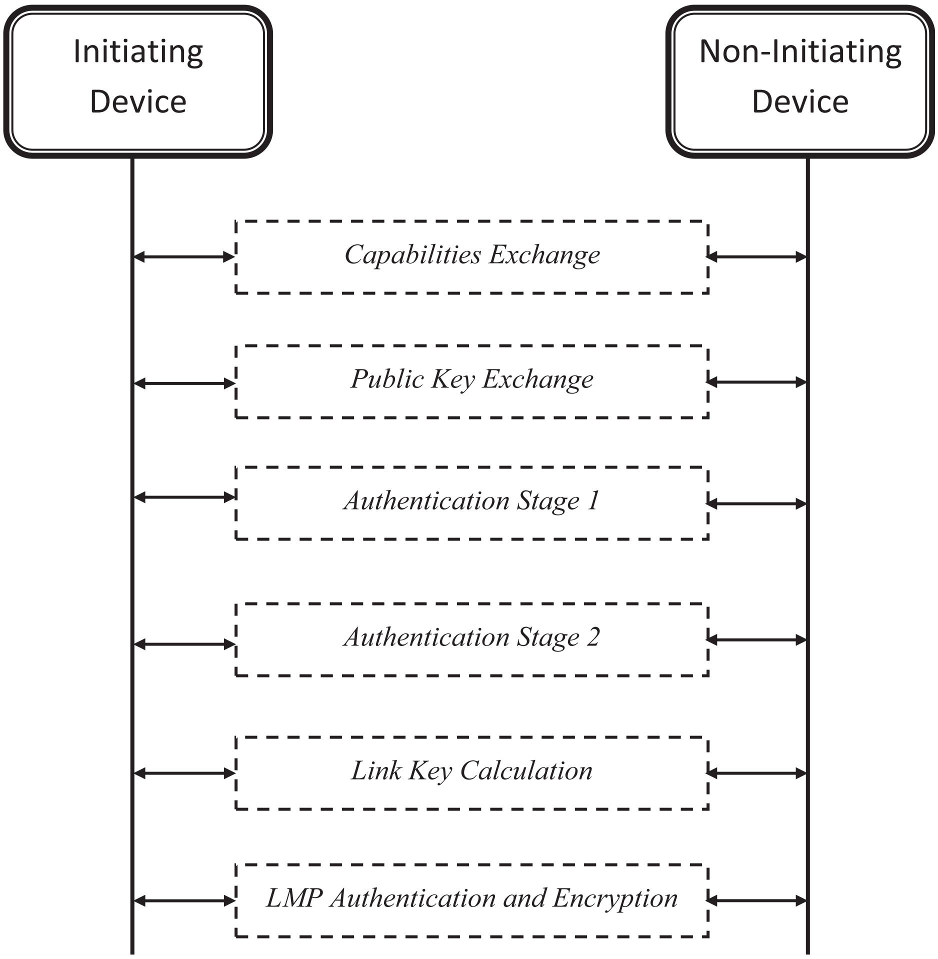


Figure 1.1: Bluetooth Secure Simple Pairing with Numerical Comparison

is no MITM in the communication between the devices. This is achieved by using a series of nonces, commitments to the nonces, and a final check of integrity checksums performed either through the OOB channel or with the help of user.

- **Authentication Stage-2:** The devices complete the exchange of values (public keys and nonces) and verify the integrity of them.

- **Link Key Calculation:** The parties compute the link key using their Bluetooth addresses, the previously exchanged values and the Diffie-Hellman key constructed in public key exchange phase.

- **Link Management Protocol Authentication and Encryption:** Encryption keys are generated in this phase, which is the same as the final steps of pairing in Bluetooth versions up to 2.0+EDR.

1.2 Why Bluetooth Security is Needed?

These days Bluetooth security is a big issue, all communication technology faces the problem of privacy and identity theft, with Bluetooth being no exception. The information and data we share through these communication technologies is both private and in many cases, critically important to us. So we can say that, providing security for bluetooth communication is very very important.

1.3 Bluetooth Security Modes

Cumulatively, the four security modes defined by various versions of Bluetooth specifications. Each version of Bluetooth supports not all, but some, of the four modes. Each Bluetooth device must operate in one of the four modes, which are described below [6].

- **Mode 1** : Security Mode 1 is non-secure. The basic security functionalities authentication and encryption are bypassed, leaving the device and connections susceptible to attackers. In effect, Bluetooth devices in this mode are promiscuous and do not employ any mechanisms to prevent other Bluetooth-enabled devices from establishing connections. Security Mode 1 is only supported in v2.0 + EDR (and earlier) devices.

- **Mode 2** : In Security Mode 2, a service level-enforced security mode, security procedures are initiated after LMP link establishment but before L2CAP channel establishment. L2CAP resides in the data link layer and provides connection-oriented and connectionless data services to upper layers. In this mode, the notion of authorization–the process of deciding if a specific device is allowed to have access to a specific service–is introduced. All Bluetooth devices can support Security Mode 2; however, v2.1 + EDR devices can only support it for backward compatibility with v2.0 + EDR (or earlier) devices.

- **Mode 3** : In Security Mode 3, the link level-enforced security mode, a Bluetooth device initiates security procedures before the physical link is fully established. Bluetooth devices operating in Security Mode 3 mandates authentication and encryption for all connections to and from the device. The authentication and encryption features are based on a separate secret link key that is shared by paired devices, once the pairing has been established. Security Mode 3 is only supported in v2.0 + EDR (or earlier) devices.

- **Mode 4** : Similar to Security Mode 2, Security Mode 4 (introduced in Bluetooth v2.1 + EDR) is a service level enforced security mode in which security procedures are initiated after link setup. SSP uses Elliptic Curve Diffie Hellman (ECDH) techniques for key exchange and link key generation. Security requirements for services protected by Security Mode 4 must be classified as one of the following: authenticated link key required, unauthenticated link key required, or no security required. Whether or not a link key is authenticated depends on the Secure Simple Pairing association model used.

1.4 Motivation

After the literature survey, it is found that MITM attacks are becoming the main problem in Bluetooth area networks. The MITM nodes are behaving like the original nodes and they can send/receive the valuable data. These MITM nodes can modify the data between the source and destination also. The attacks are based on the falsification of information sent during the input/output capabilities exchange and also the fact that the security of the protocol is likely to be limited by the capabilities of the least powerful or the least secure device type. The motivation is to achieve the solution for avoiding the MITM attacks in secure sample pairing method.

1.5 Problem Statement

The MITM first disrupts (jams) the physical layer (PHY) by hopping along with the victim devices and sending random data in every time slot. In this way, the MITM shuts down all piconets within the range of susceptibility and there is no need to use a Bluetooth chipset to generate hopping patterns. Finally, a frustrated user thinks that something is wrong with his/her Bluetooth devices and deletes previously stored link keys. After that the user initiates a new pairing process by using SSP, and the MITM can forge messages exchanged during the IO capabilities exchange phase. While using the SSP also, the MITM attacks are going to be possible by using the PHY jamming and falsification of information. The Figure 1.2 shows the problem of MITM attacks on physical layer of Bluetooth devices.

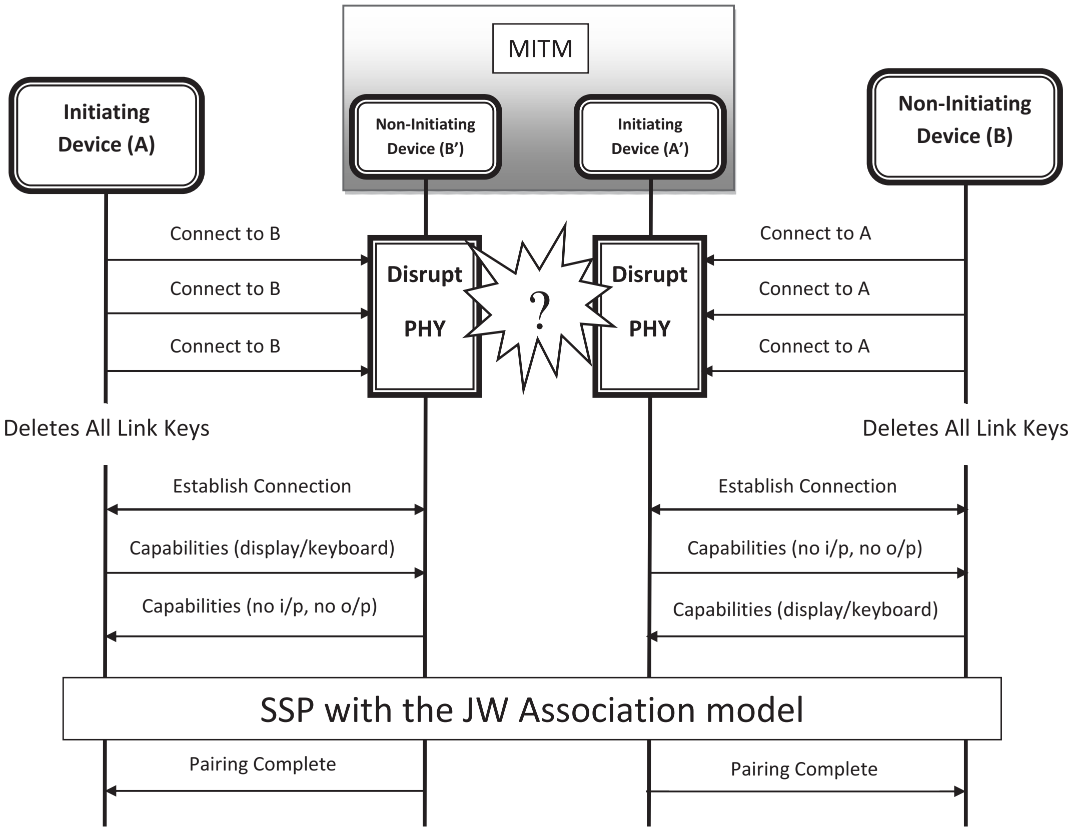


Figure 1.2: MITM Attack on Bluetooth SSP with JW Association

1.6 Thesis Organization

The rest of the thesis is organized as follows. The Bluetooth attacks and threats are summarized in Chapter 2. Chapter 3 provides existing countermeasures against all attacks and proposed countermeasure against MITM attack. Chapter 4 provides the concluding remarks.

