CIS 700/002 : Special Topics : Bluetooth: With Low Energy comes Low Security

Kamenee Arumugam
CIS 700/002: Security of EMBS/CPS/IoT
Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania

3rd Feb 2017

Overview

- Walk thru : With Low Energy comes Low Security Paper
- Ubertooth:
  - Overview Ubertooth Platform
  - Core Functionlity
  - Demo

Introduction

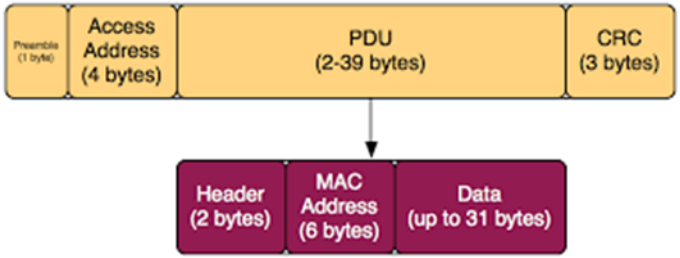
- Goal:
  - Discussion of tools : Ubertooth
  - Techniques to monitor and inject packets in BTLE

Overview Bluetooth LE

- Bluetooth Core 4.0 specification
- Wireless protocol operating 2.4GHz band
- PHY layer BTLE uses Gaussian Frequency Shift Keying (GFSK) with a 250 kHz offset
- 40 channels (37 data channels & 3 advertising channels)
- Simplified protocol – because of need to have low computation capabilities

Bluetooth LE Packet

- A packet can be 80 to 376 bits in length. Preamble: used for internal protocol management. Advertising packets have 10101010b as the preamble.
- Access Address: This is always 0x8E89BED6 for advertising packets.
- PDU: There are two PDU formats, one for advertising packets and one for data packets.
- CRC: 3 byte value calculated over PDU.



Capability of BLE sniffer

- Major contribution:
  - ability to derive the parameters needed to follow a connection that has previously been established

Eavesdropping

- To sniff a connection we need to know four values unique to that connection:
  - Hop interval (also referred to as dwell time)
  - Hop increment
  - Access address
  - CRC init

Following connections

- How ???
  - sniffer hops along the same sequence of channels at the same rate as the master and slave.
  - **Hop sequence**:nextChannel ≡ channel +hopIncrement (mod 37)
  - **Hop Interval** : master and slave will then wait for a period time before hopping to the next channel

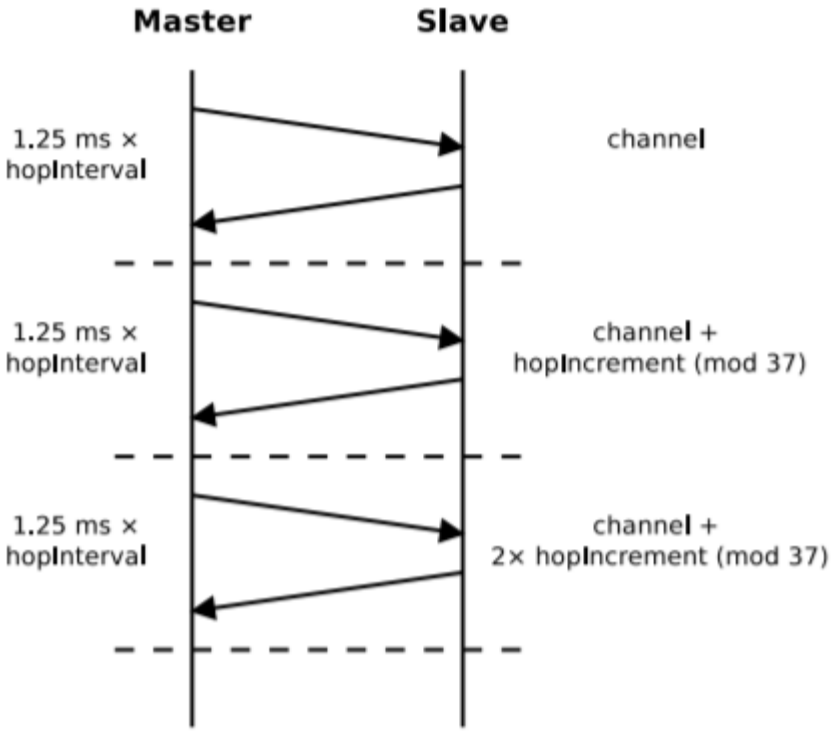


Figure 2: Master and slave each transmit on every channel, even if they have no meaningful data to exchange

Promiscuous Mode

- connection following mode- extracted from the connection initialization packet.
- promiscuous mode - recover them by exploiting properties of BTLE packets

Determining Access Address

- How ???
- monitoring an arbitrary data channel looking for empty data packets
- Empty data packets: consists of a 16 bit header and 24 bit CRC
- Identify 16 bit header, treat prior 32 bits as AA

Recovering CRCInit

- Next, filter CRCInit
- How ??
- run the bits through the LFSR in the reverse order.
- The value left in the LFSR at the end of this exercise is our candidate CRCInit

Hop Interval & Hop Increment

- Hop Interval :
  - hopInterval = Δt / 37×1.25 ms

Total time of complete cycle of hop sequence

- Hop Increment :
  - channelsHopped = Δt/ 1.25 ms×hopInterval

interarrival time of packets on two data channels (index 0 and 1).

