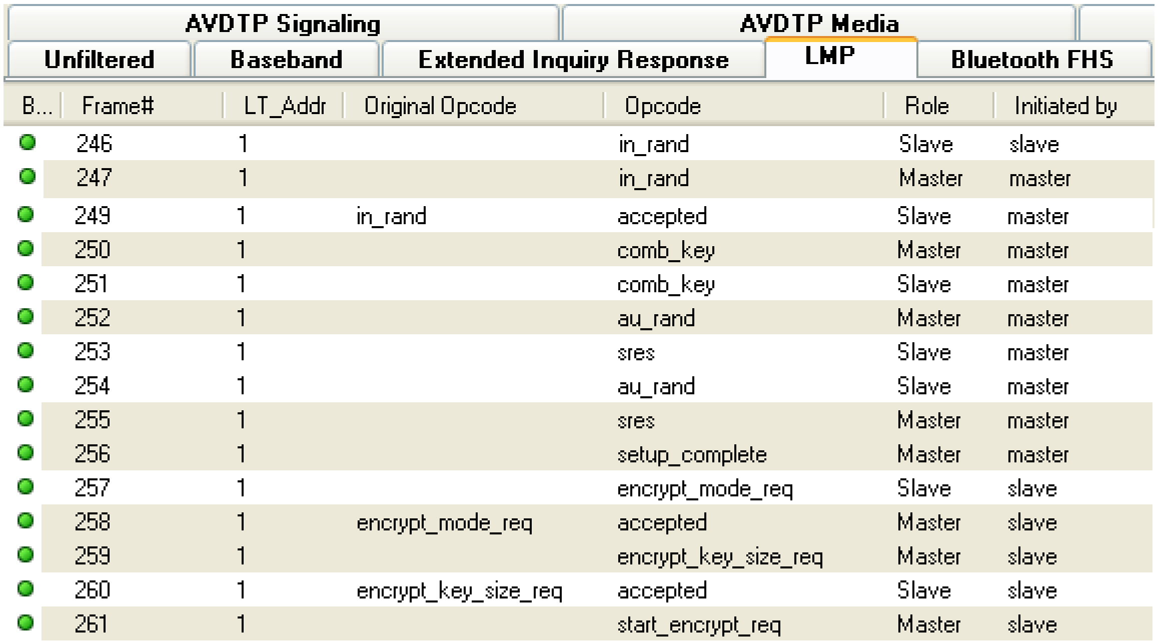
Decrypting Encrypted Bluetooth data with FTS4BT

How Encryption Works in Bluetooth:

Bluetooth devices on an encrypted link share a common link key in order to exchange encrypted data. How that link key is created depends upon the pairing method used.

Legacy Pairing (Bluetooth 2.0 and earlier)

In legacy pairing, this link key is derived from a shared PIN code, the master's Bluetooth clock, the master's BD_ADDR and a random number that is passed between the two devices. The sequence of events used to create this key, or pairing process, is shown in the FTS4BT Frame Display below.



B...	Frame#	LT_Addr	Original Opcode	Opcode	Role	Initiated by
●	246	1		in_rand	Slave	slave
●	247	1		in_rand	Master	master
●	249	1	in_rand	accepted	Slave	master
●	250	1		comb_key	Master	master
●	251	1		comb_key	Slave	master
●	252	1		au_rand	Master	master
●	253	1		sres	Slave	master
●	254	1		au_rand	Slave	master
●	255	1		sres	Master	master
●	256	1		setup_complete	Master	master
●	257	1		encrypt_mode_req	Slave	slave
●	258	1	encrypt_mode_req	accepted	Master	slave
●	259	1		encrypt_key_size_req	Master	slave
●	260	1	encrypt_key_size_req	accepted	Slave	slave
●	261	1		start_encrypt_req	Master	slave

Frame 247 is the LMP_in_rand which is where a random number generated by the master is passed to the slave. The slave acknowledges that it has accepted the number in frame 249.