Chapter 2

Bluetooth Attacks and Threats

Chapter 2

Bluetooth Attacks and Threats

Classification of threats can assist in finding threat severity, precautions, and its countermeasures. A Bluetooth Threat Taxonomy (Aboott) provides a framework for satisfying all threats. Abott consists of nine distinct classes [7] . Specifically, the Aboott classifications are surveillance, range extension, obfuscation, fuzzer, sniffing, denial of service (DoS), malware, unauthorized direct data access (UDDA), and MITM. Each attack appears in only one classification, based on its predominant characteristic, although a single attack can fall under several classifications.

2.1 Surveillance Attacks

Surveillance is an attack which is used to gather the information from the Bluetooth devices. These surveillance tools never cause the adverse effects to the target devices [7]. The threats under surveillance attacks are Blueprinting, bt_audit, Redfang, War-nibbling, Bluefish, Sdptool, BlueScanner and BTScanner.

2.1.1 Blueprinting

Blueprinting is designed for device fingerprinting. It uses the available services, device address, and other information to profile the interface, device, and host operating system. Attackers can use this service information to profile the device and get information on potential vulnerable vectors.

2.1.2 bt_audit

bt_audit scans all Protocol Service Multiplexers (PSMs) and RF Communication channels to determine if a target device has any undisclosed ports that could potentially lead to the discovery of unsecured services.

2.1.3 Redfang

Devices in non discoverable mode should be invisible, but according to prominent Bluetooth researcher Ollie Whitehouse of IT consultancy, that's not the case. Whitehouse has designed a software tool called RedFang that can discover Bluetooth devices that have been set to be non discoverable. He also says, RedFang tries to "brute-force the entire Bluetooth address space asking for a device's name",and if a legitimate name is found, even devices in nondiscoverable mode can be seen. Once the devices are discovered, they become exposed to threats such as bluesnarfing [8].

2.1.4 War-nibbling

War-nibbling can search the Bluetooth enabled devices in a particular location. By using this, the attacker can know all the profiles of the Bluetooth enabled devices in a particular area.

2.1.5 Bluefish

Bluefish takes surveillance of Bluetooth devices one step further. When a Bluefish system detects a new device, it records the Bluetooth device information and takes a photograph in the distrusted direction of the device. Each time the device reenters the range of the Bluefish running computer, the process is repeated.

2.1.6 Sdptool

Sdptool is a Service Discovery Protocol (SDP) tool, which provides the interface for performing SDP queries on Bluetooth devices, and administering a local *sdp daemon* [9].

2.1.7 BlueScanner

BlueScanner is designed for easy to use. Simply press a button when scanning a bar code. The scanned bar codes can be either send to an SMS gateway or send to an application running on the Mobile Equipment (ME) itself. BlueScanner uses Bluetooth technology for connecting with an ME like a mobile phone. BlueScanner supports service level security in Bluetooth connectivity, multiple operating modes, and audio-visual indications for the status of the BlueScanner [10].

2.1.8 BTScanner

BTScanner is designed to extract information from a Bluetooth device without having to pair with it, meaning that it operates noninvasively and therefore, invisibly. Hurman notes that the current version of btscanner can only find information about discoverable devices (such as channel information and a list of services running); but if those devices are discovered using Redfang, for example, then btscanner can learn enough about them to provide a potential weak point to a determined attacker [8].

2.2 Range Extension Attacks

The range of any wireless device will be limited to some extent. Extending a device's range might be against US Federal Communication Commission (FCC) rules, but attackers can use it to conduct attacks from a distance [7]. The threats under range extension threats are BlueSniping, Bluetooone and VERA-NG.

2.2.1 BlueSniping

BlueSniping is the term given to somebody who uses a high gain aerial connected to a computer to steal information from a bluetooth device. It is rarely found due to the nature of the equipment, but was first found to exist in Sweden [11]. BlueSniping has emerged as a method for BlueSnarfing, or simply identifying Bluetooth-enabled devices, at longer ranges than normally possible. [12].

2.2.2 Bluetooone

Bluetooone is the method involves attaching a high-gain antenna to the standard Bluetooth radio to extend ranges from meters to kilometers.

2.2.3 VERA-NG

The Very Eccentric Radio frequency Antenna - Nerf Gun (VERA-NG) combines the greatness of Nerf gun office antics with high-gain wireless auditing technology. VERA-NG is built on a Nerf Long shot CS-6, which has been equipped with two antennas (high-gain antennas made from Pringles cans) attached to both Bluetooth and Wi-Fi USB adapters. This allows for auditing of these wireless technologies from a discrete distance. VERA-NG also includes an ultra portable tablet and GPS. It is well equipped for long range wireless sniffing, mapping an area for Bluetooth and Wi-Fi devices, and shooting Nerf darts long range [13].

2.3 Obfuscation Attacks

Attackers can use obfuscation to achieve a level of anonymity for launching an attack [7]. The threats under Obfuscation threats are Bdaddr, Hciconfig and Spooftooph.

2.3.1 Bdaddr

By modifying the Bluetooth interface's firmware, the bdaddr can change addresses of the devices on certain Bluetooth chipsets. By permanently resetting the interface device address, bdaddr nullifies the assumption of the device address as a unique identifier.

2.3.2 Hciconfig

By using Hciconfig application, the users can change most of its publicly provided Bluetooth information, including their class and name. If we use this in combination with bdaddr, hciconfig lets attackers clone device addresses, names, and classes, hereby letting a laptop mask itself as a cell phone, automobile, mobile headset, and so on.

2.3.3 Spooftooph

Spooftooph simplifies the process by automatically scanning for devices in range and cloning their Bluetooth device information according to the user's selection.

2.4 Fuzzer Attacks

Fuzzing is a technique used to test application input handling. Fuzzers operate by submitting nonstandard input to an application to achieve malicious results. The threats under Fuzzer threats are BluePass, Bluetooth Stack Smasher, BlueSmack, Tanya and BlueStab [7].

2.4.1 BluePass and Bluetooth Stack Smasher

These are tools for assembling and sending packets to a target device. They help craft packets that test an application's ability to handle standard and nonstandard input data.

2.4.2 BlueSmack

BlueSmack uses a Logical Link Control and Adaptation Protocol (L2CAP) echo request, similar to an Internet Control Message Protocol (ICMP) ping. An attacker can abuse the echo request by changing its size to 600 bytes or greater.

2.4.3 Tanya

The Tanya exploit tool crafts and sends maliciously formed L2CAP quality of service messages to degrade Bluetooth service on a mobile device. The tool repeatedly sends the same maliciously crafted message to a target device and limits the throughput of the vulnerable device by forcing the device to continuously respond. Similar tools include BlueSmack and Symbian Remote Restart; both which take advantage of the L2CAP quality of service messages to overwhelm vulnerable devices [14].

2.4.4 BlueStab

BlueStab uses bad names to crash devices engaged in Bluetooth discovery address [15].

2.5 Sniffing Attacks

Sniffing is just like eavesdropping on a phone line, the process of capturing traffic in transit. Because Bluetooth broadcasts traffic wirelessly over RF, it's vulnerable to outside monitoring on specific frequencies [7]. The types of sniffers are FTS4BT, Merlin, BlueSniff, HCIDump, Wireshark and Kismet.

2.5.1 FTS4BT and Merlin

These are the two commercially available Bluetooth sniffers. These tools combine specialized hardware and software to monitor Bluetooth traffic by matching the connection's frequency hops and then capturing data in that frequency range. They log the sniffed data to a local file, which users can later view and analyze.

2.5.2 BlueSniff

Bluetooth uses frequency hopping over 79 channels in order to minimize interference and (usually) hops once every 625s, sending one packet per channel. The hopping sequence is determined by the MAC address of the master device and its clock. The master device is the one that initiates the connection, and the slave being the one connected to [16]. BlueSniff is the process of using a modified Universal Software Radio Peripheral (USRP2) motherboard to monitor all 79 channels at the same time. It monitors each channel's traffic as binary data and reassembles the data into standard Bluetooth traffic for further analysis [7].

2.5.3 HCIDump

This is a utility that can capture and read raw Bluetooth traffic by monitoring local Bluetooth interfaces and capturing data from sniffed traffic. This tool assists attackers in discovering weaknesses in protocols and services.

