3GPP TSG SA WG3 Security#32 **S3-030123**
9 – 13 February, 2004
Edinburgh, UK

Agenda Item: **6.10 (WLAN)**

Source: **Siemens**

Title: **Notes on Gauthier's replay attack on the UE functionality split scenario**

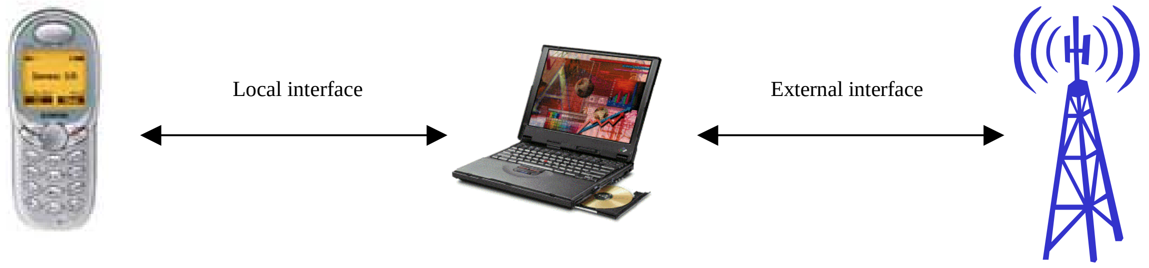
Document for: **Discussion**

Abstract

Eric Gauthier presented an attack [S3-040049] on the UE functionality split scenario as specified in TS 33.234 v100, section 4.2.4, which was briefly discussed on the SA3 mailing list between this and the last SA3 meeting. It is shown in [S3-040049] that an attacker who was able to compromise an EAP master session key MSK once can impersonate a WLAN AP an indefinite number of times. In [S3-040049], only the case was discussed where the mobile terminates EAP, derives the MSK and transfers it to the laptop. This contribution shows that the attack also works with different ways of splitting the functionality between mobile and laptop. The contribution also discusses a number of countermeasures. It is further proposed to forward [S3-040049] and this contribution to the Bluetooth security group for comment.

This document is an update of S3-040091, taking into account comments received by Tao Haukka and Peter Howard. The updated parts are marked.

The following familiar figure depicts a WLAN_UE functional split scenario:



The mobile phone (the "card holding device") on the left holds the UICC or SIM (the "card") and the laptop holds the WLAN card.

The reader is referred to ***[S3-040049]*** for a description of the attack (provided in the same zip.file as this contribution).

Abbreviation: BT = Bluetooth

1. Consideration of alternative functionality splits

We discuss now whether it makes a difference for the attack which key material is derived in the mobile and transferred to the laptop over Bluetooth.

We briefly recap the **key hierarchy**: the top level keys are the GSM session keys Kc1, Kc2, ... for EAP-SIM and the UMTS session keys CK, IK for EAP-AKA. From these, the EAP master key MK is derived. For EAP-SIM there is further random input from the EAP client to the derivation of MK, for EAP-AKA there is no further random input to the derivation of MK. (The reason is that, for EAP-SIM, the EAP client needs to provide the replay protection in the EAP authentication as it is not there in GSM, for EAP-AKA the USIM provides the replay protection in the EAP authentication.) From this point on, EAP-SIM and EAP-AKA proceed in the same way. A master session key MSK as well as further keys K_encr and K_aut used to encrypt and integrity-protect EAP messages are derived from MK in a deterministic way. MSK is sent from the EAP server to the WLAN AP, while K_encr and K_aut remain in the EAP server. The MSK is used in a deterministic way to obtain the pairwise master key PMK from which the WLAN link layer keys are derived.

Case 1: the entire EAP client is located on the mobile, MSK is transferred over BT (same as alternative 2 in S3-030747): in this alternative all EAP messages are transferred over the BT. This case was discussed in [S3-040049], the attack works as shown in [S3-040049].

Case 2: EAP-AKA, the EAP client is split over mobile and laptop, MK is transferred over BT (same as alternative 3 in S3-030747): in this alternative only the parameters needed to derive MK (UMTS challenge, EAP identity) are transferred from the laptop over the BT link, the EAP messages are handled in the laptop otherwise. This case was not discussed in [S3-040049], but the attack works in precisely the same way as shown in [S3-040049]. The reason is that

- the attacker, when impersonating the mobile, can force the laptop to use the same MK repeatedly by replaying the same encrypted data sequence over Bluetooth, as the laptop has no way of telling whether the MK is fresh.

