http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 5 (6): 1033-1037, 2006
ISSN 1812-5638

Avoidance of Intruder Attack with Changed Bluetooth Authentication Procedure

Pushpa R. Suri and Sona Rani
Department of Computer Science and Applications, Kurukshetra University, Kurukshetra, India

Abstract: This study focuses on the mechanism of Bluetooth bonding and authentication. When two units want to communicate in a secure way, the need to be paired to each other. In the pairing process units are exchanging keys and authenticate each other. We have mention about Bluetooth security breaks and holes for attacker using. Finally this study ends with behavior of attacker and countermeasures of it.

Key words: Bonding, encryption, authentication, initialization key, personal identification number

INTRODUCTION

When Bluetooth devices come within range of another, an electronic conversation takes place to determine whether the devices in range are known or whether one needs to control the other. Most Bluetooth devices do not require any form of user interaction for this to occur. If devices within range are known to one another, the devices automatically form a network-known as a pairing (Persson, 1999).

Authentication addresses the identity of each communicating device. The sender sends an encrypted authentication request frame to the receiver. The receiver sends an encrypted challenge frame back to the sender. Both perform a predefined algorithm. The sender sends its findings back to the receiver, which in turn either allows or denies the connection.

There are three different functions for authentication in bluetooth- E1, E2 and E3. E1 is used when encrypting the authorization challenge-response values. E2 is for generating different link keys. E3 is used when creating the encryption key.

Generation of initialization key: The creation of an initialization key is used when no other keys are present. The key is derived from a random number, a PIN, length of the PIN and a unit's hardware address. The PIN code can either be a factory value or the user can enter a maximum of 16 octets. There are three different scenarios how the PIN is used:

- If one device has a fixed PIN then will the unit address of the other be used when deriving a new link key.
- If both units have non-fixed key then will the hardware address of the unit that received the random number be used. The PIN code has to be entered in both devices that are going to be paired.
- If both units have fixed PIN they can't be paired.
- The unit address will be added to the PIN and the whole unit's address might not be in use. The initialization key is discarded when the link key is exchanged between the units, it is only used to protect the initial value that need to be protected before a regular key, link key, is established.

GENERATION OF LINK KEY AND LINK KEY EXCHANGE

When a link key is established between two units they will use that key for authentication. A link key is 128 bits long and a shared between two or more units (Miller, 2001). A new link key can be derived whenever to improve security. There are four types of different link keys:

Combination key: Combination key is based on information from both the units in the pairing process.

Unit key: A unit key is a key that one unit will use in all its connections with other users. Unit keys is preferred when on single unit is connecting a large group of users, it will only have to store one key instead of one key for each user.

Temporary key: A master key is only used temporarily to replace an original link key in a current session. It is used when a master unit wants to reach more than one slave using the same encryption key.

Corresponding Author: Sona Rani, Department of Computer Science and Applications, Kurukshetra University, Kurukshetra, India

Initialization key: A unit that is created if there are no other keys or the keys are lost. It is only used as link key during initialization.

The combination link key is a combination of two numbers generated in two devices.

- Each device creates a random number and encrypts it together with its hardware address.
- The random number is XORed with initialization key and sent away to the other unit.
- Now the two units have the other's random number. The hardware address is public so each unit can calculate their counter part's encrypted random number together with hardware address.
- Both units now do a bitwise modulo2 addition to combine the two units encrypted values.
- The result of the modulo2 addition is the combination key of the two units.
- A mutual authentication is required in order to confirm that both units have the correct combination key. After a successful authentication the old link key can be discarded.

AUTHENTICATION

The Bluetooth Authentication procedure is based on a challenge-response scheme. Two devices interacting in an authentication procedure are referred to as the claimant and the verifier. The verifier is the Bluetooth device validating the identity of another device. The claimant is the device attempting to prove its identity (Muller, 1999). The challenge-response protocol validates devices by verifying the knowledge of a secret key-a Bluetooth link key. The challenge-response verification scheme is depicted conceptually in Fig. 1. As shown, one of the Bluetooth devices (the claimant) attempts to reach and connect to the other (the verifier). The steps in the authentication process are the following:

Step 1: The claimant transmits its 48-bit address (BD_ADDR) to the verifier.
Step 2: The verifier transmits a 128-bit random challenge (AU_RAND) to the claimant.
Step 3: The verifier uses the E1 algorithm to compute an authentication response using the address, link key and random challenge as inputs. The claimant performs the same computation.
Step 4: The claimant returns the computed response, SRES, to the verifier.
Step 5: The verifier compares the SRES from the claimant with the SRES that it computes.
Step 6: If the two 32-bit SRES values are equal, the verifier will continue connection establishment.

