3GPP TSG SA WG3 Security — S3#33 **S3-040270**
10 - 14 May 2004
Beijing, China

Source: **BT Group**
Contact: **Colin Blanchard colin.blanchard@bt.com**
Title: **Bluetooth Security Overview for TS 33.234 WLAN interworking security specification (Annex A4)**

Document for: **Discussion and decision**
Agenda Item: **6.10**

1. Introduction

The current version of Wireless Local Area Network (WLAN) interworking security 3GPP TS 33.234 V6.0.0 (2004-03) contains an empty annex A4 that was intended to give an overview of Bluetooth security and configuration considerations when used in the following context:

1. As an alterative access technology to 802.11 interworking with 3GPP networks in the same way as HIPERLAN/2 Security architecture is described in Annex A2 of TS33.234

2. As a technology to implement the WLAN-UE Functional Split as described in section 4.2.4 of TS33.234.

This contribution provides the background material that will be used to create a CR to add appropriate text in Annex A4 of TS33.234.

2. Suggested text

A 4.1 Introduction & Background

The Bluetooth technology provides peer-to-peer communications over short distances. In order to provide usage protection and information confidentiality, the system has to provide security measures both at the application layer and the link layer. This means that in each Bluetooth unit, the authentication and encryption routines are implemented in the same way. The following provides an informational guide on how these security measures are implemented.

A 4.2 Security Modes and Levels

Bluetooth enabled devices can operate in one of three different security modes as per the Bluetooth specifications:

- **Security Mode 1** - This is the most insecure security mode in which the Bluetooth device does not initiate any security procedure. It is in a 'discovery' mode, allowing other Bluetooth devices to initiate connections with it when in range.
- **Security Mode 2** - This mode enforces security after establishment of the link between the devices at the Logical Link Control and Adaptation Protocol (L2CAP) level. This mode allows the setting up of flexible security policies involving application layer controls running in parallel with the lower protocols.
- **Security Mode 3** - This mode enforces security controls such as authentication and encryption at the Baseband level itself, before the connection is set up. The security manager usually enforces this via the Link Management Protocol (LMP).

Bluetooth allows security levels to be defined for both devices and services:

For **devices** there are two possible security levels. A remote device could either be a:

- **Trusted device** - Such a device would have access to all services for which the trust relationship has been set.

- **Untrusted device** - Such a device would have restricted access to services. Typically such devices would not share a permanent relationship with the other device.

For **services**, three levels of security have been defined.

- **Service Level 1** - services that require authorisation and authentication. Automatic access is only granted to trusted devices. Other devices need a manual authorisation.
- **Service Level 2** - services that require authentication only. Authorisation is not necessary.
- **Service Level 3** - services open to all devices; authentication is not required, no access approval required before service access is granted.

Note: The Bluetooth Architecture allows for defining security policies that can set trust relationships in such a way that even trusted devices can only get access to specific services and not to others.

A 4.3 Access Control

Fundamentally, the core Bluetooth protocols can be used to implement the following security controls to restrict access to services:

- Access to Services would need Authorisation (Authorisation always includes authentication). Only trusted devices would get automatic access.
- Access to Services would need only authentication. i.e. the remote device would need to get authenticated before being able to connect to the application.
- Access to Services would need encryption. The link between the two devices must be encrypted before the application can be accessed.

Bluetooth core protocols can only authenticate devices and not users. This is not to say that user based access control is not possible. The Bluetooth Security Architecture (through the Security Manager) allows applications to enforce their own security policies. The link layer, at which Bluetooth specific security controls operate, is transparent to the security controls imposed by the application layers. Thus it is possible to enforce user-based authentication and fine grained access control within the Bluetooth Security Framework.

A 4.4 Bluetooth Keys

Bluetooth security relies on symmetric keys for authentication and encryption. The keys involved include:

- Bluetooth Device Address – a 48 bit address, unique to each Bluetooth device (BD_ADDR)
- Random number – 128 bit random number (may be pseudo-random), changes frequently (RAND)
- Initialisation Key (INIT)
- Unit Key (UNIT)
- Link Key (LINK)
- Encryption Key (ENC)
- Authentication Key (AUTH)

A 4.5 Processes for setting up keys

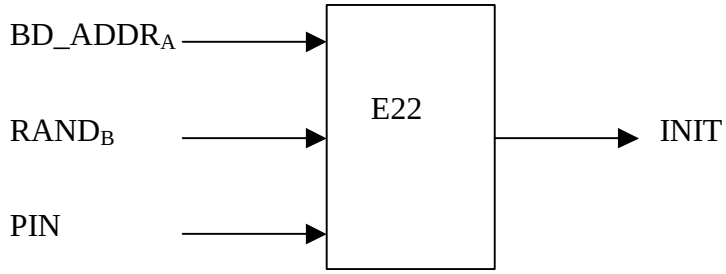
Further information on the protocols is described in Ref [36] with the full details available from Ref [41].

