

# Bluetooth:

## With Low Energy Comes Low Security

Mike Ryan  
iSEC Partners

USENIX Security / WOOT  
Aug 13, 2013

# Outline

- What is Bluetooth Low Energy?
- Protocol overview
- Sniffing Techniques
- [In]security
- Injection

# Outline

- What is Bluetooth Low Energy?
- Protocol overview
- Sniffing Techniques
- [In]security
- Injection

# What is Bluetooth Low Energy?

# What is Bluetooth ~~Low Energy~~ Smart?

- New modulation and link layer for low-power devices
- vs classic Bluetooth
  - Incompatible with classic Bluetooth devices
  - PHY and link layer almost completely different
  - High-level protocols reused (L2CAP, ATT)
- Introduced in Bluetooth 4.0 (2010)
- AKA BTLE

# Where is BTLE?

- High end smartphones
- Sports / fitness devices
- Door locks
- Upcoming medical devices

Product Name	Manufacturer	Product Type
Apple Watch	Apple	Smartwatch
Fitbit	Fitbit	Activity Tracker
Garmin	Garmin	Activity Tracker
Fitbit Charge	Fitbit	Activity Tracker
Fitbit Flex	Fitbit	Activity Tracker
Fitbit Surge	Fitbit	Activity Tracker
Fitbit Charge 2	Fitbit	Activity Tracker
Fitbit Charge 3	Fitbit	Activity Tracker
Fitbit Charge 4	Fitbit	Activity Tracker
Fitbit Charge 5	Fitbit	Activity Tracker
Fitbit Charge 6	Fitbit	Activity Tracker
Fitbit Charge 7	Fitbit	Activity Tracker
Fitbit Charge 8	Fitbit	Activity Tracker
Fitbit Charge 9	Fitbit	Activity Tracker
Fitbit Charge 10	Fitbit	Activity Tracker
Fitbit Charge 11	Fitbit	Activity Tracker
Fitbit Charge 12	Fitbit	Activity Tracker
Fitbit Charge 13	Fitbit	Activity Tracker
Fitbit Charge 14	Fitbit	Activity Tracker
Fitbit Charge 15	Fitbit	Activity Tracker
Fitbit Charge 16	Fitbit	Activity Tracker
Fitbit Charge 17	Fitbit	Activity Tracker
Fitbit Charge 18	Fitbit	Activity Tracker
Fitbit Charge 19	Fitbit	Activity Tracker
Fitbit Charge 20	Fitbit	Activity Tracker
Fitbit Charge 21	Fitbit	Activity Tracker
Fitbit Charge 22	Fitbit	Activity Tracker
Fitbit Charge 23	Fitbit	Activity Tracker
Fitbit Charge 24	Fitbit	Activity Tracker
Fitbit Charge 25	Fitbit	Activity Tracker
Fitbit Charge 26	Fitbit	Activity Tracker
Fitbit Charge 27	Fitbit	Activity Tracker
Fitbit Charge 28	Fitbit	Activity Tracker
Fitbit Charge 29	Fitbit	Activity Tracker
Fitbit Charge 30	Fitbit	Activity Tracker
Fitbit Charge 31	Fitbit	Activity Tracker
Fitbit Charge 32	Fitbit	Activity Tracker
Fitbit Charge 33	Fitbit	Activity Tracker
Fitbit Charge 34	Fitbit	Activity Tracker
Fitbit Charge 35	Fitbit	Activity Tracker
Fitbit Charge 36	Fitbit	Activity Tracker
Fitbit Charge 37	Fitbit	Activity Tracker
Fitbit Charge 38	Fitbit	Activity Tracker
Fitbit Charge 39	Fitbit	Activity Tracker
Fitbit Charge 40	Fitbit	Activity Tracker
Fitbit Charge 41	Fitbit	Activity Tracker
Fitbit Charge 42	Fitbit	Activity Tracker
Fitbit Charge 43	Fitbit	Activity Tracker