2.5.4 Wireshark

Wireshark is also a powerful wireless security analysis tool. Using Wireshark™displ-ay filtering and protocol decoders, it can easily shift through large amounts of wireless traffic to identify security vulnerabilities in the wireless network, including weak encryption or authentication mechanisms, and information disclosure risks.It can also perform intrusion detection analysis to identify common attacks against wireless networks while performing signal strength analysis to identify the location of a station or access point (AP) [17].

2.5.5 Kismet

Kismet is a computer program that allows for passive detection of wireless local area network. It enables sniffing and has some features of Intrusion Detection Schemes (IDS) for 802.11 networks. Kismet works with any Wi-Fi cards, but card must support monitor mode. Kismet allows you to capture frames, the second layer of 802.11b, 802.11a and 802.11g [18].

2.6 DoS Attacks

This is an attack on making the services unavailable. These attacks often target communication channels, but they can relate to any service the device uses, including the processor, memory, disk space, battery life, and system availability [7]. The threats under DoS threats are battery exhaustion, signal jamming, BlueSYN, Blueper, BlueJacking and vCardBlaster.

2.6.1 Battery Exhaustion

In battery exhaustion attack, the attacker keeps on sending some unnecessary data to the legitimate users; For receiving those data packets the devices lose their total energy.

2.6.2 Signal Jamming

In signal jamming attack, the attacker can send the signal to the legitimate devices, at the same time of signal transmission to them. So, by doing this, the jammer's signal and the legitimate user's signal will present in collision and the signal will not be transmitted. The legitimated users think that, there may be a network jam but it is not the real network jam. There are four types jammers [19, 20]. They are,

- **Constant Jammer:** A jammer continually emits radio signals on the wireless medium. The signals can consist of a completely random sequence of bits.

- **Deceptive Jammer:** It is similar to the constant jammer. Their similarity is due to the fact that both constantly transmit bits. The main difference is that with the deceptive jammer, the transmitted bits are not random. The deceptive jammer continually injects regular packets on the channel without any gaps between the transmissions.

- **Random Jammer:** An attacker employing random jamming, jams for t_j seconds and then sleeps for t_s seconds. During the jamming intervals, this jammer can follow any of the approaches of other jammers.

- **Reactive Jammer:** This jammer is constantly senses the channel and upon sensing a packet transmission, immediately transmits a radio signal in order to cause a collision at the receiver.

2.6.3 BlueSYN

Simultaneously attacking the device with a hping2 SYN flood, affecting the Wi-Fi interface, and a l2ping BlueSmack flood, affecting the Bluetooth interface, demonstrates a blended attack that attempts to saturate multiple communication vectors. The SYN flood propagated through a wired LAN to an access point before finding the target device, while the Bluetooth portion of the attack was launched from a Bluetooth adapter on the notebook computer directly against the targeted device. This previously undocumented attack was named the BlueSYN DoS Attack [21].

2.6.4 Blueper

Blueper is designed for mobile devices to abuse Bluetooth file transfer. It floods the target with file transfer requests. This tool sends continuous stream of pop-up messages for file transfer requests to the target. A more detrimental result is data written to a target device disk without user interaction or previous authentication, causing some devices to temporarily halt execution or crash.

2.6.5 BlueJacking

Bluejacking is usually harmless, but because blueJacked people generally don't know what has happened, they may think that their phone is malfunctioning. Usually, a blueJacker will only send a text message, but with modern phones it's possible to send images or sounds as well. With the increase in the availability of Bluetooth enabled devices, it is often reported that devices have become vulnerable to virus attacks and even complete take over of devices through a trojan horse program although most of these reports are easily debunked. Bluejacking is also confused with Bluesnarfing which is the way in which mobile phones are illegally hacked via Bluetooth [22]. Bluejacking is a technique for abusing the vCard feature on mobile phones.

2.6.6 vCardBlaster

By accepting vCards often requires no interaction on the receiver's end, opening a way for attackers to send anonymous messages without any credentials. The attack can be used to frighten users with suspicious-looking messages on their mobile devices.

2.7 Malware Attacks

Malware is a malicious form of software, often self replicating, that carries out various activities such as data mining, accessing personal files, password theft, file corruption, and system reconfiguration. Commonly known Malware subsets include viruses, worms, and Trojan horses [7]. The threats under Malware threats are BlueBag, Caribe and CommWarrior.

2.7.1 BlueBag

This vulnerability permits access to the cell phone's set of AT commands, which let an aggressor use the phone's services, including placing outgoing calls, sending, receiving, or deleting Short Message Service (SMS), diverting calls, and so on [23].

2.7.2 Caribe and CommWarrior

By using these two, the worms propagate though Bluetooth communication, infecting cell phones running Symbian OS. The user of targeted device receives a message to accept the incoming file. Based on the worm file type, once downloaded, the worm can bypass the normal user prompt for execution, installs itself in hidden directories on the host device, and set itself to autorun. It then begins to search for Bluetooth devices in range and propagates itself.

2.8 UDDA Attacks

UDDA attacks gather private information for unauthorized entities by penetrating devices through loopholes in security, allowing unauthorized access to privileged information [7]. The threats under UDDA attacks are Bloover, BlueSnarf, BlueBug, BlueSnarf++,BTCrack, btpincrack, Car Whisperer and HeloMoto.

2.8.1 Bloover

. Bloover is a tool which can launch an attack on mobile phones for reading phone books, writing phone book entries, reading or decoding the SMS stored on the device, setting call forward to a particular number and also initializing phone call. These are the services provided by the Bloover tool. This tool is not working well in Nokia phones [24].

2.8.2 BlueSnarf

This type of attack uses the OBEX (OBject EXchange) push service, which is commonly used to exchange files such as business cards. BlueSnarf allows an attacker to access the vulnerable device's phone book and calendar without authentication.

A recently upgraded version of this attack gives the attacker full read and write access [23].

2.8.3 BlueBug and BlueSnarf++

These facilitates unauthorized access to certain cell phone models, letting attackers view contacts, text messages, pictures, call records, and so on. They can also send a command to a victim device on a covert channel, thus avoiding user detection. UDDA attacks also use phone features such as short message service (SMS), Internet connection, and telephony to gain complete control of a device through its Bluetooth connection. The attacker is then free to place phone calls, copy contact lists, and reconfigure call forwarding.

2.8.4 BTCrack and btpincrack

These tools uses a brute-force method to crack the PIN. They capture packets in the pairing process and compare them with attacker-crafted packet parameters, which they generate by enumerating PINs for encrypting standard packet content. The time it takes to break a PIN is directly proportional to its length.

2.8.5 Car Whisperer

Car Whisperer automates the access to Bluetooth-enabled devices with default settings by guessing the default PIN. Once connected, the attacker can extract audio from or inject it into the target device.

2.8.6 HeloMoto

HeloMoto is a combination of BlueSnarf and BlueBag. The name of attack comes from the fact that it was originally discovered on Motorola phones [23].

2.9 MITM Attacks

MITM attacks place an attacking device between two connected devices to act as a relay (the attacker uses obfuscation to hide the attacking device). Previously paired devices send their information to the attacking device, which then relays it to its intended destination [7]. The threats under MITM attacks are BT-SSP-Printer-MITM, BlueSpooof and bthidproxy.

2.9.1 BT-SSP-Printer-MITM

The BT-SSP-Printer-MITM attack shows possible vulnerabilities in the newer Bluetooth standards. This attack focuses on the JW connection option in four association models of SSP, which lets devices pair without authentication. The BT-SSP-Printer-MITM attack sets the attacker's device as a relay point between the user's device and a printer. When the user device connects to the printer using the JW method, the attacker breaks the connection by using some form of DoS.

2.9.2 BlueSpooof

By BlueSpooof tool, The attacker can act as another Bluetooth device by using its BT address [15].

2.9.3 Bthidproxy

Bthidproxy is yet another handy piece of software. Using it MITM attack can be possible on Bluetooth connections by using two dongles and spoofing the host and device addresses. Because of virtual cabling, a one to one connection is made between device and host. This means that almost all attacks must be performed when either the device or host are off allowing anyone to take their place. This is not too much of a problem since machines get powered down often and many mice have off switches to save battery [25].

2.9.4 History of MITM Attacks

The First MITM attack on Bluetooth assumes that the link key used by two victim devices is known to the attacker was devised by Jakobsson and Wetzel [2]. This attack works for the version 1.0B and as well as all versions upto 2.0+EDR, because of no security improvements were implemented in those specifications. The authors also showed how to obtain the link key using offline PIN crunching, by passive eavesdropping on the initialization key establishment protocol.

By manipulating with the clock settings, the attacker forces both victim devices to use the same channel hopping sequence but different clocks. This is an improvement of the attack of [2] by Kugler [3]. In addition, Kugler shows how a MITM attack can be performed during the paging procedure. The attacker responds to the page request of the master victim faster than the slave victim, and restarts the paging procedure with the slave using a different clock.

Reflection (relay) attacks [4] aim at impersonating the victim devices. The attacker does not need to know any secret information, because she only relays (reflects) the received information from one victim device to another during the authentication.

The versions 2.1+EDR and 3.0+HS of Bluetooth provide protection against the MITM attacks described above, by the means of SSP. However, it has been shown that MITM attacks against Bluetooth 2.1+EDR and 3.0+HS devices are also possible [5,26–29]. Because SSP supports several association models, selection of which depends on the capabilities of the target devices, the attacker can force the devices into the use of a less secure mode by changing the capabilities information.

