Security Analysis of Bluetooth Enabled Mobile Devices

Stephanie Ho(41002023), Brian Ng(42359026), Justin Kwong(79203014) and Frank Wu(36866028)

***Abstract*— Bluetooth is a popular way for mobile devices to connect wirelessly. Such a connection allows users to transfer data between two or more devices as well as control a device remotely. However, the security of such capabilities is insufficient for widespread use of Bluetooth. Specifically, the PIN, which acts as a secret key between two devices, can be easily cracked by an attacker. In addition to this, poor implementation of security mechanisms has led to other potential attacks on mobile devices with Bluetooth. Security must be improved to provide device resources with confidentiality, integrity and availability.**

***Index Terms*—Bluetooth Security**

I. Introduction

Bluetooth has become a popular wireless connectivity mechanism that is widely used on many electronic devices such as computers, keyboards, mice, printers, PDA's, and mobile phones. As mobile phone usage is continuing to rapidly grow, an increasing amount of hardware and software applications are being integrated into mobile phones. For example, cameras are being integrated into mobile phones to allow them to take photos, shoot films and conduct video conferences. On the software side, applications allow users to check email, surf the internet, and edit documents as well as transfer them wirelessly. Such applications require data confidentiality, integrity and availability in order to retain user privacy. However, due to the nature of wireless connections, broadcasted messages can be easily intercepted and thus, Bluetooth security is crucial in enforcing security policies.

II. Bluetooth Security Mechanism

Since the development of Bluetooth, there have been many security measures taken to ensure the security of information transfer via Bluetooth. Many devices take advantage of Bluetooth technology and therefore, there are many applications that have been designed to specifically use Bluetooth as a medium for data transfer between devices. Because of the vast amount of different applications available and the increase in the need for availability of diverse and ad-hoc connections, a number of security mechanisms have also been developed to support the security of these applications. In this section, available security mechanisms as well as their implementation will be discussed.

In the Generic Access Profile for Bluetooth, there are three modes in which a Bluetooth connection can operate under. The three modes are as follows:

Security Mode 1: Non-secure
Security Mode 2: Service Level Enforced Security
Security Mode 3: Link Level Enforced Security

At the level of the non-secure mode, no authentication process will be initiated before information is transferred. This means there is no encryption, no keys, and no random numbers. For devices which operate at this level, authentication is optional.

The second mode of Bluetooth security provides moderate security through a specific channel or application. It provides flexibility for applications that could be running concurrently within a device. When applications or devices use more than one mode of security excluding non-secure mode, then security mode 2 is mandatory. At this level of security, there is no security enforcement made until either a connection has been established, or the process of establishing a connection has been initiated. Connections of this type are classified as authorization. Usually, devices will try to verify other devices that are trying to access services by means of a Bluetooth Personal Identification Number (PIN).

The third level of Bluetooth security is the Link Level Enforced Security. This is the highest level of security available that is provided by a Bluetooth connection. Like security mode 2, security mode 3 is also mandatory if more than one other security level is available excluding the non-secure mode. When a device wants to establish a connection with another Bluetooth enabled device, it sends a Link Manager Protocol (LMP) connection request. After that, the second device will establish a connection with LMP Pairing, LMP Authentication and data encryption ensuring all the security precautions have been taken before any information is sent. If any of the requirements for a secure connection has not been met, it will result in an authentication failure. When both devices feel that all security requirements have been met, they will issue a set up complete command.

Now that we know how connections are established between Bluetooth devices, we need to determine what the variables that allow for such security mechanisms are. First, to identify individual devices, each Bluetooth device is issued a unique Bluetooth device address. Under the definition defined by the IEEE, Bluetooth device addresses are 48 bits

long and are unique for each device. There are several different keys used to establish and authenticate connection between devices. The private authentication key is a key used during the process of authentication. The length of this key is 128 bits. The private encryption key is a key used for encryption and can vary from 8 to 128 bits long. Lastly, there is a random number produced by individual Bluetooth devices and is 128 bits long in length.

III. Key Management

When using Bluetooth, secure connections and information encryption all rely on the secrecy of keys. Hence, key management becomes a very concerned topic when using Bluetooth. There is one key that handles all secured transactions between two or more devices. This is called the link key and is a random number with a length of 128 bits. A link key can be classified as either a semi-permanent key or a temporary key. This classification determines the lifetime of the key. A semi-permanent key is kept and can be reused to authenticate other Bluetooth devices that share it, even after a transaction has ended. The link key is used when authenticating users as well as to derive the encryption key that will be used for the current transaction.