Fitbit Charge 44	Fitbit	Activity Tracker
Fitbit Charge 45	Fitbit	Activity Tracker
Fitbit Charge 46	Fitbit	Activity Tracker
Fitbit Charge 47	Fitbit	Activity Tracker
Fitbit Charge 48	Fitbit	Activity Tracker
Fitbit Charge 49	Fitbit	Activity Tracker
Fitbit Charge 50	Fitbit	Activity Tracker
Fitbit Charge 51	Fitbit	Activity Tracker
Fitbit Charge 52	Fitbit	Activity Tracker
Fitbit Charge 53	Fitbit	Activity Tracker
Fitbit Charge 54	Fitbit	Activity Tracker
Fitbit Charge 55	Fitbit	Activity Tracker
Fitbit Charge 56	Fitbit	Activity Tracker
Fitbit Charge 57	Fitbit	Activity Tracker
Fitbit Charge 58	Fitbit	Activity Tracker
Fitbit Charge 59	Fitbit	Activity Tracker
Fitbit Charge 60	Fitbit	Activity Tracker
Fitbit Charge 61	Fitbit	Activity Tracker
Fitbit Charge 62	Fitbit	Activity Tracker
Fitbit Charge 63	Fitbit	Activity Tracker
Fitbit Charge 64	Fitbit	Activity Tracker
Fitbit Charge 65	Fitbit	Activity Tracker
Fitbit Charge 66	Fitbit	Activity Tracker
Fitbit Charge 67	Fitbit	Activity Tracker
Fitbit Charge 68	Fitbit	Activity Tracker
Fitbit Charge 69	Fitbit	Activity Tracker
Fitbit Charge 70	Fitbit	Activity Tracker
Fitbit Charge 71	Fitbit	Activity Tracker
Fitbit Charge 72	Fitbit	Activity Tracker
Fitbit Charge 73	Fitbit	Activity Tracker
Fitbit Charge 74	Fitbit	Activity Tracker
Fitbit Charge 75	Fitbit	Activity Tracker
Fitbit Charge 76	Fitbit	Activity Tracker
Fitbit Charge 77	Fitbit	Activity Tracker
Fitbit Charge 78	Fitbit	Activity Tracker
Fitbit Charge 79	Fitbit	Activity Tracker
Fitbit Charge 80	Fitbit	Activity Tracker
Fitbit Charge 81	Fitbit	Activity Tracker
Fitbit Charge 82	Fitbit	Activity Tracker
Fitbit Charge 83	Fitbit	Activity Tracker
Fitbit Charge 84	Fitbit	Activity Tracker
Fitbit Charge 85	Fitbit	Activity Tracker
Fitbit Charge 86	Fitbit	Activity Tracker
Fitbit Charge 87	Fitbit	Activity Tracker
Fitbit Charge 88	Fitbit	Activity Tracker
Fitbit Charge 89	Fitbit	Activity Tracker
Fitbit Charge 90	Fitbit	Activity Tracker
Fitbit Charge 91	Fitbit	Activity Tracker
Fitbit Charge 92	Fitbit	Activity Tracker
Fitbit Charge 93	Fitbit	Activity Tracker
Fitbit Charge 94	Fitbit	Activity Tracker
Fitbit Charge 95	Fitbit	Activity Tracker
Fitbit Charge 96	Fitbit	Activity Tracker
Fitbit Charge 97	Fitbit	Activity Tracker
Fitbit Charge 98	Fitbit	Activity Tracker
Fitbit Charge 99	Fitbit	Activity Tracker
Fitbit Charge 100	Fitbit	Activity Tracker

Blood glucose monitor

# By The Numbers

- 186% YoY Growth for H1 2013<sup>1</sup>
- “over 7 million Bluetooth Smart ICs were estimated to have shipped for use in sports and fitness devices in the first half of 2013 alone”
- “Analysts Forecast Bluetooth Smart to Lead Market Share in Wireless **Medical** and Fitness Devices”<sup>2</sup>

<sup>1</sup><http://www.bluetooth.com/Pages/Press-Releases-Detail.aspx?ItemID=170>

