Bluetooth

Authentication - Authorisation - Encryption

General

Bluetooth is a short-range radio link intended to replace the cable(s) connecting portable and/or fixed electronic devices. Key features are robustness, low complexity, low power, and low cost.

The Bluetooth system is operating in the 2.4 GHz ISM band. In a vast majority of countries around the world the range of this frequency band is 2400 - 2483.5 MHz. Some countries have however national limitations in the frequency range. In order to comply with these national limitations, special frequency hopping algorithms have been specified for these countries. It should be noted that products implementing the reduced frequency band will not work with products implementing the full band. The products implementing the reduced frequency band must therefore be considered as local versions for a single market. The Bluetooth SIG has launched a campaign to overcome these difficulties and reach total harmonization of the frequency band.

Geography	Regulatory Range	RF Channels
USA, Europe and most other countries[1)]	2.400-2.4835 GHz	f=2402+k MHz, k=0,…,78

Table1: Operating frequency bands

Channel spacing is 1 MHz. In order to comply with out-of-band regulations in each country, a guard band is used at the lower and upper band edge.

Geography	Lower Guard Band	Upper Guard Band
USA, Europe and most other countries	2 MHz	3.5 MHz

Table2: Guard Bands

The Bluetooth system provides a point-to-point connection (only two Bluetooth units involved), or a point-to-multipoint connection, see Figure1. In the point-to-multipoint connection, the channel is shared among several Bluetooth units. Two or more units sharing the same channel form a *piconet*. One Bluetooth unit acts as the master of the piconet, whereas the other unit(s) acts as slave(s). Up to seven slaves can be active in the piconet. In addition, many more slaves can remain locked to the master in a so-called parked state. These parked slaves cannot be active on the channel, but remain synchronized to the master. Both for active and parked slaves, the channel access is controlled by the master. Multiple piconets with overlapping coverage areas form a *scatternet*. Each piconet can only have a single master. However, slaves can participate in different piconets on a time-division multiplex basis. In addition, a master in one piconet can be a slave in another piconet. The piconets shall not be frequencysynchronized. Each piconet has its own hopping channel.

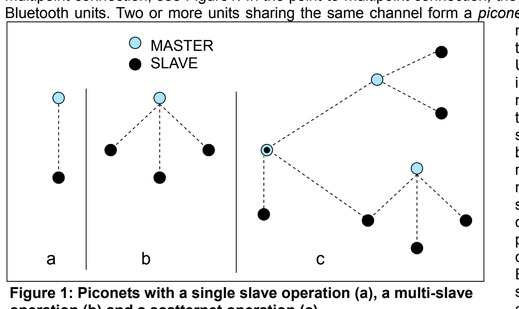

Figure 1: Piconets with a single slave operation (a), a multi-slave operation (b) and a scatternet operation (c)

The channel is represented by a pseudo-random hopping sequence hopping through the 79 or 23 RF channels. The hopping sequence is unique for the piconet and is determined by the Bluetooth device address of the master; the phase in the hopping sequence is determined by the Bluetooth clock of the master. The channel is divided into time slots where each slot corresponds to an RF hop frequency. Consecutive hops correspond to different RF hop frequencies. The nominal hop rate is 1600 hops/s. All Bluetooth units participating in the piconet are time- and hop-synchronized to the channel.

There are 13 "profiles" described in version 1.1 of the Bluetooth specification. These profiles are general behaviors through which Bluetooth units communicate with other units. The 13 profiles describe the basis for the user models and their profiles. The profiles also provide the foundation for future user models and profiles.

The Generic Access Profile defines how two Bluetooth units discover and establish a connection with each other. GAP handles discovery and establishment between units that are unconnected. The profile defines operations that are generic and can be used by profiles referring to GAP and by devices implementing multiple profiles. GAP ensures that any two Bluetooth units, regardless of manufacturer and application, can exchange information via Bluetooth in order to discover what type of applications the units support. Bluetooth units not conforming to any other Bluetooth profile must conform to GAP to ensure basic interoperability and co-existence. It also defines procedures related to use of different security levels.

