Hacking Bluetooth

Bastian Ballmann & Martin Karger

Hacking Bluetooth

Standard tools

Tools – BlueZ / OBEX

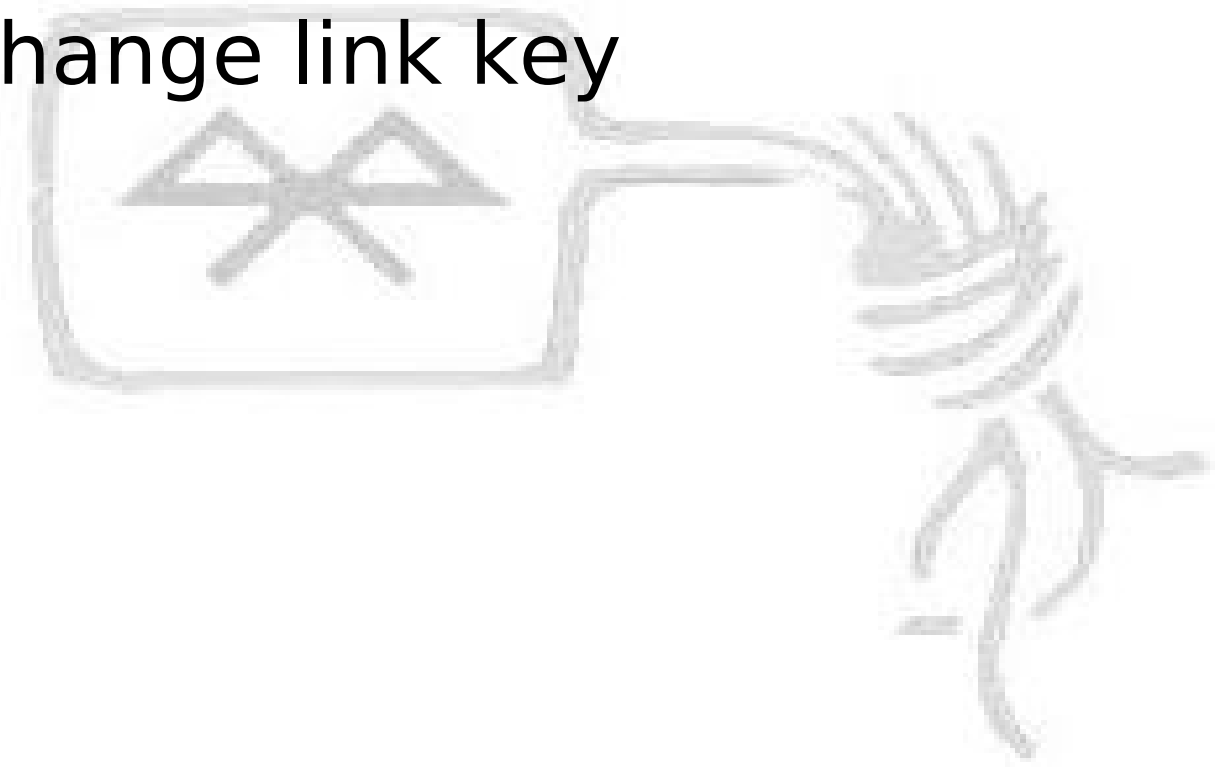
- hciconfig – Device configuration
- hcitool – Handling connections
- hcidump – Sniffing
- l2ping – L2CAP echo request
- sdptool / sdp – Service discovery
- btobex / obexftp – Object Exchange
- dfutool – Firmware up-/download
- bccmd – CSR BCCMD interface

Tools - hciconfig

- Device configuration
- noscan – Non-discoverable mode
- class 0x000204 – Claim to be a phone
- noauth – Disable authentication
- noencrypt – Disable encryption

Tools - hcitool

- scan – Scan for devices
- info – Information about remote device
- key – Change link key



Tools - hcidump

- sniff traffic directed to local devices
- Use -X to dump hex and ascii
- -A to sniff SCO audio data

Tools - l2ping

- L2CAP echo request
- -c <count>
- -s <size>
- New Ping of death for Bluetooth ^^
- Can be used to DOS some PDAs and phones (e.g. Widcomm stack)

Tools – sdptool / sdpd

- browse – Query remote SDP daemon
- search – Search for services
- Remember not every service is listed in SDP (yeah we all love Blue Bug! =)
- sdpd – start SDP daemon
- sdptool add / del – Add or delete records

Tools – btobex / obexftp

- Obex – Object Exchange protocol
- The good old Bluesnarf attack
  - btobex pb <addr> <channel>
  - btobex cal <addr> <channel>
- Bluesnarf on Sony Ericsson phones
  - obexftp -b <addr> -B 10 -g telecom/pb.vcf
- Bluejacking
  - btobex push <addr> <file>
- Directory Traversal on OBEX FTP servers

Tools - dfutool

- Up-/download firmware
- Part of USB specification (optional)
- How to get it
- cvs -d:pserver:anonymous:cvs.bluez.org:/cvsroot/bluez login
- cvs -d:pserver:anonymous:cvs.bluez.org:/cvsroot/bluez co -P utils
- How to compile
- gcc -lusb -lbluetooth csr.c dfu.c dfutool.c -o dfutool
- How to use
- dfutool upgrade muh.dfu – upload firmware
- dfutool archive new.dfu – download firmware

Tools - bccmd

- BlueCore Command Protocol
- **Danger:** can brick your hardware!
- protocol not part of the Bluetooth Spec.
- vendor specific (CSR)
- tune your chip:
  - bdaddr
  - RX/TX
  - LMP, HCI version
  - Vendor ID
  - ...