Haataja and Toivanen [1] proposed two new MITM attacks on Bluetooth SSP. The first attack is based on the falsification of information sent during the IO capabilities exchange. The second attack requires some kind of visual contact to the victim devices in order to mislead the user to select a less secure option instead of using a more secure OOB channel. Now the situation has been changed— Bluetooth devices with an adjustable Bluetooth device addresses are readily available and techniques for finding hidden (non-discoverable) Bluetooth devices have been invented. Therefore, the danger of MITM attacks has recently increased.

Table 2.1 shows the Bluetooth connection methods and the possibility of the MITM attacks on those methods.

MITM attacks can be possible on these Bluetooth connection methods— (i) SSP with just works, (ii) if one of the devices does not have IO devices or the MITM impersonates as legitimate user and tells no-input and no-output as its capabilities to connect and (iii) by creating Jam in PHY when legitimate users know each

Table 2.1: The Bluetooth connection methods and possibility of the MITM attacks

Sl. No.	Bluetooth Connection Methods	Possibility of MITM Attacks
1	SSP with Just Works	YES
2	SSP-OOB as mandatory	NO
3	SSP- Numeric comparison with both devices have IO capabilities.	NO
4	One of the devices does not have IO devices or the MITM impersonates as legitimate user and tells "no-input and no-output" as its capabilities to connect.	YES
5	By creating Jam in PHY when legitimate users know each other.	YES
6	By using RF fingerprints as Keys	NO
7	By Adding an additional window at the user interface level	NO

other. The Solutions to the above methods are— (i) by not allowing the devices for the JW option association model (the users should have key sharing) OR by allowing the devices by adding an additional window at the user interface level, (ii) OOB as a mandatory association model (i.e., the communication will be very secure by using near field communication like infrared) and (iii) by using one of Anti-Jamming techniques like frequency hopping, direct sequence spread spectrum and uncoordinated spread spectrum. The various jammers used for jamming the physical layers of Bluetooth devices are— constant jammer, deceptive jammer, random jammer, reactive jammer [19, 20]. Possible solutions to the attacks which are presented in Table 2.1 are given in Table 2.2.

Table 2.2: The possible solutions to the attacks which are presented in Table 1

Sl. No.	Problems	Solutions
1	SSP with Just Works	By not allowing the devices for the JW option association model (the users should have key sharing) OR by allowing the devices by adding an additional window at the user interface level.
2	One of the devices don't have IO devices OR The MITM impersonates as legitimate user and tells "no-input and no-output" as its capabilities to connect	OOB as a mandatory association model (i.e., the communication will be very secure by using near field communication like infrared)
3	By creating Jam in PHY when legitimate users know each other	By using one of Anti-Jamming techniques like frequency hopping, direct sequence spread spectrum and uncoordinated spread spectrum

2.10 Summary

In this chapter, all kinds of the attacks and threats have been seen which can be possible to attack on devices having Bluetooth connectivity.

Chapter 3

Countermeasures

Chapter 3

Countermeasures

A countermeasure is a measure or action taken to counter or offset another one. As a general concept it implies precision, and is any technological or tactical solution or system (often for a military application) designed to prevent an undesirable outcome in the process. To improve the security of the Bluetooth communication, we need to follow some countermeasures. The set of countermeasures are classified into three categories. They are for user, manufacturer and specification [7].

3.1 For User

There are 14 different countermeasures which have to follow by the user to avoid maximum number of the attacks discussed in the Chapter 2. The countermeasures are listed below [7].

3.1.1 Disabling Bluetooth when not in use

Bluetooth is often used for short-term inter device connections. When not in use, the best defense against attacks is to disable Bluetooth through hardware or software controls.

3.1.2 Disabling unused services

Many systems let users specify which services to enable/disable. For example, users might want to enable the audio gateway on a mobile phone but disable file transfer.

3.1.3 Placing Bluetooth devices in non-discoverable mode when not pairing

A device should only be discoverable during initial pairing. Afterwords, devices will be able to locate each other without being in discoverable mode. Devices in non-discoverable mode are much more difficult for an attacker to find.

3.1.4 Placing Bluetooth devices in security mode 2, 3, or 4, requiring authentication and encryption for communication

This often involves selecting a setting option such as "enable encryption" or "authentication required". These settings help prevent connection from unauthorized devices and make it more difficult to extract data from sniffed traffic.

3.1.5 Avoiding using JW

The JW association model doesn't protect against MITM attacks. It also facilitates device connections without any form of authentication.

3.1.6 Use alphanumeric PINs, 12 digits or greater in length

This helps prevent brute-force password guessing and makes it almost impossible for attackers to extract the password from cracking attempts on sniffed traffic.

3.1.7 Never accepting files or messages from untrusted devices

Files and messages can carry attacks against a device. Attackers can easily spoof the device name, so it's best to use a second factor of verification, such as a verbal conversation, before accepting a connection.

3.1.8 Never accepting pairing with untrusted devices

So many services are available on Bluetooth that it can be difficult to determine what users are agreeing to when a message is presented for action. Pairing is also permanent unless partnerships are later deleted. Pairing with an untrusted device can provide access to all Bluetooth services enabled on the local device.

3.1.9 Changing PINs semi frequently

This is a good practice with any form of authentication. Most Bluetooth authentication occurs just once, so changing PINs can help prevent previously trusted devices from regaining access to a device without user notification.

3.1.10 Using an additional window at the user interface level

It is recommended that an additional window, “The second device has no display and keyboard! Is it true?” should be displayed at the user interface level of SSP when the JW association model is to be used. Then the user is asked to choose “PROCEED” or “STOP”.

3.1.11 SSP-OOB as mandatory

Future Bluetooth specifications should make OOB a mandatory association model in order to radically improve the security and usability of SSP. Therefore, future Bluetooth specifications should at least strongly recommend the use of an OOB channel (e.g., NFC) to all Bluetooth device manufacturers.

3.1.12 Using RF fingerprints

There will be difference between the signals sent by the devices which are manufactured by the same company. so we can use the RF fingerprints can be used for identification.

3.2 For Manufacturer

There are some countermeasures which should be followed by the manufacturer instead of user. Because those are not able to done by the user. The countermeasures which need to follow by the manufacturer are listed bellow [7].

3.2.1 Making input validation a high priority during development of Bluetooth related tools

This basic principle applies to all software development. Software relating to the use of Bluetooth should be rigorously tested to prevent buffer overflows and illegal directory traversals.

3.2.2 Disabling all unnecessary Protocol Service Multiplexers (PSM) and RFComm channels

Closing all unused PSMs and RF Communication channels helps prevent attackers from gaining access to standard device services and back doors left open from testing.

3.2.3 Disregarding traffic not formatted to Bluetooth specification

By ignoring the traffic which is not in Bluetooth specification format, It can prevent fuzzing and enforce the Bluetooth standards.

3.2.4 Testing all products with applicable hacking tools for vulnerabilities

Using the tools such as those discussed in this article can help reveal vulnerabilities during production before the product goes on the market.

3.3 For Specification

The countermeasures which need to follow by the manufacturer offers two-factor authentication [7].

3.3.1 Offering two-factor authentication

As initial authentication often occurs only once, so second factor of authentication is warranted for devices that might have multiple users or be at risk for theft. This second form of authentication could be required for each pairing and/or service use.

3.4 Proposed Countermeasure

The proposed approach is like that— while one of the initiating or non-initiating devices is trying to connect with each other, the attacker sends wrong signals which leads to the corruption of the original signal. So, the legitimate users think that, there maybe some sort of genuine jam in the network or there may be some sort of jamming attack. Now instead of deleting all the information about the previously connected devices, the legitimate user needs to run IDS. If there is a presence of any jammer, then by running the IPS, It can be found a way to avoid the jammer and there is no need to delete the keys. After that, the process of SSP with NC association will be followed for the secure communication. The Prevention schemes of PHY layer are also called anti-jamming techniques. The algorithmic representation of the proposal is given in Algorithm 1. The Figure 3.1 shows the countermeasure against MITM attack. The algorithmic representation of the process of SSP with NC association is given in Algorithm 2 and Algorithm 3.