Depending on the application utilizing Bluetooth, the link key can be of different types. There are combination keys, unit keys, master keys, and initialization keys. The unit key is a key generated by a device when it is installed into the network. When two devices are paired, they will use a combination key. This key is a key generated from the information taken from each device. A master key is a temporary key used for a device when it wants to broadcast information to more than one device. Finally, the initialization key is used when two devices needs to establish a connection but no unit or combination keys are available yet.

When two devices attempt to pair with each other, both devices will prompt the user for a Personal Identification Number (PIN). The PIN entered can be 4-8 digits giving the PIN between 1 to 16 octets. The PIN can also be fixed such that it only needs to be entered on the device wishing to connect. The user is prompted for the PIN when there is no other previous connection between the two devices. The PIN will aid in the generation of the initialization key. The initialization key is generated using an algorithm called E22. The E22 algorithm takes the random number the device generates, the PIN, and the length of the PIN as inputs and the resulting output is the initialization key. After the processes of establishing a connection and generating a link key, the initialization key is then discarded.

In turn, the unit key is generated with the E21 algorithm. This algorithm takes the random number, and the Bluetooth device address to output the unit key. After the unit key is generated, it will be stored in non-volatile memory. Other devices can use this unit key as a link key between them. The unit key used will be determined during the initialization process. The combination key is also generated using the E21 algorithm. However, after both devices have generated a key, they will each share their random numbers so that they can generate the other device's key and calculate the combination key that shall be used between them.

The master key is a temporary key also derived using the E22 algorithm. The inputs are two 128-bit random numbers. A third random number is then sent to the slave device. The key generating algorithm and the current link key will use this number to assist both the master and slave to calculate an overlay. The master key is XORed with the overlay and sent to the slave. This allows the slave to calculate the master key.

Lastly the encryption key is generated by the E3 algorithm. The E3 takes in a random number, the link key and a 96 bit ciphering offset number to output a 128 bit encryption algorithm. The ciphering offset number is generated during the authentication process.

IV. Encryption

Bluetooth encrypts information in packets of payloads. Encryption is done using a stream cipher E0. The stream key is XOR'd with the payloads. The key stream is produced using a cryptographic algorithm based on four Linear Feedback Shift Registers. The E0 algorithm takes in the master Bluetooth device address, the random number, a slot number and an encryption key. The slot number changes with each sent packet, therefore the encryption engine has to be reinitialized before encrypting each packet.

The encryption key can vary in length between Bluetooth devices so before traffic can be encrypted, a common length for the encryption key must be negotiated before encryption can happen. The master will send a suggestion for the length of the encryption key length and the slave can either accept or decline. If the slave declines, it will make a new suggestion to the master and the master will decide if it will accept. Suggestions will travel back and forth until a common length of the encryption key can be decided. If no agreement can be reached, then no encryption will be available.

There are different encryption modes available depending on what kind of key the Bluetooth application is using. When a unit key or a combination key is used for the link key, data can only be encrypted when sending unicast traffic, however, they do not have to be. When broadcast traffic is sent, they cannot be encrypted. This results in three encryption modes:

- Encryption mode 1: nothing is encrypted
- Encryption mode 2: only unicast traffic is encrypted
- When a master key is used, a third mode is available:
- Encryption mode 3: all traffic (unicast and broadcast) are encrypted

V. Authentication

Authentication is performed in a Bluetooth system by using symmetrical keys. The whole system relies on the assumption that only the two devices that wish to communicate with each other have access to the key. Unfortunately, Bluetooth's authentication scheme is very weak and cracking the PIN is actually very easily accomplished.

Bluetooth authentication is completed in three main steps. The entire process is called the "pairing process". Each step involves data transmission and key generation. Key generation is described in the previous section and will only be briefly mentioned in this section.

The first step of authentication is the generation of the initialization key. When two devices wish to authenticate each other, the device deemed the master sends a random 128-bit number to the slave. Then, the key is calculated using the random number (IN_RAND), the Bluetooth device address of the slave (BD_ADDR), and the user-entered PIN. Through the E22 algorithm the initialization key (K_{init}) is calculated and used for the next authentication step. The figure below shows the first authentication step.