Tools - bccmd

- How to get it
- cvs -d:pserver:anonymous:cvs.bluez.org:/cvsroot/bluez login
- cvs -d:pserver:anonymous:cvs.bluez.org:/cvsroot/bluez co -P utils
- How to compile
- gcc -lusb -lbluetooth csr.c csr_3wire.c csr_bcsp.c csr_h4.c csr_hci.c csr_usb.c ubcsp.c bccms.c -o bccmd
- How to use
- bccmd pslist
- bccmd psset 0x0001 0x08 0x07 0x06 0x05 0x04 0x03 0x02 0x01
- Sets Bluetooth address 01:02:04:08:05:06

Hacking Bluetooth

Bluetooth sniffing

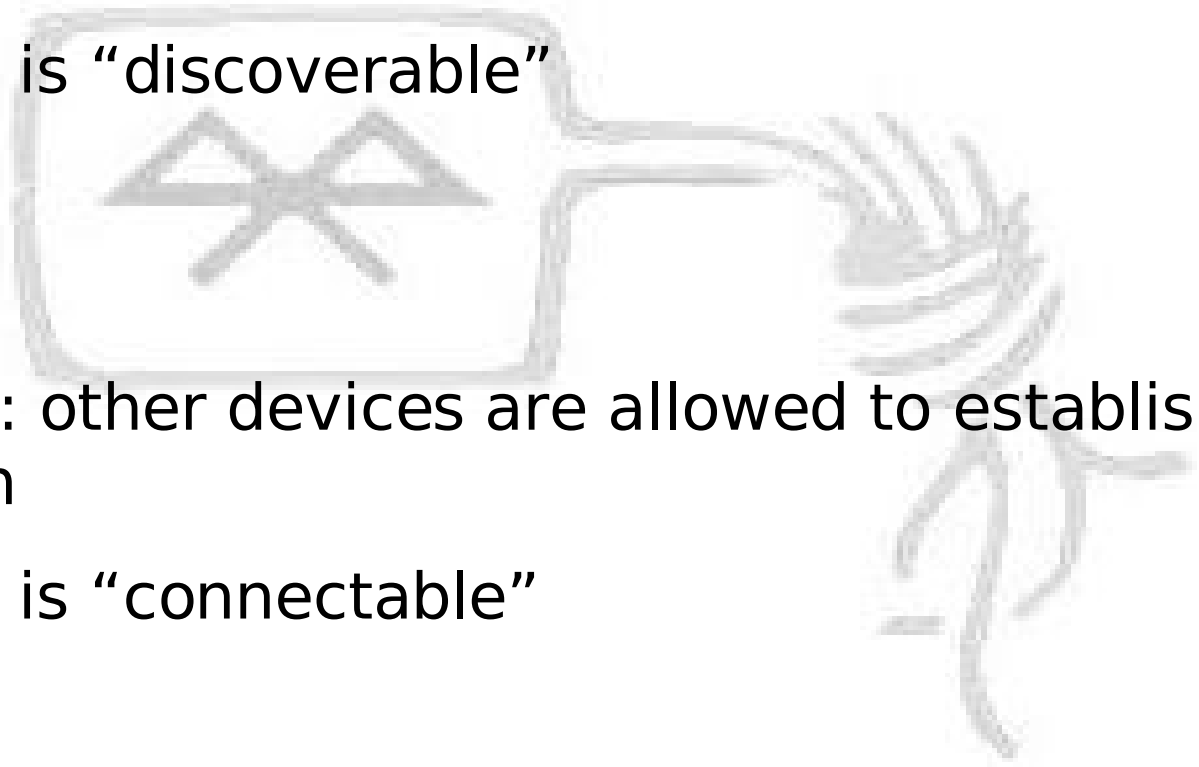
Sniffing - Inquiry and Page Scan

Inquiry Scan

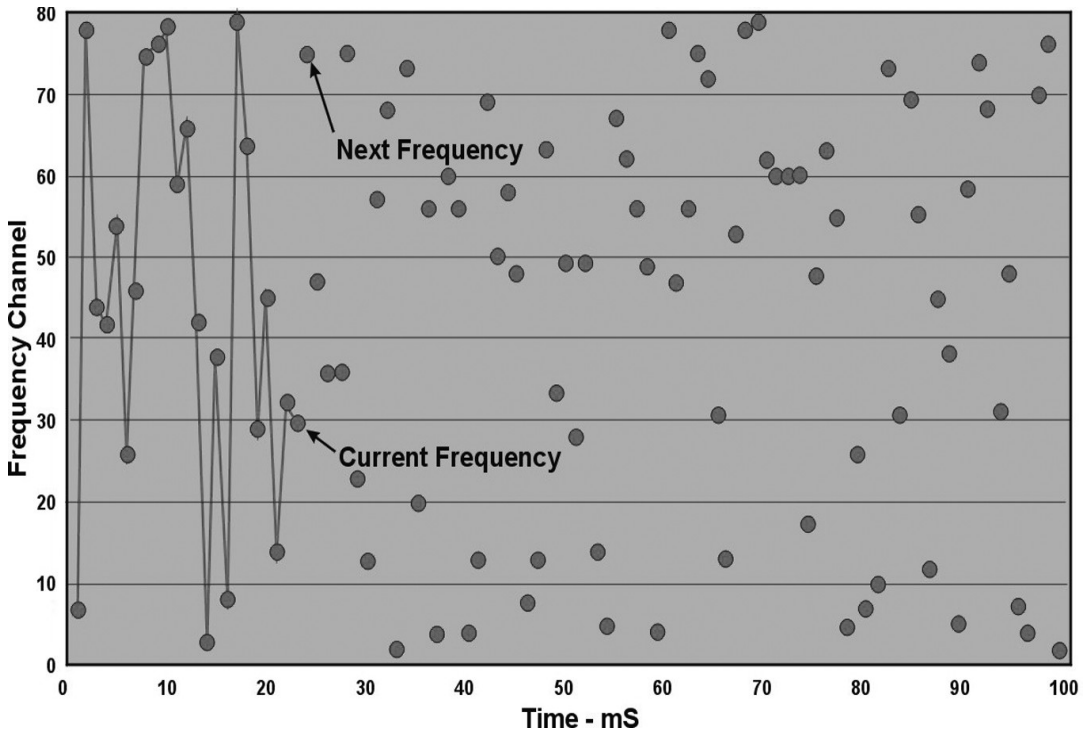