A 4.5.1 Initialisation Key Establishment

This protocol is used to exchange a temporary initialisation key, which is used to encrypt information during the generation of the encryption key.
For devices A and B:

1. A PIN is manually entered to each device.
2. Device A, having detected device B (and sees B's Bluetooth device address) sends a random number to device B.
3. Both Bluetooth devices calculate an initialisation key, based on the random number sent by A, the Bluetooth device address of B and the shared PIN (uses algorithm E22).

4. Verification: A chooses a new random number and calculates a number based on the initialisation key, the new random number and B's Bluetooth device address. This is sent to B.
5. B reverses the process using its Bluetooth device address, the initialisation key and the number sent and returns this.
6. A can now confirm the keys were shared successfully.
7. Repeat the last 3 steps with roles reversed, so B can confirm the same



Link key generation – Option 1 (Unit Key)

This is to share a link key, having established an initialisation key as above. In this case, one device is limited in memory (device A), so a 'short cut' is employed:

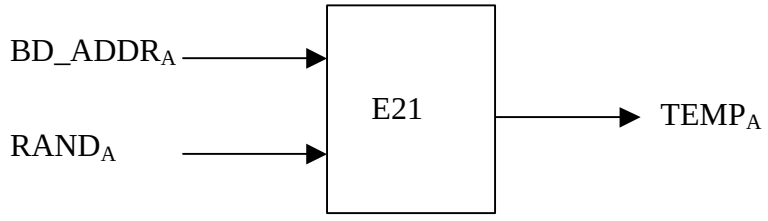
1. A encrypts its unit key with the initialisation key and sends this to B.
2. B decrypts the message with the initialisation key.
3. Both devices now have A's unit key, and they use this as the link key. The initialisation key is now discarded.

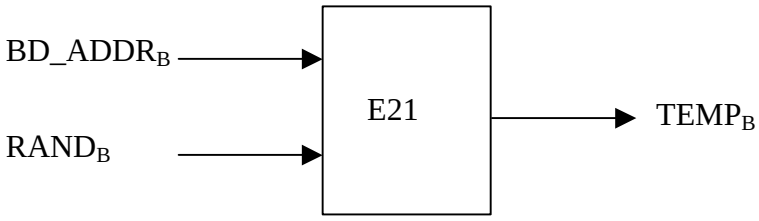
The problem with this is that if A now communicates with another device, say C, then this pair will use the same encryption key and B can read all their communications and impersonate A.

Link key generation – Option 2 (Combination Key)

This is an alternative to Option 1, and is recommended, assuming both devices are sufficiently capable. The result is a combination key.

1. Both devices generate a random number.
2. Device A computes a number based on its random number and Bluetooth device address, using algorithm E21.
3. Device B does the same with its own keys.
4. Both units encrypt their calculated numbers with their shared initialisation key and send them to each other.
5. Both devices now have both calculated numbers and combine them to create the link key – in this case, a combination key.
6. The link key is mutually verified. The initialisation key is no longer needed.



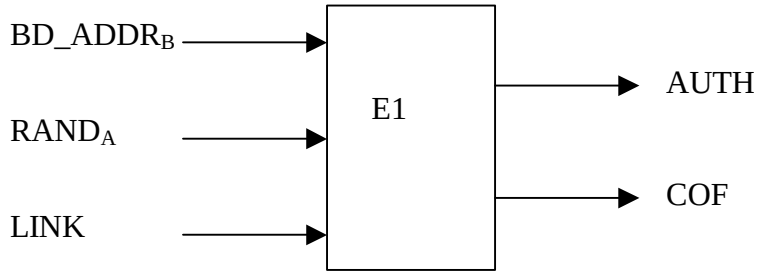


$$Temp_A \oplus Temp_B \rightarrow LINK$$

A 4.6 Authentication

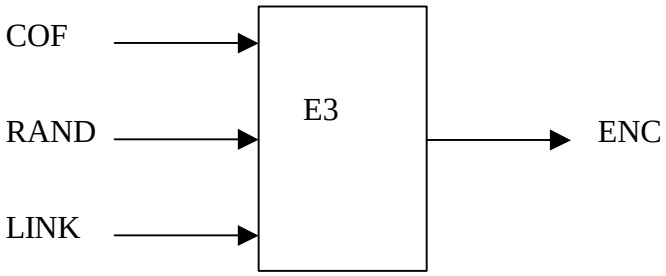
Once the link key has been set up, authentication can start. Here, device A is authenticating device B.