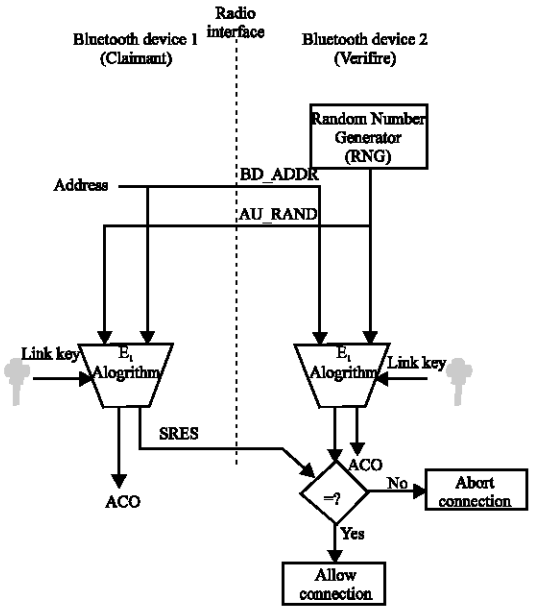

Fig. 1: Authentication process

If authentication fails, a Bluetooth device will wait an interval of time before a new attempt can be made. This time interval will increase exponentially to prevent an adversary from repeated attempts to gain access by defeating the authentication scheme through trial-and-error with different keys. However, it is important to note that this "suspend" technique does not provide security against sophisticated adversaries performing offline attacks to exhaustively search PINs.

Again, the Bluetooth standard allows both uni-directional and mutual authentication to be performed. The E1 authentication function used for the validation is based on the SAFER+ algorithm.

The Bluetooth address is a public parameter that is unique to each device. This address can be obtained through a device inquiry process. The private key, or link key, is a secret entity. The link key is derived during initialization, is never disclosed outside the Bluetooth device and is never transmitted over the air interface. The random challenge, obviously a public parameter, is designed to be different on every transaction. The random number is derived from a pseudo-random process within the Bluetooth device (SIG, 2001). With knowledge of the challenge and response parameters, it should be impossible to predict the next challenge or derive the link key.

PROBLEM IN THE CURRENT SYSTEM

When connection is made between the Bluetooth devices, an intruder device can be there in different ways

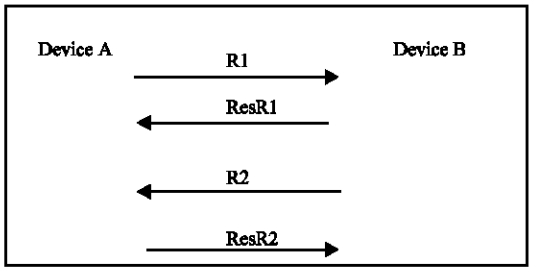

Fig. 2: Messages in existing Authentication process

(Jakobsson and Security, 2001). An intruder can act as the fake device in the different roles. The fake device can behave as false slave or false master. Similarly the intruder can be a active (changing the contents of the information) intruder or passive one. (simply coping the information and sending the same information to the another end). It can continue the connections to the both communicating (original) devices or detach the one end (the another end is considering the intruder as the real one communicating device). Messages sent by intruder C are shown in Fig. 3.

In the existing authentication scheme of bluetooth technology, mutual authentication is performed.

First one device sends the random number for authentication to device second. Then the second device sends the response and sends another random number for the verification of first device. Then the first device sends the response of random number send by second device. In this way the identification of both the devices is done.