<sup>2</sup><http://www.bluetooth.com/Pages/Press-Releases-Detail.aspx?ItemID=165>

# Outline

- What is Bluetooth Low Energy?
- Protocol overview
- Sniffing Techniques
- [In]security
- Injection



# Protocol Stack



# PHY Layer

- GFSK, +/- 250 kHz, 1 Mbit/sec
- 40 channels in 2.4 GHz
- Hopping

# Physical Channels

→ Advertising: 3 channels

→ Data: 37 channels

RF Channel	RF Center Frequency	Channel Type	Data Channel Index	Advertising Channel Index
0	2402 MHz	Advertising channel		37
1	2404 MHz	Data channel	0	
2	2406 MHz	Data channel	1	
...	...	Data channels	...	
11	2424 MHz	Data channel	10	
12	2426 MHz	Advertising channel		38
13	2428 MHz	Data channel	11	
14	2430 MHz	Data channel	12	
...	...	Data channels	...	
38	2478 MHz	Data channel	36	
39	2480 MHz	Advertising channel		39

# Hopping

- Hop along 37 data channels
- One data packet per channel
- Next channel  $\equiv$  channel + hop increment (mod 37)
- Time between hops: hop interval

3 → 10 → 17 → 24 → 31 → 1 → 8 → 15 → ...  
hop increment = 7

# Link Layer



Figure 2.1: Link Layer packet format

- PDU min of 2 bytes due to 2 byte header
- LLID: Control vs Data
- Length

# L2CAP and Beyond

- Use existing decoders for this
- Not a Hard Problem™

# Recap



# Outline

- What is Bluetooth Low Energy?
- Protocol overview
- Sniffing Techniques
- [In]security
- Injection