1. A sends a random 128 bit challenge to B.
2. B calculates a number using the challenge, its Bluetooth device address and the link key, under algorithm E1.
3. B returns just the 32 most significant bits to A.
4. A can now check these bits to authenticate B.
5. The remaining 96 bits are the Ciphering Offset Number (COF), used in encryption.
6. The roles of A and B can now be reversed.



A 4.7 Encryption (Confidentiality)

Every time this pair of Bluetooth devices starts an encrypted session, they calculate an encryption key. They use a random number, the link key and the Ciphering Offset Number (generated during authentication).



All data is encrypted, using algorithm E0 and the encryption key to encrypt the packets sent between devices providing confidentiality between the communicating devices.

A 4.8 Configuration Considerations

Ref	Consideration	Recommendation
1	Any key in Bluetooth depends either directly on its generation or for protective reasons on the Initialisation Key, which is built from a secret PIN. So if an attacker is able to capture the communications from the initialisation sequence onwards the attacker only has to find the right PIN to break the security of all keys, including the link encryption keys. A link key is used temporarily during initialization, known as the initialization key. This key is derived from the BD_ADDR, a PIN code, the length of the PIN (in octets), and a random number IN_RAND which a transmitted in clear over the air. This derived key becomes the CURRENT LINK KEY The encryption engines in both devices must then be synchronized An LMP_in_rand message is sent carrying the random number; both sides then use that to initialise their encryption engines Next the verifier sends and LMP_au_rand message containing the random number to be authenticated by the claimant. The claimant encrypts this number using its CURRENT LINK KEY and then returns the encrypted number in a secure response message LMP_sres. The verifier encrypts the random number from LMP-au_rand with its CURRENT LINK KEY and compares it with the encrypted version in LMP_sres. Thus the verifier can decide whether both sides share the same link key without the link key ever being transmitted on air. Once Master and Slave know that they share a secret key, they could use that key for encrypting traffic. But if data with a pattern is sent then it is possible to eventually crack the link key. Hence the use of dynamic derived keys either unit and combination keys. The combination key is the combination of two numbers generated in device A and B, respectively. Each device generates a random number which are protected during the on air exchange by XORing with the CURRENT LINK KEY The same procedure is invoked regularly during normal operation to refresh the link keys and prior to encryption start to modify the encryption keys to address the key stream repeat issue. Hence other than the PIN, all other information that contributes to the authentication /ciphering is publicly known or protected with a strength equal to that of the PIN	The full 16 octet PIN shall be used which shall be unique to each device. Out of band secure distribution methods shall be considered. Ref: [33] [34] [35] [36] [37]

Ref	Consideration	Recommendation
2	Unit keys are static and only changed when the Bluetooth device is reset. If an attacker is able to authenticate, or at least perform the first 3 steps of the initialisation procedure, he is able to learn the Unit Key. As this is the Link Key that the attacked device also uses for all other connections the attacker can masquerade as the attacked device, or eavesdrop later encrypted transmissions.	Combination keys must be used. Ref [35] [37]
3	Key stream reuse The clock value is also used to calculate a new seed, and therefore a new key stream, for each packet. A key stream reuse will occur after approximately one day. The clock value is a 28-bit counter that is incremented every 312.5 s, so 228 * 312.5 s = 23.30 h. The key stream also depends on a random value, which is exchanged when encryption is enabled. So to prevent encryption under the same key stream more than once, Bluetooth devices do not need to generate a new encryption key, it would be sufficient if they would restart the encryption once a day, to use a new random number. The Bluetooth 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. Bluetooth 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.	The connection shall be terminated and restarted at least once a day to force the use of a new random number from a command from the network The encryption key generation could be changed so as to give assurance of encryption key freshness also to the slave. Ref: [37] [38]