- Inquiry Scan: device will respond to other devices which are „Searching for Bluetooth devices...“ (Inquiring)
- the device is “discoverable”

Page Scan

- Page Scan: other devices are allowed to establish a connection
- the device is “connectable”

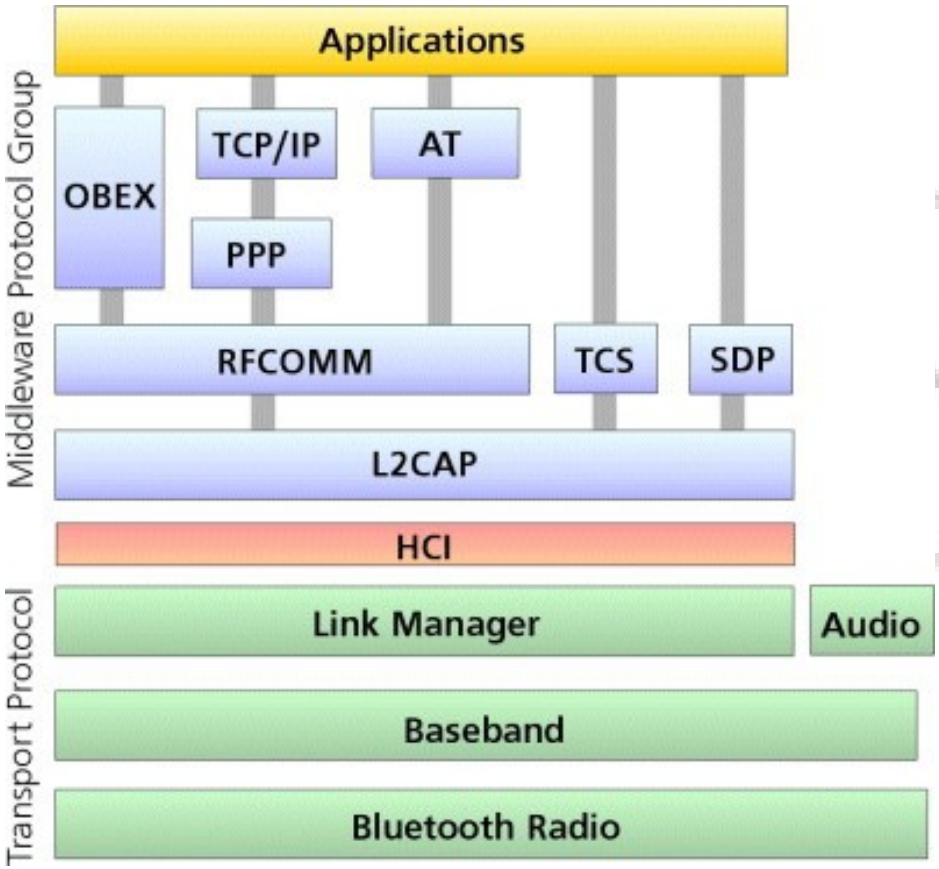


Sniffing - FHS



- 79 channels
- up to 1600 hops/sec
- Piconet hopping sequence:
  - channels: BD-ADDR Master
  - sync in time: Clock Master

Sniffing - HCI



Host Controller Interface

- hardware abstraction layer
- only minimal control over hardware
- no possibility to influence the hopping sequence
- no support for RAW packets