sniffing  
Bluetooth  
is  
hard



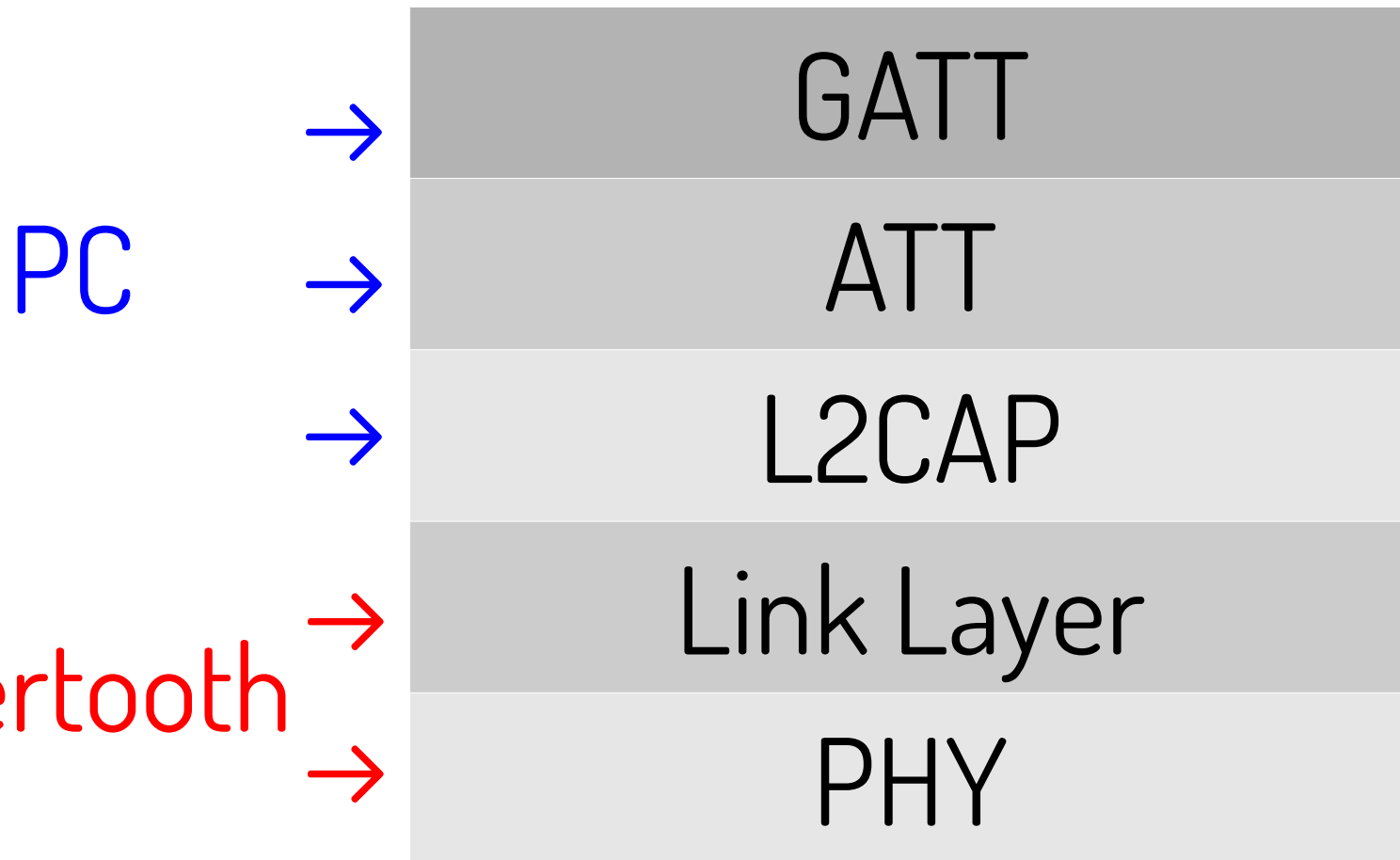
sniffing

Bluetooth LE

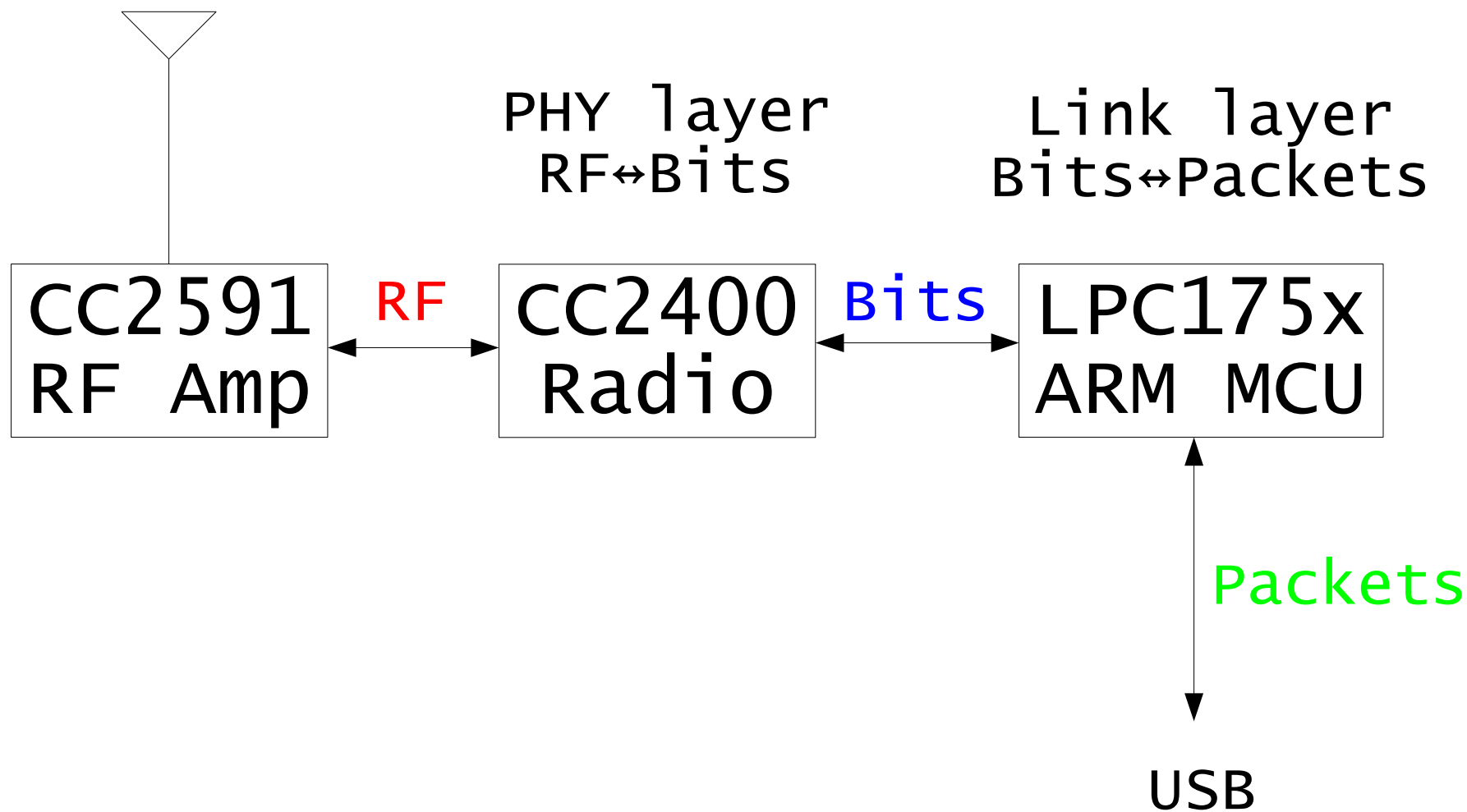
is slightly  
less hard

# How do we sniff it?

Start at the bottom and work our way up:



# Ubertooth Block Diagram



# Capturing: PHY Layer

- Configure CC2400
  - Set modulation parameters to match Bluetooth Smart
  - Tune to proper channel
- Follow connections according to hop pattern
  - Hop increment and hop interval, sniffed from connect packet or recovered in promiscuous mode
- Hand off bits to ARM MCU



# What Info Do We Need?

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

## → Access Address

→ Advertising: Fixed **0x8E89BED6**

→ Connection: Varies

## → Channel number

→ Hop interval

→ Hop increment

→ Nice to have: CRCInit

Where?

Easy mode:  
Connect packet!

22

# Link Layer



Figure 2.1: Link Layer packet format

What we know: Access Address

What we have: Sea of bits

What we want: Start of PDU

CC2400 does this  
**FO FREE**

10001110111101010101  
10011100000100011001  
11100100110100011101

# PHY Layer.. Link Layer..

Field name	Relation	Value (Protocol)