Ref.	Consideration	Recommendation
4	Replay of old messages due to Lack of Integrity protection in the Bluetooth security design. Just taking over an authenticated connection will not be so easy if the connection is encrypted, as the encryption key is based on the link key. Therefore a Bluetooth device knows that valid encrypted packets can only be generated by a device in possession of the valid link key (either itself or the authenticated device). If different link keys are established for each combination of two Bluetooth devices this means the attacker cannot generate new messages. But as the integrity of packets is not protected an attacker might replay old messages. Bluetooth Clock: the Bluetooth clock value is input to the encryption algorithm, so the attacker needs to reset the Bluetooth clock before replaying a message to the target. The Bluetooth master controls the Bluetooth clock and can reset it.	Ensure that encryption is applied and managed according to recommendations outlined in this document. Support enhancement of the Bluetooth security specification with Integrity by message authentication code. Ref: [37] [38]
5	Loss of location privacy in discoverable mode The Bluetooth device's unique base address is freely broadcasted for example during the inquiry procedure. As this is a permanent unique identifier of a personal device, tracking is easy if the device is in discoverable mode. By observing the time, rate, length, maybe even source or destination of messages an attacker can deduce confidential information. Privacy issues arise if the attacker can observe a fixed source identifier, which could be traced and associated with a user. An attacker sends messages to the wireless network or actively initiates communication sessions. Then by observing the time, rate, length, sources or destinations of messages on the wireless transmission medium an attacker can deduce confidential information. An attacker does not require reading the actual data, but for some users the sheer information that they are communicating is considered to be confidential.	A warning should be implemented to inform users about vulnerabilities that are inherent with Bluetooth devices in discoverable mode. c.f. Bluesnarfing and Bluejacking Separate Bluetooth interface/software stack that cannot be placed in discoverable mode by the user once the pairing process is complete. What the end user does with the other interface is then up to the end user. Ref: [34] However, non discoverable mode can also be attacked see concern 6 below.
6	Finding non-discoverable Bluetooth devices by brute forcing the last six bytes of the devices Bluetooth address and sending a read_remote_name (Redfang Tool)	Implement a warning to users about vulnerabilities that are inherent with Bluetooth devices in non discoverable mode Review 3GPP requirement for Anonymity Mode Ref: [39] [40]

Ref.	Consideration	Recommendation
7	Use of Narrow band Jammer to force Bluetooth V1.2 devices to “sterilise” all channels on the assumption that they need to be avoided due to interference from 802.11 I devices	Need to ensure that that all frequencies are not used up.
8	Bluetooth V1.1 has a problem with the Inquiry protocol in that there was a 1 in 10 chance that the devices would not connect.	In the context of 3GPP WLAN Interworking only Bluetooth Version 1.2 shall be used.

3 References (to be added to TS33.234 section 2)

[33] Bluetooth™ Security White Paper Bluetooth SIG Security Expert Group
http://grouper.ieee.org/groups/1451/5/Comparison%20of%20PHY/Bluetooth_24Security_Paper.pdf

[34] Security Weaknesses in Bluetooth Markus Jakobsson and Susanne Wetzel
http://www.rsasecurity.com/rsalabs/staff/bios/mjakobsson/bluetooth/bluetooth.pdf

[35] Security Comparison: BluetoothTM Communications vs. 802.11 Thomas G. Xydis Ph.D. Simon Blake-Wilson
http://www.ccss.isi.edu/papers/xydis_bluetooth.pdf

[36] Bluetooth Security Juha T. Vainio Department of Computer Science and Engineering
Helsinki University of Technology
http://www.niksula.cs.hut.fi/~jiitv/bluesec.html

[37] Security Requirements for Wireless Networks and their Satisfaction in IEEE 802.11b and Bluetooth
Henrich C. Poehls
http://www.2000grad.de/impressum/Security_Requirements_for_Wireless_Networks_and_their_Satisfaction_in_IEEE_802_11b_and_Bluetooth.pdf

[38] LS on “Attack and countermeasures in a User Equipment functionality split scenario using Bluetooth”
http://www.3gpp.org/ftp/tsg_sa/WG3_Security/TSGS3_32_Edinburgh/Docs/ZIP/S3-040164.zip

[39] Red fang the Bluetooth hunter
http://www.atstake.com/research/tools/info_gathering/

[40] News - Red Fang "Bluetooth hack" not much use" - TDK
http://www.newswireless.net/articles/0300910-bluestake.html

[41] “Specification of the Bluetooth System”, Bluetooth, http://www.bluetooth.com/