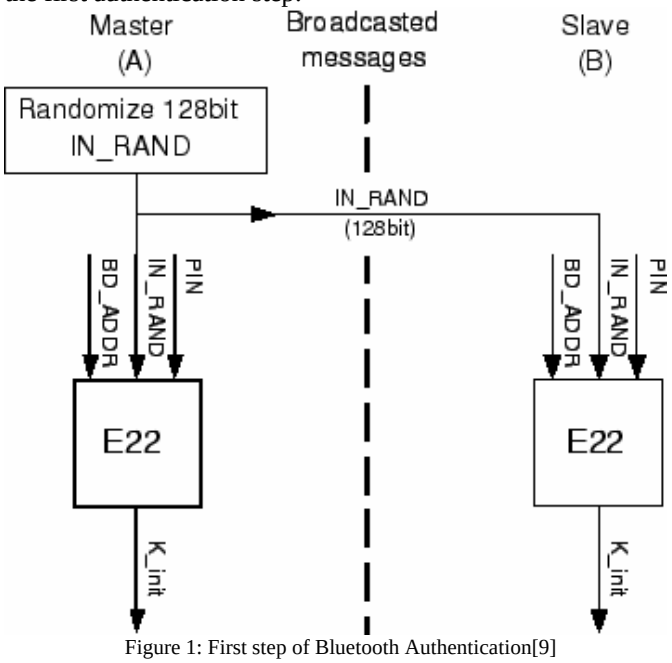

Figure 1: First step of Bluetooth Authentication[9]

The second step of the process begins with a transmission of another random number XOR'd with the initialization key transmitted from master to slave (LK_RAND_A). The slave transmits a similar signal back to the master (LK_RAND_b). The unit key (K_{ab}) is then created using values derived from these two XOR'd numbers, BD_ADDR and the E21 algorithm. The unit key is stored in each device's memory to be used for future authentication.

Finally, to verify that both devices have the same unit key (and thus PIN as well), the master generates another random number (AU_RAND) and sends it to the slave. Then, both devices generate a value (SRES) using the unit key, the Bluetooth device address, and the random number utilizing the E1 algorithm. The slave then transmits this value to the master who verifies that this number corresponds with that which was generated by the master. Should the values not correspond, the authentication process is halted. After a certain number of invalid values, the whole authentication process will be restarted. This is important for cracking the PIN (described later). Should the values correspond with each other, then this step of the authentication process is repeated with the roles of the master and slave switched. Authentication is completed if the second SRES values correspond to each other. The figure below shows the first part of the final authentication step.

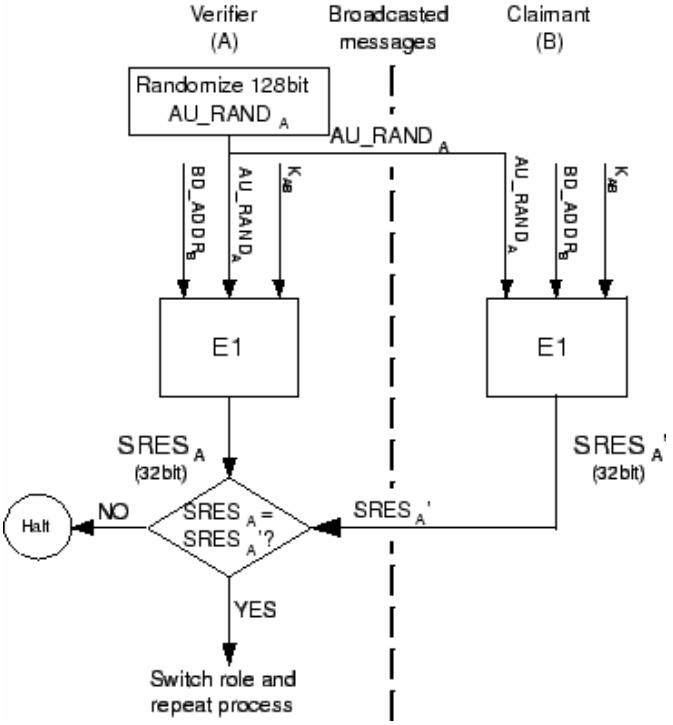

Figure 2: Final Authentication Step

VI. Cracking the PIN

In order to crack the PIN between two devices, the attacker must first eavesdrop on the entire authentication process. Since Bluetooth transmits data wirelessly, eavesdropping is merely a matter of have an antennae and being within range. As mentioned before, Class One Bluetooth devices have a range of 100m, so eavesdropping on the authentication signal is an easy task given the right tools. The transmitted values are summarized in the Figure 3.

