3GPP TSG SA WG3 Security#32 **S3-040049**
9 - 13 February, 2004
Edinburgh, UK

Agenda Item:	6.10 (WLAN)
Source:	Orange
Title:	A man-in-the-middle attack using Bluetooth in a WLAN interworking environnment
Document for:	Discussion
Author:	Eric Gauthier (eric.gauthier@orange.ch, Tel. +41 78 787 53 08)

Abstract

During the SA3-31 meeting in Munich, it was decided that the Bluetooth link between peripheral devices did not require integrity protection (see section 6.1.1 of [1]). This contribution indicates that a man-in-the-middle attack may be possible on the bluetooth link in a WLAN interworking environment. The attacker lures the victim to connect to a malicious WLAN access point. The attack does not require to know the Bluetooth link key. The attacker can repeat this attack on the same victim many times in any WLAN network. A discussion of countermeasures against this attack can be found in a companion contribution [2].

1 Introduction

We present an attack against a victim that connects a device, such as a laptop, to a WLAN network and authenticates over Bluetooth using another device, such as a mobile phone, holding a SIM or a USIM card [3]. The mobile is authenticated by an Authentication Server (AS) connected to the WLAN network by an IP network. The goal of the attack is to connect the victim laptop to a fake WLAN Access Point (AP). This is achieved by a man-in-the-middle attack on the Bluetooth link between the laptop and the mobile station. The attacker has a device able to receive Bluetooth packets in promiscuous mode and send forged ones to the mobile and the laptop of the victim, as shown in Figure 1.

We make the following further assumptions about the attack:

- **The laptop is the Bluetooth slave.** We suppose for simplicity that the mobile station is the master and the laptop is the slave of the Bluetooth link. This assumption is not strong since the Bluetooth protocol allows the master and slave to switch their role. The attacker device acting as the man-in-the-middle could force a master-slave switch.
- **Authentication is terminated in the mobile.** The mobile and the AS implement the EAP-AKA authentication method [4]. The attack works also for EAP-SIM [5]. Both the mobile and the AS derive two keys: a master key from the UMTS ciphering and integrity keys [4], and a Master Session Key (*MSK*) from the master key [6].
- **Some access points can be compromised.** We suppose the attacker can compromise at least one access point to obtain the *MSK*. This is perhaps the strongest assumption.

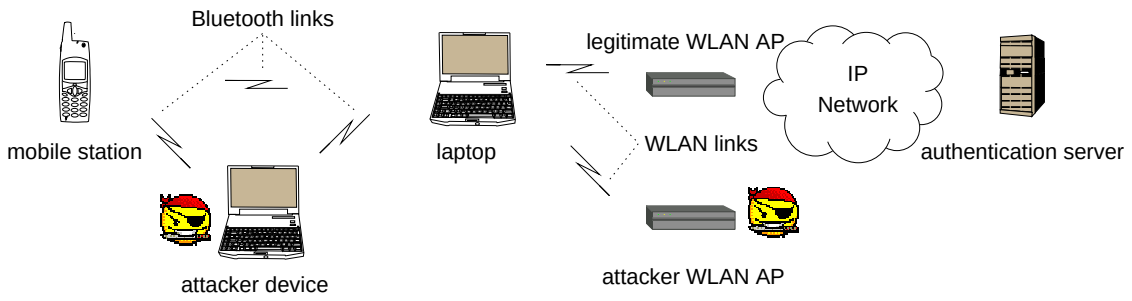

Figure 1: Elements of the attack: the laptop requires access to an IP based service through the network WLAN infrastructure. Authentication is provided by the mobile that is linked to the laptop by Bluetooth. Authorisation is provided by an authentication server that communicates with the WLAN access point over an IP based network. The attacker requires two devices: a Bluetooth device that intercepts the communication between the victim mobile and the laptop and a WLAN access point.

> However, some access points have shown recently to leak some keying information [7]. Furthermore, many access points have still low physical security and could be tampered with.

The *MSK* is used as the ciphering key for the WLAN link. The *MSK* is transmitted by the AS to the WLAN Access Point (AP) using an AAA protocol such as RADIUS [3]. We assume the communication between the AS and the AP is properly protected using TLS or IPSec. The mobile transmits the *MSK* to the laptop using an unspecified protocol over Bluetooth.

