Multilevel Security Algorithm for Bluetooth Technology

Bhanvadiya Manthan A¹, Dr. Ajay shanker Shingh²

¹M.Tech Student Computer Engineering, R K University, Rajkot ²Associate Professor, School of Computer Science and Engineering, R K University, Rajkot ¹Manthan.bhanvadiya@gmail.com

²ajaysingh@rku.ac.in

Abstract:- Bluetooth is a low cost, short range and low power radio technology, which is used to connect the devices such as mobile phone handsets, portable computers etc. to each other without cables or any other physical medium. Bluetooth technology is used primarily to establish wireless personal area networks (WPAN). In my research we discuss and innovating novel security solution in Bluetooth. We will also describe vulnerabilities in Bluetooth technologies and threats against those vulnerabilities. Based on the common vulnerabilities and threats, recommendations for possible countermeasures that can be used to improve Bluetooth security are also made.

Key Word: – Secure and Fast Encryption Routine, Authentication, Pairing, Encryption and Decryption.

I. INTRODUCTION

Bluetooth is a short range wireless communication technology. developed to use for home, office and mobile Personal Area Networks [1]. Today, Bluetooth is successfully integrated into mobile phones, Personal Digital Assistants (PDAs) and other consumer devices to communicate them. Bluetooth technology was invented in 1994 by Ericson but version 1.0 of Bluetooth came out in 1999. Today, more than one billion Bluetooth devices are used by the consumers all over the world [2].

Bluetooth Technology is a combination of hardware and software. The hardware is riding on a radio chip. On the other hand, the main control and security protocols have been implemented in the software. By using both hardware and software Bluetooth has become a smart technology for efficient and flexible wireless communication system. Bluetooth radio chip supports communication among a group of electronic devices. Once the hardware radio chips are installed into the electronic devices then after wireless communication can be established among these devices. The operating distance between two Bluetooth devices ranges from 10 and 100 meters. By using a directional antenna and an amplifier the range of Bluetooth can be extended over a mile away. One of the major advantages of Bluetooth technology is that it operates in a license-free Industrial, Scientific and Medical (ISM) band ranging from 2.4 to 2.4835 MHz. This band is divided into 79 channel each being 1MHz wide. Using Fast Frequency Hopping Sequence (FFHS) a Bluetooth device hops from one channel to another channel up to 1600 times in one second [3].

In this paper, we will provide some background information about Bluetooth system, its applications and various security issues involve in Bluetooth, mainly authentication, encryption, and key management. We will also describe vulnerabilities in Bluetooth technologies and threat against those vulnerabilities. Based on the common vulnerabilities and threats, recommendations for possible counter measures that can be used to improve Bluetooth security mechanism. This provides better understanding of the security problem, current solution space, and future research scope to resolve various security issues involve in Bluetooth security.

1.1 BLUETOOTH TECHNOLOGY

A piconet contains maximum eight active BT devices that include one master and seven active slaves at a time. The master manages and schedules data transmission and channel allocation to its slaves [2]. A master sends packet to slave in even-numbered slots and slave replies to master in subsequent odd-numbered slot. Slaves are allowed only to send packet in response of master's packet. All slaves listen the master, slave is allowed to transmit only after being addressed by the master.

Bluetooth is a wireless RF communication system using mainly omni-directional antennas. Communication with other Bluetooth devices is possible within the range and no direct line-of-sight between the communicating Bluetooth devices is required. This capability makes Bluetooth communication much easier to use than the traditional cable-based communication or very short range direct line-of-sight infrared communication, but on the other hand it also makes eavesdropping much easier. The technology was originally designed for short range personal area networks, but the widespread use of Bluetooth interfaces in consumer portable electronics has opened the door to new forms of exploitation. That is, instead of pointto-point communication, using message broad-casting. 95% of mobile phones sold today are Bluetooth enabled, which is massive increase from 2008, when a mere 5% had the Bluetooth technology. Statistics also show that over 70% of consumers leave their Bluetooth switched on throughout the day, meaning more people than ever are making use of Bluetooth's magic [1].

There are several security algorithms available to ensure the security in wireless network devices. Some of the major methods are AES, DES, Triple DES, IDEA, BLOWFISH, SAFER+,ECDH etc. The SAFER+ algorithm is based on the existing SAFER family of ciphers. Although SAFER+ is the most widely used algorithm, it seems to have some vulnerabilities. This proves that proposed SAFER+ algorithm has better data throughput and frequency than the existing algorithms.