In frames 250 and 251, the combination keys are passed between master and slave. In frame 252, the master sends its LMP_au_rand. This is the random number that has been encrypted using the link key that master has calculated. The slave then responds with frame 253, an LMP_sres confirming that it was able to compute the same number. That process is repeated in the other direction (slave to master) in frames 254 and 255. Then the setup_complete message is sent and they're slave requests encryption mode in frame 257, and the master accepts in frame 258. The actual encryption starts after the start encryption request in frame 261.

In order for FTS to decrypt an encrypted Bluetooth conversation, FTS must compute the same link key being used by the devices being sniffed. Since this link key is never sent over the air, FTS must have all of the same information the devices being sniffed have so that it can calculate the same link key that each of the two devices does. To decrypt successfully, FTS must know the PIN code and capture:

- The LMP_in_rand
- Both LMP_comb_keys
- Both LMP_au_rand/LMP_sres pairs.

If any of these are missed, FTS4BT will not be able to decrypt. If you capture encrypted data and find that everything captured after the LMP_start_encryption_request is in error, look back at the LMP frames previous to that and you'll probably find one or more of these missing.

Creating a link key with Secure Simple Pairing (SSP)

To capture and decrypt data between two Bluetooth devices using Secure Simple Pairing you have two choices. If one of your devices can be put into Secure Simple Pairing Debug Mode, all that needs to be done in I/O Settings is to choose your master and slave. It doesn't matter what's been selected in the Pairing Method drop down, FTS will see the debug messages being captured and automatically use SSP debug mode.

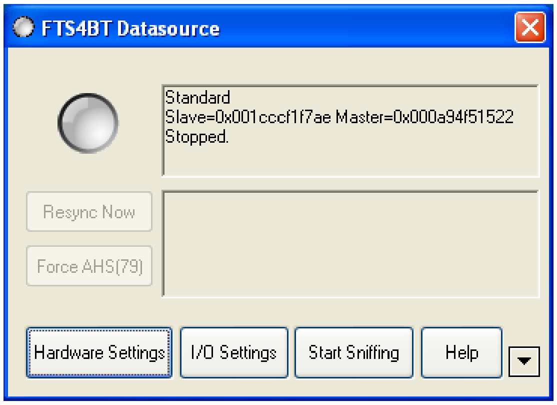
The Private Key edit box should be left blank.

If neither of your devices can be put into debug mode, you'll need to know the Private Key being used by one of your devices. If that is the case, enter the Private Key into the box provided.

Once the link key has been created, decryption operates the same way it does in legacy pairing.

How to Capture and Decrypt Data with FTS4BT

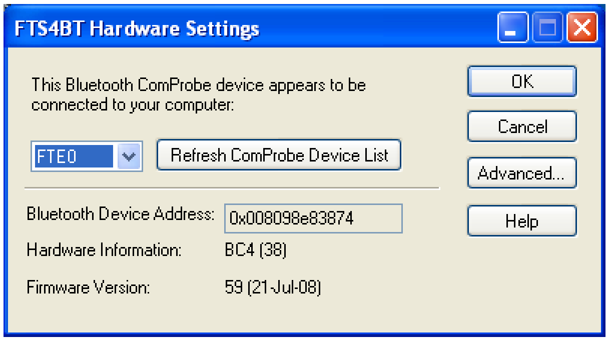
1. Open FTS4BT "Air Sniffer (Basic) in the desktop folder.



Standard
Slave=0x001cccf1f7ae Master=0x000a94f51522
Stopped.

This will bring up the FTS4BT Datasource. This is where parameters are set for sniffing, including synchronization mode, and the devices to be sniffed.

Hardware Settings.
Select the Hardware Settings button.