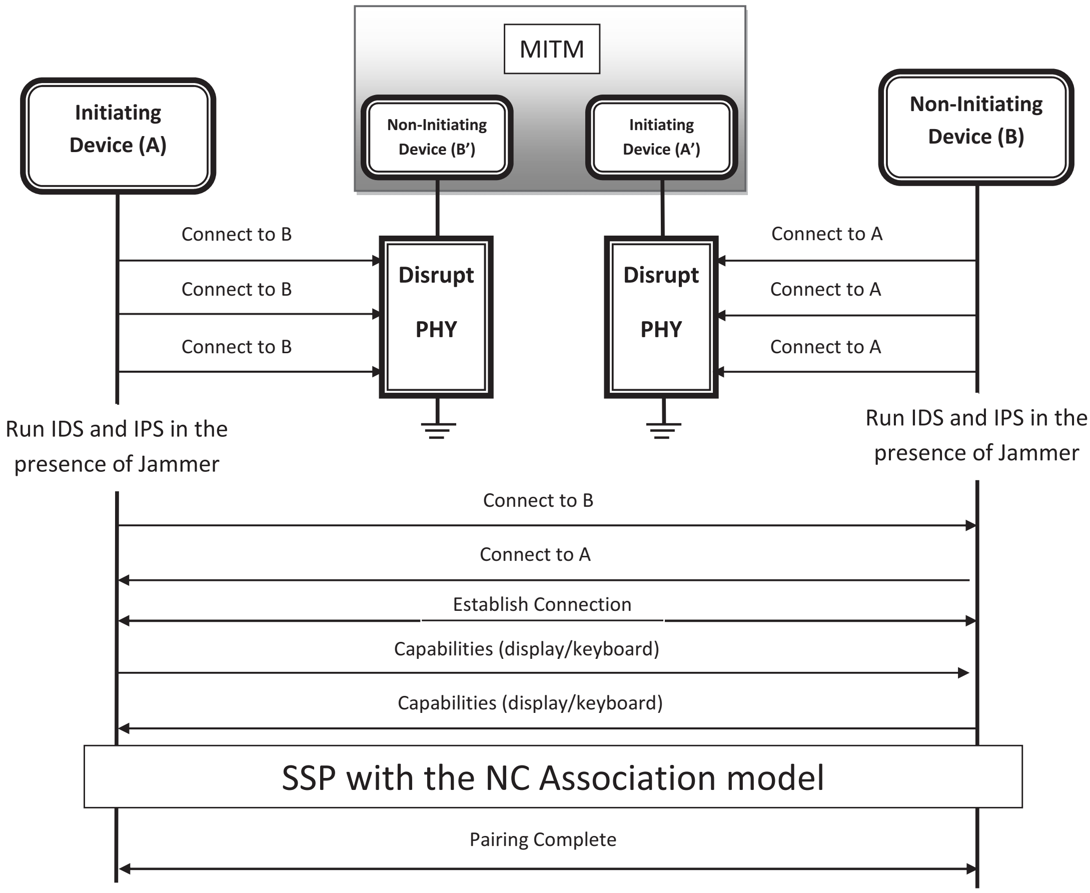

MITM
Initiating Device (A)
Non-Initiating Device (B')
Initiating Device (A')
Non-Initiating Device (B)
Connect to B
Connect to B
Connect to B
Disrupt PHY
Disrupt PHY
Connect to A
Connect to A
Connect to A
Run IDS and IPS in the presence of Jammer
Run IDS and IPS in the presence of Jammer
Connect to B
Connect to A
Establish Connection
Capabilities (display/keyboard)
Capabilities (display/keyboard)
SSP with the NC Association model
Pairing Complete

Figure 3.1: Countermeasure to MITM Attack

Algorithm 1 Proposed Connection Establishment Algorithm between Users

```
REQUIRE Assumptions
Both the users Already Communicated Before
Both the Users Knows the Address of Each Others
Send Signals for Establishment of the Connection
if Basic Connection Established then
    Run SSP with NC Association Algorithm
else
    if Found Network Jamming then
        Run IDS Schemes
        if Found the Attacker then
            Run IPS Schemes
            Run SSP with NC Association Algorithm
        else
            Delete All Link Keys and Go for a Fresh Connection
        end if
    end if
end if
```

Algorithm 2 SSP with NC Association (Part 1)

```
REQUIRE Notations
p : large prime number
g : generator of order p-1
IOcapX : Input and Output capabilities of user X
SKx : Private Key of User X
DHKey : Diffie-Hellman Key
Nx : Nonce generated by the user X
rx : Random number generated by user X for this algorithm it is set to 0
Cx : Commitment value from user X
Vx : verification value from user X
fun1 : One-way function used to compute commitment
fun2 : One-way function used to compute numeric check values
fun3 : One-way function used to compute check values
LK : Link Key
BD_ADDRx : 48-bit Bluetooth address of device X
Step1 :
$userA$ : Send the IOcapA to $userB$
$userB$ : Send the IOcapB to $userA$
$UserAandB$ :
if both users have both input and output capabilities then
    goto step 2
else
                                    ▷ For some reason we need to break here!
```

Algorithm 3 SSP with NC Association (Part 2)

```
    end the connection
end if
Step 2 :
$UserA$ : enters a private key $SKa$
$PKa$= pow($g$,$SKa$) mod $p$ , send the $PKa$ to $userB$
$UserB$ : enters a private key $SKb$
$PKb$=pow($g$,$SKb$) mod $p$ , send the $PKb$ to $userA$
$UserA$ : $DHKey$ = pow($PKb$,$SKa$) mod $p$
$UserB$ : $DHKey$= pow($PKa$,$SKb$) mod $p$
Step 3 :
$UserA$ : Generate a random number $Na$ and set $ra$ to 0, send $Na$ to $userB$
$UserB$ : Generate a random number $Nb$ and set $br$ to 0
$Cb$=$fun1(PKb, PKa, Nb, 0)$ ,send $Cb$ and $Nb$ to $userA$
$UserA$ :
if $Cb == fun1(PKb, PKa, Nb, 0)$ then
    $Va = fun2(PKa, PKb, Na, Nb)$ , send $Va$ to $userB$
end if
$UserB$ : $Vb = fun2(Pka, Pkb, Na, Nb)$ , send $Vb$ to $userA$
$UserAandB$ :
if $Va == Vb$ then
    goto step 4
else
    end the connection
end if
Step 4 :
$UserA$ : $Ea = fun3(DHKey, Na, Nb, 0, IOcapA, A, B)$ and send $Ea$ to $userB$.
$UserB$ : $Eb = fun3(DHKey, Na, Nb, 0, IOcapB, B, A)$
if $Ea = fun3(DHKey, Na, Nb, 0, IOcapA, A, B)$ then
    send $Eb$ to $userA$
else
    end the connection
end if
$UserA$ :
if $Eb = fun3(DHKey, Na, Nb, 0, IOcapB, B, A)$ then
    goto step 5
else
    end the connection
end if
Step 5 :
$UserAandB$ : $LK = fun4(DHKey, Na, Nb, “btlk'', BD_ADDRa, BD_ADDRb)$
Step 6 :
$UserAandB$ : Use $LK$ for encryption and decryption of data.
```

3.4.1 Using Intrusion Detection Schemes

The traditional techniques of IDS of wired line networks are directly taken to Bluetooth communication also. Table 3.1 shows the types of IDS. With these IDS listed in Table 3.1, one can be able to detect all types of jammers and overcome the problem of distinguishing between network dynamics and jamming attacks. However, there are still open issues. For example, the frequency of the location advertisements can significantly affect the performance of the location consistency check system. In addition, wireless propagation effects (e.g., Fading) should be taken into consideration for accurately computing the false alarm rate of the IDS.

Table 3.1: Discoverability of various jammers using different IDS

Sl. No.	Intrusion Detection Schemes [19, 20, 30]	Constant Jammer	Deceptive Jammer	Random Jammer	Reactive Jammer
1	Signal Strength Measurements	Yes	Yes	No	No
2	Carrier Sensing Time	Yes	Yes	No	No
3	Measuring the Packet Delivery Ratio (PDR)*	Yes	Yes	Yes	Yes
4	Consistency Checks**	Yes	Yes	Yes	Yes

* PDR measurements can not always distinguish between jamming and network failures and/or poor link conditions.
** Consistency Checks introduce two detection techniques:
(a) Signal Strength Consistency Check
(b) Location Consistency Check

3.4.2 Intrusion Prevention Schemes

There are 5 schemes. The Simple PHY Layer Techniques ans Directional Antennas do not perform any processing of the transmitted signal while, The rest of the schemes perform processing of transmitted signal.

Simple PHY Layer Techniques

By reducing the distance between legitimate transceiver pair or by increasing the transmission power, we can reduce the jamming-to-signal ratio and make the link more robust to jamming attacks.

Directional Antennas

Jamming interference coming from directions other than the direction of transmission does not stimulate transmission deferrals due to carrier sensing. [31]

Spread Spectrum

The most well known techniques are based on the use of Spread Spectrum communications. Here signal processing techniques used as jamming countermeasures. [32]

Cyber Mines and FEC (Forward Error Correction)

Low energy long-lived jamming units are called cyber-mines. For handing these there are some methods like Low Density Parity Codes (LDPC) and Turbo-Codes etc. [33, 34]

Use of covert channels in the presence of a jammer

When the reception of a packet is affected by jammer, the receiver can identify the reception of a (corrupted) packet. By encoding data based on the inter-arrival times between received corrupted packets, a low rate channel under jamming can be established. [35, 36]

3.4.3 Security Services

The main security requirements are Integrity ,Authenticity and Confidentiality.

Integrity

Integrity means that data cannot be modified undetectably. In our proposed method, the MITM attacker is not able to launch the attack because we are using IDS and IPS methods to detect and prevent the attacker. So we can say that, Integrity exists.

Authenticity

Authenticity is the process of validation whether the two parties which are communicating are genuine or not. In our proposed method, we are using Diffie/Hellman key exchange in second stage of SSP process and also there are two authentication stages. So we can say that, the authenticity exists.

