Journal of Computer Science 5 (9): 646-650, 2009
ISSN 1549-3636

Bluetooth Wireless Network Authentication Using Radio Frequency Communication Protocol

Ghossoon M. Waleed, M. Faizal, R.B. Ahmad, M. Fareq B. Abd Malek and M. Alif Ghani
School of Computer and Communication Engineering, University Malaysia Perlis,
Kompleks Pusat Pengajian Seberang Ramai (Blok A) No. 12 and 14, Jalan Satu,
Taman Seberang Jaya Fasa 3, P.O.S. 02000 Kuala Perlis, Malaysia

Abstract: Problem statement: Bluetooth has emerged as very popular ad hoc network standard today. Bluetooth network applications include wireless synchronization, e-mail/internet/intranet access using local personal computer connections, hidden computing through automated applications and networking. Due to this ability, it is not impossible for the network to receive any attack. **Approach:** In this study, we developed an algorithm to build new software that secures connections between two Bluetooth platforms, which included authentication, authorization and confidentiality. There is no authentication when using the protocol in connecting the Bluetooth platform. When there is no security, so that all users can directly connect to the server without any permission because n most of the application nowadays that refers to RFCOMM protocol is not. **Results:** The test environment for the non-secure connections is also being setting up before build the software. The purpose of this study was to build a secured Bluetooth application in connecting Bluetooth Platform using Radio Frequency Communication (RFCOMM) protocol. **Conclusion:** Furthermore focusing on setting up the test environment for a non-secure connection application.

Key words: Security mode, radio frequency communication, cellular digital packet data and global system for mobile communications

INTRODUCTION

Wireless networks serve as the transport mechanism between devices. Wireless networks can be categorized into three group based on their coverage range: Wireless Wide Area Network (WWAN), Wireless Local Area Network (WLAN) and Wireless Personal Area Networks (WPAN). WWAN includes wide coverage area technologies such as 2G cellular, Cellular Digital Packet Data (CDPD) and Global System for Mobile Communications (GSM). WLAN includes 802.11 and Hyper LAN.WPAN, representing wireless personal area network technologies such as Bluetooth and Infrared (IR). All of these technologies receive and transmit information using Electromagnetic (EM) waves[1].

The problem when most of the application nowadays that refers to RFCOMM protocol is not, there is no authentication when using the protocol in connecting the Bluetooth platform. When there is no security, so that all users can directly connect to the server without any permission.

Security features of Bluetooth per the specifications: Bluetooth has three different modes of security. Each Bluetooth device can operate in one mode only at a time. These three modes are the following:

- Security mode 1-non-Secure mode
- Security mode 2-service-level enforced security mode
- Security mode 3-link-level enforced security mode

In security mode 1, a device will not initiate any security procedures. In this non-secure mode, the security functionality (authentication and encryption) is completely bypassed. In effect, the Bluetooth device in Mode 1 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.

In security mode 2, the service-level security mode, security procedures are initiated after channel establishment at the Logical Link Control and Adaptation Protocol (L2CAP) level. L2CAP resides in the data link layer and provides connection-oriented

Corresponding Author: Ghossoon M. Waleed, School of Computer and Communication Engineering, University Malaysia Perlis, Kompleks Pusat Pengajian Seberang Ramai (Blok A) No. 12 and 14, Jalan Satu, Taman Seberang Jaya Fasa 3, P.O.S. 02000 Kuala Perlis, Malaysia

and connectionless data services to upper layers. For this security mode, a security manager control access to services and to devices. The centralized security manager maintains polices for access control and interfaces with other protocol and device users. Varying security polices and "trust" level to restrict access may be defined for applications with different security requirement operating in parallel. Therefore, it is possible to grant access to some service without providing access to other services.

In security Mode 3, 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.

MATERIALS AND METHODS

The methodology used two ways to establish the connection of Bluetooth Wireless Network Authentication Using RFCOMM protocol and satisfy the secure connection (authentication) in Linux- Ubuntu and Windows- XP. For the security part, security mode 3 is being used, which is the link-level security mode. A Bluetooth device initiates security procedures before the channel is established. This mode supports authentication and encryption. These features are based on a secret link key that is shared by a pair of devices.

