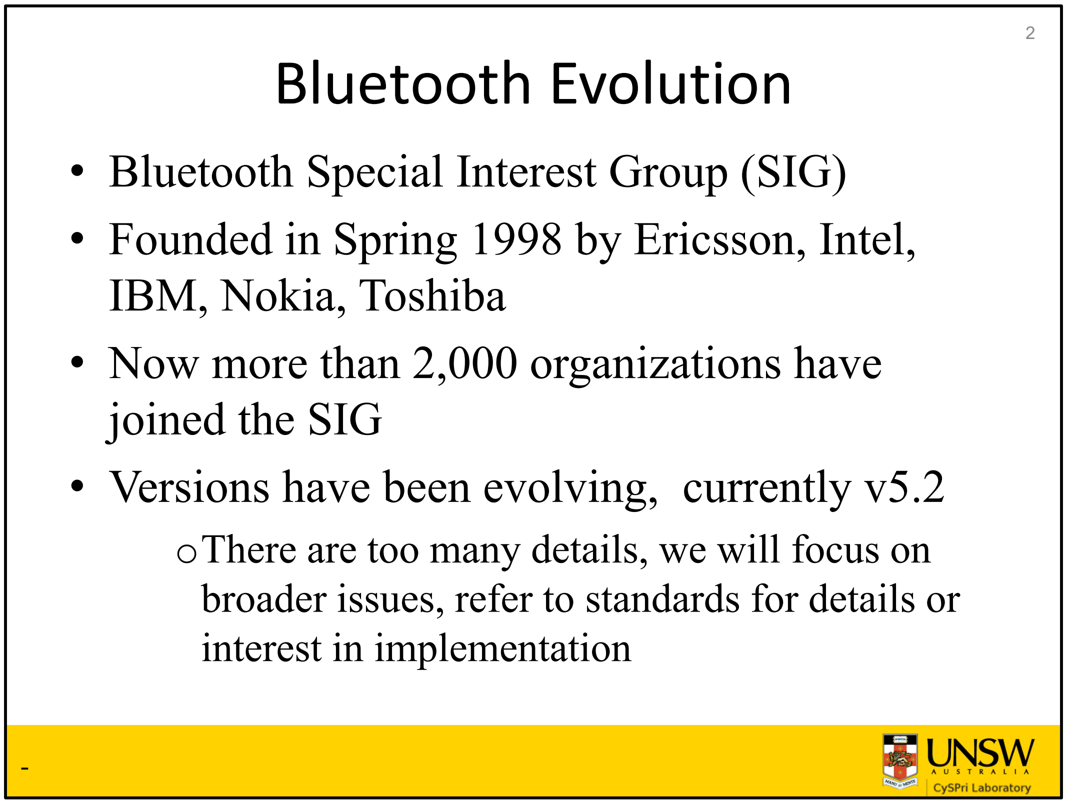
WK09-03-Bluetooth Security

Securing Fixed and Wireless Networks COMP4337/9337

Never Stand Still

School of Computer Science and Engineering, UNSW



Bluetooth Evolution

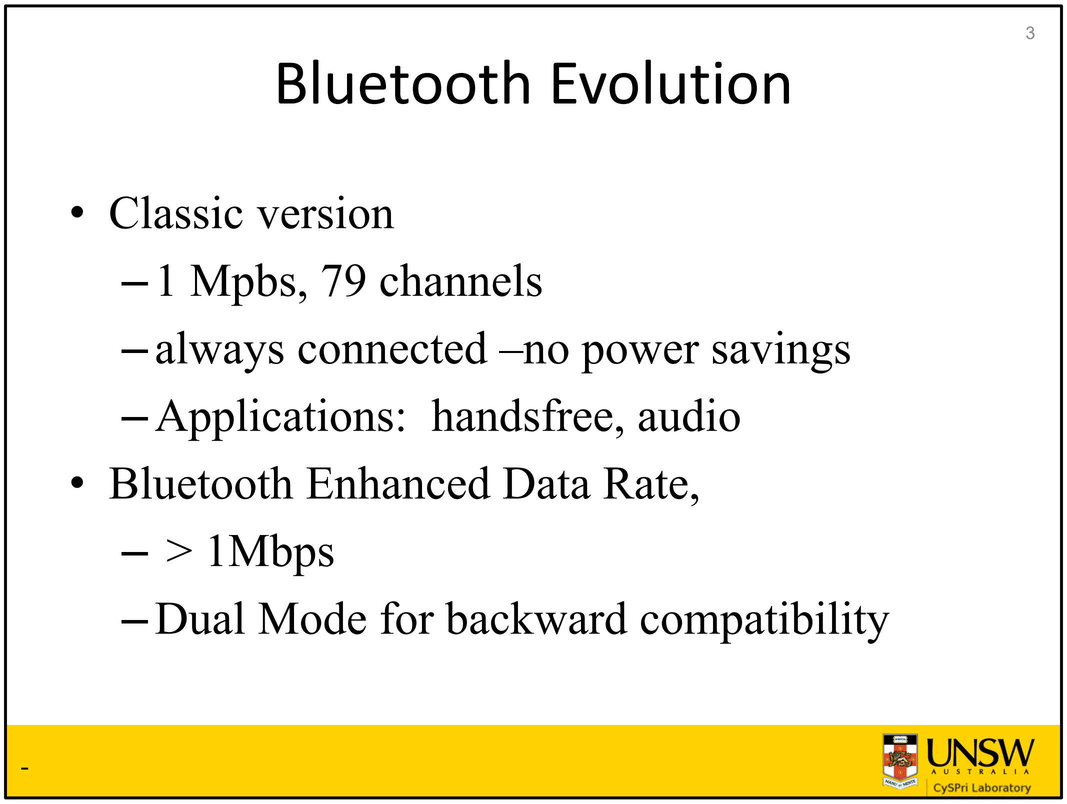
- Bluetooth Special Interest Group (SIG)
- Founded in Spring 1998 by Ericsson, Intel, IBM, Nokia, Toshiba
- Now more than 2,000 organizations have joined the SIG
- Versions have been evolving, currently v5.2