Notice that five of the seven transmissions are in plaintext. These transmissions are used by an attacker to launch a brute force on the PIN. Recall that the initialization key is produced from the Bluetooth address, the PIN and the random number through the E22 algorithm. The random number is transmitted in plaintext, so the attacker will have this value; and the Bluetooth address can be obtained by a simple query command. The only missing value is the PIN, so the attacker just needs to exhaust a list of possible PIN's ($10^4+10^5+10^6+10^7+10^8$ = 111,110,000 possibilities) to find the K_{ab} that produces $SRES_a$ and $SRES_b$ given the random numbers AU_RAND_a and AU_RAND_b respectively [9].

The most difficult task of the attacker is not actually cracking the PIN, but to record the whole authentication process. Because unit keys are stored, only the last step of the authentication process is usually broadcasted. However, certain specifications of Bluetooth security allow an attacker to force the two devices to authenticate from the very first step. Signals sent at certain stages of the authentication process will force the two devices to re-pair [9].

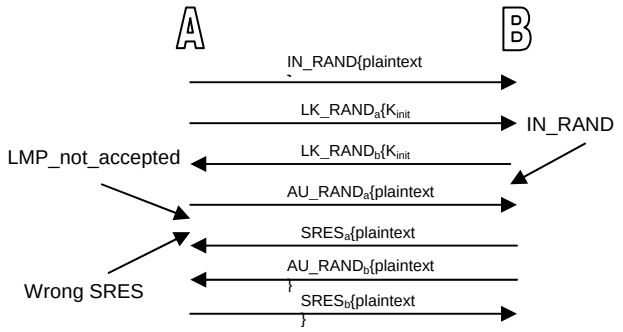

Figure 3: Transmissions in Authentication Process and Signals Used to Force Re-Pairing

In the figure above, there are three third-party signals added to the authentication process. Only one of these signals needs to be transmitted to force a re-pairing. One such signal is an "IN_RAND" sent to the slave. Upon receiving this signal, the slave thinks that the master is trying to pair with it and so the slave will discard its unit key in order to re-pair with the master. Another signal that forces re-pairing is "LMP_not_accepted". This signal is typically used by a slave device to tell a master device that the two have not yet been paired and hence share no unit key. This may happen frequently due to unit key discarding in order to free memory for storing unit keys shared with other devices. Such a scenario occurs more often with smaller devices with limited non-volatile memory. Should the "LMP_not_accepted" signal be sent to the master right after it sends its AU_RAND value (in order to verify the slave), a re-pairing will begin. Instead of sending a "LMP_not_accepted" signal however, several wrong SRES values can also be sent in its place to force a re-pairing. Incorrect SRES values will make the master think that the slave does not have the right key and hence, re-pair with the device [9]. In all the cases, a successful re-pairing can be eavesdropped on by an attacker to start cracking the PIN. Results for the time it requires to crack a PIN are shown below.

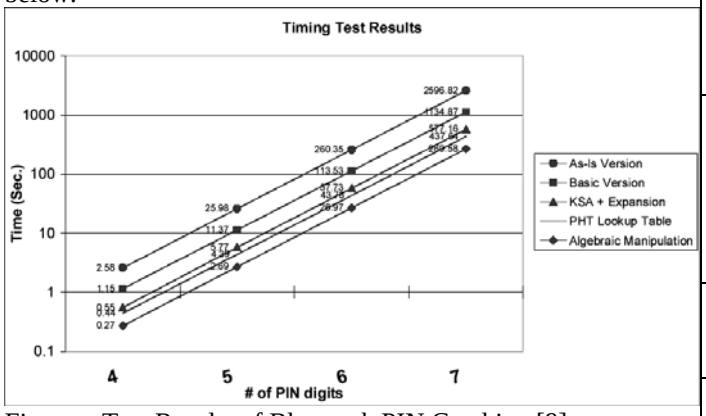

Figure : Test Results of Bluetooth PIN Cracking [9]