Confidentiality

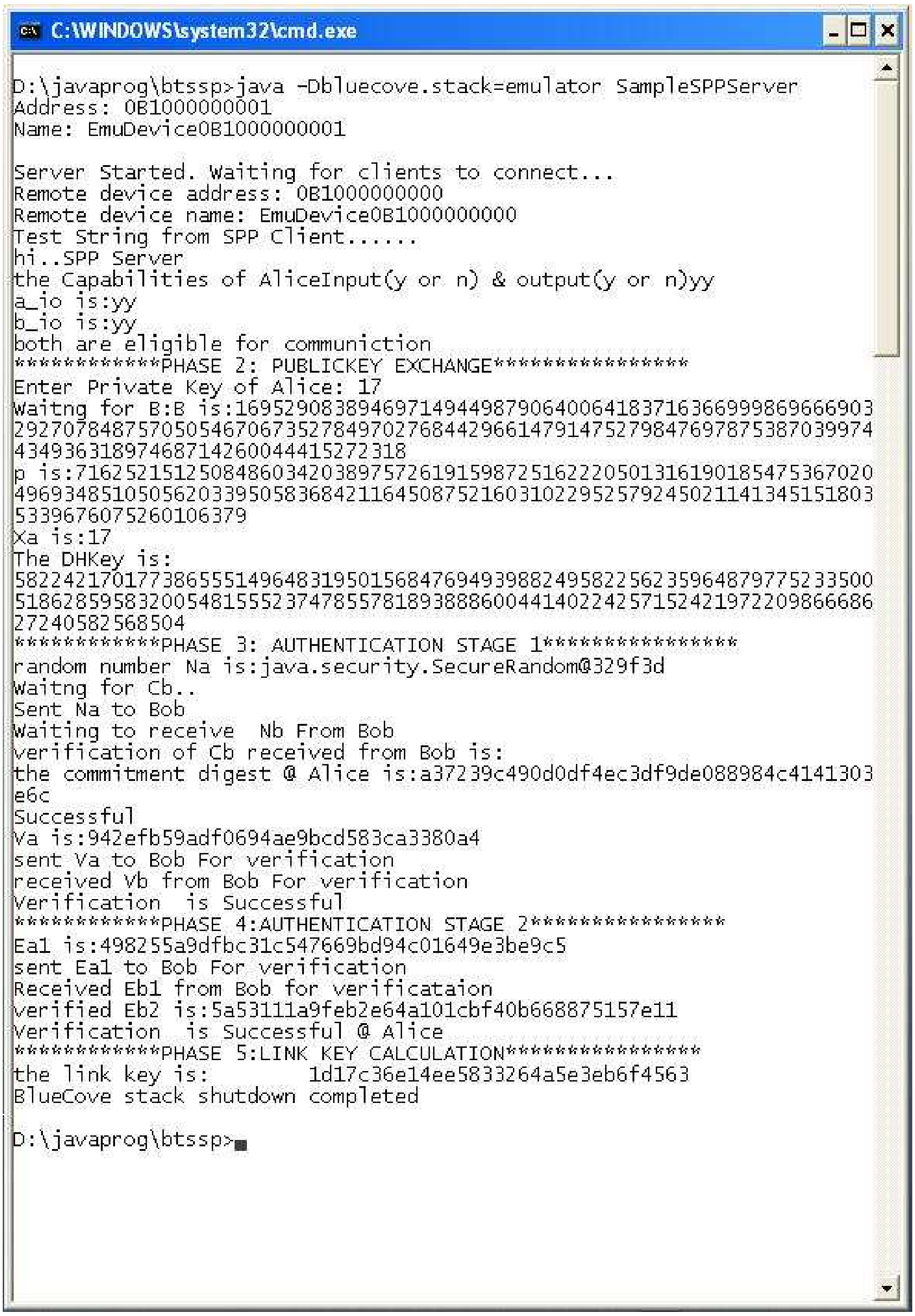
Confidentiality is the term used to prevent the disclosure of information to unauthorized individuals or systems. In our proposed method, we are encrypting the message or data which is going to be transfered with a link key which will be known to only the communicating parties. Even though some one can intercept, but he can not decode. So we can say that, the confidentiality exists.

3.4.4 Simulation Details

BlueCove [37] is a Java library for Bluetooth (JSR-82 implementation) that currently interfaces with the Mac OS X, WIDCOMM, BlueSoleil and Microsoft Bluetooth stack found in Windows XP SP2 or Windows Vista and WIDCOMM and Microsoft Bluetooth stack on Windows Mobile. BlueCove-GPL is additional GPL licensed module to support BlueCove runtime on Linux BlueZ. BlueCove JSR-82 Emulator module is additional module for BlueCove to simulate Bluetooth stack. BlueCove can be used in Java 2 Platform, Standard Edition (J2SE) 1.1 or newer. **bluecove-emu** is additional module for BlueCove to simulate Bluetooth stack. **bluecove-emu** is a pure Java implementation of JSR-82 without Bluetooth hardware. Fully tested using TCK JSR-82 TCK test results.

3.4.5 Results & Discussions

Java is a secure programming language, so it is not feasible to create adversary node using java. To do that we need an additional hardware support. So we have simulated the proposed countermeasure partially. The simulation of successful Secure Simple Pairing method is shown in Figures 3.2 and 3.3. The Figure 3.2 is the non-initiating device's execution and Figure 3.3 is the initiating devices execution. The simulation of unsuccessful Secure Simple Pairing method is shown in Figure 3.4.

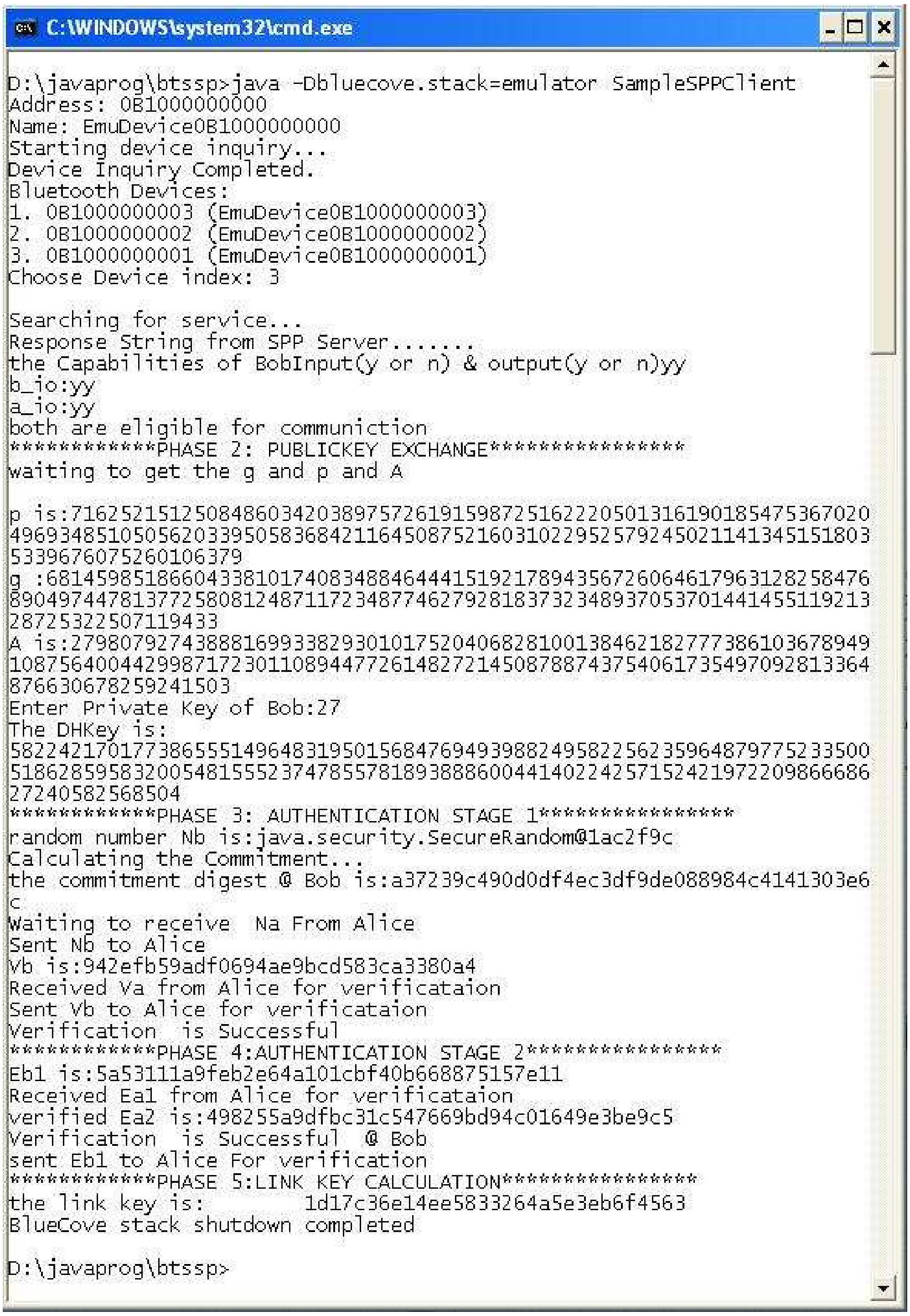


```
D:\javaprog\btssp>java -Dbluecove.stack=emulator SampleSPPServer
Address: 0B1000000001
Name: EmuDevice0B1000000001

Server Started. Waiting for clients to connect...
Remote device address: 0B1000000000
Remote device name: EmuDevice0B1000000000
Test String from SPP Client......
hi..SPP Server
the Capabilities of AliceInput(y or n) & output(y or n)yy
a_io is:yy
b_io is:yy
both are eligible for communiction
************PHASE 2: PUBLICKEY EXCHANGE****************
Enter Private Key of Alice: 17
Waitng for B:B is:1695290838946971494498790640064183716366999869666903
2927078487570505467067352784970276844296614791475279847697875387039974
434936318974687142600444152723l8
p is:7162521512508486034203897572619159872516222050131619018547536702
0496934851050562033950583684211645087521603102295257924502114134515180
3533967607526010637
Xa is:17
The DHKey is:
5822421701773865551496483195015684769493988249582256235964879775233500
5186285958320054815552374785578189388860044140224257152421972209866686
27240582568504
************PHASE 3: AUTHENTICATION STAGE 1****************
random number Na is:java.security.SecureRandom@329f3d
Waitng for Cb..
Sent Na to Bob
Waiting to receive  Nb From Bob
verification of Cb received from Bob is:
the commitment digest @ Alice is:a37239c490d0df4ec3df9de088984c4141303
e6c
Successful
Va is:942efb59adf0694ae9bcd583ca3380a4
sent Va to Bob For verification
received Vb from Bob For verification
Verification  is Successful
************PHASE 4:AUTHENTICATION STAGE 2****************
Ea1 is:498255a9dfbc31c547669bd94c01649e3be9c5
sent Ea1 to Bob For verification
Received Eb1 from Bob for verificataion
verified Eb2 is:5a53111a9feb2e64a101cbf40b668875157e11
Verification  is Successful @ Alice
************PHASE 5:LINK KEY CALCULATION****************
the link key is:        1d17c36e14ee5833264a5e3eb6f4563
BlueCove stack shutdown completed

D:\javaprog\btssp>
```