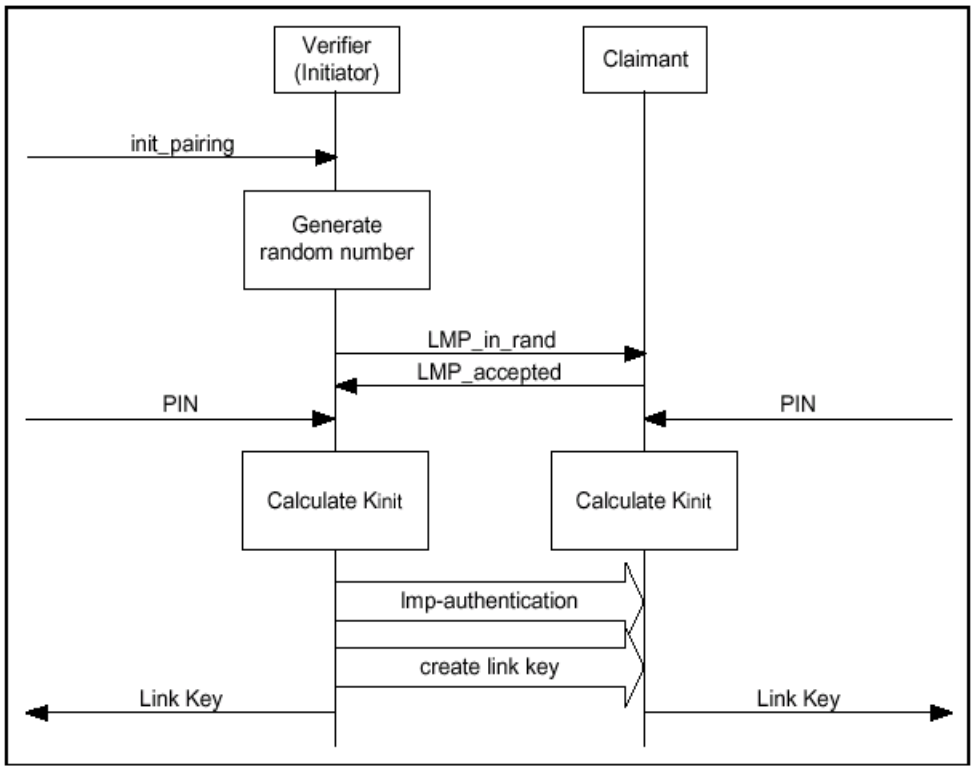
Sniffing - IDEA!!!

- implementing a custom firmware supporting raw access and control over frequency hopping
- sell it for $bignum EUR
- really works, ask Max:
  - “Transforming a consumer Bluetooth Dongle into a Bluetooth Sniffer”
- Frontline Test Equipment http://www.fte.com/
  - software & firmware: download
  - hardware & serial: ask your dealer for testing version

Sniffing – Sync Piconet

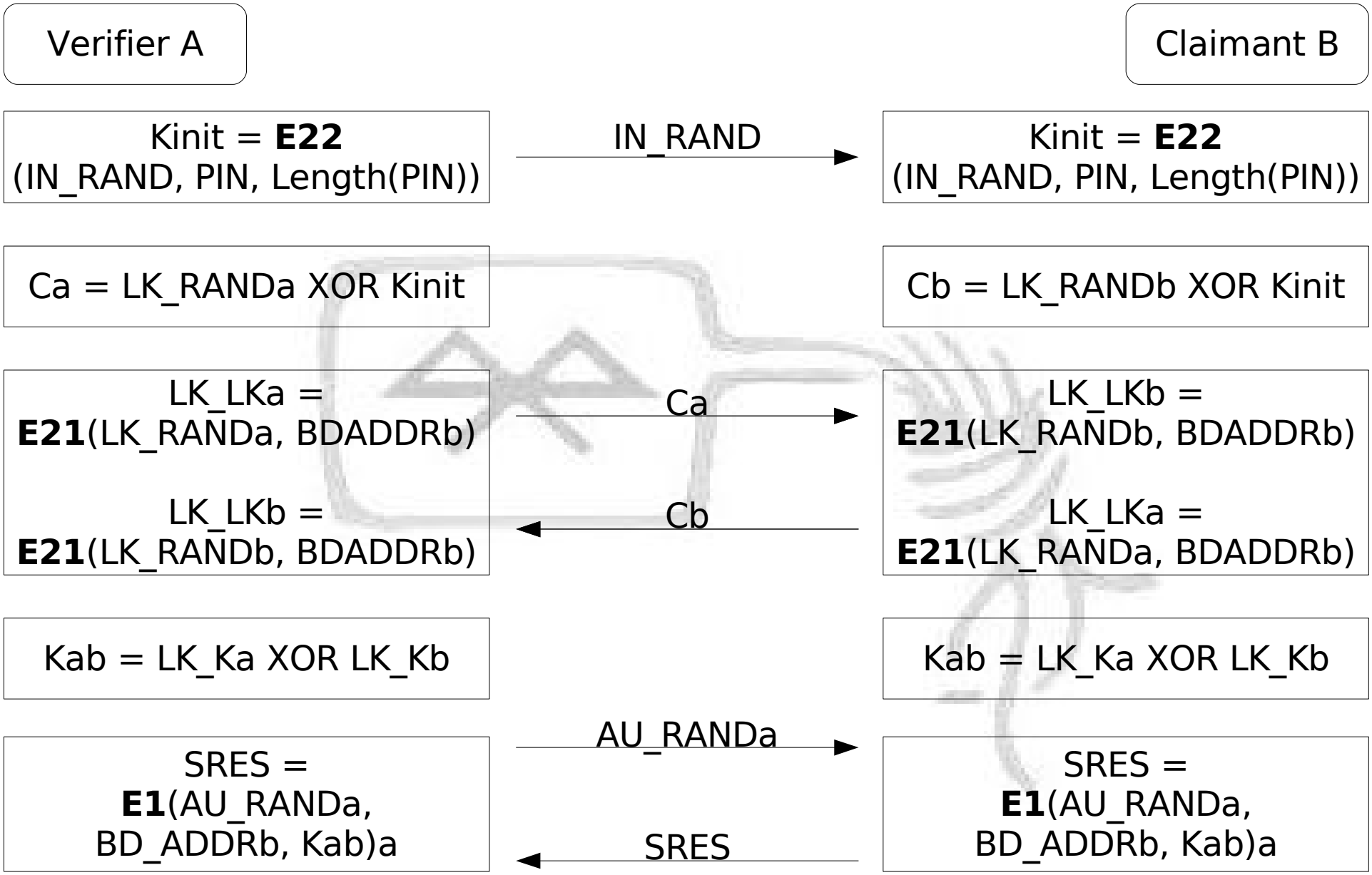
- Requirements
  - BD-ADDR must be known
  - Clock (can be obtained by inquiry or page
  - who is master/slave?
- Frontline
  - inquiry on slave -> sniffer following slave's hopping sequence
  - master paging slave -> sniffer following master's hopping sequence (Piconet)