RFCOMM protocol: The RFCOMM protocol provides emulation of serial ports over the L2CAP protocol. This protocol is based on the ETSI standard TS 07.10. Only a subset of the TS 07.10 standard is used and some adaptations of the protocol are specified in the Bluetooth RFCOMM specification. RFCOMM is a simple transport protocol, which provides emulation of RS232 serial ports over the L2CAP protocol. The protocol is based on the ETSI standard TS 07.10. Only a subset of the TS 07.10 standard is used and an RFCOMM-specific extension is added, in the form of a mandatory credit based flow control scheme.

Device type: Basically two device types exist that RFCOMM must accommodate:

- Type 1 devices are communication end points such as computers and printers
- Type 2 devices are those that are part of the communication segment; e.g., modems

Null modem emulation: RFCOMM is based on TS 07.10. When it comes to transfer of the states of the non-data circuits, TS 07.10 does not distinguish between DTE and DCE devices. The RS-232 control signals are sent as a number of DTE/DCE independent signals. The way in which TS 07.10 transfers the RS-232 control signals creates an implicit null modem when two devices of the same kind are connected together. No single null-modem cable wiring scheme works in all cases; however the null modem scheme provided in RFCOMM should work in most cases.

Multiple emulated serial port: Two BT devices using RFCOMM in their communication may open multiple emulated serial ports. RFCOMM supports up to 60 open emulated ports; however the number of ports that can be used in a device is implementation-specific. A Data Link Connection Identifier (DLCI) identifies an ongoing connection between a client and a server application. The DLCI is represented by 6 bits, but its usable value range is 2…61. The DLCI is unique within one RFCOMM session between two devices.

To account for the fact that both client and server applications may reside on both sides of an RFCOMM session, with clients on either side making connections independent of each other, the DLCI value space is divided between the two communicating devices using the concept of RFCOMM server channels.

If a BT device supports multiple emulated serial ports and the connections are allowed to have endpoints in different BT devices, then the RFCOMM entity must be able to run multiple TS 07.10 multiplexer sessions. Note that each multiplexer session is using its own L2CAP Channel ID (CID). The ability to run multiple sessions of the TS 07.10 multiplexer is optional for RFCOMM.

System requirement: There are few requirements that need to be fulfilled. This project required two laptops with Bluetooth device (specific to WIDCOMM/BROADCOM Bluetooth stack), BROADCOM BlueChat application for testing the RFCOMM protocol connection, Microsoft Visual Studio 2008 and BROADCOM Bluetooth Software Development Kit (SDK) for building the secured software.

WIDCOMM/BROADCOM: WIDCOMM was the first Bluetooth stack for the Windows operating system. The stack was initially developed by a company named WIDCOMM Inc., which was acquired by BROADCOM Corporation in April 2004. BROADCOM continues to license the stack for inclusion with many Bluetooth-powered end-user devices.

Fig. 1: WIDCOMM and BROADCOM logo

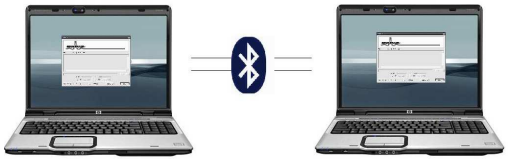

Fig. 2: Test environment for BlueChat application

BROADCOM Corporation is an American supplier of Integrated Circuits (ICs) for broadband communications. Founded in 1991 by Henry Samueli (chairman and CTO) and Henry Nicholas, it became a public company in 1998 and now employs over 7,400 people worldwide. BROADCOM is among the Worldwide Top 20 Semiconductor Sales Leaders Fig. 1.

BROADCOM BlueChat: The BlueChat application creates an RFCOMM connection and Permits the client and server to chat. BlueChat is installed on two platforms. One is configures as the client and the other is configured as the server by the user.

Algorithm for GUI:

- The GUI allows modem control signals to be set/read and error conditions to be send to the other side
- The user establishes the connection (with one PC as the server and the other as the client). The server PC creates a new service and then waits for the client to connect
- The client performs a device inquiry and service discovery
- The two sides connect and communicate, Fig. 2

System design and implementation: The implementation of "Bluetooth Wireless Network Using RFCOMM Protocol" can be divided into two ways, either in Windows Operating System or in the Linux environment.