This document does not discusses countermeasures to this attack. Such discussion can be found in a companion document [2].

We describe the attack in Section 2 and discuss why it works in Section 3.

2 The attack

We suppose initially that the mobile and the laptop are already Bluetooth paired and have derived a Bluetooth link key K that is semi-permanent. The attacker does not know K.

We divide the attack in two phases. The first phase allows the attacker to passively record the Bluetooth session during which the victim mobile sends the *MSK* to the victim laptop. The attacker also obtains the *MSK* by compromising the access point used by the victim. In the second phase, the attacker forces the laptop to use the compromised *MSK* by replaying the session recorded during the first phase. As a result, the victim laptop connects – without being aware of it – to the attacker's access point that uses the compromised *MSK*. The attacker can repeat the second phase of the attack on the same victim many times in any WLAN network. We now describe both phases of the attack in more details.

2.1 Recording the Bluetooth session

We describe the session that the attacker records on the Bluetooth link between the victim mobile and laptop. Initially, the mobile and the laptop of the victim mutually authenticate each other using the Bluetooth Link Management Protocol (LMP), as shown in Figure 2. The mobile (the master) sends a LMP au_rand message with a challenge *RAND1* to the laptop. The laptop computes the response *RES1* and sends it back to the mobile using a LMP sres message. Similarly the laptop authenticates the mobile sending the challenge *RAND2* and verifying the received response *RES2*. Then the mobile (the master) initiates the Bluetooth encryption by sending a LMP start_encryption that carries a random number *EN_RAND*.

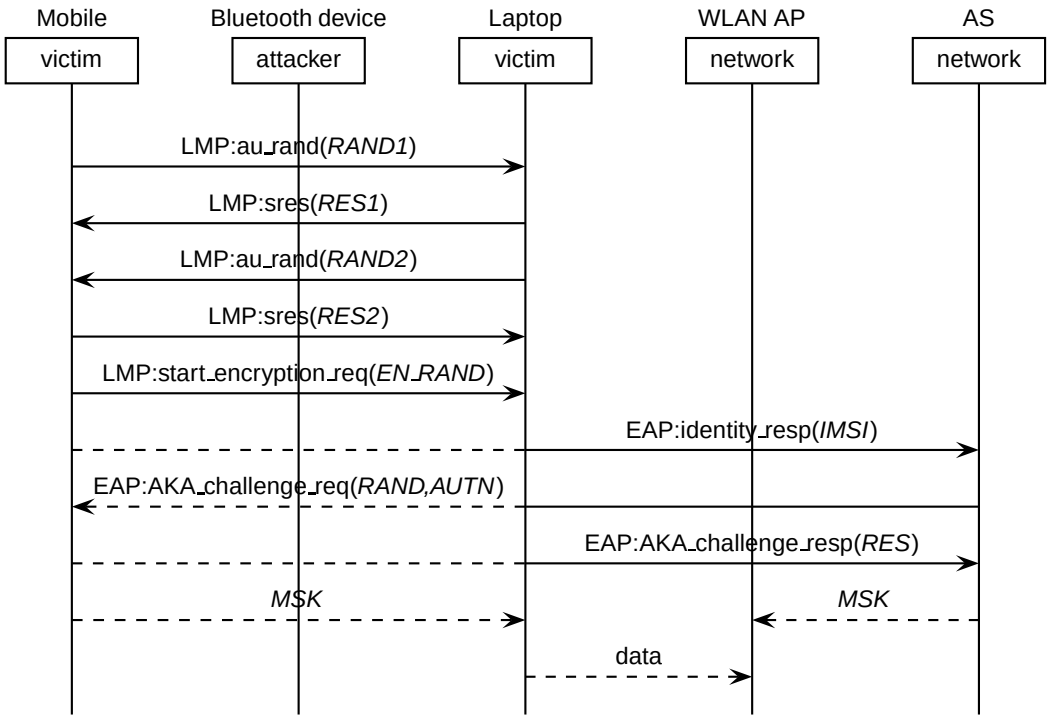

Figure 2: First phase of the attack: the attacker records the Bluetooth packets between the victim mobile and laptop. The packets captured must include the authentication, the encryption command and the encrypted communication of the Master Session Key (*MSK*). Encrypted messages are shown as dashed lines. The attacker also obtains the *MSK* by compromising the access point used by the victim.

