Secure Simple Pairing Explained

Introduction

The first version of *Bluetooth* pairing, based on a PIN-code, didn't provide any real level of security against sniffing. A sniffer such as the Ellisys *Bluetooth* Explorer 400 is capable of automatically and instantly determining the PIN-code and deducing the Link Key, just by passively sniffing the pairing process.

With *Bluetooth* becoming more and more widespread, a secure pairing method became a hard requirement for ensuring the long term success of the technology. Introduced in the *Bluetooth* 2.1 specification, Secure Simple Pairing (SSP) fixes all of the issues of the previous pairing method, and makes pairing *Bluetooth* devices simpler than ever.

Stronger security also means new challenges for *Bluetooth* engineers. Debugging off-the-shelf devices in the field becomes difficult to impossible.

Things are not as bad as they may seem however. This document is aimed at introducing the basics of SSP and clearing up some misconceptions commonly found in the *Bluetooth* community.

Pairing process

The process of pairing devices is aimed at creating a **shared secret** between two *Bluetooth* devices: the **Link Key**. This Link Key is then used to authenticate devices to each other and encrypt exchanged data. The data is actually not directly encrypted with the Link Key; a temporary **Encryption Key** is derived from the Link Key and from random numbers that are exchanged shortly before the start of the encrypted traffic. This Encryption Key is then used for encrypting the data in both directions. It can be changed at any time while the connection is active and will be discarded as soon as the connection is closed, or if the encryption is stopped.

The *Bluetooth* specification defines two standard pairing procedures, LMP-pairing (aka PIN-code based), and SSP. Non-standard pairing methods are also possible, but require both devices to be from the same manufacturer. The result of any pairing method is the same though: creating the shared Link Key.

Once two devices own the same Link Key, this shared secret can be used to re-authenticate both devices with each other at a later time. When reconnecting, devices quickly verify that they both have the same Link Key by exchanging numbers that are derived from it. If Link Keys match it is possible to go on with creating the Session Key. Otherwise, the pairing process (either LMP-pairing or SSP) has to be restarted from the very beginning, resulting in the creation of a brand new Link Key.

LMP pairing (aka PIN-code based)

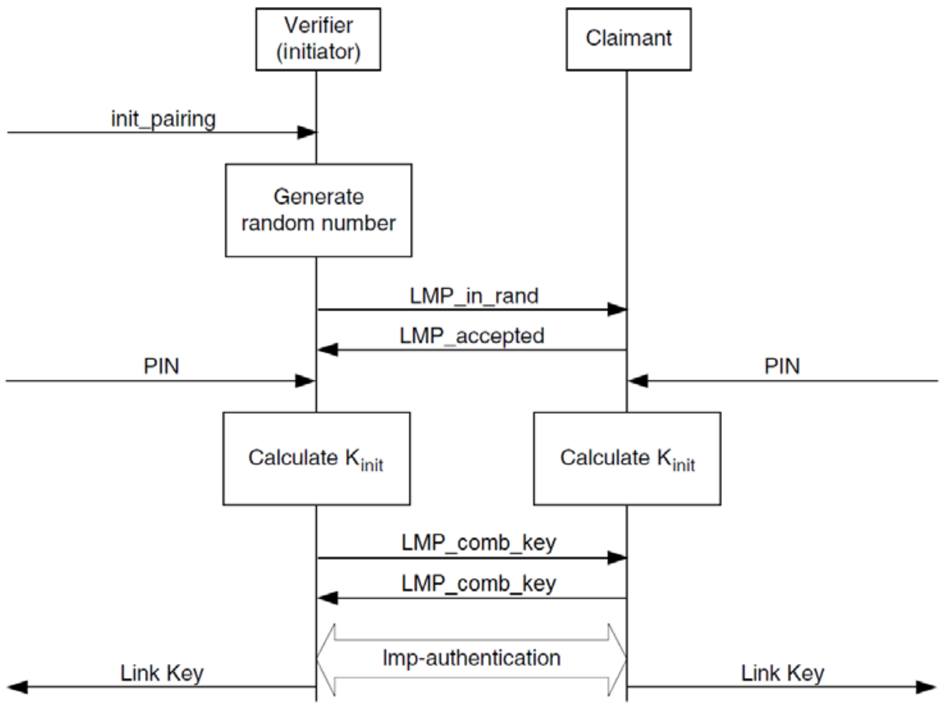
The inputs of the algorithm used to create the Link Key for an LMP pairing are the following:

- The BDADDR of the two devices
- A 16-byte random number created by the initiator
- A PIN code entered by the user on both devices (except for devices with "fixed PIN" where the user can not change the PIN)

These numbers are used to first create a temporary shared initialization key, which is then converted into a Link Key using LMP pairing key generation procedure.

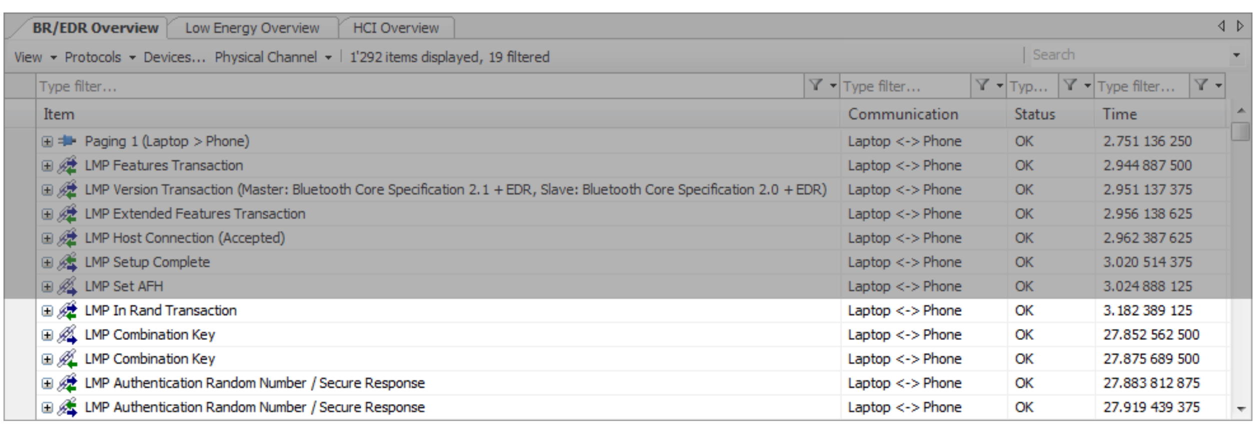
Since the only undisclosed information is the PIN code, the number of possible secret Link Keys is limited by the number of possible PIN codes. If a 4 digit PIN code is used by the devices, an attacker would only have to try 10’000 different Link Keys at maximum before being able to decrypt the traffic. This is where the weakness of the LMP pairing resides.

The LMP pairing process is described by the following chart:



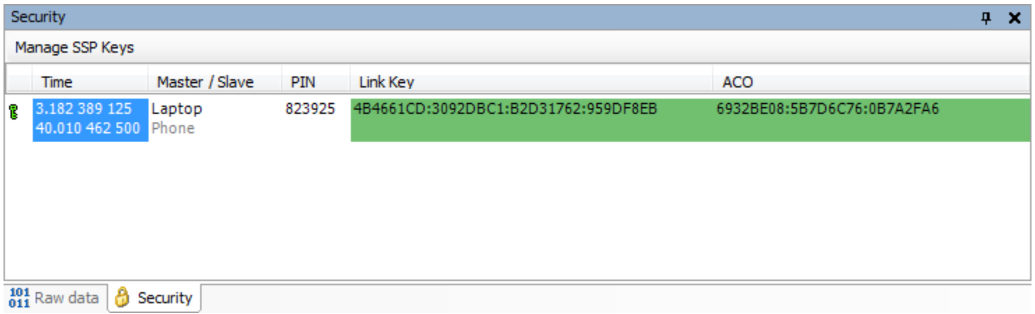
The only information which is not transmitted over-the-air is the PIN code.