Figure 3.2: Simulation of Server Side Screen Shot for a Successful SSP connection



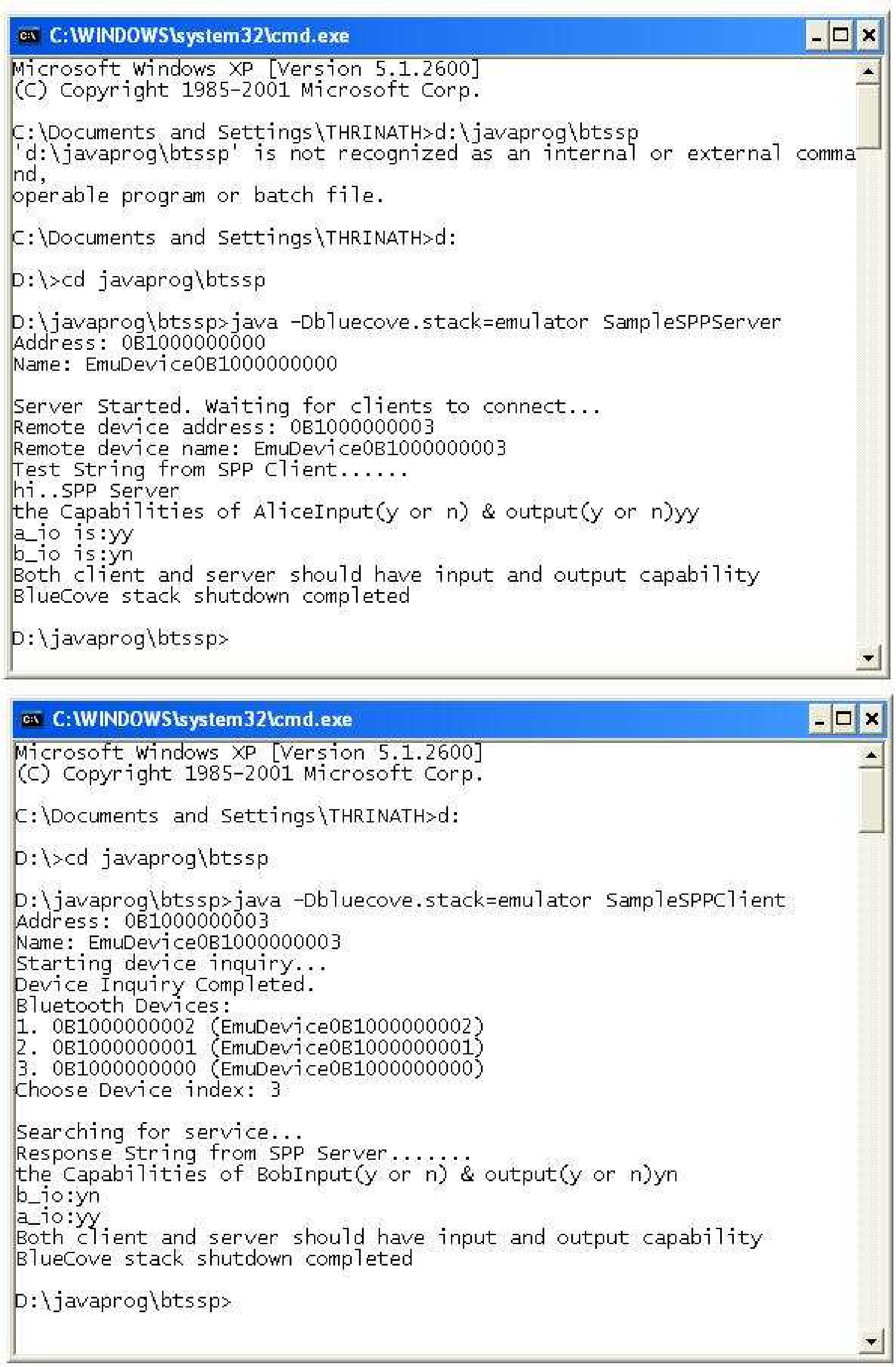
```
D:\javaprog\btssp>java -Dbluecove.stack=emulator SampleSPPClient
Address: 0B1000000000
Name: EmuDevice0B1000000000
Starting device inquiry...
Device Inquiry Completed.
Bluetooth Devices:
1. 0B1000000003 (EmuDevice0B1000000003)
2. 0B1000000002 (EmuDevice0B1000000002)
3. 0B1000000001 (EmuDevice0B1000000001)
Choose Device index: 3

Searching for service...
Response String from SPP Server.......
the Capabilities of BobInput(y or n) & output(y or n)yy
b_io:yy
a_io:yy
both are eligible for communiction
************PHASE 2: PUBLICKEY EXCHANGE****************
waiting to get the g and p and A

p is:716252151250848603420389757261915987251622205013161901854753670204
96934851050562033950583684211645087521603102295257924502114134515180
3533967607526010637 9
g :681459851866043381017408348846444151921789435672606461796312825847 6
890497447813772580812487117234877462792818373234893705370144145511921 3
28725322507119433
A is:279807927438881699338293010175204068281001384621827773861036789491
08756400442998717230110894477261482721450878874375406173549709281336 4
87663067825924150 3
Enter Private Key of Bob:27
The DHKey is:
582242170177386555149648319501568476949398824958225623596487977523350 0
518628595832005481555237478557818938886004414022425715242197220986668 6
27240582568504
************PHASE 3: AUTHENTICATION STAGE 1*****************
random number Nb is:java.security.SecureRandom@1ac2f9c
Calculating the Commitment...
the commitment digest @ Bob is:a37239c490d0df4ec3df9de088984c4141303e6
c
Waiting to receive  Na From Alice
Sent Nb to Alice
Vb is:942efb59adf0694ae9bcd583ca3380a4
Received Va from Alice for verificataion
Sent Vb to Alice for verificataion
Verification  is Successful
************PHASE 4:AUTHENTICATION STAGE 2*****************
Eb1 is:5a53111a9feb2e64a101cbf40b668875157e11
Received Ea1 from Alice for verificataion
verified Ea2 is:498255a9dfbc31c547669bd94c01649e3be9c5
Verification  is Successful  @ Bob
sent Eb1 to Alice For verification
************PHASE 5:LINK KEY CALCULATION****************
the link key is:        1d17c36e14ee5833264a5e3eb6f4563
BlueCove stack shutdown completed

D:\javaprog\btssp>
```

Figure 3.3: Simulation of Client Side Screen Shot for a Successful SSP connection



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\THRINATH>d:\javaprog\btssp
'd:\javaprog\btssp' is not recognized as an internal or external comma
nd,
operable program or batch file.

C:\Documents and Settings\THRINATH>d:

D:\>cd javaprog\btssp

D:\javaprog\btssp>java -Dbluecove.stack=emulator SampleSPPServer
Address: 0B1000000000
Name: EmuDevice0B1000000000

Server Started. Waiting for clients to connect...
Remote device address: 0B1000000003
Remote device name: EmuDevice0B1000000003
Test String from SPP Client......
hi..SPP Server
the Capabilities of AliceInput(y or n) & output(y or n)yy
a_io is:yy
b_io is:yn
Both client and server should have input and output capability
BlueCove stack shutdown completed

D:\javaprog\btssp>
```

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\THRINATH>d:

D:\>cd javaprog\btssp

D:\javaprog\btssp>java -Dbluecove.stack=emulator SampleSPPClient
Address: 0B1000000003
Name: EmuDevice0B1000000003
Starting device inquiry...
Device Inquiry Completed.
Bluetooth Devices:
1. 0B1000000002 (EmuDevice0B1000000002)
2. 0B1000000001 (EmuDevice0B1000000001)
3. 0B1000000000 (EmuDevice0B1000000000)
Choose Device index: 3

Searching for service...
Response String from SPP Server.......
the Capabilities of BobInput(y or n) & output(y or n)yn
b_io:yn
a_io:yy
Both client and server should have input and output capability
BlueCove stack shutdown completed

D:\javaprog\btssp>
```

Figure 3.4: Simulation of Screen Shot for an Unsuccessful SSP connection

3.5 Summary

In this chapter, we have seen all kinds of the countermeasures for the threats and attacks which are discussed on Chapter-2 and also proposed a countermeasure to avoid the MITM attack when the both communicating parties are known to each other.