Note that the test results above are plots of different implementations of the PIN cracking algorithm. The faster implementations contain more optimization techniques. The test shows the time it takes to crack a PIN using a Pentium III 450MHz. With the fastest implementation, cracking the PIN takes approximately 0.27 seconds for a four digit PIN and 4.5 minutes for a seven digit PIN. With a Pentium IV 4Ghz computer, however, it only takes 0.063 seconds for a four digit PIN and 1.25 minutes for a 7 digit PIN. This means that an attacker can eavesdrop on both the devices almost in real-time when a four-digit PIN is used. The possibility of having a PIN cracked this quickly demonstrates that current Bluetooth security is extremely insufficient.

VII. Other Security Weaknesses

As Bluetooth becomes more popular and widely used, security issues have been discovered and are increasing in numbers, especially those that are in mobile phones. Security issues and vulnerabilities of Bluetooth security have brought on many threats and attacks targeting mobile phones. The table below provides a list of current security problems with Bluetooth.

	Security weaknesses/vulnerability
Encryption	Encryption of data is optional.
Default settings	Sometimes, the default configuration settings of a device are not secure. For example, functions such as authentication and encryption may be disabled or the PIN may be set to "0000". It may also be difficult to modify the default security settings [10].
Weak Pins	Pins are used for generating encryption keys and links between users. Short and trivial pins are considered to be weak PIN's; they are easy to guess or break [10].
Ways to generate and distribute Pins	Difficulties of distributing and establishing pins arise in large networks. Also, PIN's are subject to typical password problems such as: passwords getting written down, changed infrequently, forgotten or shared [11].
Unit key (link key)	Unit key is reused: after a unit key is generated, the same key is used for every connection between the two devices, thus making connections insecure. So if an attacker successfully obtains the unit key, he can monitor all traffic between the two devices [10].
Random number generation	Bluetooth currently does not have a specific mechanism to generate random numbers. So the manufacturers need to have their own way of generating random numbers. Thus the quality of such numbers may vary among different manufacturers [10].
User authenticat-ion	Currently only device authentication is provided, Bluetooth does not provide any form of user authentication [11].
Bluetooth pairing process	Pairing is recommended to proceed in a private location. Pairing in public is vulnerable to a PIN cracking attack.
Link security	End-to-end security is not provided. Only individual links are encrypted and authenticated [11].
Limited security	Audit, non-repudiation, and other services are not provided [11].

services	

Table 1: Current Bluetooth Security Problems

VIII. Attacks on Bluetooth Security

Different kinds of attacks and flaws have been discovered and will be discussed in the following section. These attacks target the vulnerabilities of Bluetooth mobile phones and bring loss of resource confidentiality, integrity, and availability.

A. Bluesnarfing

This attack allows illegal connection to a device silently without alerting the owner of the device and the need of the approval from the owner. Attackers can then gain access to any stored data and even to the restricted area of the memory, including the phonebook, pictures, settings, messages, call history, and phone serial numbers. Attackers typically use the Bluesnarfing attack to copy the content of the phone which may result in "phone cloning", where another cellular phone makes phone calls under the attacked mobile account. The attack can also be used to make phone calls and send text messages without the user's consent. It is usually possible only when the device is switched to "discoverable" or "visible" mode from "invisible" mode when Bluetooth is enabled. However, it has recently been found that tools that allow the attack to be possible even in "invisible" mode are available on the internet [12].

The procedure of Bluesnarfing becomes easier with software assistance. The software used for the procedure had to be run on a laptop in its early stages of development, but this required the attack to be done within short range from the targeted device, thus making the attack subject to a great risk of being noticed. Today, the attack is lest suspicious because "Bluesnarfing software written in Java can run on any J2ME-enabled cell phone." [13]

Bluesnarfing works through the mechanism for exchanging objects, using the OBEX protocol. OBEX is used to exchange all kind of objects such as files, pictures, calendar entries, etc. [14] The Bluesnarfing software program tries to connect to the target Bluetooth device through the OBEX Push profile of Bluetooth. But it uses the "pull" function instead of the "push" function to obtain any stored data on the device [13]. The OBEX Push profile of Bluetooth is a severe vulnerability of Bluetooth technology which was implemented in earlier Bluetooth mobile phones. OBEX Push can be performed without authenticating with the device. This is because it was originally designed to send non-malicious data; particularly business cards. Mobile phone engineers deemed it unnecessary to authenticate for such simple exchanges and so, the implementation of business card exchanges bypassed the Bluetooth security mechanisms. Firmware upgrades are now available to correct the problem [13].