- the attacker, when impersonating the EAP server, can replay the same EAP messages towards the laptop, as they do not contain any session dependent variable parameters. In particular, the MAC computed using K_aut on the message “EAP-Request/AKA-Challenge” can be replayed;

- the derivation of the keys MSK, K_encr and K_aut from MK is deterministic.

Case 2’: EAP-SIM, the EAP client is split over mobile and laptop, MK is transferred over BT (same as alternative 3 in S3-030747): in this alternative only the parameters needed to derive MK (GSM challenges, EAP identity, NONCE_MT, Version List, Selected Version) are transferred from the laptop over the BT link, the EAP messages are handled in the laptop otherwise. For this case, the attack does **not** work. The reason is that the laptop generates a nonce NONCE_MT and sends it to the EAP server. The message “EAP-Request/SIM/Challenge”, which is sent from the EAP server to the EAP client on the laptop, contains a MAC which is computed also over NONCE_MT. Consequently, the MAC needs to be different for each protocol run and cannot be replayed, and the attacker will not be able to successfully impersonate the network side.

Case 3: EAP-AKA, the EAP client is split over mobile and laptop, CK, IK is transferred over BT (same as alternative 1 in S3-030747): in this alternative only the parameters needed to derive the UMTS keys CK, IK (UMTS challenge RAND, AUTN) are transferred from the laptop over the BT link, the EAP messages are handled in the laptop otherwise. This was not discussed in [S3-040049], but the attack works in the same way as shown in [S3-040049]. The reason is the same as for case 2.

Case 3’: EAP-SIM, the EAP client is split over mobile and laptop, Kc1, Kc2, ... are transferred over BT: in this alternative only the parameters needed to derive the GSM keys Kc1, Kc2, ... (GSM challenges RAND) are transferred from the laptop over the BT link, the EAP messages are handled in the laptop otherwise. for this case, the attack does not work. The reason is the same as for case 2’.

Please note, that, when cases 3 and 3’ were considered in the Siemens contribution S3-030747, it was found that these cases implied significant security risks as a compromise of the laptop would affect other UMTS and GSM domains as well.

2. Prerequisites for the attack

Key freshness: ~~from a cryptographic point of view, the attack works because a BT slave has no assurance of encryption key freshness (while the master does). Please note that BT does provide mutual entity authentication and mutual key authentication.~~from a cryptographic point of view, the attack works when the laptop has no assurance of encryption key freshness. The BT master always has assurance of encryption key freshness as it contributes a nonce to the computation of the encryption key at the start of encryption.
BT provides mutual entity authentication and mutual key authentication. Mutual authentication is performed as a succession of two unilateral authentications. A value ACO is computed as a result of an authentication. The initiator of a unilateral authentication inputs a nonce to the computation of ACO, the responder does not. The ACO value from the authentication performed last is used to derive the encryption key. So, the initiator of the last authentication also has assurance of encryption key freshness, as long as it can be assured to have initiated the last authentication.

Compromise of an MSK: it can, of course, be speculated how likely it is to compromise a WLAN AP and obtain an MSK from this. Probably, it should not be ruled out. But the main point is that a security protocol which is insecure forever in the future if once a session key was compromised is quite weak. So, the effect of the compromise of an AP reaches far beyond the time the attacker actually controls the AP and the attack does not need to make use of any legitimate AP.

EAP keys K_encr and K_aut: please note that it is not necessary for the attacker to obtain these keys. They remain in the background authentication server and would be much more difficult to compromise.

Bluetooth Clock: the BT clock value is input to the encryption algorithm, so the attacker needs to reset the BT clock before replaying a message to the target. This may be not easy to achieve in practice, but please note that the BT master controls the BT clock and can reset it. We assume for the purpose of this contribution that an attacker can achieve this.

3. Countermeasures

Change BT baseband specification: the BT encryption key generation could be changed so as to give assurance of encryption key freshness also to the slave. This could be realistic at best in the long term as it would required hardware changes. It is not considered an option for the 3GPP Rel6 or maybe even Rel 7 timeframe.

Laptop is master: the attack works only when the laptop is the slave. The attack could be prevented if the laptop always played the role of master. This would be compatible with the current BT specifications. It is envisaged that an enhanced version of the SIM Access Profile would be used to transfer data between mobile and laptop over BT. A Role Control Unit on the laptop would then have to enforce that, for every BT connection over which the enhanced SIM Access Profile is run, the laptop is the BT master. The BT security group should comment on the practicability of this solution. If practical, it would constitute an easy fix of the problem.

