Bluetooth Smart, But Not Smart Enough

Mike Ryan
iSEC Partners

iSEC Open Forum

Jan 31, 2012

iSECpartners
part of nccgroup

Slides and More Info

http://lacklustre.net/bluetooth/

Overview

- Three parts
  - what is LE
  - how do we sniff it → demo!
  - security analysis

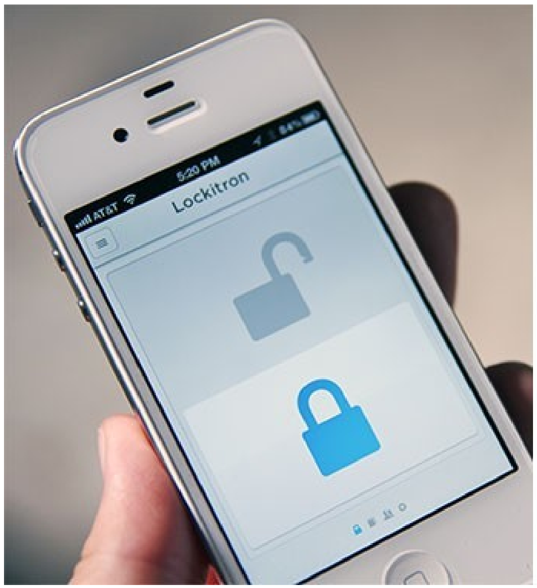
What is Bluetooth LE?

- New modulation mode for low-power devices
- Introduced in Bluetooth 4.0
- AKA Bluetooth Smart
- Almost completely different from classic Bluetooth
- Designed to operate for a long time off a coin cell

Where is LE used?

- Sports devices (heart monitor, pedal cadence)
- Sensors (e.g., thermometer)
- Wireless door locks
- Upcoming medical devices

epic foreshadowing



How does LE compare to Classic BT?

- Master/slave architecture
- Reuses high-level protocols

- Different modulation parameters
- Different channels (still in 2.4 GHz ISM)
- Different channel hopping scheme
- Different packet format
- Different whitening

sniffing
Bluetooth
is
hard

sniffing
Bluetooth LE
is slightly
less hard

How do we sniff it?

Start at the bottom and work our way up:

GATT
ATT
L2CAP
→ Link Layer
→ PHY

PHY Layer

- GFSK modulation
- 40 x 1 MHz channels spaced 2 MHz apart
- Handled entirely by CC2400

RF → bits

Link Layer

LSB MSB

Preamble (1 octet)	Access Address (4 octets)	PDU (2 to 39 octets)	CRC (3 octets)

Figure 2.1: Link Layer packet format

octets you say?

Link Layer

LSB			MSB
Preamble (1 octet)	Access Address (4 octets)	PDU (2 to 39 octets)	CRC (3 octets)

Figure 2.1: Link Layer packet format

What we have: Sea of bits

What we want: Start of PDU

What we know: AA

10001110111110101010 1
1001110000010001100 1
1110010011010001110 1

L2CAP and Beyond

```
06 0b 07 00 04 00 1b 11
00 16 58 b8 02 62 fb b2
```

- RTFM
- It's actually quite readable!

L2CAP and Beyond

GATT
ATT
L2CAP
Link Layer
PHY

Example Packet

06 0b	
07 00	L2CAP length: 7
04 00	channel 4: LE Attribute Protocol
1b	Handle Value Notification
11 00	Attribute Handle
16	flags
58	**heart rate: 88 bpm**
b8 02	**RR-interval: 696 ms**
62 fb b2	

So we can turn RF into packets

- Now what?
- BTLE doesn't sit on a single channel, it hops!

Let's follow connections!

How Connections Work

- Hop along data channels
- One data packet per timeslot

$3 \rightarrow 10 \rightarrow 17 \rightarrow 24 \rightarrow 31 \rightarrow 1 \rightarrow 8 \rightarrow 15 \rightarrow \ldots$

hop increment = 7

Following Connections

The four things you need to follow a connection are:

1. AA
2. CrcInit
3. Time slot length
4. Hop increment

How do I get these values?