B. Backdoor Attack

This attack also involves with illegal connection which will lead to disclosure of personal data, the only difference is that a trust relationship needs to be established through the Bluetooth "pairing" process. The attacking device is then deleted from the target's pair list once the connection is successful in order to ensure that it is no longer recorded in the paired device register. Since the owner of the device is not likely to be observing the pair list all the time or in the exact moment when the connection is establishing, it is difficult for them to notice the connection which is no longer trusted. The attacker is then able to continue accessing the phone freely with the privilege for a trusted relationship. The additional privilege includes accessing the Internet, WAP or GPRS gateways other than just the stored data in the phone [15].

C. Bluejacking

Bluejacking is very different from other attacks. The purpose of Bluejacking is to allow the attacker to send anonymous messages to a device instead of collecting data, with the means of OBEX's "push" function as mentioned earlier because authentication is not required. The message containing the device's name will be displayed on the targeted device during the Bluetooth initialization pairing process. Bluejacking is often harmless and does not involve with any data accessing, removal or alternation. The attack usually performs as a joke to get the reaction from some users by changing the device's name. However, it is also possible for bluejacking to be used for malicious activities. If the attacker has purposely changed the device's name to something like "Click accept if you are smart!" It is easy for the targeted user to fall for the trick and allow the attacker to gain access to his device [16].

D. Bluebug Attack

This attack creates a serial connection to the victim's phone which allows the attacker to access the phone command using Bluetooth technology. The procedure of the attack usually begins with OBEX pushing, as with Bluesnarfing and Bluejacking, which does not require any authentication or pin entry. "Then, due to flaws in the phone's Bluetooth implementations, a Bluebugger could interrupt the sending process, and the Bluebugger's phone would remain listed in the victim's phone as a "trusted" device."[13] The Bluebugger could then enter Ascii Terminal (AT) commands to control the phone [13]. AT commands are very common for the configuration and control of telecommunication devices. Attackers will then be allowed to make phone calls, send and receive messages, access and edit the phonebook and phone settings, and connect to the Internet (similar to Bluesnarf and Backdoor attacks). In addition, the Bluebug attack allows the attacker to eavesdrop on a phone conversation. It has been discovered that it is even possible to track other nearby phone

conversation if the targeted (attacked) device is on a GSM network [1], [12].

E. Cabir Worm

Bluetooth mobile phone is also vulnerable to worms and viruses, as with computers. The Cabir worm is malicious software classified as a self-replicating worm. It tries to pair the Bluetooth device it is on with other targeted device. When the pairing is successful, it installs itself to the targeted device and repeats this process to other similar vulnerable devices. The downside of Cabir worm is that it drains the battery of the device whenever it's searching for other enabled Bluetooth devices, making its presence somewhat detectable in mobile devices [1], [16].

F. Denial of Service (DoS) attacks

DoS attacks are also possible on Bluetooth mobile phones. This works the same as the traditional Dos attacks; the attackers simply continue sending invalid requests to the Bluetooth enabled device and occupying the Bluetooth channel of the device. These invalid requests are considered as invalid Bluetooth OBEX messages. DoS attack will not only drain the battery of the device but keep the Bluetooth channel of the device busy disabling communication with other Bluetooth devices. This attack greatly affects the availability of Bluetooth networks [1], [16].

IX. Insight and Concluding Remarks

Bluesnarfing, Backdoor, and Bluebugging attacks are the result of implementation errors by the manufacturers of mobile phones. Work has been carried out to correct faulty models and to ensure the same vulnerability will not be suffered by future products. Most of these attacks are not likely to happen when Bluetooth on the mobile phone disabled [1]. Other weaknesses of Bluetooth mobile phone such as Bluejacking and worm issues can be avoided if the device is carefully used, because these attacks require the user to accept the pairing process [16]. Although attacks can be avoided, the issue of easy to crack PIN's still remains.