Chapter 4

Conclusion

Achievements and Limitations of the Work

Chapter 4

Conclusion

4.1 Achievements and Limitations of the Work

In this thesis, we have shown the MITM attack that is very harmful in Bluetooth pairing. Our proposal includes the support of IDS and IPS to protect against MITM attacks which was never been used in Bluetooth pairing till date. We have shown the simulation screen shots of the unsuccessful and successful SSP when there is no attacker. We have partially implemented the proposed countermeasure as there are no devices with adjustable BD ADDRs. Except sophisticated and expensive protocol analyzers, no one can perform the MITM attacks. In near future, If the adjustable BT ADDR devices are available, our proposal will defend and withstand the attackers against the attack.

Bibliography

[1] K. Haataja and P. Toivanen. Two practical man-in-the-middle attacks on bluetooth secure simple pairing and countermeasures. *Wireless Communications, IEEE Transactions on*, 9(1):384–392, Jan. 2010.

[2] Markus Jakobsson and Susanne Wetzel. Security weaknesses in bluetooth. In David Naccache, editor, *Topics in Cryptology, CT-RSA 2001*, Lecture Notes in Computer Science, pages 176–191. Springer Berlin / Heidelberg, 2001.

[3] Kugler and Dennis. “man in the middle attacks” on bluetooth. In *Financial Cryptography*, volume 2742 of *Lecture Notes in Computer Science*, pages 149–161. Springer Berlin / Heidelberg, 2003.

[4] Albert Levi, Erhan Çetintaş, Murat Aydos, Çetin Kaya Koç, and M. Ufuk Çağlayan. Relay attacks on bluetooth authentication and solutions. In Cevdet Aykanat, Tugrul Dayar, and Ibrahim Körpeoglu, editors, *Computer and Information Sciences - ISCIS 2004*, volume 3280 of *Lecture Notes in Computer Science*, pages 278–288. Springer Berlin / Heidelberg, 2004.

[5] K. Haataja. *Security threats and countermeasures in Bluetooth-enabled systems.* PhD thesis, University of Kuopio, Department of Computer Science, Feb. 2009.

[6] Karen Scarfone and John Padgette. *Guide to Bluetooth Security.* Special Publication 800-121, Recommendations of the National Institute of Standards and Technology, 2008. http://csrc.nist.gov/publications/nistpubs/800-121/SP800-121.pdf.

[7] John Paul Dunning. Taming the blue beast: A survey of bluetooth based threats. *IEEE Security and Privacy*, 8:20–27, 2010. http://doi.ieeecomputersociety.org/10.1109/MSP.2010.3.

[8] Peter Piazza. From Bluetooth to RedFang. http://www.securitymanagement.com/article/bluetooth-redfang?page=0%2C2.

[9] Maxim Krasnyansky. sdptool. http://linuxcommand.org/man_pages/sdptool1 .html.

[10] Overview of blue scanner. http://nesttech.com/Portals/0/BlueScan.pdf.

[11] bluesnipping. http://www.urbandictionary.com/define.php ?term=bluesniping.

[12] bluesnipping. http://en.wikipedia.org/wiki/Bluesniping.

[13] Randy Marchany. Vt it security news, april 2010. http://www.security.vt.edu/downloads/newsletters/hotsheetV3N2.pdf.

[14] Terrence J. OConnor. Bluetooth intrusion detection, 2008. http://repository.lib.ncsu.edu/ir/bitstream/1840.16/235/1/etd.pdf.

[15] Lisa Phifer. Don't get bitten by bluetooth. http://export.imix.co.za/node/48833.

[16] Dominic Spill and Andrea Bittau. Bluesniff: Eve meets alice and bluetooth. In *Proceedings of the first USENIX workshop on Offensive Technologies*, pages 5:1–5:10, Berkeley, CA, USA, 2007. USENIX Association.

[17] Wireless sniffing with wireshark. http://www.willhackforsushi.com/books/37 7_eth_2e_06.pdf.

[18] Adam Ziaja. Setting up the bluetooth gps to work with kismet, November 2009. http://az.linux.pl/2009/11/gps-bluetooth-kismet.html.

[19] K Pelechrinis, M Iliofotou, and V. Krishnamurthy. Denial of service attacks in wireless networks: The case of jammers. *Communications Surveys Tutorials, IEEE*, 99:1–13, 2010.

[20] Wenyuan Xu, Wade Trappe, Yanyong Zhang, and Timothy Wood. The feasibility of launching and detecting jamming attacks in wireless networks. In *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc'05, pages 46–57, New York, NY, USA, 2005. ACM. http://doi.acm.org/10.1145/1062689.1062697.

[21] Timothy K. Buennemeyer and Randy C. Marchany Michael A. Gora. Battery polling and trace determination for bluetooth attack detection in mobile devices. IEEE.

[22] Bluejacking. http://en.wikipedia.org/wiki/Bluejacking.

[23] Merloni C. Carettoni L. and Zanero S. Studying bluetooth malware propagation: The bluebag project. IEEE, April 2007. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4140986.

[24] Marcel Holtmann Adam Laurie and Martin Herfurt. Hacking bluetooth enabled mobile phones and beyond full disclosure, December 2004. http://trifinite.org/Downloads/21c3_Bluetooth_Hacking.pdf.

[25] RobotSkirts. Bluetooth keyboard attacks, Feb 2010. http://www.robotskirts.com/2010/02/06/bluetooth-keyboard-attacks/.

[26] Jani Suomalainen, Jukka Valkonen, and N. Asokan. Security associations in personal networks: A comparative analysis. In Frank Stajano, Catherine Meadows, Srdjan Capkun, and Tyler Moore, editors, *Security and Privacy in Ad-hoc and Sensor Networks*, volume 4572 of *Lecture Notes in Computer Science*, pages 43–57. Springer Berlin / Heidelberg, 2007. http://dx.doi.org/10.1007/978-3-540-73275-4_4.

[27] K. Hypponen and K.M.J. Haataja. Nino: man-in-the-middle attack on bluetooth secure simple pairing. In *3rd IEEE/IFIP International Conference in Central Asia on Internet, 2007, ICI-2007*, pages 1–5, Sep. 2007. http://dx.doi.org/10.1109/CANET.2007.4401672.

[28] K. Haataja and K. Hypponen. Man-in-the-middle attacks on bluetooth: a comparative analysis, a novel attack, and countermeasures. In *Proc. IEEE Third International Symposium on Communications, Control and Signal Processing (ISCCSP-2008).* St. Julians, Malta, Mar 2008.

[29] K. Haataja and P. Toivanen. Practical man-in-the-middle attacks against bluetooth secure simple pairing. In *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on*, pages 1 –5, Oct. 2008. http://dx.doi.org/10.1109/WiCom.2008.1153.

[30] Wenyuan Xu, Ke Ma, W. Trappe, and Yanyong Zhang. Jamming sensor networks: attack and defense strategies. *Network, IEEE*, 20(3):41 – 47, May-Jun. 2006. http://dx.doi.org/10.1109/MNET.2006.1637931.

[31] Noubir and Guevara. On connectivity in ad hoc networks under jamming using directional antennas and mobility. In Peter Langendoerfer, Mingyan Liu, Ibrahim Matta, and Vassilis Tsaoussidis, editors, *Wired/Wireless Internet Communications*, volume 2957 of *Lecture Notes in Computer Science*, pages 521–532. Springer Berlin / Heidelberg, 2004. http://dx.doi.org/10.1007/978-3-540-24643-5_17.

[32] A. J. Viterbi. *Principles of Spread Spectrum Communication.* Addison-Wesley Wireless Communications Series. Addison-Wesley, 1995.

[33] Guevara Noubir and Guolong Lin. Low-power dos attacks in data wireless lans and countermeasures. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7:29–30, Jul. 2003. http://doi.acm.org/10.1145/961268.961277.

[34] Guolong Lin and Guevara Noubir. On link layer denial of service in data wireless lans: Research articles. *Wirel. Commun. Mob. Comput.*, 5:273–284, May 2005.

[35] Wenyuan Xu, Wade Trappe, and Yanyong Zhang. Anti-jamming timing channels for wireless networks. In *Proceedings of the first ACM conference on Wireless network security*, WiSec'08, pages 203–213, New York, NY, USA, 2008. ACM. http://doi.acm.org/10.1145/1352533.1352567.

[36] F.R.K. Chung, J.A. Salehi, and V.K. Wei. Optical orthogonal codes: design, analysis and applications. *Information Theory, IEEE Transactions on*, 35(3):595 –604, May 1989.

[37] BlueCove Team. Bluecove, 2004-2008. http://bluecove.org/index.html.

[38] Behrouz A. Forouzan. *Cryptography & Network Security.* The McGraw-Hill Companies, Tata McGraw-Hill Publishing Company Limited, New Delhi, 2007.

Dissemination of Work

Accepted

1. Thrinantha R Mutchukota, Saroj Kumar Panigrahy, and Sanjay Kumar Jena, "Man-in-the-Middle Attack and its Countermeasure in Bluetooth Secure Simple Pairing", in *Fifth International Conference on Information Processing (ICIP-2011)*, 5-7th August, 2011, Bangalore, India.