Here is the same traffic captured by the Ellisys sniffer:



Item	Communication	Status	Time
Paging 1 (Laptop > Phone)	Laptop <-> Phone	OK	2.751 136 250
LMP Features Transaction	Laptop <-> Phone	OK	2.944 887 500
LMP Version Transaction (Master: Bluetooth Core Specification 2.1 + EDR, Slave: Bluetooth Core Specification 2.0 + EDR)	Laptop <-> Phone	OK	2.951 137 375
LMP Extended Features Transaction	Laptop <-> Phone	OK	2.956 138 625
LMP Host Connection (Accepted)	Laptop <-> Phone	OK	2.962 387 625
LMP Setup Complete	Laptop <-> Phone	OK	3.020 514 375
LMP Set AFH	Laptop <-> Phone	OK	3.024 888 125
LMP In Rand Transaction	Laptop <-> Phone	OK	3.182 389 125
LMP Combination Key	Laptop <-> Phone	OK	27.852 562 500
LMP Combination Key	Laptop <-> Phone	OK	27.875 689 500
LMP Authentication Random Number / Secure Response	Laptop <-> Phone	OK	27.883 812 875
LMP Authentication Random Number / Secure Response	Laptop <-> Phone	OK	27.919 439 375

Based on this captured information, the Ellisys software is capable of automatically determining the PIN code and computing the Link Key, without any user interaction. Here is the result in the Ellisys software:



After this, the Ellisys software will automatically decrypt the data of any subsequent secure connections. This process is described in the Authenticated Connection chapter below.

Secure Simple Pairing

SSP uses a much more elaborate mechanism, known as elliptic curve cryptography, that avoids the use of a PIN code as part of the Link Key calculation process (PIN codes or other user numbers can still be used as part of the authentication process though), but rather use extremely large random number for seeding Link Key calculation. The number of possible Link Keys is thus no longer limited to less than 2^{128} possibilites, which is far beyond any realistic attacker capabilities.

In order to realize this, the SSP process starts by establishing a different kind of shared secret in between the two devices. This shared secret is known as the Diffie-Hellman key (DHKey) and is a 192-bit random number. As a prerequisite, both devices have each a private key and a public key. The public key is transmitted over-the-air and can be known by anyone, but the private key will never be disclosed. We will call these two keys the SSP Public / Private key pair, but these are also known as Diffie-Hellman Public / Private key pair (Diffie and Hellman being the two persons who developed the algorithm).