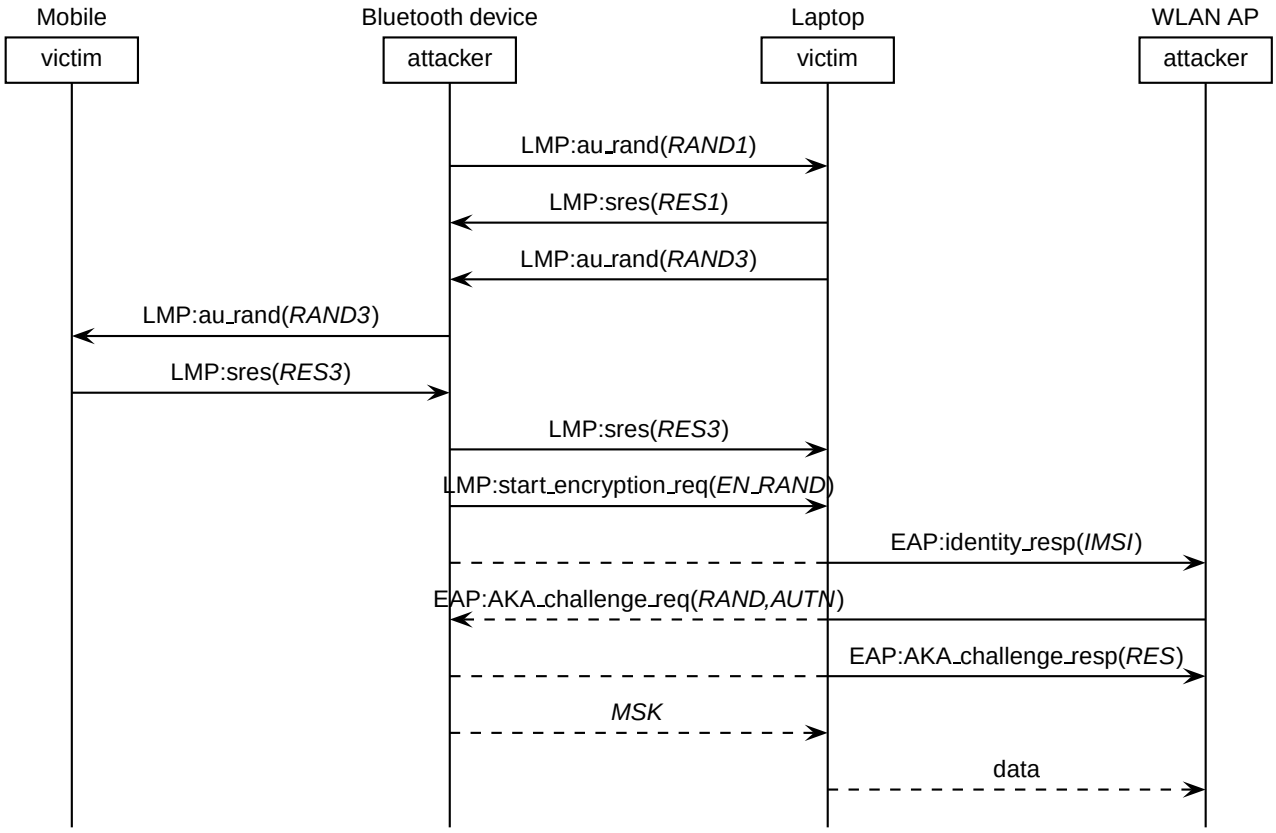

Figure 3: Second phase of the attack: the attacker replays the Bluetooth traffic to the victim laptop. The attacker acts also as a false laptop since mutual authentication is used on the Bluetooth link. The victim laptop connects to the attacker access point without being aware of it. The attacker can repeat this phase of the attack on the same victim many times in any WLAN network.

Then the mobile station and the AS mutually authenticate using EAP authentication. The mobile sends a EAP identity_resp message to the AS that indicates the identity *IMSI* of the card held by the mobile. The AS sends back a EAP AKA_challenge_req message to the mobile with an AKA challenge *RAND* and a network authentication token *AUTN*. The mobile station verifies the authentication token, computes a response *RES* and sends it back to the AS using the EAP AKA_challenge_resp message.