Finding AA

- Sit on data channel waiting for empty data packets
- Collect candidate AA's and pick one when it's been observed enough

100011101111101010101
10011100000100011001
10000000000000001101
10100011000110000101

Not depicted: whitening!

Finding CRCInit

- Filter packets by AA
- Plug CRC into LFSR and run it backward

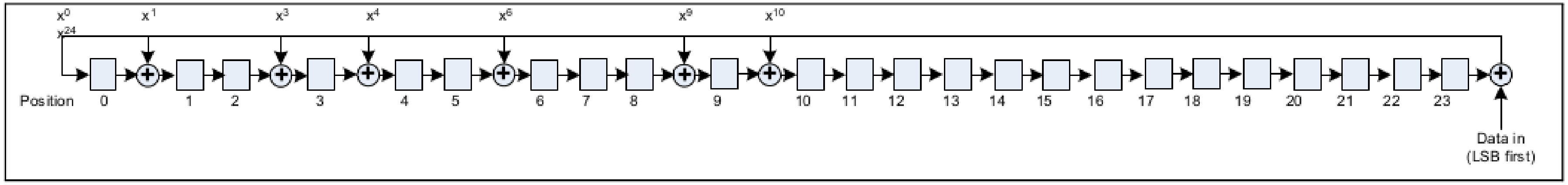


Figure 3.2: The LFSR circuit generating the CRC

See also "Bluesniff: Eve meets Alice and Bluetooth"

Finding time slot length

- Observation: 37 is prime
- Sit on data channel and wait for two consecutive packets

$$\frac{\Delta t}{37} = time\ slot\ length$$

Finding Hop Increment

- Start on data channel 0, jump to data channel 1 when a packet arrives
- We kn[illegible] hop in[illegible], so we [illegible] calcul[illegible] how m[illegible]ny [illegible] a[illegible] were h[illegible]op[illegible] betwe[illegible] 0 and 1
- I w[illegible]n't [illegible]e y[illegible] with [illegible]e matl[illegible]
- We use [illegible] LU[illegible] conve[illegible] that [illegible] hop a[illegible]unt

MATH

Promiscuous: Summary

The four things you need to follow a connection are:

- ✓ AA
- ✓ CrcInit
- ✓ Hop interval
- ✓ Hop amount

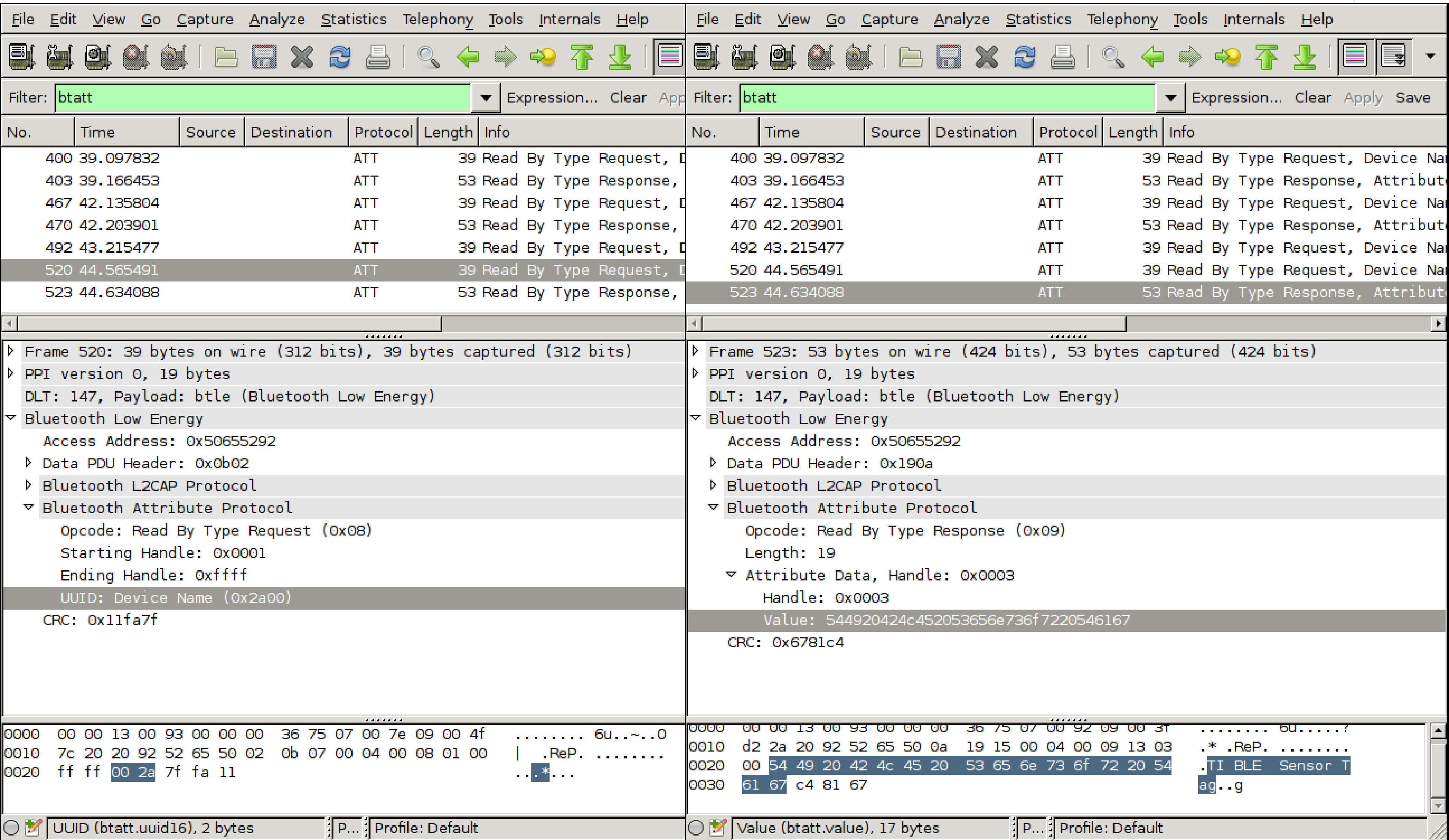
Current Status

- Sniff new connections
- Sniff already-established connections (promiscuous)
- Wireshark protocol dissector

- Grab the git!
- Available in Gentoo! (thanks Zero_Chaos)

- Everything implemented in-firmware

Wireshark!