Profile
Generic Access Profile
Service Discovery Application Profile
Cordless Telephony Profile
Intercom Profile
Serial Port Profile
Headset Profile
Dial-up Networking Profile
Fax Profile
LAN Access Profile
Generic Object Exchange Profile
Object Push Profile
File Transfer Profile
Synchronisation Profile

Table 3: Profiles

Bluetooth Security

In every Bluetooth device, there are four entities used for maintaining the security at the link level. The Bluetooth device address (BD_ADDR), which is a 48-bit address that is unique for each Bluetooth device and defined by the Institute of Electrical and Electronics Engineers (IEEE). Private authentication key, which is a 128-bit random number used for authentication purposes. Private encryption key, 8-128 bits in length that is used for encryption. And a random number (RAND), which is a frequently changing 128-bit random or pseudo-random number that is made by the Bluetooth device itself.

Entity	Size
BD_ADDR	48 bits
Private user key, authentication	128 bits
Private user key, encryption configurable length (byte-wise)	8 – 128 bits
RAND	128 bits

Table 4: Entities used in authentication and encryption procedures

Security modes

In Bluetooth Generic Access Profile, the Bluetooth security is divided into three modes:

Security mode 1 (non-secure)

When a Bluetooth device is in security mode 1 it shall never initiate any security procedure

Security mode 2 (service level enforced security)

When a Bluetooth device is in security mode 2 it shall not initiate any security procedure before a channel establishment request has been received or a channel establishment procedure has been initiated by itself. Whether a security procedure is initiated or not depends on the security requirements of the requested channel or service. A Bluetooth device in security mode 2 should classify the security requirements of its services using at least the following attributes:
This information is stored in the service database of the security manager. If no registration has taken place, a

Authorisation Required	Access is only granted automatically to trusted devices (i.e., devices marked as such in the device database) or untrusted devices after an authorisation procedure. Authorisation always requires authentication to verify that the remote device is the right one.
Authentication Required	Before connecting to the application, the remote device must be authenticated
Encryption Required	The link must be changed to encrypted mode, before access to the service is possible

Table 5: Security Level of Services

default security level is used. This default is:
Incoming Connection: Authorisation and Authentication required
Outgoing Connection: Authentication required

Note: Security mode 1 can be considered (at least from a remote device point of view) as a special case of security mode 2 where no service has registered any security requirements.

Security mode 3 (link level enforced security)

When a Bluetooth device is in security mode 3 it shall initiate security procedures before the channel is established.

Authorisation and Authentication

We distinguish between authentication and authorisation. The terms are defined as follows:

Authentication

Authentication is the process of verifying 'who' is at the other end of the link. Authentication is performed for devices (BD_ADDR). In Bluetooth this is achieved by the authentication procedure based on the stored link key or by pairing (entering a PIN).

The entity authentication used in Bluetooth uses a challenge-response scheme in which a claimant's knowledge of a secret key is checked through a 2-move protocol using symmetric secret keys. The latter implies that a correct claimant/ verifier pair share the same secret key, for example K. In the challenge-response scheme the verifier challenges the claimant to authenticate a random input (the challenge), denoted by AU_RAND_A, with an authentication code, denoted by , and return the result SRES to the verifier, see Figure 2. This figure shows also that in Bluetooth the input to E_1 consists of the tuple AU_RAND_A and the Bluetooth device address (BD_ADDR) of the claimant. The use of this address prevents a simple reflection attack[2)]. The secret K shared by units A and B is the current link key.