FTS4BT Hardware Settings

This Bluetooth ComProbe device appears to be connected to your computer:

FTE0

Bluetooth Device Address: 0x008098e83874

Hardware Information: BC4 (38)

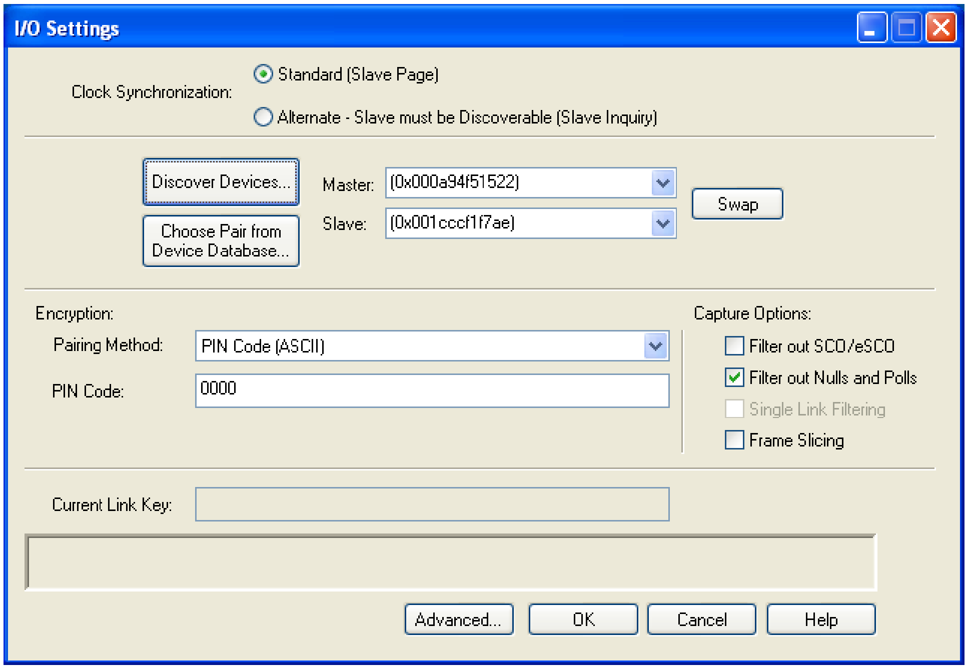
Firmware Version: 59 (21-Jul-08)

Here the Bluetooth ComProbe may be selected and tested.

- Click Advanced.
- If you have previously captured the pairing of devices, a list of link keys that have been previously calculated by FTS4BT will be displayed here. Two devices that have previously paired will use the same link keys until they are paired again.

2. Open the I/O Parameter window on the FTS4BT Datasource.

How to setup the I/O Parameters.



I/O Settings

Clock Synchronization: ● Standard (Slave Page) ○ Alternate - Slave must be Discoverable (Slave Inquiry)

Master: (0x000a94f51522)

Slave: (0x001cccf1f7ae)

Encryption:

Pairing Method: PIN Code (ASCII)

PIN Code: 0000

Capture Options:

☐ Filter out SCO/eSCO

☑ Filter out Nulls and Polls

☐ Single Link Filtering

☐ Frame Slicing

Current Link Key:

- Make both of your devices discoverable.
- Press the “Discover Devices” button. FTS4BT will find any discoverable Bluetooth devices within its range. You will then be able to select your devices from the master and slave drop down lists. If one or both of your devices can’t be made discoverable, you may type in the BD_ADDR(s) directly.

1. Select the synchronization mode that best suits your application. Clock synchronization is how FTS4BT learns the clock that is being used on the piconet.

Standard (Slave Page)

This is how FTS learns the clock of a device that is NOT discoverable. For example, after a phone and a headset have paired, often the headset will not be discoverable. If the headset is a slave device and it is not discoverable, then FTS4BT will not be able to synchronize to that device using Slave Inquiry mode. If we know the headset (slave) BD-ADDR then by using Slave Page mode, FTS4BT will be able to page the device, (but will never complete the connection during the page session). Once FTS4BT learns the clock information during the paging process, FTS4BT will discontinue the paging process and will now be synchronized to the undiscoverable slave's clock.