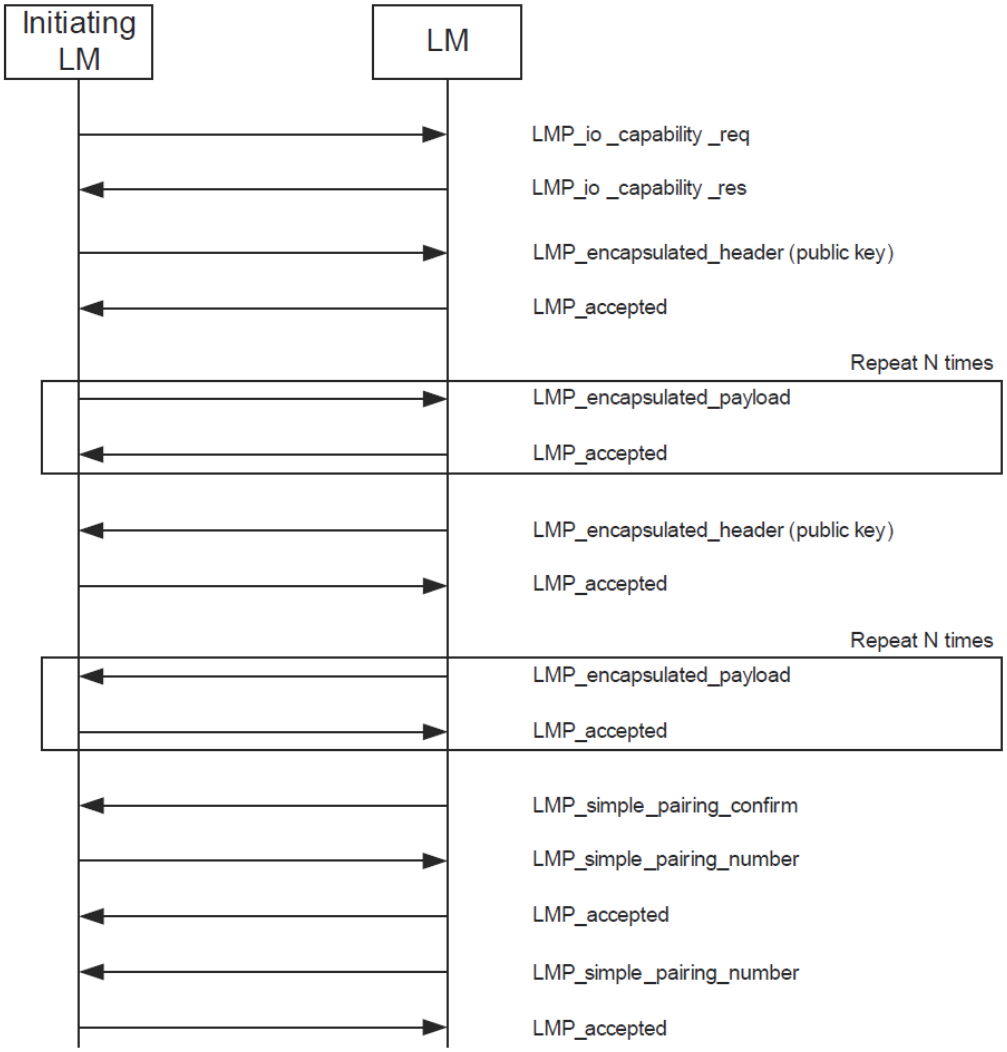
The carefully chosen mathematical space and algorithms used for creating the SSP key pairs are such that:

- It is very hard (i.e. impossible using current state-of-the-art computers) to calculate the Private Key using the Public Key (but it is easy to calculate the Public Key based on the Private one)

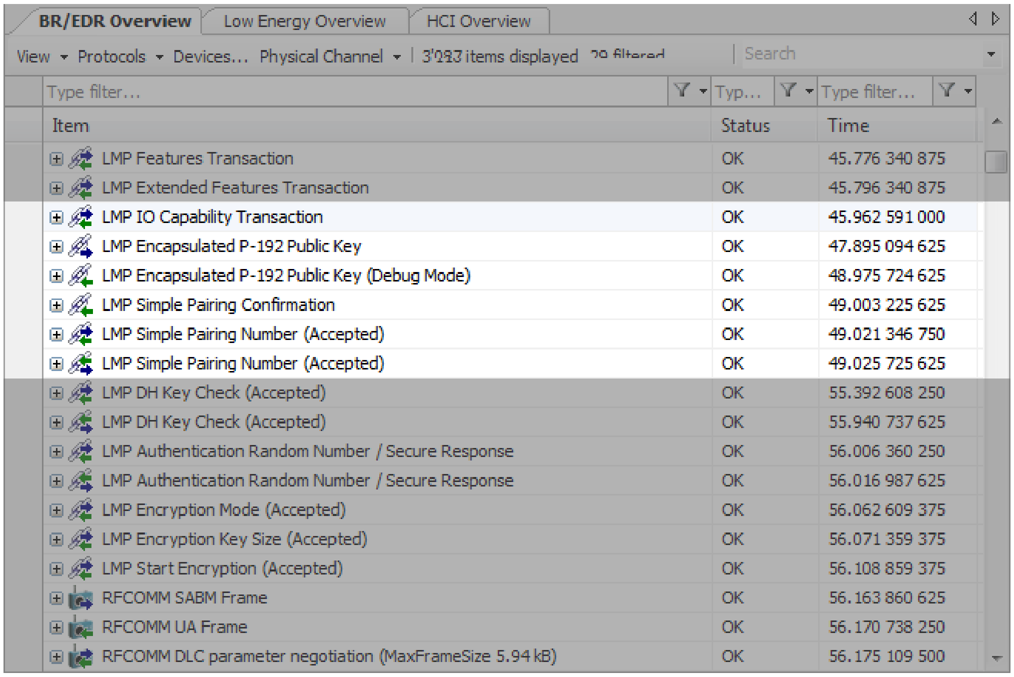
- Given two SSP key pairs A and B, there exists a well-known function F such that F(PublicA, PrivateB) = F(PublicB, PrivateA). The result of this function is the DHKey. Only the very two devices owning A and B are able to calculate the same DHKey.