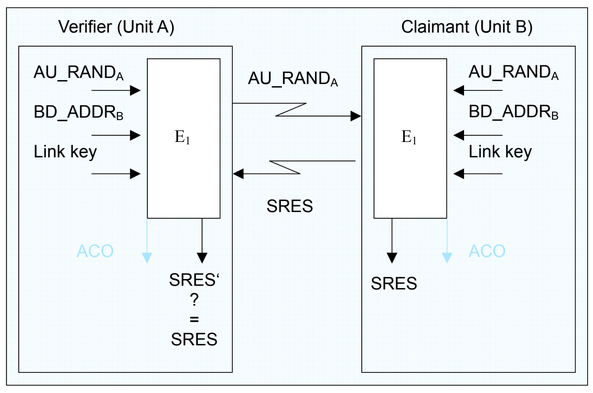

Figure 2: Challenge-response for the Bluetooth

In the Bluetooth, the verifier is not necessarily the master. The application indicates who has to be authenticated by whom. Certain applications only require a one-way authentication. However, in some peer-to-peer communications, one might prefer a mutual authentication in which each unit is subsequently the challenger (verifier) in two authentication procedures. The LM coordinates the indicated authentication preferences by the application to determine in which direction(s) the authentication(s) has to take place. For mutual authentication with the units of Figure 2, after unit A has successfully authenticated unit B, unit B could authenticate unit A by sending a AU_RAND_B (different from the AU_RAND_A that unit A issued) to unit A, and deriving the SRES and SRES' from the new AU_RAND_B, the address of unit A, and the link key K_{AB}. If an authentication is successful the value of ACO as produced by E_1 should be retained.

When the authentication attempt fails, a certain waiting interval must pass before the verifier will initiate a new authentication attempt to the same claimant, or before it will respond to an authentication attempt initiated by a unit claiming the same identity as the suspicious unit. For each subsequent authentication failure with the same Bluetooth address, the waiting interval shall be increased exponentially. That is, after each failure, the waiting interval before a new attempt can be made, for example, twice as long as the waiting interval prior to the previous attempt[3)]. The waiting interval shall be limited to a maximum. The maximum waiting interval depends on the implementation. The waiting time shall exponentially decrease to a minimum when no new failed attempts are being made during a certain time period. This procedure prevents an intruder to repeat the authentication procedure with a large number of different keys.

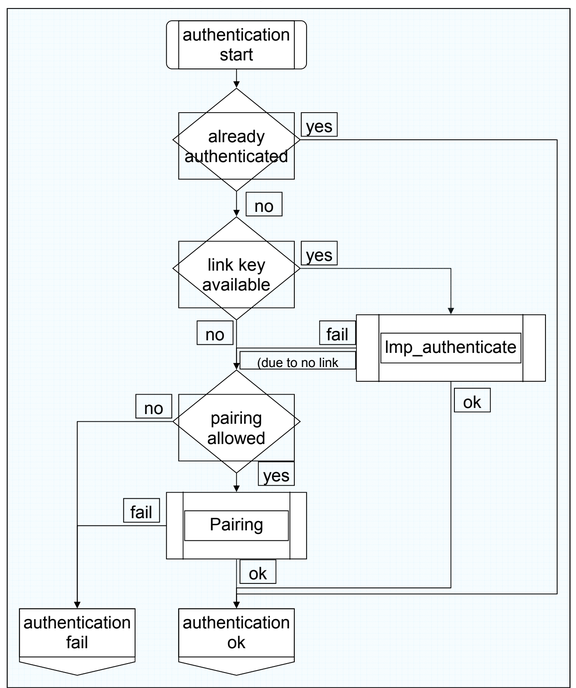

Figure 3: Example Flow Chart for Authentication Procedure

To make the system somewhat less vulnerable to denial-of-service attacks, the Bluetooth units should keep a list of individual waiting intervals for each unit it has established contact with. Clearly, the size of this list must be restricted only to contain the N units with which the most recent contact has been made. The number N can vary for different units depending on available memory size and user environment.

The authentication function E_1 proposed for the Bluetooth is a computationally secure authentication code, or often called a MAC. uses the encryption function called SAFER+. The algorithm is an enhanced version[4)] of an existing 64-bit block cipher SAFER-SK128, and it is freely available.

Authorisation

Authorisation is the process of deciding if device X is allowed to have access to service Y. This is where the concept of 'trusted' exists. Trusted devices (authenticated and indicated as "trusted"), are allowed access to services. Untrusted or unknown devices may require authorisation based on user interaction before access to services is granted. This does not principally exclude that the authorisation might be given by an application automatically. Authorisation always includes authentication.