  - o There are too many details, we will focus on broader issues, refer to standards for details or interest in implementation

-

UNSW AUSTRALIA CySPri Laboratory

Bluetooth was founded in 1998 and It has a Bluetooth Special Interest Group (SIG) founded by consortium of companies including Ericsson, Intel, IBM etc.

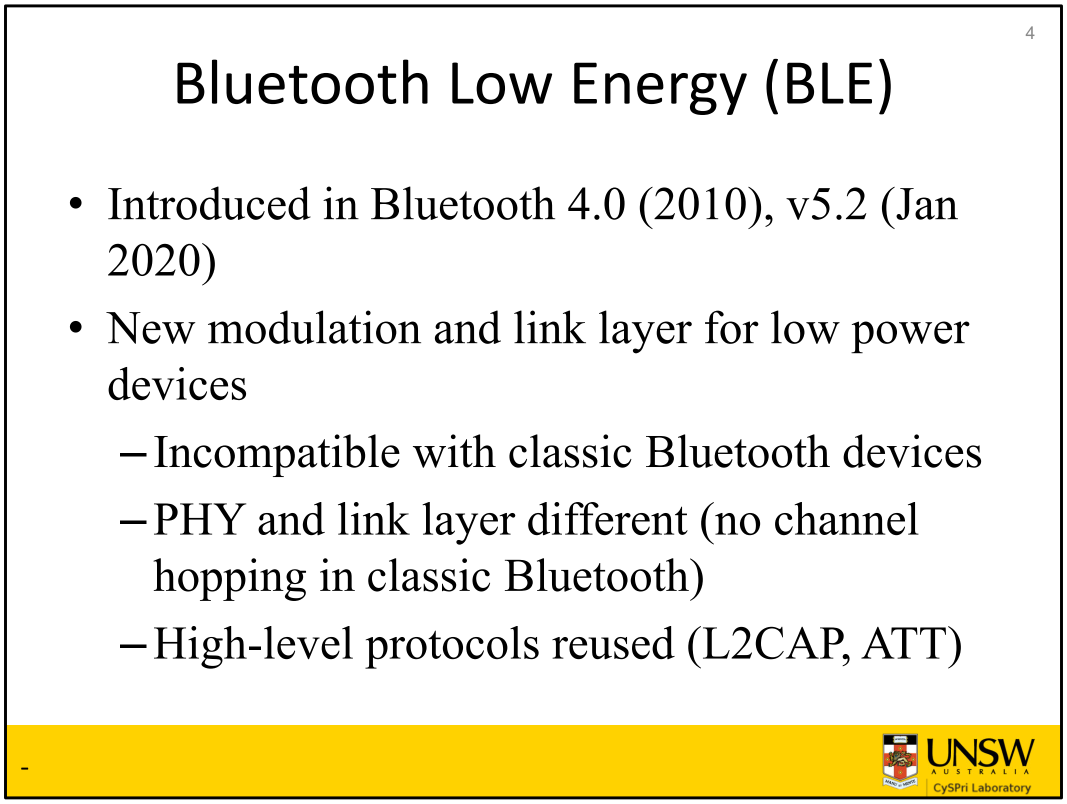
Now more than 2000 industrial organizations have joined this special interest group to evolve the standard and develop Bluetooth related devices and applications.