Sniffing - PIN, Link-Key & Pairing



- PIN: User Input: „1234“
- Link-Key (K_{ab}, comb_key)
  - the real shared secret (not the pin)
  - derived from PIN
  - „0x6f924dead517fa6f781ef0beef86a7e7“
- Pairing
  - creation of a shared Link-Key
  - following connections rely on Link-Key

Sniffing - Pairing & Authentication



Dongle Cloning – Shopping List

- BT-Dongle
  - CSR Chipset
  - Type: Flash or External using Flash
  - ideal: CSR BC4 Chipset
  - 15 – 30€
- Bluez CVS: dfutool, bccmd (, bdaddr)

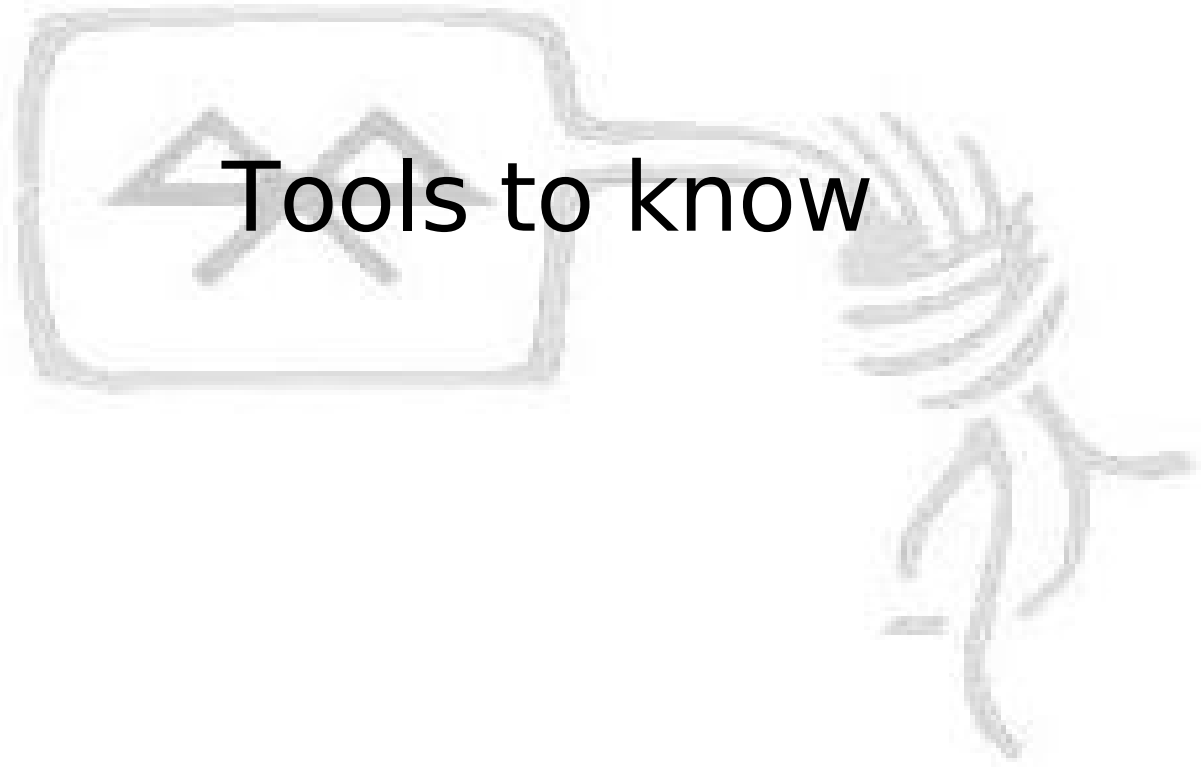
Playing with packets

Playing with packets - L2CAP

- Protocol Multiplexing (like IP)
- QoS (like ICMP)
- Segmentation / Reassembly (like TCP)