<ul style="list-style-type: none"><li>▶ BT ATT - Bluetooth Attribute Protocol</li><li>▶ BT AVCTP - Bluetooth AVCTP Protocol</li><li>▶ BT AVDTP - Bluetooth AVDTP Protocol</li><li>▶ BT AVRCP - Bluetooth AVRCP Profile</li><li>▶ BT BNEP - Bluetooth BNEP Protocol</li><li>▶ BT DUN - Bluetooth DUN Packet</li><li>▶ BT GNSS - Bluetooth GNSS Profile</li><li>▶ BT HCRP - Bluetooth HCRP Profile</li><li>▶ BT HFP - Bluetooth Handsfree Profile</li><li>▶ BT HID - Bluetooth HID Profile</li><li>▶ BT L2CAP - Bluetooth L2CAP Protocol</li></ul>	<ul style="list-style-type: none"><li>is present</li><li>==</li><li>!=</li><li>&gt;</li><li>&lt;</li><li>&gt;=</li><li>&lt;=</li><li>contains</li><li>matches</li></ul>	<input type="text"/> Predefined values: <input type="text"/> Range (offset:length) <input type="text"/>



# Capturing Packets... To PCAP!

- ubertooth-btle speaks packets
- libpcap → dump raw packet data
- PPI header (similar airodump-ng and kismet)
  
- We have a DLT for Bluetooth Smart
  - Unique identifier for the protocol
  - Public release of Wireshark plugin Coming Soon™

# Wireshark Awesomeness

The image displays two side-by-side screenshots of the Wireshark network protocol analyzer interface, showing packet captures and their details.