II. BLUETOOTH SECURITY ATTACKS

A. THREADS AND VULNERABILITIES IN

BLUETOOTH TECHNOLOGY

Bluetooth devices are exposed to malicious intervention during the process of pairing with another device. These weaknesses are primarily due to flaws in the link key establishment protocol, which is required for devices to pair, and the fact that the encryption of a session is optional and created at the end of the pairing process. It means that the various types of attacks can be performed well before pairing is complete. Even after the pairing is complete, the attackers can still sniff the airwaves to gain enough information to steal link keys so that they can deceptively authenticate or perform Man-in-the-Middle (MITM) attacks to impersonate other devices.

Some other reported attacks on the Bluetooth security are (1) MAC spoofing attack, (2) PIN cracking attack, (3) Man-in-the-Middle/Impersonation attack, (4) BlueJacking attack, (5) BlueSnarfing attack, (6) BlueBugging attack, (7) BluePrinting attack, (8) Blueover attack, (9) off-line PIN recovery attack, (10) brute-force attack, (11) reflection attack, (12) backdoor attack, (13) DoS attack, (14) Cabir worm, (15) Skulls worm, and (16) Lasco worm [4-7].

III. OVERVIEW OF BLUETOOTH SECURITY

This papers deals with the mechanisms used in Bluetooth Security Mode 3: The Link-level security mode. In this mode, a Bluetooth device will initiate security measures before a channel is established. This is a built-in mechanism, that is used regardless of the application layer security that may also be used. In security mode 3 terminology, establishing a channel between two Bluetooth devices is called pairing or bonding [8].

A. THE BLUETOOTH PAIRING & AUTHENTICATION PROCESS

The Bluetooth initialization procedures consists of 3 or 4 steps:

- 1. Creation of an initialization key (K_{init}) .
- 2. Creation of a link key (K_{ab}) .
- 3. Authentication.

After the 3 pairing steps are completed, the devices can derive an encryption key to hide all future communication in an optional fourth step.

Before the pairing process can begin, the PIN code must be entered into both Bluetooth devices. Note that in some devices (like wireless earphones) the PIN is fixed and cannot be changed. In such cases, the fixed PIN is entered into the peer device. If two devices have a fixed PIN, they cannot be paired, and therefore cannot communicate. In the following sections we go into the details of the steps of the pairing process.

1) CREATION OF KINIT

The K_{init} key is created using the E_{22} algorithm, whose inputs are:

- 1. BD_ADDR.
- 2. the PIN code and its length.
- 3. 128 bit random number *IN_RAND*.

This algorithm outputs a 128 bit word, which is referred to as the initialization key (K_{init}).

Figure <u>1</u> describes how K_{init} is generated using E_{22} . Note that the PIN code is available at both Bluetooth devices, and the 128 bit IN_RAND is transmitted in plaintext. As for the BD_ADDR : if one of the devices has a fixed PIN, they use the BD_ADDR of the peer device. If both have a variable PIN, they use the PIN of the slave device that receives the IN_RAND . In Figure <u>1</u>, if both devices have a variable PIN, BD_ADDR_B shall be used. The Bluetooth device. This is usually done before connection establishment begins. A detailed explanation of the inner design of E_{22} algorithm can be found in figure 8.

This initialization key (K_{init}) is used only during the pairing process. Upon the creation of the link key (K_{ab}) , the K_{init} key is discarded.



Fig. 1: Generation of K_{init} using E_{22}

```
2) CREATION OF K_{AB}
```

After creating the initialization key, the devices create the *link key* K_{ab} . The devices use the initialization key to exchange two new 128 bit random words, known as LK_RAND_A and LK_RAND_B . Each device selects a random 128 bit word and sends it to the other device after bitwise xoring it with K_{init} . Since both devices know K_{init} , each device now holds both random numbers LK_RAND_A and LK_RAND_B . Using the E_{21} algorithm, both devices create the *link key* K_{ab} . The inputs of E_{21} algorithm are:

- 1. a*BD_ADDR*.
- 2. The 128 bit random number *LK_RAND*.

Note that E_{21} is used twice is each device, with two sets of inputs. Figure <u>2</u> describes how the link key K_{ab} is created. A detailed explanation of the inner design of E_{21} can be found in Figure 9.