This is the magic behind SSP: the two devices will be capable of pairing without the need to transmit any critical information over-the-air, and without the need to share this information by an out-of-band mechanism (such as typing it on a keyboard). The DHKey will be used as a seed for calculating the Link Key. The rest of the pairing process is similar to LMP pairing.

The SSP pairing process is described by the following chart:



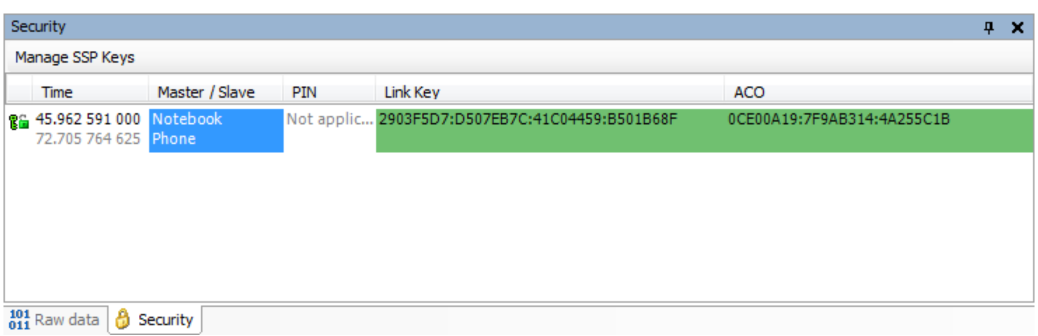
Here is the same traffic captured by the Ellisys sniffer:



The only information the sniffer does not know (in order to compute the Link Key from traffic transmitted over-the-air) is the SSP Private Keys. Actually, only one of the two SSP Private Keys is required to determine the DHKey and hence the Link Key. If the user provides the SSP Private Key of his device to the Ellisys analysis software, then Link Keys from pairings of this device against any other device will be deduced automatically.

Another approach is to use the SSP Debug Mode. As we now understand the basics of SSP, understanding the SSP Debug Mode is easy. A device placed in SSP Debug Mode will not use its usual SSP Private / Public key pair, but will use the SSP Debug Mode Private / Public key pair instead. If either of the two devices is placed in SSP Debug Mode, the Ellisys sniffer will be capable of automatically deducing the Link Key resulting from the pairing, by recognizing the Debug Mode Public key sent over the air and using the corresponding well-known Private key. Using SSP Debug Mode or providing one of the two SSP Private Keys is exactly equivalent.