Device Trust Level

We distinguish between two different device trust levels:

Trusted Device	The device has been previously authenticated, a link key is stored and the device is marked as "trusted" in the Device Database.
Untrusted Device	The device has been previously authenticated, a link key is stored but the device is not marked as "trusted" in the Device Database
Unknown Device	No security information is available for this device. This is also an untrusted device.

Table 6: Device turst levels

There will be a database table maintained in the security manager. This database might be maintained for all services together or separately for each service or group of services.

Authenticate trusted device

The verification is done using the authentication procedure, defined in the LMP and Baseband specifications. A device is verified as trusted, if a positive authentication response is given and the trusted flag is set.

Set-up of the trusted relationship

A trusted relationship is established during the pairing procedure. This is usually performed during the bonding procedure but could be performed at connection set-up. When an untrusted device is authorised to use a service, it is also possible to add it to the list of trusted devices during the same procedure. This of course requires an active selection by the user.

Authenticate untrusted device

Authentication of untrusted devices is done similarly as for trusted devices with the exception that the device is not marked as trusted in the internal database.

Encryption

User information can be protected by encryption of the packet payload; the access code and the packet header are never encrypted. The encryption of the payloads is carried out with a stream cipher called E_0 that is re-synchronized for every payload. The overall principle is shown in Figure 5. The stream cipher system E_0 consists of three parts. One part performs the initialization (generation of the payload key), the second part generates the key stream bits, and the third part performs the encryption and decryption. The payload key generator is very simple - it merely combines the input bits in an appropriate order and shift them into the four LFSRs used in the key stream generator. The main part of the cipher system is the second, as it also will be used for the initialization. The key stream bits are generated by a method derived from the summation stream cipher generator attributable to Massey and Rueppel. The method has been thoroughly investigated, and there exist good estimates of its strength with respect to presently known methods for cryptanalysis. Although the summation generator has weaknesses that can be used in so-called correlation attacks, the high re-synchronization frequency will disrupt such attacks.

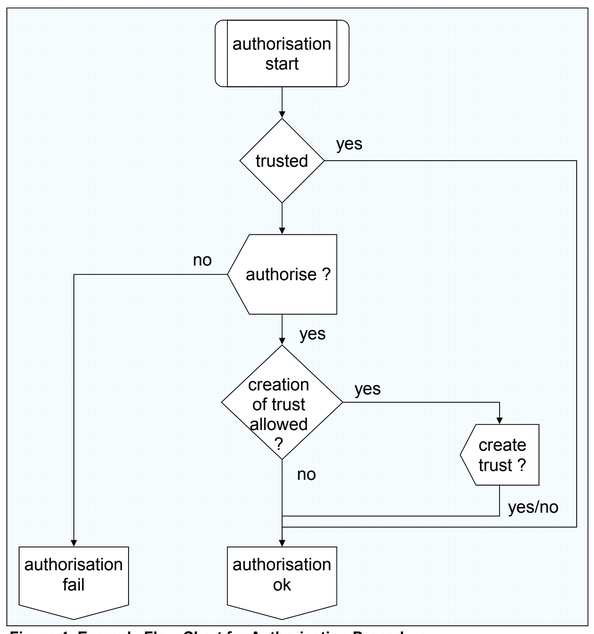

Figure 4: Example Flow Chart for Authorisation Procedure

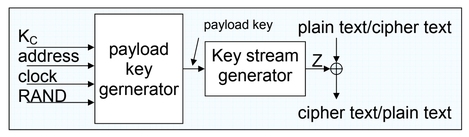

Figure 5: Stream ciphering for Bluetooth with E_0

Encryption key size negotiation

Each Bluetooth device implementing the baseband specification needs a parameter defining the maximal allowed key length, $L_{max}, 1 \le L_{max} \le 16$ (number of octets in the key). For each application, a number L_{min} is defined indicating the smallest acceptable key size for that particular application. Before generating the encryption key, the involved units must negotiate to decide what key size to actually use.