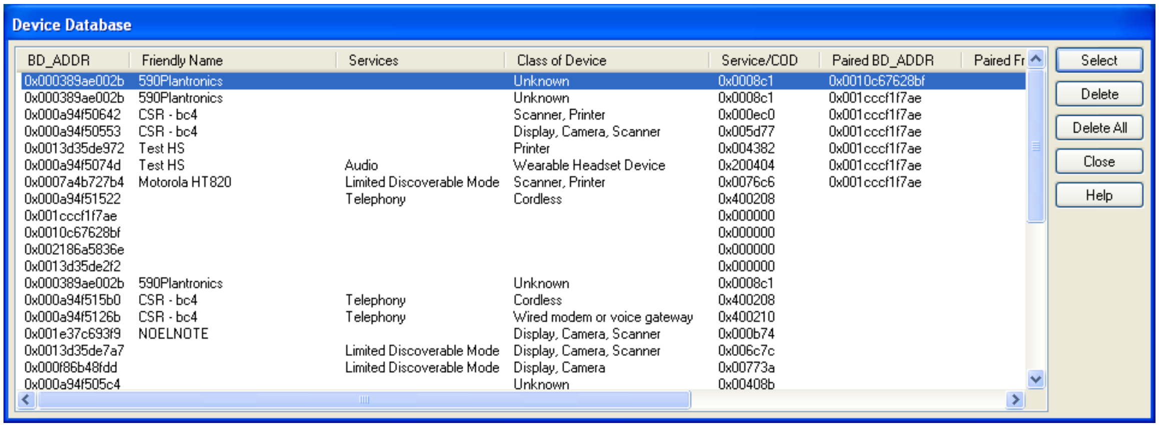
Slave Page is highly recommended, especially when the link is encrypted. Slave Page locks on to the master's clock faster than Master Inquiry and so is less likely to miss any of the encryption set up information FTS4BT needs to decrypt the subsequent data.

Alternate – Slave must be discoverable (Slave Inquiry)

FTS4BT will perform an inquiry of the slave and determine its clock. In this mode, the slave must be discoverable.

4. Inform FTS4BT which device is going to be master and which device is going to be slave. **Note that it is necessary to select both a master and slave device if the link you are sniffing is using encryption**.

5. Enter Encryption PIN code. As mentioned above, FTS4BT needs the PIN code in order to calculate the link key the two devices are using. Alternately, you may enter the Link Key manually if it is known. FTS4BT also keeps a database of the link keys it's previously calculated, which may be accessed by clicking the "Choose Pair from Device Database button:

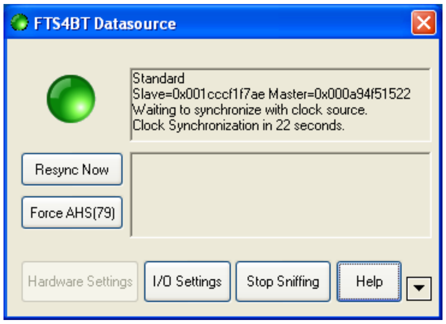


BD_ADDR	Friendly Name	Services	Class of Device	Service/COD	Paired BD_ADDR	Paired Fr
0x000389ae002b	590Plantronics		Unknown	0x0008c1	0x0010c67628bf	
0x000389ae002b	590Plantronics		Unknown	0x0008c1	0x001cccf1f7ae	
0x000a94f50642	CSR - bc4		Scanner, Printer	0x000ec0	0x001cccf1f7ae	
0x000a94f50553	CSR - bc4		Display, Camera, Scanner	0x005d77	0x001cccf1f7ae	
0x0013d35de972	Test HS		Printer	0x004382	0x001cccf1f7ae	
0x000a94f5074d	Test HS	Audio	Wearable Headset Device	0x200404	0x001cccf1f7ae	
0x0007a4b727b4	Motorola HT820	Limited Discoverable Mode	Scanner, Printer	0x0076c6	0x001cccf1f7ae	
0x000a94f51522		Telephony	Cordless	0x400208		
0x001cccf1f7ae				0x000000		
0x0010c67628bf				0x000000		
0x002186a5836e				0x000000		
0x0013d35de2f2				0x000000		
0x000389ae002b	590Plantronics		Unknown	0x0008c1		
0x000a94f515b0	CSR - bc4	Telephony	Cordless	0x400208		
0x000a94f5126b	CSR - bc4	Telephony	Wired modem or voice gateway	0x400210		
0x001e37c693f9	NOELNOTE		Display, Camera, Scanner	0x000b74		
0x0013d35de7a7		Limited Discoverable Mode	Display, Camera, Scanner	0x006c7c		
0x000f86b48fdd		Limited Discoverable Mode	Display, Camera	0x00773a		
0x000a94f505c4			Unknown	0x00408b		

6. Select the **Filter Out** (eSCO/SCO, NULL, POLL) set up. These filters are low level hardware filters. There are also Display Filters that you may use later to filter the captured data. Any data filtered out here will not be captured at all. In most cases the default set up (filter out nulls and polls, leave SCO data in) will be OK.

7. The OK button should now be available. If OK is grayed out, there is something set up incorrectly in the I/O Parameter window. For example, if you selected PIN code in the Encryption drop down but you neglected to fill in the PIN code, then OK will be grayed out.

8. Press OK, and then Start Sniffing. When you start sniffing, the Status Icon will go red (not yet synchronized). After a short time (a few seconds) the icon will turn green and the status will change to "Waiting to synchronize with clock source". At this point FTS4BT is synchronized and waiting for a baseband connection.



Standard
Slave=0x001cccf1f7ae Master=0x000a94f51522
Waiting to synchronize with clock source.
Clock Synchronization in 22 seconds.

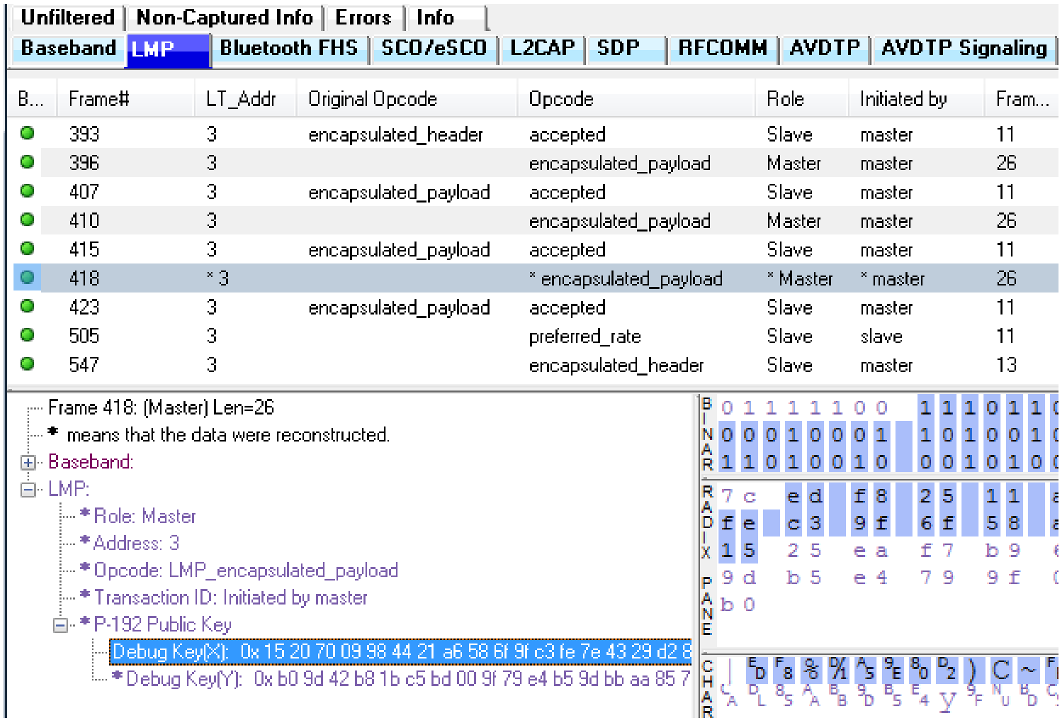
Note that there is a clock counting down from 30 seconds. When it reaches 5 seconds, the icon will turn yellow, to warn you that at 0 it will resync. This is to avoid the ComProbe's clock from drifting to far in relation to the master's clock. While it's resyncing, FTS4BT will not be capturing data, so make sure to connect your devices while the icon is green. **It is recommended that you click the Resync Now button just before you start the pairing process.** This will ensure that the clocks are closely synchronized.