File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: btatt
Expression... Clear Apply Save
No. Time Source Destination Protocol Length Info
400 39.097832 ATT 39 Read By Type Request, Device Na
403 39.166453 ATT 53 Read By Type Response, Attribut
467 42.135804 ATT 39 Read By Type Request, Device Na
470 42.203901 ATT 53 Read By Type Response, Attribut
492 43.215477 ATT 39 Read By Type Request, Device Na
520 44.565491 ATT 39 Read By Type Request, Device Na
523 44.634088 ATT 53 Read By Type Response, Attribut
Frame 520: 39 bytes on wire (312 bits), 39 bytes captured (312 bits)
PPI version 0, 19 bytes
DLT: 147, Payload: btle (Bluetooth Low Energy)
Bluetooth Low Energy
Access Address: 0x50655292
Data PDU Header: 0x0b02
Bluetooth L2CAP Protocol
Bluetooth Attribute Protocol
Opcode: Read By Type Request (0x08)
Starting Handle: 0x0001
Ending Handle: 0xffff
UUID: Device Name (0x2a00)
CRC: 0x11fa7f
0000 00 00 13 00 93 00 00 00 36 75 07 00 7e 09 00 4f .... 6u..~..O
0010 7c 20 20 92 52 65 50 02 0b 07 00 04 00 08 01 00 | .ReP. ....
0020 ff ff 00 2a 7f fa 11 ...*...
UUID (btatt.uuid16), 2 bytes
Profile: Default
Frame 523: 53 bytes on wire (424 bits), 53 bytes captured (424 bits)
PPI version 0, 19 bytes
DLT: 147, Payload: btle (Bluetooth Low Energy)
Bluetooth Low Energy
Access Address: 0x50655292
Data PDU Header: 0x190a
Bluetooth L2CAP Protocol
Bluetooth Attribute Protocol
Opcode: Read By Type Response (0x09)
Length: 19
Attribute Data, Handle: 0x0003
Handle: 0x0003
Value: 544920424c452053656e736f7220546167
CRC: 0x6781c4
0010 d2 2a 20 92 52 65 50 0a 19 15 00 04 00 09 13 03 .* .ReP. ....
0020 00 54 49 20 42 4c 45 20 53 65 6e 73 6f 72 20 54 .TI BLE Sensor T
0030 61 67 c4 81 67 ag..g
Value (btatt.value), 17 bytes
Profile: Default

Demo

→ demo

→ demo

→ demo

→ demo

→ demo

→ demo

→ demo

→ demo

→ demo

Security

- Good news: there is encryption
- Bad news: depending on your situation it's probably not very effective

Key Exchange

- Pairing mode determines temporary key (TK)
  - Just Works
  - 6 digit PIN
  - OOB

Not DH!

- Just works: no passive eavesdropper protection
- 6 digit PIN: easily brute forceable
- OOB provides the only meaningful security

Eavesdropping Scenario

- Alice pairs with her brand new LE device
- Eve observes pairing / key exchange
- Just Works or 6 digit PIN: Eve recovers TK
- With TK and pairing data: Eve recovers STK
- With STK and key exchange: Eve recovers LTK

LTK = Session Key = GAME OVER

Well, not quite..

- Each connection uses a different nonce, so Eve has to witness connection setup
- The LTK is exchanged once and reused for many connections

Active Attacks

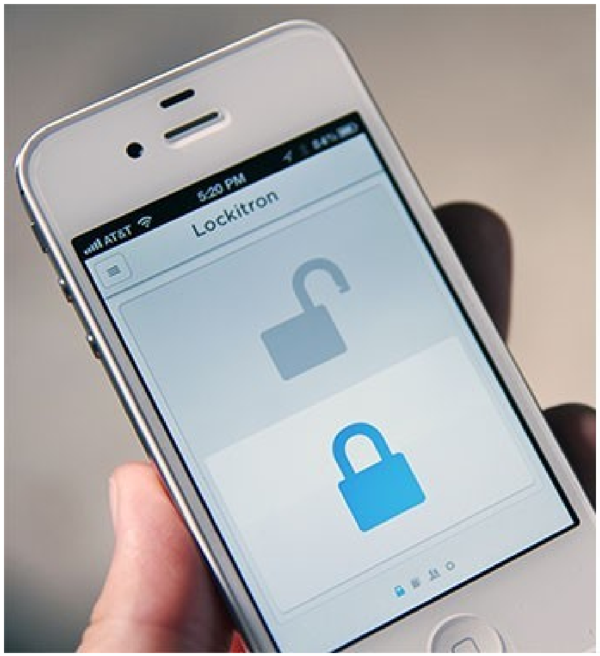
- How do you witness a connection setup?
  - Force a reconnect!
  - Should be as simple as jamming the connection
- What about connections that use a pre-shared LTK?
  - Inject message LL_REJECT_IND (reject LTK)

"My Bad"

"**None of the pairing methods provide protection against a passive eavesdropper** during the pairing process as predictable or easily established values for TK are used."