Finally, the mobile and the AS compute the session key *MSK*. The *MSK* is transmitted by the AS to the WLAN Access Point (AP) in a RADIUS access_accept message. The mobile transmits the *MSK* to the laptop using an unspecified message encrypted by Bluetooth. The laptop and the AP then exchange data encrypted using *MSK* over the WLAN link.

2.2 Replaying the Bluetooth session

We now describe how the attacker can replay the recorded session to force the laptop to reuse the compromised *MSK*. Initially, the attacker and the laptop of the victim mutually authenticate each other as shown in Figure 3. The attacker first sends the challenge *RAND1* recorded in the first phase to the laptop and receives the response *RES1*. The laptop then sends a new challenge *RAND3* to the attacker that forwards it to the mobile. The mobile computes the response *RES3*, sends it to the attacker that forwards it to the laptop. The attacker starts the Bluetooth encryption using the same challenge *EN_RAND* recorded during the first phase. The attacker then replays the EAP authentication sequence that the laptop simply forwards to the WLAN AP controlled by the attacker. The EAP messages are not forwarded by the attacker AP. Finally, the attacker sends to the laptop the message containing the compromised *MSK* recorded during the first phase and Bluetooth encrypted. The victim sends encrypted data to the attacker WLAN thinking being connected to a legitimate network.

3 Why it works

This attack is based on the fact that Bluetooth does not provide a way to verify the integrity and freshness of messages. We now show that the attacker can replay Bluetooth encrypted messages. During the Bluetooth authentication, the mobile sends a Bluetooth challenge *RAND1* to the laptop that answers with the Bluetooth response *RES1*. Upon receiving the response, the mobile computes the Authentication Ciphering Offset (*ACO*) as follows:

$$ACO = E_1(K, RAND1, ADD_{laptop})$$

where ADD_{laptop} is the address of the Bluetooth adapter in the laptop and E_1 is a hash function specified in [8]. The laptop sends also a challenge *RAND2* to the mobile if mutual authentication is used. The mobile then sends back a response *RES2* to the laptop. It appears however that Bluetooth does not use *RAND2* in the computation of *ACO* [8].

Then the mobile sends an encryption command to the laptop indicating a random number *EN_RAND*. This number is used to compute the value of the Bluetooth KC as follows:

$$KC = E_3(K, EN\_RAND, ACO)$$

where E_3 is specified in [8]. Finally, the ciphering keystream K_{cipher} is generated using the following equation:

$$K_{cipher} = E_0(KC, CLK_{MS}, ADD_{MS})$$

where CLK_{MS}, ADD_{MS} are the master clock and the address of the Bluetooth adapter of the mobile and E_0 is specified in [8].

The K_{cipher} does not depend on any random number generated by the laptop. Therefore the attacker device can replay the sequence of Bluetooth encrypted messages to the laptop.

4 Acknowledgements

The author would like to thank Jari Arkko, Florent Bersani, Benoît Calmels, Sharat Chander, Sylvie Fouquet, Guenther Horn and Stefan Schröder for their comments that improved significantly the document.

5 References

[1] 3GPP Technical Report on (U)SIM Security Reuse by Peripheral Devices on Local Interfaces, TR ab.cde V0.7.0 32, Oct 2003.

[2] Siemens, “Notes on Gauthier’s replay attack on the UE functionality split scenario”, Contribution S3-xxxxxx, SA3-32 Meeting, Edinburgh, Feb 2004.

[3] 3GPP Technical Specification Group Service and Systems Aspects, TS 33.234, Nov 2003

[4] J. Arkko and H. Haverinen, “EAP AKA Authentication”, Internet Draft, draft-arkko-pppext-eap-aka-11.txt, Oct 2003.

[5] H. Haverinen, J. Salowey, ”EAP SIM Authentication”, draft-haverinen-pppext-eap-sim-12.txt, Oct 2003

[6] Aboba, B., et al., “EAP Key Management Framework”, Internet Draft, draft-ietf-eap-keying-02, Nov 2003.

[7] Cisco Security Advisory, “SNMP Trap Reveals WEP Key in Cisco Aironet Access Point”, http://www.cisco.com//warp/public/707/cisco-sa-20031202-SNMP-trap.shtml, Dec 2003.

[8] Specification of the Bluetooth System, version 1.1, Feb 2001.