Improving Bluetooth security should be a priority. The current level of security offered is no better than WEP encryption for wireless networking. Since Bluetooth and wireless networking have overlapping issues, we can examine improvements made by WPA and WPA2 on WEP encryption in order to improve Bluetooth security.

The main problem with Bluetooth security is the ease of cracking PIN. Hence, it is necessary to increase the PIN length. In addition to this, it should not be possible to derive the PIN from the authentication process. A solution to this problem would be to create temporary keys derived from the PIN such that the PIN is not directly used. Automatic re-keying should also be implemented such that even if a key is compromised, the key cannot be used to decipher all communication between two devices. This has been implemented in WPA2 and has proven to be effective in keeping the master key (in this case, the PIN) safe from attacks. The PIN length should be increased in such a way such that by the time a key derived from the PIN can be compromised, the re-keying process should have already created a new key for future communications. This will prevent an attacker from successfully collecting any data.

Another issue that needs to be addressed is that of fail-safe defaults. When two devices are paired, the two devices are given full access rights to each other's resources. Thus, when security is compromised, all devices within its network are also compromised. Bluetooth security should have access control which limits the potential hazards resulting from a compromise.

References

[1] The Bluetooth Special Interest Group. *Bluetooth: The Official Bluetooth Membership Site*. [Online]. Available: http://www.bluetooth.com

[2] Kraemer. James. (2005, August). *Testing and Qualifying a Bluetooth Design. Smart Modular Technologies Inc.* [Online]. Available: http://www.eetasia.com/ARTICLES/2005AUG/B/2005AUG01_RFD_DT_TA.pdf

[3] Pravin Bhagwat and Srinivasa Rao. *On the Characterization of Bluetooth Scatternet Topologies*. Submitted for publication.

[4] Juha T. Vainio (2000, May 25). *Bluetooth Security*. [Online]. Available: http://www.niksula.cs.hut.fi/~jiitv/bluesec.html

[5] Tom Karygiannis, Les Owens (2002, Nov.) *Wireless Network Security* [Online]. Available: http://csrc.nist.gov/publications/nistpubs/800-48/NIST_SP_800-48.pdf

[6] M Freitas (2003, Jan. 10). *What is Bluetooth?* [Online] Available: http://www.geekzone.co.nz/content.asp?contentid=108

[7] David Blankenbeckler (2005, Dec. 2). *An Introduction to Bluetooth* [Online] Available: http://www.wirelessdevnet.com/channels/bluetooth/features/bluetooth.html

[8] John Howie (2005, March). *Bluetooth Security Essentials* [Online]. Available: http://www.windowsitpro.com/Article/ArticleID/45210/45210.html

[9] Y. Shaked and A. Wool. Cracking the Bluetooth PIN. In Proc. 3rd USENIX/ACM Conf. Mobile Systems, Applications, and Services (MobiSys), pages 39-50, Seattle, WA, June 2005.

[10] Bundesamt für Sicherheit in der Informationstechnik. (2003). *Bluetooth: Threats and Security Measure*. Available: http://www.bsi.bund.de/

[11] Karygiannis, Tom & Owens, Les. *Wireless Network Security*. National Institute of Standards and Technology Special Publication pp. 800-48.

[12] Newitz, Annalee. "They've Got Your Number…" *Wired*. December 2004, 92.

[13] Legg, Gary. (2005, Aug 4). *The Bluejacking, Bluesnarfing, Bluebugging Blues: Bluetooth Faces Perception of Vulnerability*. [Online]. Available: http://www.techonline.com/community/tech_topic/bluetooth/38467

[14] Janssens, Sil. (2005). *Bluetooth Security Tools*. [Online]. Available: http://student.vub.ac.be/~sijansse/2e%20lic/BT/Tools/Tools.html#tth_chAp2

[15] Laurie, Adam and Laurie, Ben. (2003). "Serious flaws in Bluetooth security lead to disclosure of personal data", *TheBunker*. [Online]. Available: http://www.thebunker.net/security/bluetooth.htm

[16] Walsh, Stephen, Wan , Jun , & Sadlier, Arran. "Bluetooth Security", *Technology Survey*. [Online]. Available: http://ntrg.cs.tcd.ie/undergrad/4ba2.05/group15/