A second reason why the attack would be no longer possible is that the attacker would have no possibility any more to reset the clock.

Laptop initiates last authentication before encryption starts: the laptop need not be the BT master to do that, but it can only do that for a point-to-point link. Then the laptop could gain assurance of encryption key freshness (cf. section 2). But an attack could still be possible if an attacker, impersonating the mobile, could initiate yet another authentication of the laptop, and re-start encryption, just before replaying a message to the laptop. In this way, the attacker could regain control of the ACO, and, if the attacker was also the Bluetooth master, of the encryption key. In order for the laptop to spot this attack, the Bluetooth link layer on the laptop would have to indicate to the application (e.g. an enhanced SIM access profile) that the mobile had initiated an authentication of the laptop. Would such an indication be available in Bluetooth systems, as currently specified? Alternatively, a logic implemented at link layer could take care that an authentication initiated by the mobile is always followed by an authentication initiated by the laptop. Comments from the Bluetooth security group would be appreciated.

Modify link key: [BT, vol2, part H, section 3.2.7] describes a procedure to modify an existing link key. Both master and slave can initiate a modification of the link key. This modification can be performed without repeating the pairing procedure (unless the link key is a unit key, which is not recommended anyhow). When the link key changes, also any encryption key derived from it will be different, so no replay of encrypted BT sequences is possible. So, the attack would be prevented if the laptop initiated a modification of the link key before a transaction using the enhanced SIM Access Profile is started. Of course, this would introduce a delay in the 3G-WLAN authentication procedure. The BT security group could perhaps assist here in indicating the order of magnitude of the delay caused by a modification of the link key. It would also be useful to know if an interface was available for the application to trigger a modification of the link key.
If the delay in modifying the link key is significant then one solution would be to update the link key immediately after the transaction with the (U)SIM. A disadvantage doing the link key update immediately after the 3G-WLAN authentication procedure is that the attacker could block the link key update that happens immediately after the 3G-WLAN authentication. To prevent this, the laptop would have to remember to update the link keys before the 3G-WLAN authentication in the case that the Bluetooth link was blocked after the previous 3G-WLAN authentication. This

may add some undesirable complexity to the implementation. I should be mentioned that regular link key update prevents also other attacks. In fact, the SIM access profile recommends that, for security reasons, the link key is updated every time the profile is accessed.

Integrity by BT encryption and keyless hash function: assume that the laptop includes a nonce in every request to the mobile over the encrypted BT link, and the mobile applies a keyless hash function (e.g. SHA-1) to the key material and the nonce and sends the key material (e.g. the MK) and the hash back over the encrypted BT link. This would thwart the attack under the assumptions made in [S3-040049]. This solution would have the advantage that there are no requirements on the BT baseband, the solution could be realised entirely as part of the application protocol between the laptop and the mobile (an enhanced SIM access profile). It would also come with a moderate computational and no key management cost.
Weakness: it should be noted that the method is susceptible to known plaintext attacks. Whenever a pair of key material and nonce becomes known to the attacker the attacker can decrypt future nonces sent by the laptop and compute the correct hash. Then the attack in [S3-040049] can be performed as described there. However note also that, for the attacker to recover the plaintext key material and the nonce, he would have to break the BT cipher, or compromise the laptop. But if the key material and nonce on the laptop was compromised then perhaps also the BT link key could be compromised.

Integrity by message authentication code: integrity with replay protection could be specified as part of an enhanced SIM access profile using message authentication codes and sequence numbers or nonces for replay protection. The most difficult part here is expected to be the key management for the MAC. The keys could be either derived from the BT link layer keys (would an interface be available to use these keys?), or new keys would be used. But then a new key establishment procedure for these MAC keys would be needed, similar to the BT pairing procedure.

Conclusion

This contribution contains remarks on the scope of the attack in [S3-040049] and discusses a number of possible countermeasures. No recommendation for one of these countermeasures is made here, as it is felt that more study and input from the BT security group is needed. Which countermeasure should be preferred depends, among other criteria, on BT performance and implementation issues, and on the threat model (compromise of laptop).

References

[BT] Bluetooth specification version 1.2, www.bluetooth.org.

[S3-040049] Orange, “A man-in-the-middle attack using Bluetooth in a WLAN interworking environment”, contribution to SA3#32.