Fig 2. Generation of K_{ab} using E_{21}

3) MUTUAL AUTHENTICATION

Upon creation of the link key K_{ab} , mutual authentication is performed. This process is based on a challenge-response scheme. One of the devices, the verifier, randomizes and sends (in plaintext) a 128 bit word called AU_RAND_A . The other device, the claimant, calculates a 32 bit word called *SRES* using an algorithm E_I . The claimant sends the 32 bit *SRES* word as a reply to the verifier, who verifies (by performing the same calculations) the response word. If the response word is successful, the verifier and the claimant change roles and repeat the entire process. Figure <u>3</u> describes the process of mutual authentication. The inputs to E_I are:

- 1. The random word AU_RAND_A .
- 2. The link key K_{ab} .
- 3. Its own Bluetooth device address (BD_ADDR_B) .

A detailed explanation of the inner design of E_1 can be found in Figure 10.



Figure 3: Mutual authentication process using E_1

B. BLUETOOTH CRYPTOGRAPHIC PRIMITIVES

As we described above, the Bluetooth pairing and authentication process uses three algorithms: E_{22} , E_{21} , E_1 . All of these algorithms are based on the SAFER+ cipher [9], with some modifications. Here we describe features of SAFER+ that are relevant to our attack.

1) DESCRIPTION OF SAFER+

SAFER+ is a block cipher [9] with a block size of 128 bits and three different key lengths: 128, 192 and 256 bits. Bluetooth uses SAFER+ with 128 bit key length. In this mode, suppose the use of 128 bit key length number of SAFER+ round is 8 use of 192 bits and 256 bits key number of round 12 and 16 can be used SAFER+ Consist of:

This addition constitutes the output transformation for safer+ encryption. The encrypted text is a cipher text. The input for the decryption of the safer+ is the cipher text block of 16-bytes. The decryption begins with the input transformation that undoes the output transform in the encryption process. This block then process through the r rounds of decryption, round1 of which undoes the round of encryption, round r undoes the encryption of round1 of encryption to produce the original plaintext. The round sub

keys used for decryption used same as encryption but applied in reverse order.



The key scheduling algorithm (KSA)

The key scheduling algorithm used in SAFER+ produces 17 different 128-bit subkeys, denoted K_1 to K_{17} . Each SAFER+ round uses 2 subkeys, and the last key is used in the SAFER+ output transformation. The important details for our discussion are that in each step of the KSA, each byte is cyclic-rotated left by 3 bit positions, and 16 bytes (out of 17) are selected for the output subkey. In addition, a 128 bit bias vector, different in each step, is added to the selected output bytes.

The SAFER+ Round

As depicted in Figure 4, SAFER+ consists of 8 identical rounds. Each round calculates a 128 bit word out of two subkeys and a 128 bit input word from the previous round. The central components of the SAFER+ round are the 2-2 Pseudo Hadamard Transform (PHT), the Armenian Shuffles, and the substitution boxes denoted $\mathbf{\tilde{e}}$ and $\mathbf{\tilde{l}}$.

The Pseudo Hadamard Transform takes two input bytes and produces two output bytes, as follows:

 $PHT[a, b] = [(2a + b) \mod 256, (a + b) \mod 256]$

The Armenian Shuffle is a permutation of 16 bytes. See Figure $\frac{5}{2}$ for the Armenian shuffle order.

The substitution boxes $\mathbf{e}^{"}$ and $\mathbf{l}^{"}$ are non-linear, both replace an input byte with an output byte. Their implementation is given in equations (1) and (2):

$$e(x) = (45^{x} \pmod{257}) \mod 256$$
 (1)

$$l(x) = y \quad s.t. \quad e(y) = x \tag{2}$$

Figure 5 describes the structure of one SAFER+ round.



🕀 - bitwise xor

Fig 5: Structure of one SAFER+ round

VI. SAFER+ MODIFIED VERSION

Proposed SAFER Plus Algorithm

The Existing SAFER+ algorithm is modified to provide higher data throughput and frequency. The modified SAFER+ algorithm has three modifications when compared to the existing one.

(i) Rotation block is introduced between every round. Rotation is towards left for encryption and towards right for decryption.

(ii) The input of round 1 and the output of round 2 are Xor/Add Modulo 16 byte-by-byte to form the input of round 3. Similarly the input of round 5 and the output of round 6 are Xor/Add Modulo 16 byte-by-byte to form the input of round 7.