"A future version of this specification will include elliptic curve cryptography and Diffie-Hellman public key exchanges that will provide passive eavesdropper protection."

Why should I care about LE security?



Pacemaker hack can deliver deadly 830-volt jolt

Pacemakers and implantable cardioverter-defibrillators could be manipulated for an anonymous assassination

By Jeremy Kirk

October 17, 2012 12:40 AM ET

+ Briefcase More

IDG News Service - Pacemakers from several manufacturers can be commanded to deliver a deadly, 830-volt shock from someone on a laptop up to 50 feet away, the result of poor software programming by medical device companies.

The new research comes from Barnaby Jack of security vendor IOActive, known for his analysis of other medical equipment such as insulin-delivering devices.

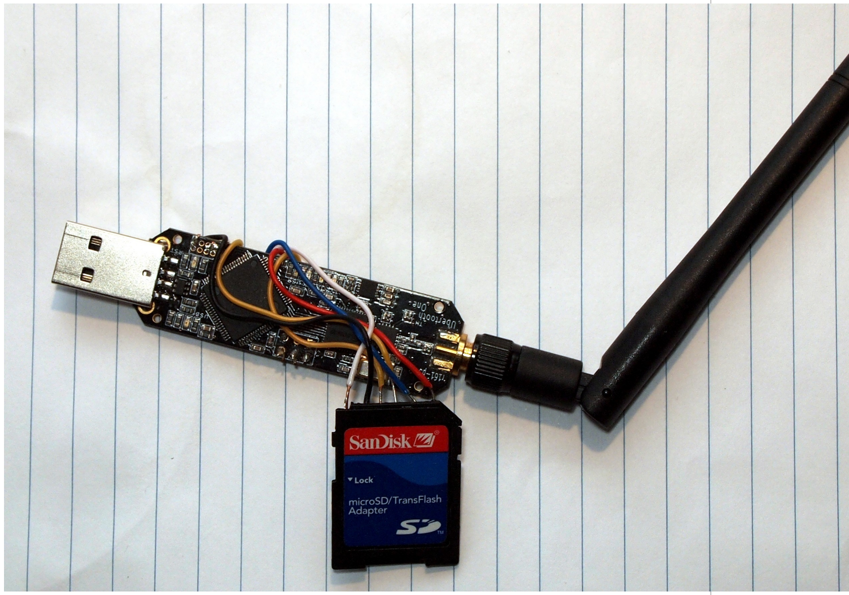
Jack, who spoke at the Breakpoint security conference in Melbourne on Wednesday, said the flaw lies with the programming of the wireless transmitters used to give instructions to pacemakers and implantable cardioverter-defibrillators (ICDs), which detect irregular heart contractions and deliver an electric shock to avert a heart attack.

Take-Away

- LE security compromised by design
- If security matters, use OOB pairing
- Alternatively: BYOE   see also: The end-to-end principle

Future Work

- Wireshark capture source
- Demonstrate encryption attacks
- Master/slave on dongle
- SD + battery
- Channel maps that don't use all 37 channels



Thanks

Mike Ossmann
Dominic Spill

Mike Kershaw (dragorn)
#ubertooth on freenode
bluez
Bluetooth SIG

Facebook / iSEC

Thank You

Mike Ryan

@mpeg4codec

mikeryan@isecpartners.com

http://lacklustre.net/

http://ubertooth.sf.net/

Related Work

- TI CC2540EMK-USB – $49
- BlueRadios BlueSniff™ – $249
  - “Only available to BlueRadios Clients who purchased our modules for use”
- Ellisys Bluetooth Explorer 400+LE – $N0,000

None support sniffing already-established connections!

Slave Device Lifecycle

- When connected
  - Hop along data channels
  - One data packet per timeslot
- When not connected
  - periodically announce existence on advertising channel
  - respond to requests from master

3 → 10 → 17 → 24 → 31 → 1 → 8 → 15 → ...

hop amount = 7