Bluetooth Evolution

- Classic version
  - 1 Mpbs, 79 channels
  - always connected –no power savings
  - Applications: handsfree, audio
- Bluetooth Enhanced Data Rate,
  - > 1Mbps
  - Dual Mode for backward compatibility

A very quick look at older Bluetooth developments to see how it evovled



Bluetooth Low Energy (BLE)

- Introduced in Bluetooth 4.0 (2010), v5.2 (Jan 2020)
- New modulation and link layer for low power devices
  - Incompatible with classic Bluetooth devices
  - PHY and link layer different (no channel hopping in classic Bluetooth)
  - High-level protocols reused (L2CAP, ATT)

-

UNSW AUSTRALIA CySPri Laboratory

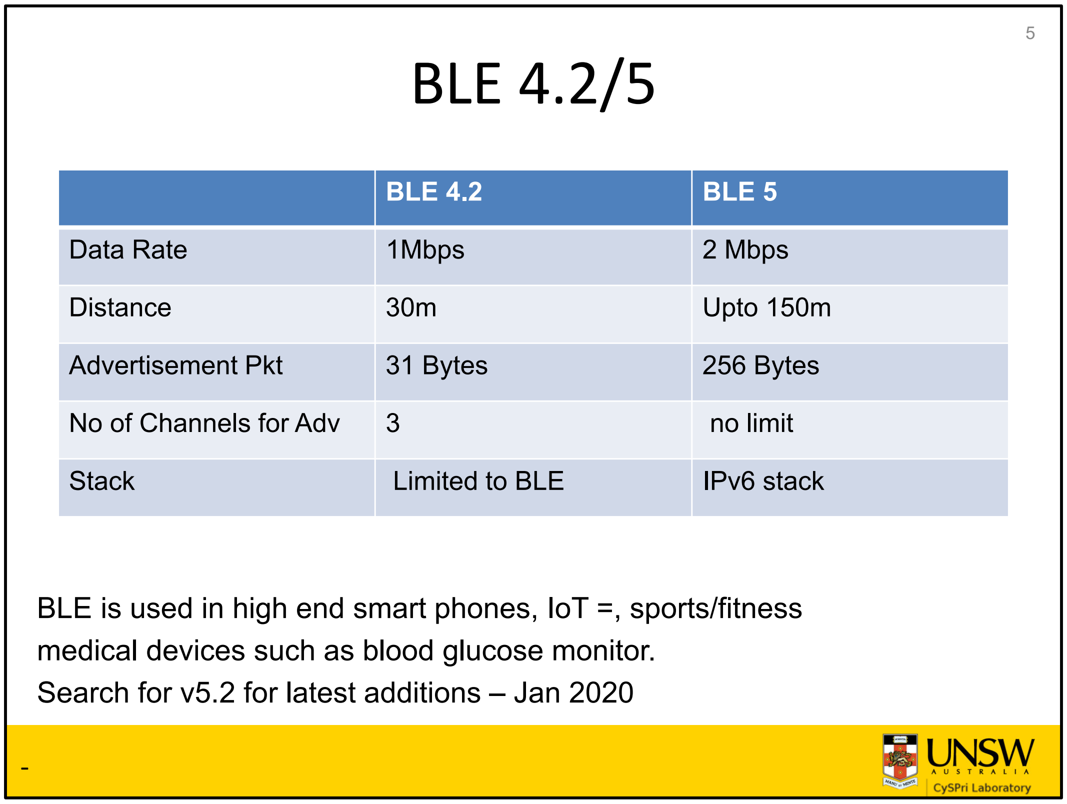
Channel hopping is used in Bluetooth Low Energy protocol introduced in 2010. Also called Bluetooth 4.0