- Groupmanagement (like IGMP)
- Simple packetgenerator
  - code
  - ident
  - header size
  - http://www.datenterrorist.de/devel/l2cap-packet.c

Playing with packets - L2CAP

- Possible solutions for implementing a L2CAP connection resetter?
- We assume that there is no encryption or the link key is known
- Interessting packet types
  - L2CAP_COMMAND_REJ
  - L2CAP_CONN_RESP
    - 0x2 – 0x4 connection refused
  - L2CAP_CONF_REQ and MTU 0
  - L2CAP_CONF_REQ and QoS no traffic

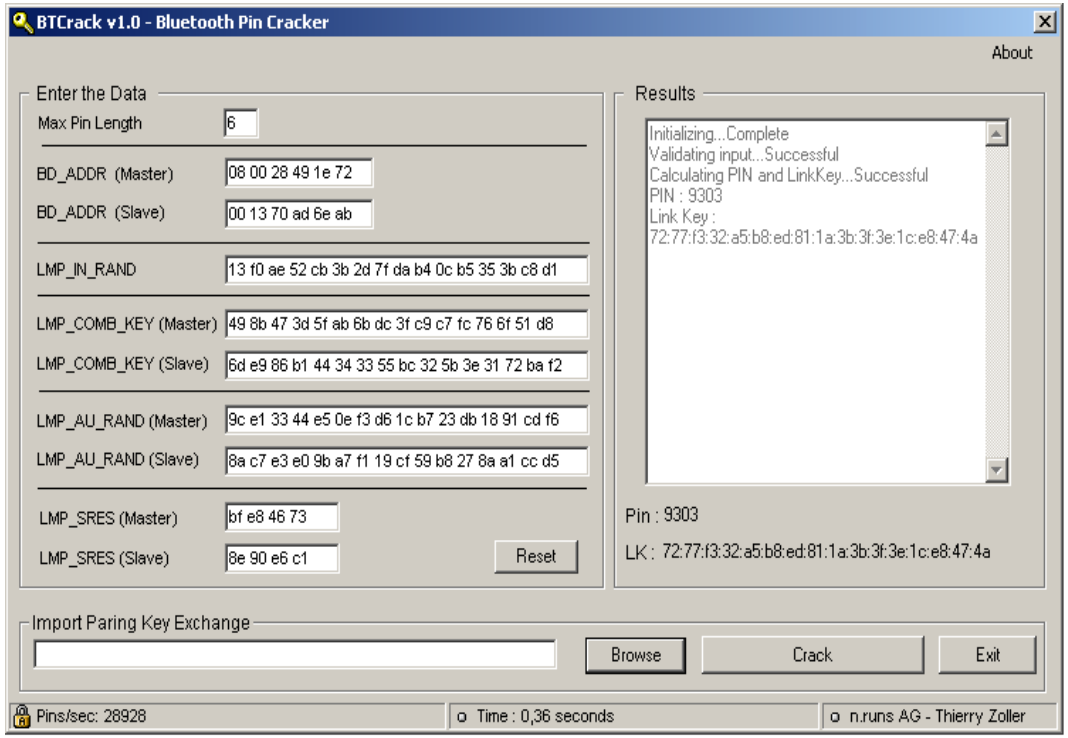


Tools to know

Tools to know

- BTCrack – Thierry Zoller
- carwhisperer – Martin Herfurt
- Hidattack – Collin Mulliner
- BSS – Pierre Betouin
- Bluediving – Bastian Ballmann