In the above Fig. 2, device A sends a random number R1 to device B for authentication of device B. The device B sends the Res_{R1} to device A. Then the device B sends random number R2 and the device B sends the Res_{R2} to device B

Behavior of intruder C in existing scheme: Suppose an intruder C wants to make connection in between the both devices A and B.

The following messages will be sent by the intruder device C (behaving as fake B device to A and fake A to device B):

- Device A sends the random number R1 to fake device B.
- Fake device B now behaves as fake device A and sends the same random number R1 to device B.
- Now the device B sends the response Res_{R1} to fake device A (intruder C).
- Intruder's C sends the same response Res_{R1} received from device B to device A.
- Then the device B sends the authentication random number R2 to fake device A.

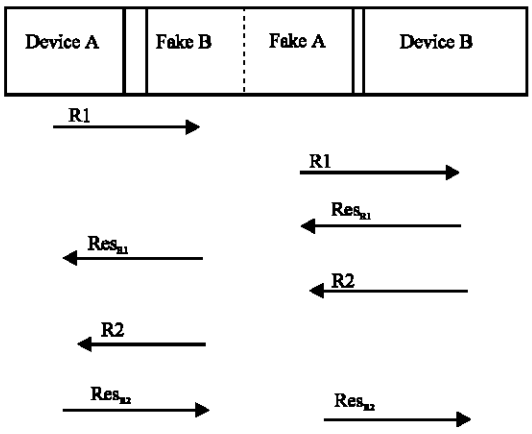

Fig. 3: Messages in existing Authentication process by intruder

- The intruder C sends the same random number R2 to device A.
- The device A sends the Res_{R2} to fake device B.
- The fake device A now sends the same response Res_{R2} to device B.
- Hence in this way the intruder device make the connections with both the devices A and B (Fig. 3).

IMPROVED AUTHENTICATION METHOD

In the Bluetooth scheme, mutual authentication is performed exclusively between master and slave. First, one is authenticated with AU_RAND (challenge) and SRES (response) exchange. Then the other is authenticated again using a challenge/response mechanism. We propose to change this authentication message exchanges in a form such that first both parties exchange their authentication random values and claimant does not send its response before getting the response from the verifier. This message exchange is shown in the Fig. 4. In this method, the attacker cannot obtain SRES values from the victims, since both victims first wait for the SRES value from the other party (i.e., from the attacker). Since the attacker acts as a verifier in both piconets, its authentication challenge is responded with another authentication challenge from the genuine entities.

With the improved authentication method, if message exchanges in a nested form such that first both parties exchanges their random values and claimant does not send its response before getting the response from the verifier. The messages are shown as below:

Now there are two cases in this authentication procedure:

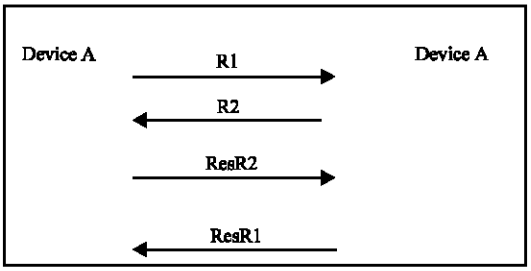

Fig. 4: Messages in new Authentication process

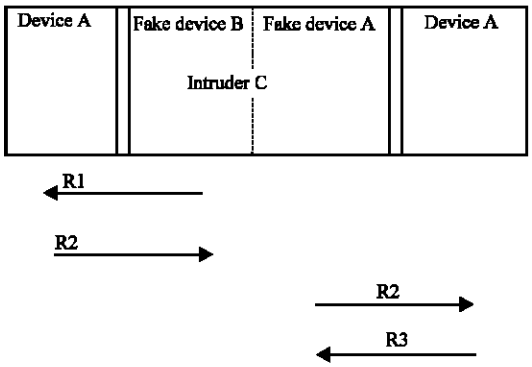

Fig. 5: Messages in improved authentication process by intruder (Case 1)

- When the request for connection is generated from the intruder device C to device A.
- When the request for connection is made from the device A to device C.