Introduce new modulation and link layer for low power devices It is not compatible with classic Bluetooth devices

High level protocols have been reused.

Bluetooth Low Energy (BLE) 4.1 and higher (40 channels), Event based (low power), can support mesh network (earlier only, master slave), competes with Zigbee for IoT, Better security, Application: Medical,   FHSS (frequency hopping spread spectrum), more on channel hopping later.

Latest version 5.2 has introduced some more new features – again you can find out these detais if interested.   Some interesting power-control features useful in localisaiton.



BLE 4.2/5

	BLE 4.2	BLE 5
Data Rate	1Mbps	2 Mbps
Distance	30m	Upto 150m
Advertisement Pkt	31 Bytes	256 Bytes
No of Channels for Adv	3	no limit
Stack	Limited to BLE	IPv6 stack

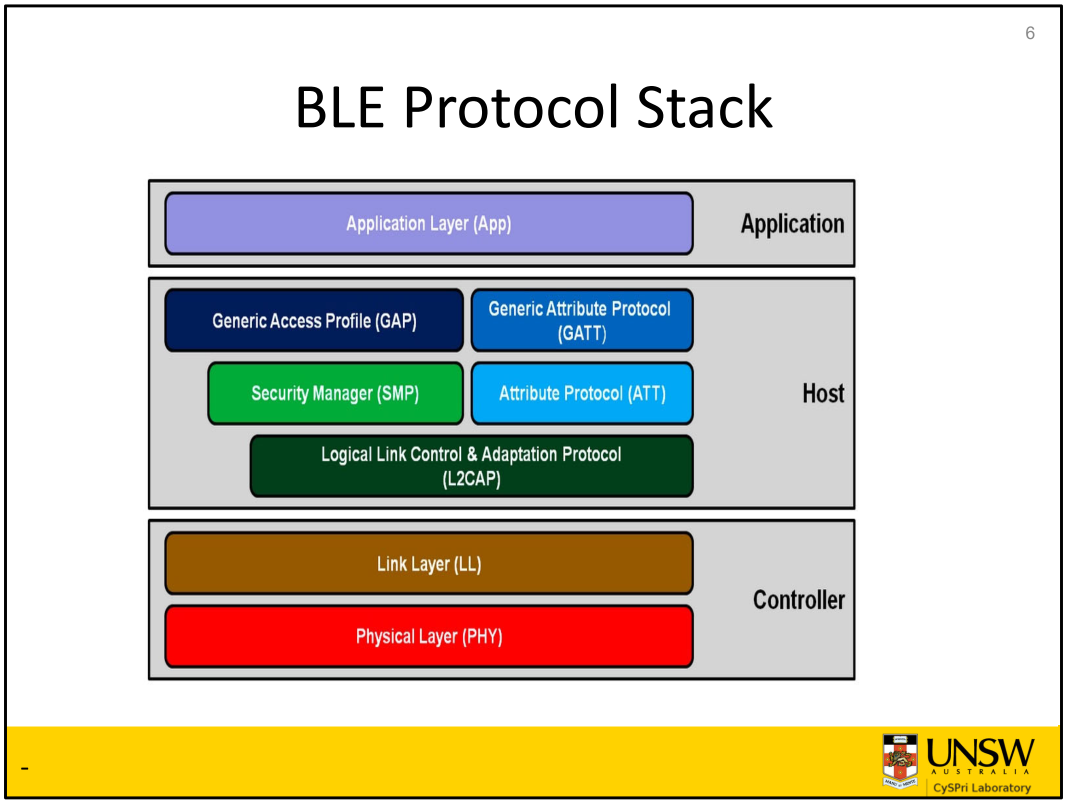
BLE is used in high end smart phones, IoT =, sports/fitness
medical devices such as blood glucose monitor.
Search for v5.2 for latest additions – Jan 2020