**Left Screenshot (Packet 520):**

- Filter:** btatt
- Packet List:**

No.	Time	Source	Destination	Protocol	Length	Info
400	39.097832			ATT	39	Read By Type Request, Device Name
403	39.166453			ATT	53	Read By Type Response, Attribute
467	42.135804			ATT	39	Read By Type Request, Device Name
470	42.203901			ATT	53	Read By Type Response, Attribute
492	43.215477			ATT	39	Read By Type Request, Device Name
520	44.565491			ATT	39	Read By Type Request, Device Name
523	44.634088			ATT	53	Read By Type Response, Attribute
- Packet Details:**
  - 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
- Packet Bytes:**

```

0000 00 00 13 00 93 00 00 00 36 75 07 00 7e 09 00 4f  .... 6u...~..0
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  ..*...

```
- Bottom Panel:** UUID (btatt.uuid16), 2 bytes | Profile: Default

**Right Screenshot (Packet 523):**

- Filter:** btatt
- Packet List:**

No.	Time	Source	Destination	Protocol	Length	Info
400	39.097832			ATT	39	Read By Type Request, Device Name
403	39.166453			ATT	53	Read By Type Response, Attribute
467	42.135804			ATT	39	Read By Type Request, Device Name
470	42.203901			ATT	53	Read By Type Response, Attribute
492	43.215477			ATT	39	Read By Type Request, Device Name
520	44.565491			ATT	39	Read By Type Request, Device Name
523	44.634088			ATT	53	Read By Type Response, Attribute
- Packet Details:**
  - 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: 544920424c4520536556e736f7220546167
      - CRC: 0x6781c4
- Packet Bytes:**

```

0000 00 00 13 00 93 00 00 00 36 75 07 00 92 09 00 3f  .... 6u.....?
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

```
- Bottom Panel:** Value (btatt.value), 17 bytes | Profile: Default

# Promiscuous Mode

- Techniques for recovering
  - Access Address
  - CRCInit
  - Hop Interval
  - Hop Increment

# Recovering Access Address

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

10001110111101010101  
10011100000100011001  
1000000000000000001101  
10100011000110000101

Not depicted: whitening!

# Recovering CRCInit

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

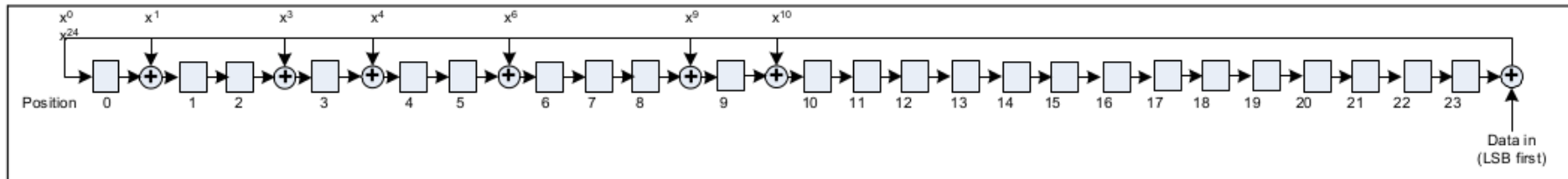


Figure 3.2: The LFSR circuit generating the CRC

See also “Bluesniff: Eve meets Alice and Bluetooth”, USENIX WOOT '07

# Recovering Hop Interval

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

$$\frac{\Delta t}{37} = \textit{hop interval}$$

# Recovering Hop Increment

- Start on data channel 0, jump to data channel 1 when a packet arrives
- We know hop interval, so we can calculate how many channels were hopped between 0 and 1