When your connection is established, the Status Icon will turn blue, signifying that a baseband link has been established and data should start to appear in the Frame Display.

How to tell if one of your devices is in Secure Simple Pairing Debug Mode

When a device is configured in SSP debug mode, FTS4BT will decode and display the debug key in the Encapsulated Payload message. There will be an Encapsulated Payload message sent from both the master and the slave. The message from the device that is in debug mode will show the debug key, the other will show the public key. Remember, only one of the devices needs to be in SSP debug mode.

Encapsulated Payload message from a device in SSP Debug Mode:



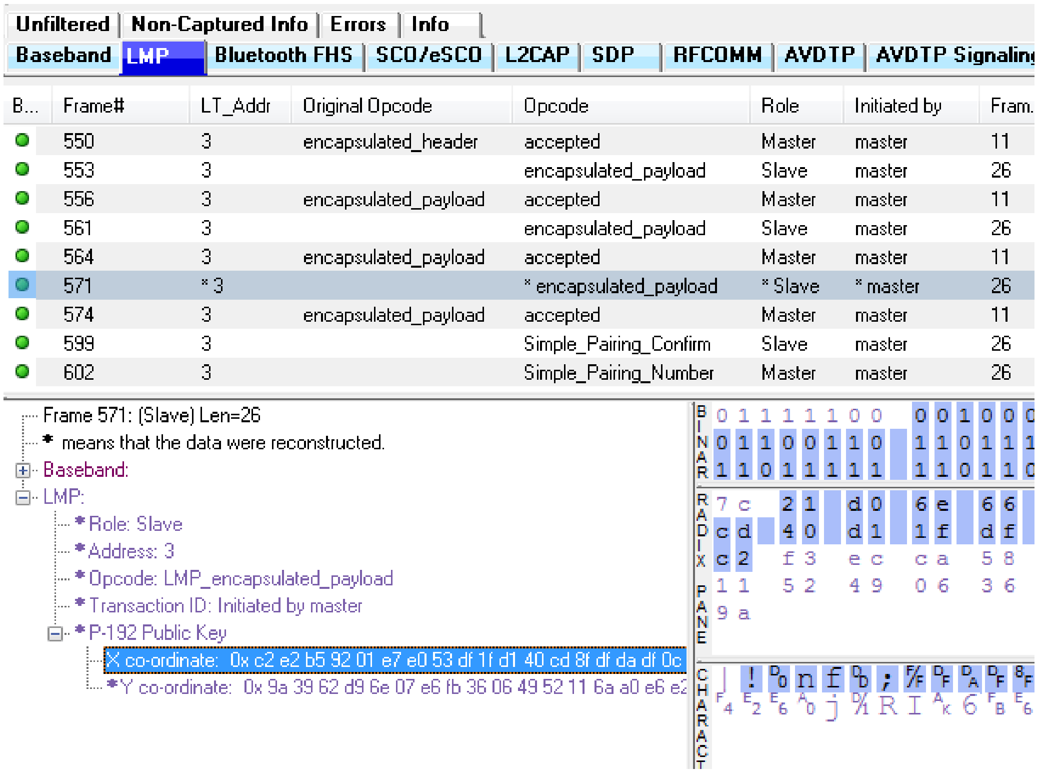
B...	Frame#	LT_Addr	Original Opcode	Opcode	Role	Initiated by	Fram...
●	393	3	encapsulated_header	accepted	Slave	master	11
●	396	3		encapsulated_payload	Master	master	26
●	407	3	encapsulated_payload	accepted	Slave	master	11
●	410	3		encapsulated_payload	Master	master	26
●	415	3	encapsulated_payload	accepted	Slave	master	11
●	418	* 3		* encapsulated_payload	* Master	* master	26
●	423	3	encapsulated_payload	accepted	Slave	master	11
●	505	3		preferred_rate	Slave	slave	11
●	547	3		encapsulated_header	Slave	master	13

```
Frame 418: (Master) Len=26
* means that the data were reconstructed.
Baseband:
LMP:
    * Role: Master
    * Address: 3
    * Opcode: LMP_encapsulated_payload
    * Transaction ID: Initiated by master
    * P-192 Public Key
        Debug Key(X):  0x 15 20 70 09 98 44 21 a6 58 6f 9f c3 fe 7e 43 29 d2 8
        * Debug Key(Y):  0x b0 9d 42 b8 1b c5 bd 00 9f 79 e4 b5 9d bb aa 85 7
```

```
B 0 1 1 1 1 1 0 0   1 1 1 0 1 1
N 0 0 0 1 0 0 0 1   1 0 1 0 0 1
R 1 1 0 1 0 0 1 0   0 0 1 0 1 0

R 7 c   e d   f 8   2 5   1 1
D f e   c 3   9 f   6 f   5 8
X 1 5   2 5   e a   f 7   b 9
P 9 d   b 5   e 4   7 9   9 f
E b 0
```