-

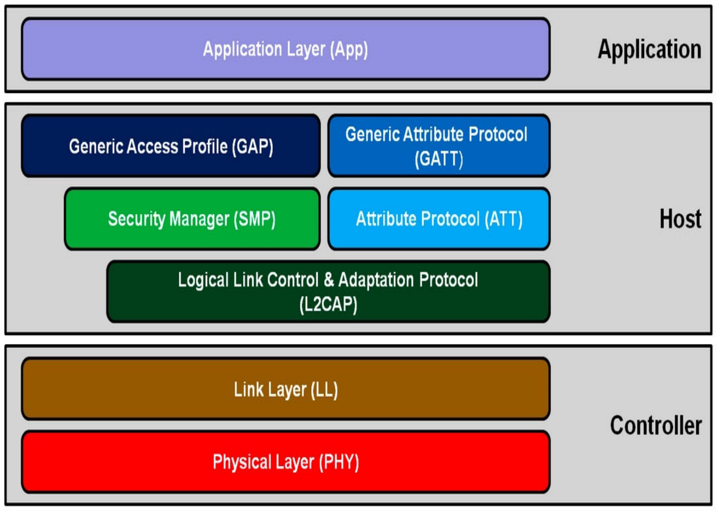
UNSW AUSTRALIA CySPri Laboratory

Table is self-explanatory.

Applicatons:High end smart phones
Sports/fitness devices
Door locks
Upcoming medical devices (e.g., blood glucose monitor)



BLE Protocol Stack



-

UNSW AUSTRALIA CySPri Laboratory

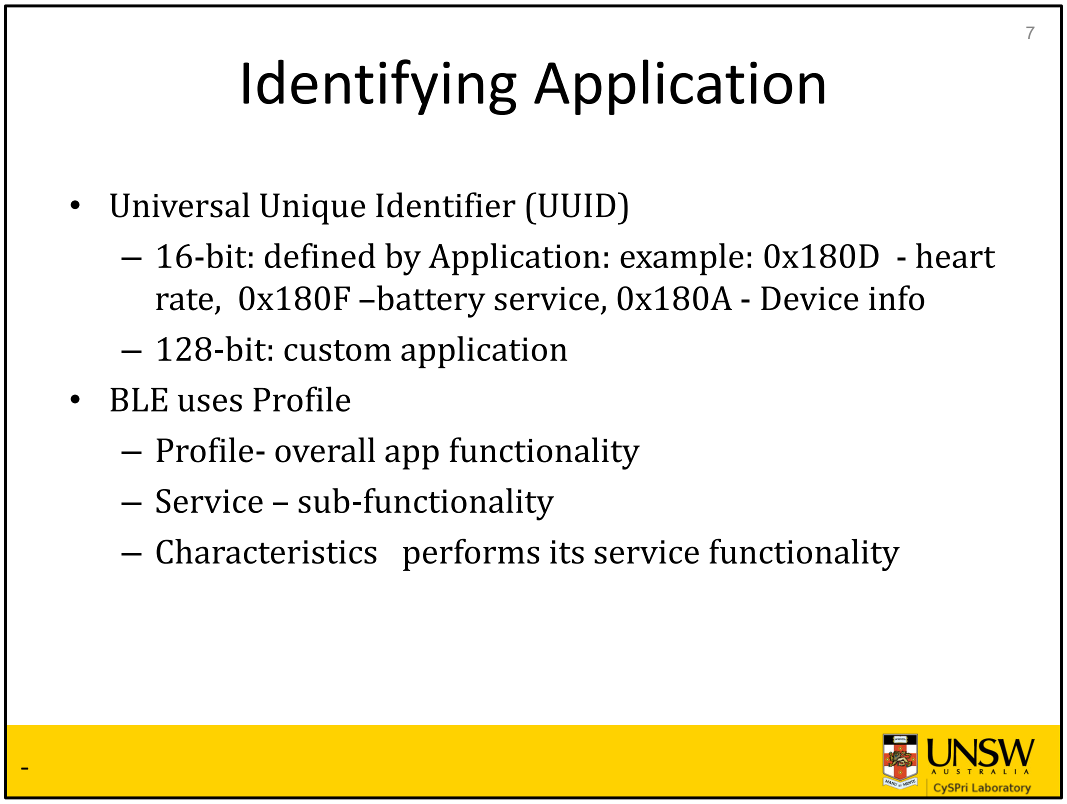
Details not needed to be memorized, good to have broader appreciation. Our focus is security so we are not going to discuss these in any details.