Linux operating system: In Linux OS, Bluez Bluetooth stack is being used. There are few coding that has been used to satisfy the connection between server and client. The first coding is to discover the Bluetooth service within area. The result of the application shows the address of the Bluetooth devices.

The address is then being inserted in the second coding which is for the client device. The client will directly connect to the server and gives a simple message as a result.

Application service discovery: The program reserves system Bluetooth resources, scans for nearby Bluetooth devices and then looks up the user friendly name for each detected device. A more detailed explanation of the data structures and functions used follows.

Server-client connection: The address of the detected device will be added in the client coding, which lets the client to connect with the server. The client will transfer some data to the server and disconnect.

Windows operating system: In Windows environment, two laptops with Bluetooth device are available. The device is being specific to WIDCOMM/BROADCOM since this protocol stack is available in most laptops nowadays.

BlueChat application for test environment: For implementing the RFCOMM protocol, BlueChat application is used. This application is some sort of nowadays Instant Messaging (IM) application, but the different is that it connects and communicates using Bluetooth. The testing environment allows user to prove that most of the available applications in market are not secure. Any user who has discovered the available service can directly connect to it and start communicate.

Build secured application (RFCOMM protocol): The core of the project, building a secured application, only certain user can connect to the available service. The PIN key should bond by the user before communication. The Microsoft Visual Studio 2008 is being used for the programming side while the BROADCOM Bluetooth SDK is the guide for building the application. This RFCOMM protocol provides emulation of serial ports over the L2CAP protocol. The protocol is based on the ETSI standard TS 07.10. Only a subset of the TS 07.10 standard is used and an RFCOMM-specific extension is added, in the form of a mandatory credit based flow control scheme.

The RFCOMM protocol supports up to 60 simultaneous connections between two BT devices. The number of connections that can be used simultaneously in a BT device is implementation-specific. For the purposes of RFCOMM, a complete communication path involves two applications running on different devices (the communication endpoints) with a communication segment between them.

Testing secured application: Testing the secured application, two Bluetooth platforms and a secured application are used. The different between the test environments is that the user must first bond with the server in order to communicate.

Establish connection:

- One platform is act as a server. Server will always be on the idle mode so that it can always listen for any client to connect to it
- The client is the one who can request to connect to the server
- Upon connecting the server, the client must insert the pin key for a security measure
- The server is as well required to insert the same pin key that the server has inserted

RESULTS

The results have achieved upon completing the objective project as the following:

- The Bluetooth connect directly without authentication with test environment for non secure connection in both operating system using Linux-Ubuntu and Windows XP
- Developed an algorithm to the secure application for Bluetooth connection in both server and client by authentication using the PINKEY
- The test application divided in two parts, result for test environment and result for secured application

Test environment result: The testing environment has two modes which are server mode and client mode.

- Server Mode always listening for the connection of client and to satisfy the connection required Act as Chat Server check box, BlueChat starts a session and performs the following tasks:
  - Adds a service record using DK class CSdpService in function CBlueChatDlg; DoCreateSeerviceRecord()
  - Opens an RFCOMM server port using OpenServer()
  - Waits for a client connection, Fig. 3
- Client mode search for server to connected: BlueChat starts a client session. The list of servers is populated automatically with the Bluetooth devices found using device inquiry:
  - Select a Chat Server dialog box that appears, select a host from the list
  - Click Discover to send a discovery request to determine if the host is a Chat Server
  - Click OK, The dialog window closes and an RFCOMM client connection is opened with the chat server. BlueChat is ready to communicate
  - To send a message, type the message in the edit field and then click Send
- The BlueChat main window will display the information for client and server mode at the end of session:
  - Messages send and received are echoed in the log window
  - RFCOMM event display in the log window, Fig. 4

Secured application: The main different between the secured application and the test application is that the two laptops have to bond with each other using the PIN key, Fig. 5.