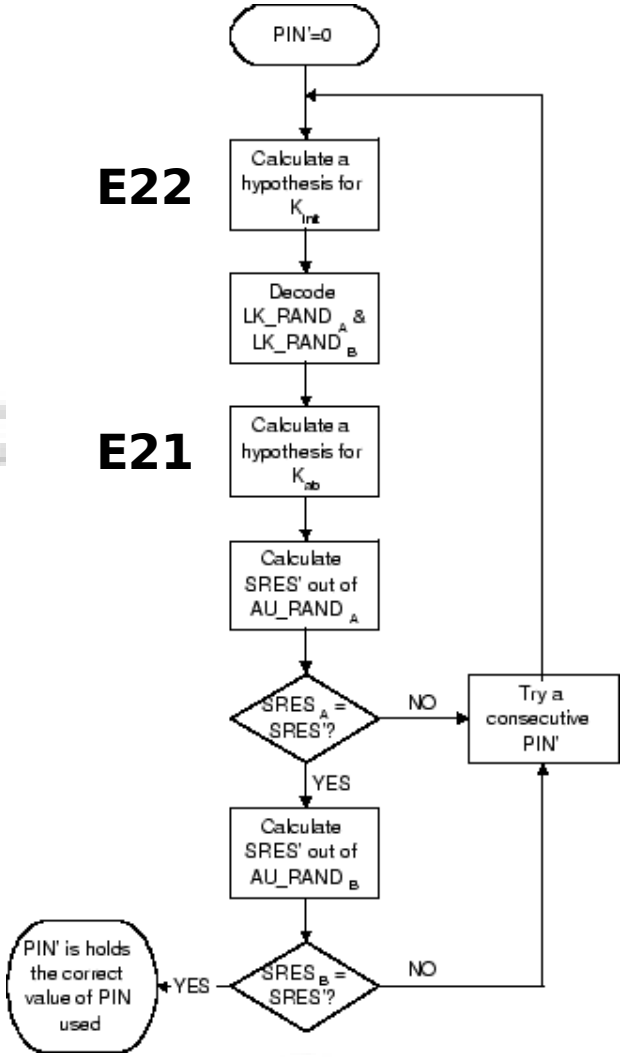
Tools - btcrack



- implementing attack on pairing process
- some issues right now (v1.0)
- source code to be released very soon

Tools - btcrack

```
Pin =-1;
Do
{
     PIN++;
     CR_K = E22(RAND, PIN,
       length(PIN));
     CR_RANDA = CA xor CR_K;
     CR_RANDB = CB xor CR_K;
     CR_LKA = E21(CR_RANDA, ADDRA);
     CR_LKB = E21(CR_RANDB, ADDRB);
     CR_LKAB = CR_LKA xor CR_LKB;
     CR_SRES = (CH_RAND, ADDRB,
       CR_LKAB);
}
while (CR_SRES == SRES)
```



Tools - btcrack: reality check

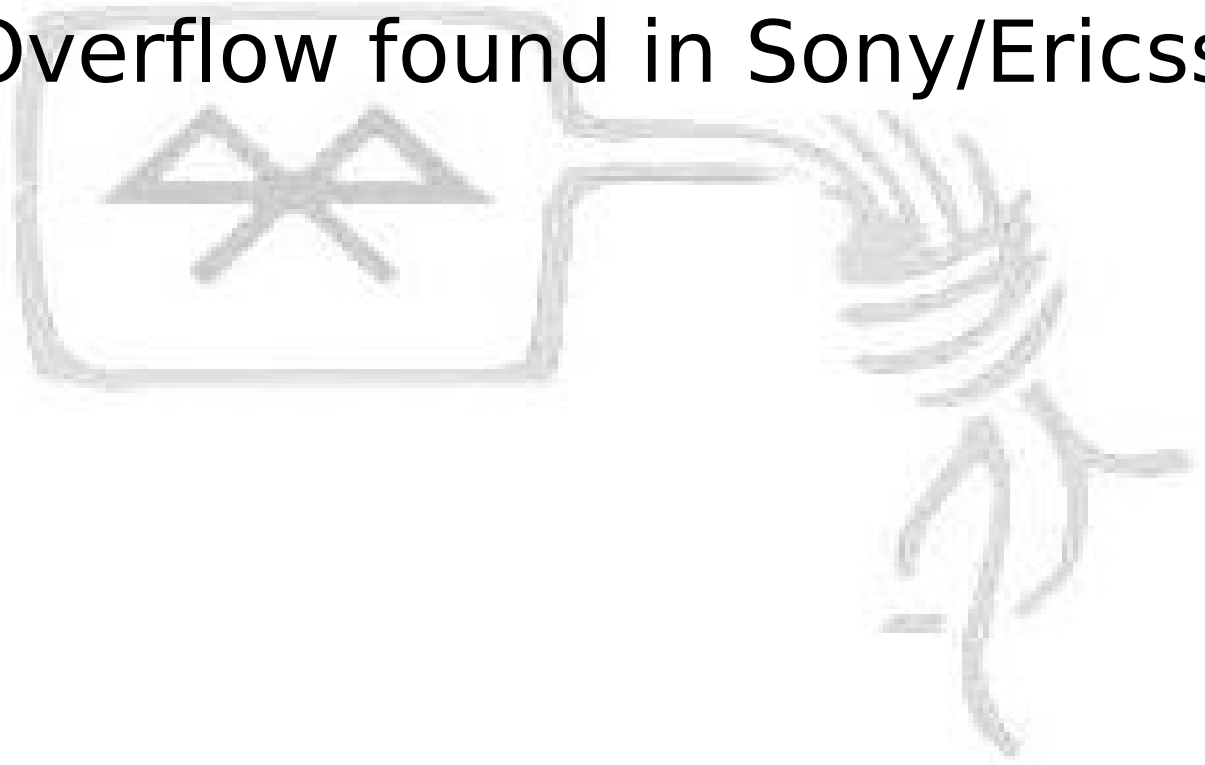
- re-pairing must be forced
- BT-ADDR(s) must be known
- Master/Slave roles must be known
- distance master <-> sniffer should be minimal
- even under good circumstances synchronization might become difficult
- until now: just an attack for your lab
- long range sniffing possible?

