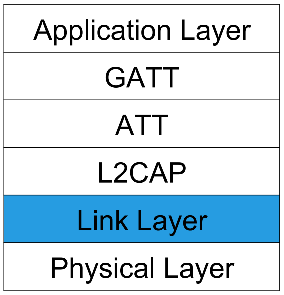
BLE Security

EECS 582 -- Spring 2015

Overview

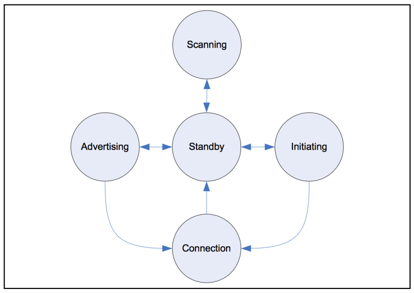
BLE Refresher
Attacks
Improvements
Authentication
Privacy
Discussion

BLE: Quick/Simplified Refresh



Application Layer
GATT
ATT
L2CAP
Link Layer
Physical Layer

Link Layer State Machine



Link Layer Connections - Steps

1. Initiate Connection
2. Exchange keys <- Attack!
3. Authenticate
4. Send encrypted messages

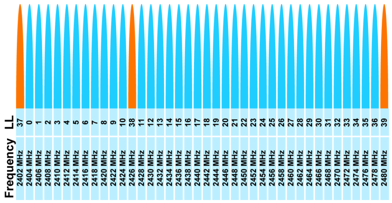
BLE CONNECT_REQ Packet

Payload		
InitA (6 octets)	AdvA (6 octets)	LLData (22 octets)

LLData									
AA (4 octets)	CRCInit (3 octets)	WinSize (1 octet)	WinOffset (2 octets)	Interval (2 octets)	Latency (2 octets)	Timeout (2 octets)	ChM (5 octets)	Hop (5 bits)	SCA (3 bits)

Initiating a BLE Connection

- Peripheral advertises
- Initiator starts connection
  - *hopInterval*
  - *hopIncrement*
  - *accessAddress*
  - *crcInit*
- Initiator and peripheral move to next channel



LL	37	0	1	2	3	4	5	6	7	8	9	10	38	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	39
Frequency	2402 MHz	2404 MHz	2406 MHz	2408 MHz	2410 MHz	2412 MHz	2414 MHz	2416 MHz	2418 MHz	2420 MHz	2422 MHz	2424 MHz	2426 MHz	2428 MHz	2430 MHz	2432 MHz	2434 MHz	2436 MHz	2438 MHz	2440 MHz	2442 MHz	2444 MHz	2446 MHz	2448 MHz	2450 MHz	2452 MHz	2454 MHz	2456 MHz	2458 MHz	2460 MHz	2462 MHz	2464 MHz	2466 MHz	2468 MHz	2470 MHz	2472 MHz	2474 MHz	2476 MHz	2478 MHz	2480 MHz

Sniffing an on going connection

- Eliminate false positives (how do you know what is a packet)
  - Look for 16-bit header for empty packet, take prior 32-bits as AA
  - *crcInit* can be reversed, by running the packet through the LFSR in reverse (magic, magic, math, math...)
  - Access Address is set in each packet.
- Wait on a channel and observe subsequent packets, record time between

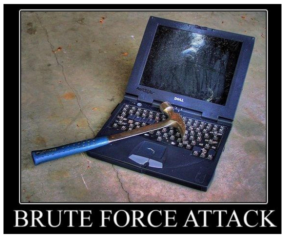
$$hopInterval = \frac{\Delta t}{37 \times 1.25\ \mathrm{ms}}$$

- Wait for a packet on two separate data channels

$$channelsHopped = \frac{\Delta t}{1.25\ \mathrm{ms} \times hopInterval} \qquad hopIncrement \equiv channelsHopped^{-1} \pmod{37}$$

Encryption - BLE 4.0 & 4.1

- Custom key exchange
  - Select TK (128 bit AES key)
  - Use TK to agree upon LTK
- What's TK?
  - Just Works^{TM}: key == 0
  - 6-digit passkey: key in 0-999,999
  - Out of Band: You're on your own.

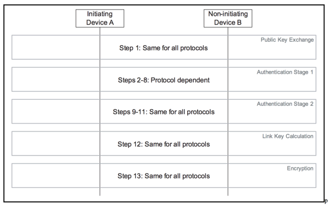


BLE 4.2 - Secure Simple Pairing

- Elliptic Curve Diffie Hellman
  - 96 bits of entropy with P-192 or 128 bits with P-256
- Protects against passive eavesdropping
- Does not protect against MITM

- Association models (anti-MITM)
  - Numeric comparison
  - Out of Band
  - Passkey

- Secure Connections Only Mode



Initiating Device A

Non-initiating Device B

Public Key Exchange

Step 1: Same for all protocols

Authentication Stage 1

Steps 2-8: Protocol dependent

Authentication Stage 2

Steps 9-11: Same for all protocols

Link Key Calculation

Step 12: Same for all protocols

Encryption

Step 13: Same for all protocols

Link Layer Encryption

- TCP/IP
  - No encryption
  - No authentication
  - Relies on application layer
  - Vulnerable to passive listener

- BLE
  - Node-to-node encryption
  - Impractical authentication (for many IoT)
  - Simply Secure is safe from passive listener

Could I be tracked?

- Device Address Randomization
  - Access Address generated by identity key (IRK)
  - IRK exchanged during bonding

- Do people use it?
  - "We do not currently employ Bluetooth Smart in this capability."
  - "...we do not use randomize device address."
  - "As far as we are aware, our two products that use BLE do not utilize this feature."

Summary

- Proven link-layer encryption scheme node to node (in 4.2)
- No protection against MITM without traditional I/O
- Option for randomizing device address

Wishlist

- Better way to do authentication
  - Many IoT class devices don’t have classical I/O
  - How to I control what devices are connected to my gateway?
  - How can I control what gateways I connect to?
- Multihop communication
  - Do I trust the nodes in between the gateway and destination?
  - What happens if one of my devices is compromised?
- Do I trust my gateway?

References

https://lacklustre.net/bluetooth/
Ryan_Bluetooth_Low_Energy_USENIX_WOOT.pdf
https://eprint.iacr.org/2013/309.pdf
https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx?
doc_id=286439

What does IoT need?

- Confidentiality
  - I don’t want people monitoring my habits at home
    - ...but people can already see if my lights are on…
  - Communication between nodes should be kept secret
- Authentication
  - We want to know what nodes are on our network and that they’re legit.
- Preventing pivots
  - If a node is compromised, it should be hard for that node to pop other devices.
- Do I want people to know what devices I have in my house?
- Prevent neighbors from turning off lights
- General framework that different classes of devices can “inherit” from: medical IoT can specify something that fitness IoT needn’t have.