The master sends a suggested value, $L^{(M)}_{sug}$ to the slave. Initially, the suggested value is set to $L^{(M)}_{max}$. If $L^{(S)}_{min} \le L^{(M)}_{sug}$, and, the slave supports the suggested length, the slave acknowledges and this value will be the length of the encryption key for this link. However, if both conditions are not fulfilled, the slave sends a new proposal, $L^{(S)}_{sug} < L^{(M)}_{sug}$, to the master. This value should be the largest among all supported lengths less than the previous master suggestion. Then, the master performs the corresponding test on the slave suggestion. This procedure is repeated until a key length agreement is reached, or, one unit aborts the negotiation. An abortion may be caused by lack of support for L_{sug} and all smaller key lengths, or if $L_{sug} < L_{min}$ in one of the units. In case of abortion Bluetooth link encryption can not be employed.
The possibility of a failure in setting up a secure link is an unavoidable consequence of letting the application decide whether to accept or reject a suggested key size. However, this is a necessary precaution. Otherwise a fraudulent unit could enforce a weak protection on a link by claiming a small maximum key size.

Encryption modes

If a slave has a semi-permanent link key (i.e. a combination key or a unit key), it can only accept encryption on slots individually addressed to itself (and, of course, in the reverse direction to the master). In particular, it will assume that broadcast messages are not encrypted. The possible traffic modes are described in Table 7. When an entry in the table refers to a link key, it means that the encryption/decryption engine uses the encryption key derived from that link key.

Mode	Broadcast traffic	Individually addressed traffic
1	No encryption	No encryption
2	No encryption	Encryption, Semi-permanent link key

Table 7: Possible traffic modes for a slave using a semi-permanent link key

If the slave has received a master key, there are three possible combinations as defined in Table 8. In this case, all units in the piconet uses a common link key, K_{master} . Since the master uses encryption keys derived from this link key for all secure traffic on the piconet, it is possible to avoid ambiguity in the participating slaves on which encryption key to use. Also in this case the default mode is that broadcast messages are not encrypted. A specific LM-command is required to activate encryption - both for broadcast and for individually addressed traffic.

Mode	Broadcast traffic	Individually addressed traffic
1	No encryption	No encryption
2	No encryption	Encryption, K_{master}
3	Encryption, K_{master}	Encryption, K_{master}

Table 8: Possilbe encryption modes for a slave in possession of a master key

The master can issue an LM-command to the slaves telling them to fall back to their previous semi-permanent link key. Then, regardless of the previous mode they were in, they will end up in the first row of Table 7, i.e. no encryption.

Encryption concept

For the encryption routine, a stream cipher algorithm will be used in which ciphering bits are bit-wise modulo-2 added to the data stream to be sent over the air interface. The payload is ciphered after the CRC bits are appended, but, prior to the FEC encoding.

Each packet payload is ciphered separately. The cipher algorithm E_0 uses the master Bluetooth address, 26 bits of the master real-time clock (CLK_{26-1}) and the encryption key K_C as input, see Figure 6 (where it is assumed that unit A is the master).

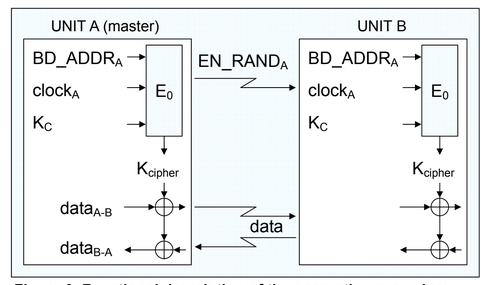

Figure 6: Functional description of the encryption procedure

The encryption key, K_C, is derived by algorithm E_3 from the current link key, a 96-bit Ciphering OFfset number (COF), and a 128-bit random number (see Figure 7). The COF is determined in one of two ways. If the current link key is a master key, then COF is derived from the master BD_ADDR. Otherwise the value of COF is set to the value of ACO as computed during the authentication procedure. The random number is issued by the master before entering encryption mode. Note that EN_RAND_A is publicly known since it is transmitted as plain text over the air.