Tools - carwhisperer

- Inject audio to cars and headphones
- Record audio
- Realtime patch can be found under
- http://www.digitalmunition.com/carwhisper-realtime.tar

Tools - BSS

- Bluetooth Stack Smasher
- L2CAP fuzzer
- Buffer Overflow found in Sony/Ericsson phones



Tools – Hidattack

- Hijacking bluetooth keyboards
- currently no realtime support :/
- Our device must be a HID device
- hciconfig hci0 class 0x002540
- We must add a SDP keyboard record
- sdpd; spdtool add hid

Tools - Bluediving

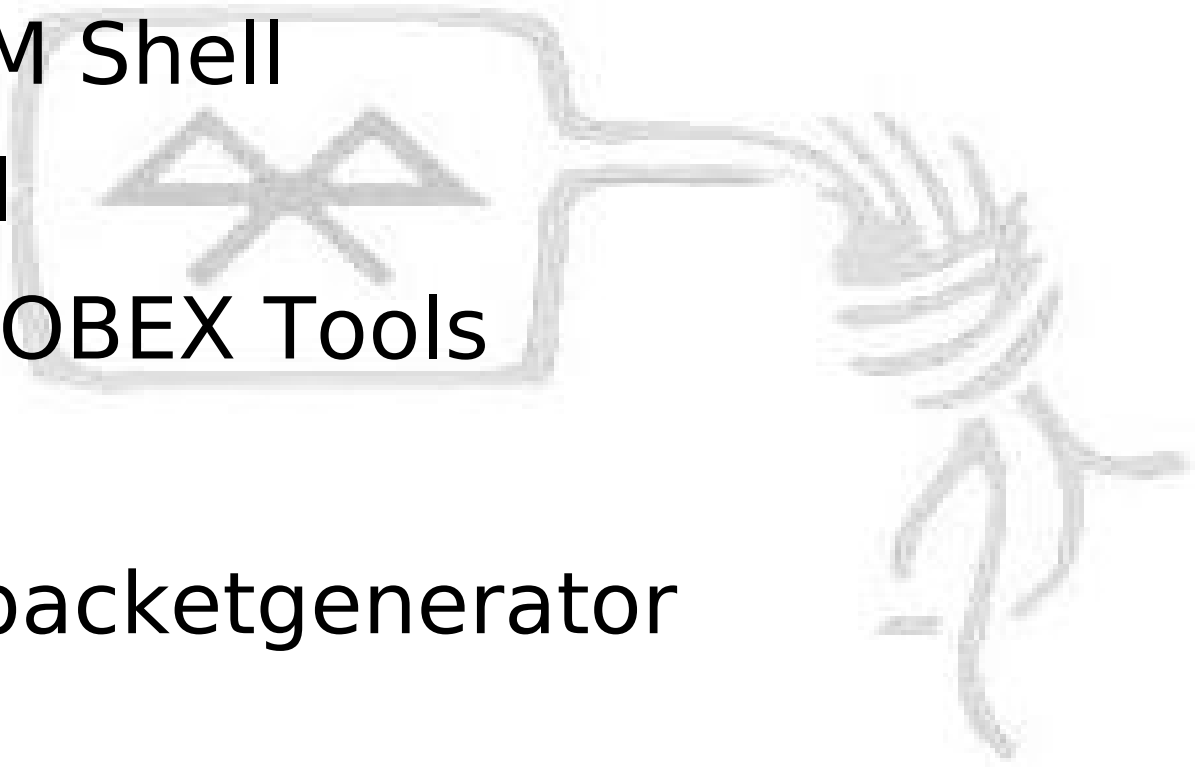
- Linux and FreeBSD version
- Search for devices
- Implements several exploits
- Can automatically attack devices based on vendor part of MAC and SDP scan
- Bluetooth address spoofing
- RFCOMM scanner
- Frontend to common tools

Bluediving - Exploits

- Blue Snarf / Blue Snarf++
- Blue Bug
- Helo Moto
- Blue Smack
- Symbian DOS (malicious device name)
- OBEX Overflow

Bluediving – Implemented tools

- Redfang
- Carwhisperer (with realtime patch)
- RFCOMM Shell
- AT Shell
- BlueZ / OBEX Tools
- BSS
- L2CAP packetgenerator



Hacking Bluetooth

BT 2.1 - Secure Simple Pairing

- Secure Simple Pairing
  - Elliptic Curve Diffie-Hellman (ECDH)
  - MITM Protection
  - Passive Eavesdropping Protection
  - multiple Association Models
    - Numeric Comparison
    - Just Works
    - Out Of Band (e.g. NFC)
    - Passkey Entry

Links to know

- www.holtmann.org
- www.trifinite.org
- www.mulliner.org
- www.digitalmunition.com
- www.zoller.lu
- www.datenterrorist.de
- www.evilgenius.de
- www.chaostal.de

Hacking Bluetooth

Happy hacking out there!