# Recovering Hop Increment (math)

$$0 + \text{hopIncrement} \times \text{channelsHopped} \equiv 1 \pmod{37}$$

$$\text{hopIncrement} \equiv \text{channelsHopped}^{-1} \pmod{37}$$

$$\text{channelsHopped}^{-1} \equiv \text{channelsHopped}^{37-2} \pmod{37}$$

We use a LUT to convert that to hop increment



# Sniffing Summary

- Connection following
- Promiscuous: Recovering the four values
  - Access address
  - CRCInit
  - Hop interval
  - Hop Increment

# Outline

- What is Bluetooth Low Energy?
- Protocol overview
- Sniffing Techniques
- [In]security
- Injection

# Encryption

- Provided by link layer
- Encrypts and MACs PDU
- AES-CCM

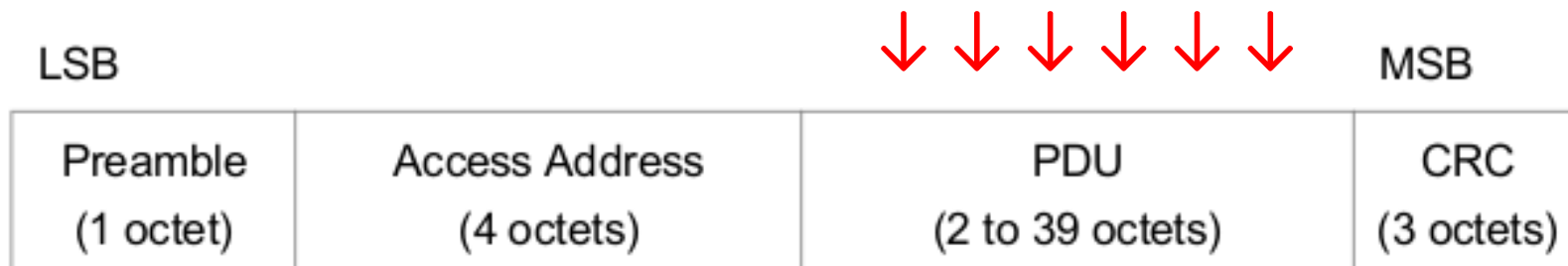


Figure 2.1: Link Layer packet format

# Custom Key Exchange Protocol

- Three stage process
- 3 pairing methods
  - Just Works™
  - 6-digit PIN
  - OOB
- “None of the pairing methods provide protection against a passive eavesdropper” -Bluetooth Core Spec

# Cracking the TK

confirm  
=  
AES(TK, AES(TK, rand XOR p1) XOR p2)

GREEN = we have it  
RED = we want it

TK: integer between 0 and 999,999  
Just Works™: always 0!

# Cracking the TK – With *crackle*

Total time to crack:  
< 1 second

# And That's It

- TK → STK
- STK → LTK
- LTK → Session keys

KEY EXCHANGE = BROKEN  
100% PASSIVE

# LTK Reuse

- Good for security: pair in a faraday cage
- Counter-mitigation: Active attack to force re-pairing



# Decrypting

- Assumption: Attacker has LTK – reused!
- Procedure
  - Attacker passively capturing packets
  - Connection established
  - Session information captured

# Decrypting – With *crackle*

- Yes, crackle does that too!
- crackle will decrypt
  - a PCAP file with a pairing setup
  - a PCAP file with an encrypted session, given an LTK

# Am I Affected?

- Probably
- Exception: Some vendors implement their own security on top of GATT
  - Did they talk to a cryptographer?

# Security Recap

- Key exchange broken
- LTK reuse means all communication is effectively compromised
- 99% passive
  - Worst case scenario: one active attack with off-the-shelf hardware

# Outline

- What is Bluetooth Low Energy?
- Protocol overview
- Sniffing Techniques
- [In]security
- Injection

# Injection

- Pretty much the same as receiving, opposite direction
- Follow the spec!
  - Link layer header
  - Payload data
- Hand that off to Ubertooth
  - Calculate CRC
  - Whiten
- Devil is in the CC2400 details