Copyright © IJRTS



Fig.6: Proposed SAFER+ for encryption

As stated before, all of the algorithms used during Bluetooth pairing and authentication process, use SAFER+ as is, or the modified version of SAFER+. In the remainder of this paper, SAFER+ is denoted as A_r , and the modified version of SAFER+ is denoted as A_r . Next subsections describe how E_{22} , E_{21} , E_1 are implemented using SAFER+.



Fig 7: Proposed SAFER+ for Decryption

V. SAFER+ BASED ALGORITHMS

A. Inner Design of E_{22}

 E_{22} is used to generate the initialization key. The inputs used are:

- 1. aBD_ADDR (48 bits long).
- 2. the PIN code and its length L.
- 3. a 128 bit random number *IN_RAND*.

At first, the PIN and the BD_ADDR are combined to create a new word: if the PIN contains less than 16 bytes, some of the BD_ADDR bytes are appended to the PIN. If the PIN is less than 10 bytes long, all bytes of BD_ADDR shall be used. Let **PIN'** denote the new word created, and **L'** denote the number of bytes the new word contains. Now, if L' is less than 16, the new word is cyclic expanded till it contains 16 bytes. Let **PIN''** denote this second new word. PIN'' is used as the 128 bit input key of A_r . IN_RAND is used as the 128 bit input data, after xoring the most significant byte with L'. Figure 8 describes the inner design of E_{22} .

Copyright © IJRTS



Fig. 8: Inner design of E_{22}

B. Inner design of E_{21}

 E_{21} is used to generate the link key. The inputs used are:

- 1. aBD_ADDR (48 bits long).
- 2. a 128 bit random number *LK_RAND*.

At first, the *BD_ADDR* is cyclic expanded to form a 128 bit word which is used as the input data of A_r . The key used for A_r consists of the 128 bit random number *LK_RAND*, after xoring its most significant byte with 6 (result denoted *LK_RAND'*). Figure 9 describes the inner design of E_{21} .



Fig 9: Inner design of E_{21}

C. Inner design of E_1

 E_1 is used to perform mutual-authentication. The inputs used are:

- 1. A random word AU_RAND_A .
- 2. The link key K_{ab} .
- 3. aBD_ADDR (48 bits long).

The inner design of E_1 contains both A_r and A_r' . The link key is used twice. Once, it is supplied as is for the key input of A_r . Later, it goes through a transformation denoted **Offset** and supplied as the key input of A_r' . The ``Offset" transformation consists of adding and xoring its bytes with some constants [16] .As for the *BD_ADDR*, it is cyclic expanded to form a 128 bit word denoted *BD_ADDR'*. The inner design of E_1 is depicted in figure 10.





- 16 xor operations

Fig10: Inner design of E_1

Advantages of SAFER+

- •A proven track record of security
- •Speed and simplicity
- Transparency
- Flexibility of Use
- · Flexibility of Environment

Limitations of SAFER+

•No proof of complete security

Encryption/Decryption Dissimilarity

Copyright © IJRTS

VI. REFERENCES

- J. Dunning. Taming the Blue Beast: A Survey of Bluetooth Based Threats. IEEE Security & Privacy, 8(2):20–27, Mar-Apr. 2010.
- [2] K. Scarfone and J. Padgette. Guide to Bluetooth Security. NIST Special Publication 800-121, Sep 2005.
- [3] Jochen Schiller, "Mobile Communications", Second Edition, Addison Wesley Publications, 2003, pp. 290-292
- [4] Keijo Haataja, "Security Threats and Countermeasures in Bluetooth-Enabled Systems", Kuopio University Library, 2009, pp. 68-80
- [5] Colleen Rhodes, "Bluetooth Security", East Carolina University, pp.6-9
- [6] Karen Scarfone and John Padgette, (Bluetooth Threats)
 "Guide to Bluetooth Security", Computer Security Division - National Institute of Standards and Technology, US Department of Commerce, 2008, pp. 25-26
- [7] Raquel Hill and Billy Falotico, "Bluetooth Wireless Technology Security Threats and Vulnerabilities", Indiana University Bloomington, 2008, pp. 7-8
- [8] Specification of the Bluetooth system, v.1.2. Core specification, available from <u>http://www.bluetooth.org/spec</u>, 2003.
- [9] J. L. Massey, G. H. Khachatrian, and M. K. Kuregian. SAFER+. In Proc. First Advanced Encryption Standard Candidate Conference. National Institute of Standards and Technology (NIST), 1998.