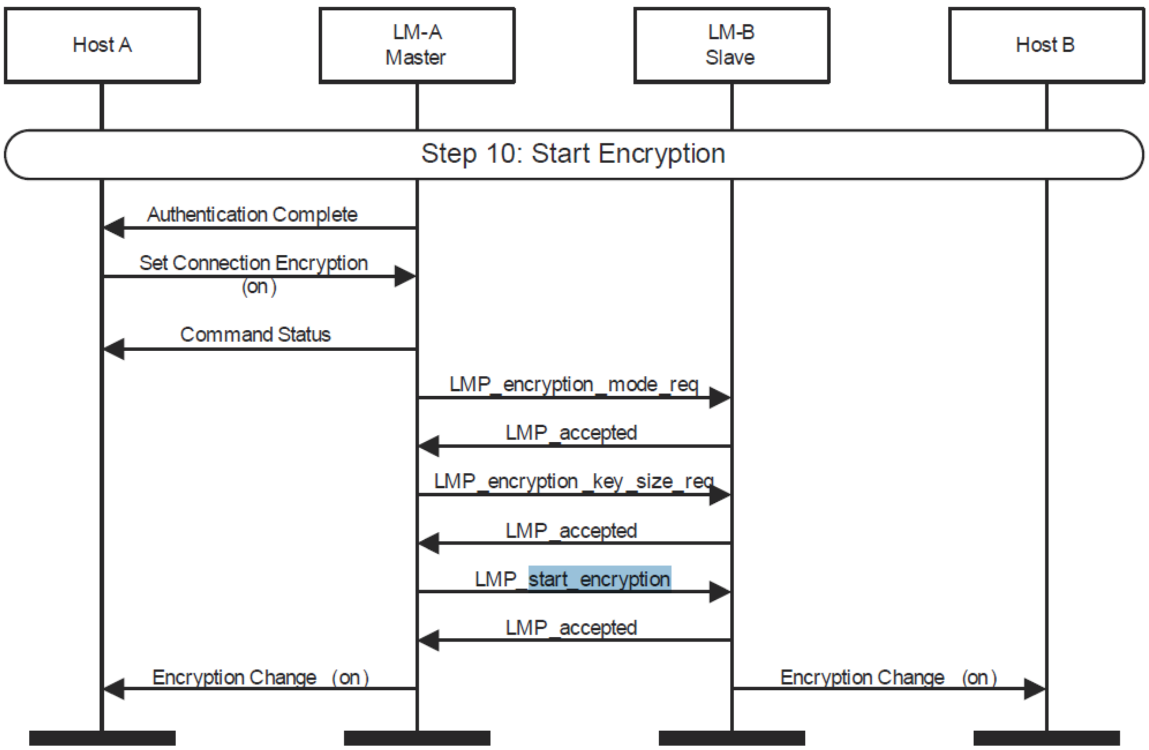
The following screenshot shows the Link Key found by the Ellisys software when SSP Debug Mode is used:



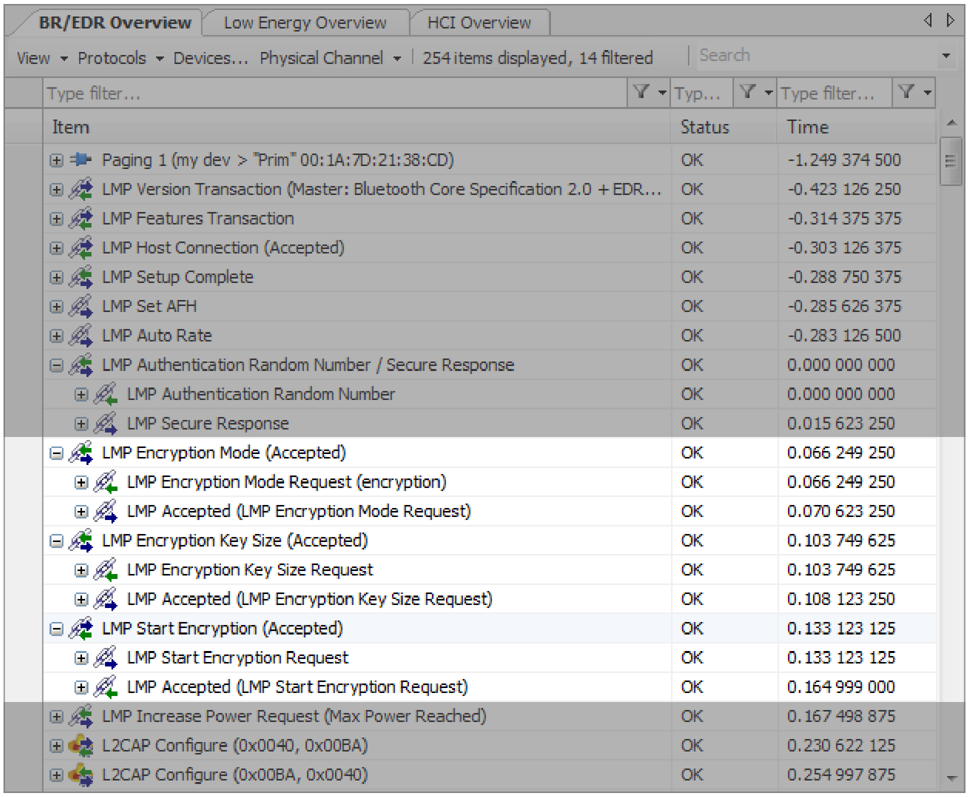
Authenticated Connection

After pairing, the two devices are sharing the same Link Key. These devices can then use the Link Key for authentication (to make sure that the other device is really who it says it is) and to derive the encryption key (to protect data exchanged by the two devices).