# Demo

→ D

→ e

• m

- o



# Capabilities

- Ubertooth
  - Passively intercept Bluetooth Smart
  - Promiscuous mode
  - Injection
- Wireshark plugins
- crackle
  - Crack TK's sniffed with Ubertooth
  - Decrypt PCAP files with LTK



# Software

- Ubertooth and libbtbb
  - <http://ubertooth.sourceforge.net/>
- crackle
  - <http://lacklustre.net/projects/crackle/>
- nano-ecc (8-bit ECDH and ECDSA)
  - <https://github.com/iSECPartners/nano-ecc>

# Thanks

Mike Ossmann  
Dominic Spill

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

# Thank You

Mike Ryan

iSEC Partners

@mpeg4codec

mikeryan@isecpartners.com

<http://lacklustre.net/>

# Apocrypha (extra)

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: `btle.ll_control_opcode` Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
1365	0.000			Bluetooth LE	41	LL Control PDU: LL_CHANNEL_MAP_REQ

.....

- ▶ Frame 1365: 41 bytes on wire (328 bits), 41 bytes captured (328 bits) on interface 1
- ▶ PPI version 0, 24 bytes
  - DLT: 147, Payload: btle (Bluetooth Low Energy)
- ▼ Bluetooth Low Energy
  - Access Address: 0xaf9aa518
  - ▶ Data PDU Header: 0x080f
    - LL Control Opcode: LL\_CHANNEL\_MAP\_REQ (0x01)
    - LL Control Data: 00fcfffe1f0101
    - CRC: 0x9bb2c0

0x1FFEFFFC00: remove channels 12, 27-36

# Encryption Mitigation (extra)

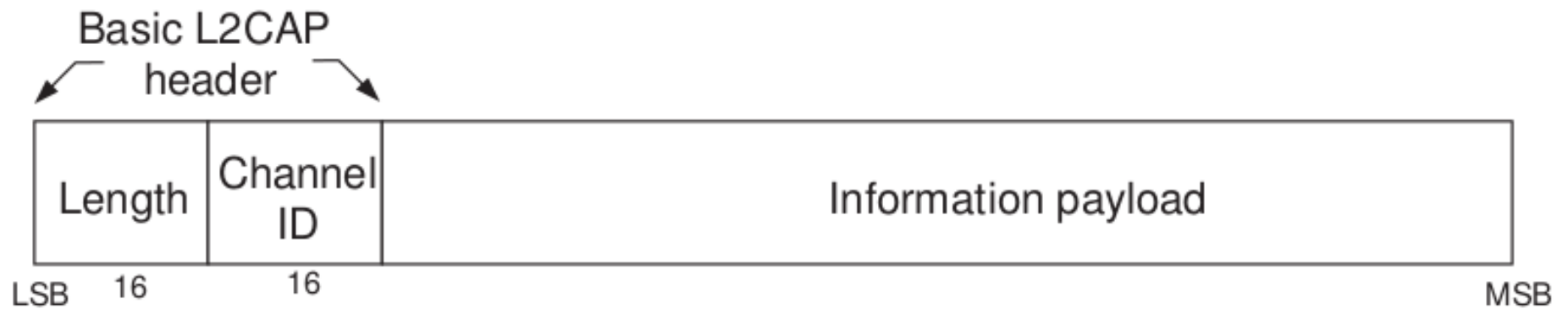
- Every session uses a different session key
- Every session uses several nonces

Solution: jam the connection to restart a session

- LTK exchanged once, used many times

Solution: inject LTK\_REJECT\_IND message

# L2CAP (extra)



# ATT/GATT (extra)

- Services: groups of characteristics
- Characteristics
  - Operations
- Everything identified by UUID
  - 128 bit
  - Sometimes shortened to 16 bits

# Example GATT Service: Heart Rate (extra)

- Service: **0x180D**
- Characteristic 1: **0x2A37** – Heart Rate
  - Can't read or write
  - Notify: subscribe to updates
- Characteristic 2: **0x2A38** – Sensor Location
  - Readable: 8 bit int, standardized list
- Other characteristics: **0x2803, 0x2902, ...**