However, some of the protocols are useful for building applications in a standard way so that say a data-reader can read values (e.g. hear-rate from a heart-rate monitory use standard higher layer protocols.

No need to memorise these

ATT (Attribute protocol) – how to discover/read/write attributes on a peer device

GATT (Generic Attribute Profile) – how to discover and provide services based on ATT
L2CAP (Logical Link Control and Adaptation Protocol) – packet segmentation and reassemble



Identifying Application

- Universal Unique Identifier (UUID)
  - 16-bit: defined by Application: example: 0x180D - heart rate, 0x180F –battery service, 0x180A - Device info
  - 128-bit: custom application
- BLE uses Profile
  - Profile- overall app functionality
  - Service – sub-functionality
  - Characteristics performs its service functionality

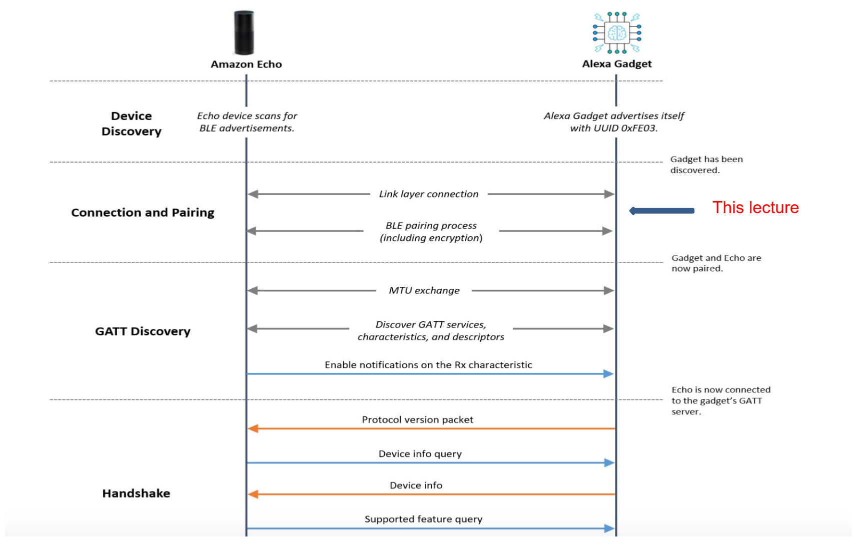
-

UNSW AUSTRALIA CySPri Laboratory

Not much to do with security. You can skip this, more for data access/application.

Simply if you come across UUID etc in protocol fields, you can relate to this.