Encapsulated Payload message from a device **NOT** in SSP Debug Mode:



B...	Frame#	LT_Addr	Original Opcode	Opcode	Role	Initiated by	Fram.
●	550	3	encapsulated_header	accepted	Master	master	11
●	553	3		encapsulated_payload	Slave	master	26
●	556	3	encapsulated_payload	accepted	Master	master	11
●	561	3		encapsulated_payload	Slave	master	26
●	564	3	encapsulated_payload	accepted	Master	master	11
●	571	* 3		* encapsulated_payload	* Slave	* master	26
●	574	3	encapsulated_payload	accepted	Master	master	11
●	599	3		Simple_Pairing_Confirm	Slave	master	26
●	602	3		Simple_Pairing_Number	Master	master	26

```
Frame 571: (Slave) Len=26
* means that the data were reconstructed.
Baseband:
LMP:
    * Role: Slave
    * Address: 3
    * Opcode: LMP_encapsulated_payload
    * Transaction ID: Initiated by master
    * P-192 Public Key
        X co-ordinate:  0x c2 e2 b5 92 01 e7 e0 53 df 1f d1 40 cd 8f df da df 0c
        *Y co-ordinate:  0x 9a 39 62 d9 6e 07 e6 fb 36 06 49 52 11 6a a0 e6 e2
```

```
BINARY
0 1 1 1 1 1 0 0   0 0 1 0 0 0
0 1 1 0 0 1 1 0   1 1 0 1 1 1
1 1 0 1 1 1 1 1   1 1 0 1 1 0

RADIX PANE
7 c   2 1   d 0   6 e   6 6
c d   4 0   d 1   1 f   d f
c 2   f 3   e c   c a   5 8
1 1   5 2   4 9   0 6   3 6
9 a

CHARACT
]  !  D0  n  f  DD  ;  F/F  DF  DA  DF  8F
F4  E2  E6  A0  j  D/1  R  I  Ak  6  FB  E6
```