Fig. 3: BlueChat server mode

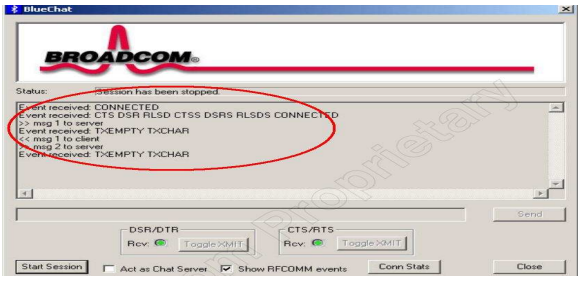

Fig. 4: BlueChat log window

Fig. 5: Secure application main interface

Table 1: Bluetooth security requirements

Security recommendation	Requirement
Choose PIN codes that are sufficiently random and long. Avoid static and weak PINs, such as all zeroes.	PIN codes should be random so that they cannot be easily guessed by malicious users. Longer PIN codes are more resistant to brute force attacks. For Bluetooth v2.0 (or earlier) devices, an eight-character alphanumeric PIN should be used, if possible. The use of a fixed PIN is not acceptable for sensitive Bluetooth connections.
Ensure that link keys are based on combination keys rather than unit keys.	The use of shared unit keys can lead to successful MITM attacks. The use of unit keys for security was deprecated in Bluetooth v1.2.
Bluetooth devices should be configured by default as and remain, undiscoverable except as needed for pairing.	Bluetooth interfaces should be configured as non-discoverable, which prevents visibility to other Bluetooth devices except when discovery is specifically needed. Also, the default self-identifying or discoverable names provided on Bluetooth devices should be changed to anonymous, unidentifiable names.
Establish a "minimum key size" for any key negotiation process.	Establishing minimum key sizes ensures that all keys are long enough to be resistant to brute force attacks. Preferably, keys should be at least 128 bits long.

The secure application, its consist of two modes (server mode and client mode), in the server mode the same steps applied to listen for connection from client, the Client mode bond the two laptops: Firstly, click the Start Session button, then select a chat server window will pops up:

- Click the bonding button and bond window will be shown
- Enter the PIN key and click the bond button
- The Server must key in the same PIN key as the client inserted. And start chatting

DISCUSSION

The main different between the secured application and the test application is that the two laptops found the bond with each other using the PIN key. Secured application has been build to solve the non-secure environment and the requirements for Bluetooth security as shown to explain the details as shown in Table 1.

CONCLUSION

From the implementation of the project titled Bluetooth Wireless Networks Authentication using RFCOMM Protocol and the effect of the performance result, it leads to the following conclusion:

- The test environment is not secured and can connect directly without authentication in both operating system using Linux-Ubuntu and Windows XP
- Authentication algorithm developed using C++ and Bluetooth API coding to the Bluetooth connection secured
- One of the main advantage of Bluetooth faster data transfer and should be secure when achieved connection
- The test application of Bluetooth includes the test environment and secured application

The test environment for non-secure connection has been set up and found the result.

REFERENCES

1. Mohammad Ilyas, 2002. The Handbook of Ad hoc Wireless Networks. CRC Press, ISBN: 0849313325, pp: 624.
2. Miller, M., 2001. Discovering Bluetooth. Sybex Inc., ISBN: 10: 0782129722, pp: 288.
3. Karygiannis, T. and L. Owens, 2004. Wireless network security-802.11, Bluetooth and handheld devices. http://www.developers.net/node/view/136
4. Erasala, N. and David C. Yen, 2002. Bluetooth technology: A strategic analysis of its role in global 3G wireless communication era. Comput. Stand. Interfaces, 24: 193-206. http://portal.acm.org/citation.cfm?id=635215
5. Rysavy, P., 2001. Break free with wireless LANs network computing, mobile and wireless technology feature. http://www.rysavy.com/Articles/BreakFree/BreakFree.htm
6. Blue Tomorrow, 2008. The history of Bluetooth. http://www.bluetomorrow.com/content/section/11/38/
7. Providing information on hacking Bluetooth. http://www.crypto.ruhr-uni-bochum.de/imperia/md/content/seminare/itsss07/lidesbluetoothsecurityandhacks.pdf
8. NIST, 2007. Computer security resource center. http://csrc.nist.gov/publications
9. Wikipedia, 1960. Information on Ishikawa diagram. http://en.wikipedia.org/wiki/Ishikawa_diagram
10. Wikipedia, 1988. Information on spiral model. http://en.wikipedia.org/wiki/Spiral_model
11. NSAUDITOR, 2009. BlueAuditor-scan and monitors Bluetooth devices in a wireless network! http://www.nsauditor.com/bluetooth_network_scanner.html