Case 1: Request from C to A: In this case when the intruder C will initiate the connection establishment procedure with device A. The following messages will be sent (Fig. 5):

- The fake device sends the random number R1 to device A.
- The device A does not sends the response for R1. It sends the another random number R2 to fake device B for authentication and waits for the response for R2.
- Suppose the fake device is trying to get the response from device B. It sends the same random number to devices B.
- The device B does not sends the response of R2, firstly it verifies the fake device A. and sends the one another random number R3 and waits for the response of R3 from fake device A.
- Hence in this case the attacker can't involve itself into the devices A and B.

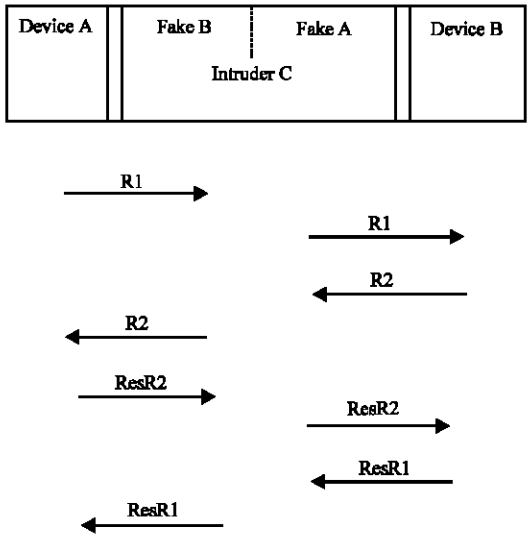

Fig. 6: Messages in improved Authentication process by intruder (Case 2)

Case 2: Request from A to C: In this case when the request is made from device A. The following messages are exchanged between the devices A, C and B (Fig. 6):

- First the authentication random number R1 is sent by device A.
- The device C sends the same random number R1 to device B.
- The device C does not send the response of R1, it sends another random number R2 to fake device A for authentication.
- The device C transmits the random number R2 to device A and waits for the response of R2 from A.
- The device A gives the response 'ResR2' of R2.
- The device C gives the ResR2 to device B.
- The device B sends the ResR1 in response to the number R1 to device C
- The ResR1 is sent as it is to device A by the device C.

Hence the connection is made between the devices A and C and C and B. but this is only possible when the request is initiated by the device A and simultaneously there is a connection between the device A and B.

In this type of attack the messages are sent as it is. The intruder copies the contents and can't alter the message. Integrity is maintained, as the intruder has no right to change the contents. But the confidentiality is disturbed.

The solution to this problem is to add some form of timestamp to the communication between devices A and

B. For example when the device A starts communication it can receive the response of its challenge within the specified time period t. If the device C will try to communicate in-between the devices A and B, it sends the same random number to device B and a time lag from the specified time is there due to exchange of messages. So by giving the specified time limit for the response time solves the problem of confidentiality.

CONCLUSIONS

While Bluetooth has several nice features, it fails to be a secure replacement for wires. As we have shown that Bluetooth is susceptible to the attacks by intruders independent of security mechanisms. If an unknown device wants to make connections or request for a service. Then proper authentication is followed by authorization and encryption. But authentication process should be such that unknown device would not get response of any random number until and unless it will give response to the random number which is sent by the device with which it wants to make connections.

The attack is based on the fact that the Bluetooth does not provide a way to verify the integrity of messages and the intruder can attack on the one slave device or more simultaneously in the same network. But if we give the provision that not any single slave will response until it verifies the identity of other device and another method is that one device can estimate the delay by observing the response time given by the verifier, so in this way we can check the identity of the device and can improve the security.

REFERENCES

Bluetooth, S.I.G., 2001. Specification of the Bluetooth system, Core", Version 1.1. available at http://www.bluetooth.com/.

Jakobsson, M. and S.W. Security, 2001. Weaknesses in bluetooth available at http: //www. rsasecurity. com/ rsalabs/staff/bios/mjakobsson/bluetooth/bluetooth .pdf .

Miller, B. 2001. IEEE 802.11 and Bluetooth wireless technology. available at http://www-106. ibm. com/ developerworks/wireless/library/wi-phone/

Muller, T. 1999. Bluetooth White Paper: Bluetooth Security Architecture, Version 1.0.

Persson, J., 1999. Bluetooth baseband security concept. Proceedings Bluetooth' 99, London, June 1999.