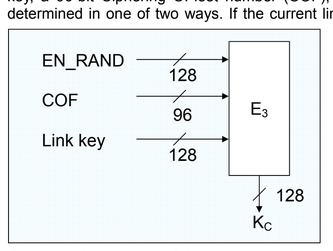

Figure 7: Generation of the encryption key

The real-time clock is incremented for each slot. The E_0 algorithm is re-initialized at the start of each new packet (i.e. for Master-to-Slave as well as for Slave-to-Master transmission). By using CLK_{26-1} at least one bit is changed between two transmissions. Thus, a new keystream is generated after each reinitialization. For packets covering more than a single slot, the Bluetooth clock as found in the first slot is being used for the entire packet.

The encryption algorithm E_0 generates a binary keystream, K_{cipher}, which is modulo-2 added to the data to be encrypted. The cipher is symmetric; decryption is performed in exactly the same way using the same key as used for encryption.

This poster is part of the course Multimedia and Security at the Department of Computer Science (Institute of Technical and Business Information Systems / Research Group Multimedia and Security), Otto-von-Guericke-University of Magdeburg in summer term 2003. Alexander Opel, toengel@toengel.net, http://www.toengel.net

Appendix

Notes

1) The Bluetooth Specification includes a special frequency hopping pattern to provide provisions for compliance with national limitations like in France. The frequency range for France is 2.4465 - 2.4835 GHz and the corresponding RF channels are f = 2454 + k MHz, k = 0,...,22..

2) The reflection attack actually forms no threat in Bluetooth because all service requests are dealt with on a FIFO bases. When preemption is introduced, this attack is potentially dangerous.

3) An other appropriate value larger than 1 may be used.

4) It is presently one of the contenders for the Advanced Encryption Standard (AES) submitted by Cylink, Corp, Sunnyvale, USA

List of Acronyms and Abbreviations

Abbreviation or Acronym	Meaning	Abbreviation or Acronym	Meaning
ACO	Authenticated Ciphering Offset	ISM	Industrial Scientific Medicine
BD_ADDR	Bluetooth Device Address	LFSR	Linear Feedback Shift Register
COF	Ciphering OFfset number	LM	Link Manager
CRC	Cyclic Redundancy Check	LMP	Link Manager Protocol
FEC	Forward Error Correction	LSB	Least Significant Bit
FIFO	First In First Out	MSB	Most Significant Bit
GAP	Generic Access Profile	RF	Radio Frequency
IEEE	Institution of Electrical and Electronics Engineers	SIG	Special Interest Group

Table 9: List of acronyms and abbreviations

Bluetooth Device Address (BD_ADDR)

Each Bluetooth transceiver is allocated a unique 48-bit Bluetooth device address (BD_ADDR). This address is derived from the IEEE802 standard. This 48-bit address is divided into three fields:
- LAP field: lower address part consisting of 24 bits
- UAP field: upper address part consisting of 8 bits
- NAP field: non-significant address part consisting of 16 bits

The LAP and UAP form the significant part of the BD_ADDR. The total address space obtained is 2^{32}.

LSB MSB

company_assigned						company_id					
LAP						UAP		NAP			
0000	0001	0000	0000	0000	0000	0001	0010	0111	1011	0011	0101

Figure 8: Format of BD_ADDR

References

- Specification Volume 1: Specification of the Bluetooth System – Core, Version 1.1, February 22, 2001, [Bluetooth_1_1_vol1.pdf] – http://www.bluetooth.org
- Specification Volume 2: Specification of the Bluetooth System – Profiles, Version 1.1, February 22, 2001, [Bluetooth_11_Profiles_Book.pdf] – http://www.bluetooth.org
- Bluetooth Security Architecture, Version 1.0, July 15, 1999 [Security_Architecture.pdf] - http://www.bluetooth.org
- Juha T. Vainio, Bluetooth Security - http://www.niksula.cs.hut.fi/~jiitv/bluesec.html
- SwedeTrack System, The Bluetooth Profiles - http://www.swedetrack.com/usblue4.htm
- c't magazine 20/2001, 16/2002, 7/2003, 11/2003, 12/2003