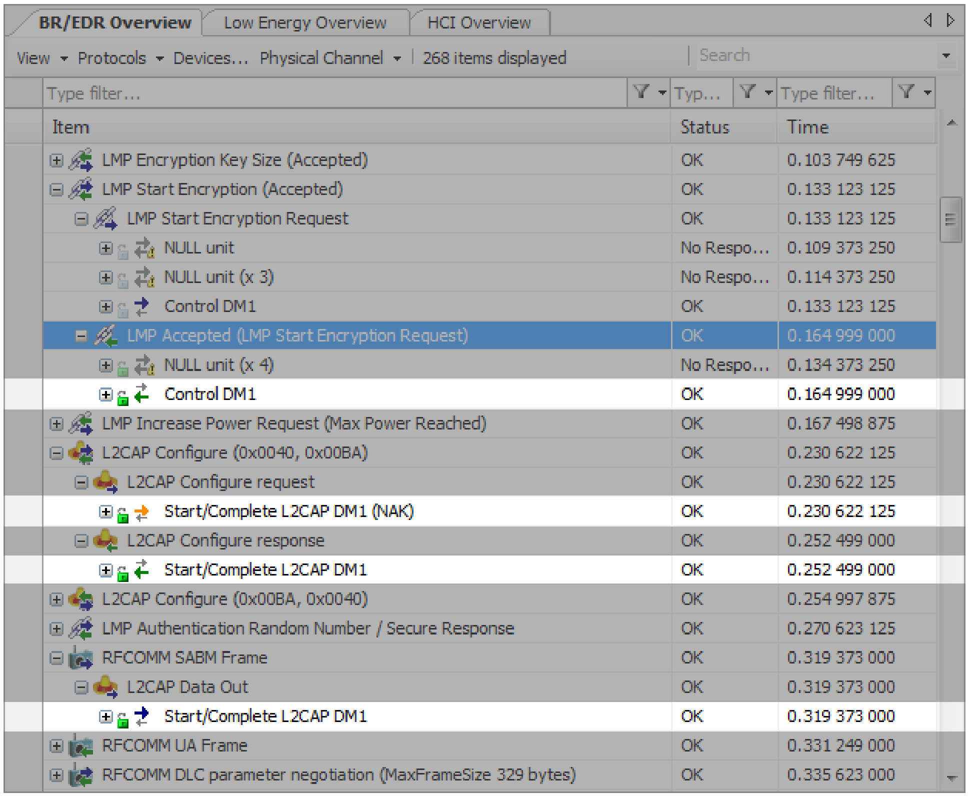
The data is encrypted right after the LMP_start_encryption request. The complete Start Encryption procedure is depicted in the following chart:



Here is the same traffic displayed in the Ellisys software:



It is interesting to note that the packets will be encrypted right after the LMP_start_encryption request, so even the LMP_accepted handshake will already be encrypted. The following screenshot shows which packets are encrypted. When the lock icon is blue, this means that the packets are plain (not encrypted). When the lock icon is green, this means that the packets have been successfully decrypted.



Feedback

Feedback on our Expert Notes is always appreciated. To provide comments or critiques of any kind on this paper, please feel free to contact us at expert@ellisys.com.

Other interesting readings

- EEN_BT01 - Capturing Bluetooth Traffic, the Right Way
- EEN_BT03 - Your First Wide-Band Capture
- EEN_BT06 - Bluetooth Security - Truths and Fictions
- More Ellisys Expert Notes available at: http://www.ellisys.com/technology/expert_notes.php

Rev. A. Updated 2011-05-16