Injection

- send undirected advertising messages broadcasting the existence of a device with a user-specified MAC address
- theory of operation:
  - craft an undirected advertising packet
  - whiten the data and send it to the CC2400 to be transmitted

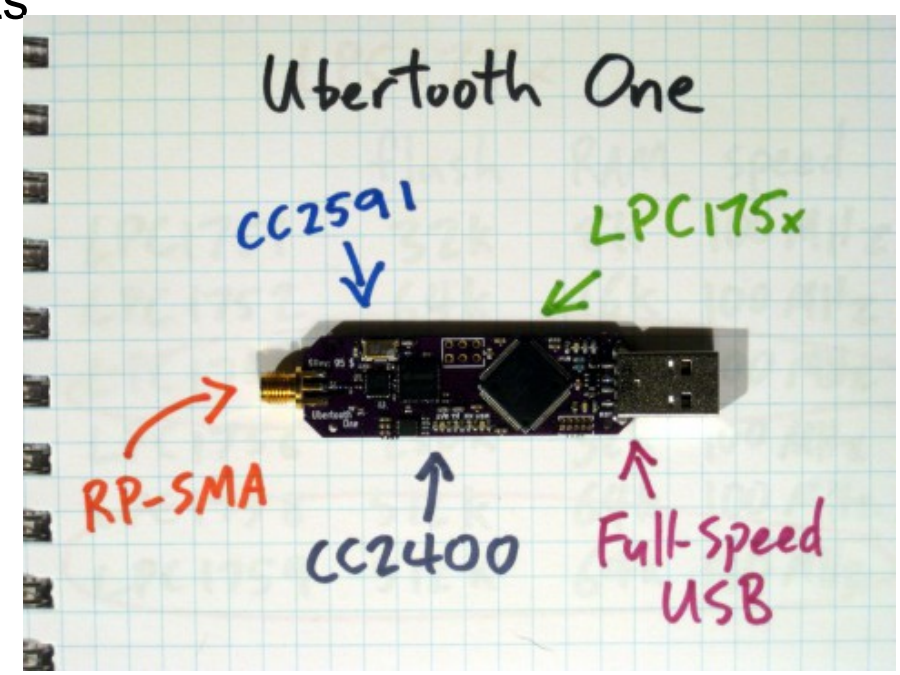
Bypassing The Encryption

- BTLE uses AES – CCM
- Our attack targets the key exchange rather than the encryption itself
- establish a shared secret known as a longterm key (LTK) thru through a **key exchange protocol**
- simplistic brute force algorithm to guess TK (128 bit AES key): calculate the confirm for every possible TK value between 0 and 999,999

Ubertooth Platform

- Designed by Michael Ossmann
- 2.4GHz experimentation platform
- Bluetooth 1.x, Low energy, 802.11 FHSS
- Hardware: -
  - RP-SMA RF connector: connects to test equipment, antenna, or dummy load.
  - CC2591 RF front end.
  - CC2400 wireless transceiver.
  - LPC175x ARM Cortex-M3 microcontroller with Full-Speed USB 2.0

standard Cortex Debug Connector (10- pin 50-mil JTAG) and ISP serial connector.



Ubertooth Functionality

- Bluetooth development platforms
- operate in monitor mode, monitoring Bluetooth traffic in real-time
- Ubertooth Utilities: -

Components	Functionality
Ubertooth-scan	Active (bluez) device scan and inquiry supported by Ubertooth. Perform equivalent of “hci scan”.
Ubertooth-specan-ui	This shows a GUI window with a spectrum analyzer for the 2.4 GHz band. It is very useful to see at what frequencies there are signals.
Ubertooth-follow	CLK discovery and follow for a particular UAP/LAP
Ubertooth-btle	passive Bluetooth Low Energy monitoring
Ubertooth-rx	Passive Bluetooth discovery/decode

Ubertooth: 3rd party software

- In order to sniff Bluetooth LE, we need to use "ubertooth-btle" utility and couple of 3rd party software:
  - Crackle : cracks Bluetooth Smart (BLE) encryption. It exploits a flaw in the pairing mechanism that leaves all communications vulnerable to decryption by passive eavesdroppers.
  - Kismet : Kismet is a wireless network detector, sniffer, and intrusion detection system. Capability to sniff Bluetooth can be expanded using ubertooth plugin.
  - Wireshark : Ubertooth provide Wireshark BTBB and BR/EDR plugins allow Bluetooth baseband traffic that has been captured using Kismet to be analyzed and dissected within the Wireshark GUI.

Ubertooth Demo

- Ubertooth-scan
- Ubertooth-specan-ui
- Ubertooth-follow
- Ubertooth-btle