Example Bluetooth Link

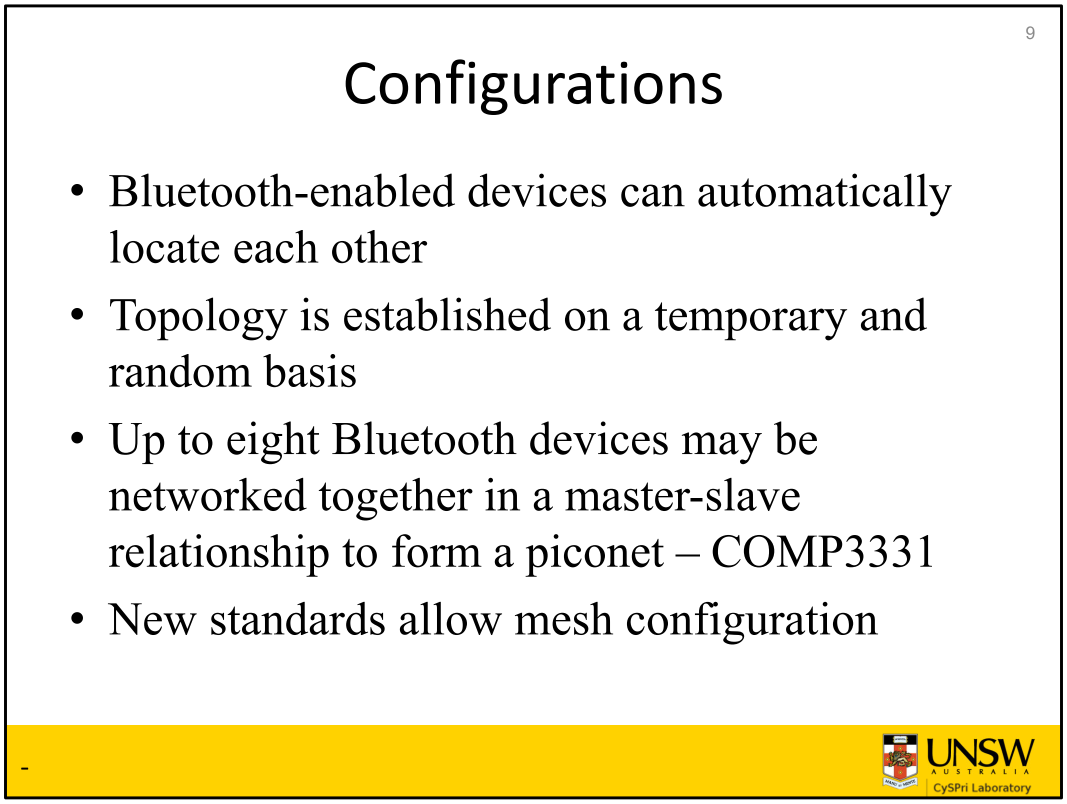

Amazon Echo
Alexa Gadget
Device Discovery
Echo device scans for BLE advertisements.
Alexa Gadget advertises itself with UUID 0xFE03.
Gadget has been discovered.
Connection and Pairing
Link layer connection
BLE pairing process (including encryption)
This lecture
Gadget and Echo are now paired.
GATT Discovery
MTU exchange
Discover GATT services, characteristics, and descriptors
Enable notifications on the Rx characteristic
Echo is now connected to the gadget's GATT server.
Handshake
Protocol version packet
Device info query
Device info
Supported feature query

https://developer.amazon.com/en-US/docs/alexa/alexa-gadgets-toolkit/bluetooth-le-pair-connect.html

-

UNSW AUSTRALIA
CySPri Laboratory

You can see the whole process – I recommend you follow this link (not for exam). Shows you various phases. Non-security details in not within our scope.



Configurations

- Bluetooth-enabled devices can automatically locate each other
- Topology is established on a temporary and random basis
- Up to eight Bluetooth devices may be networked together in a master-slave relationship to form a piconet – COMP3331
- New standards allow mesh configuration

-

UNSW AUSTRALIA CySPri Laboratory

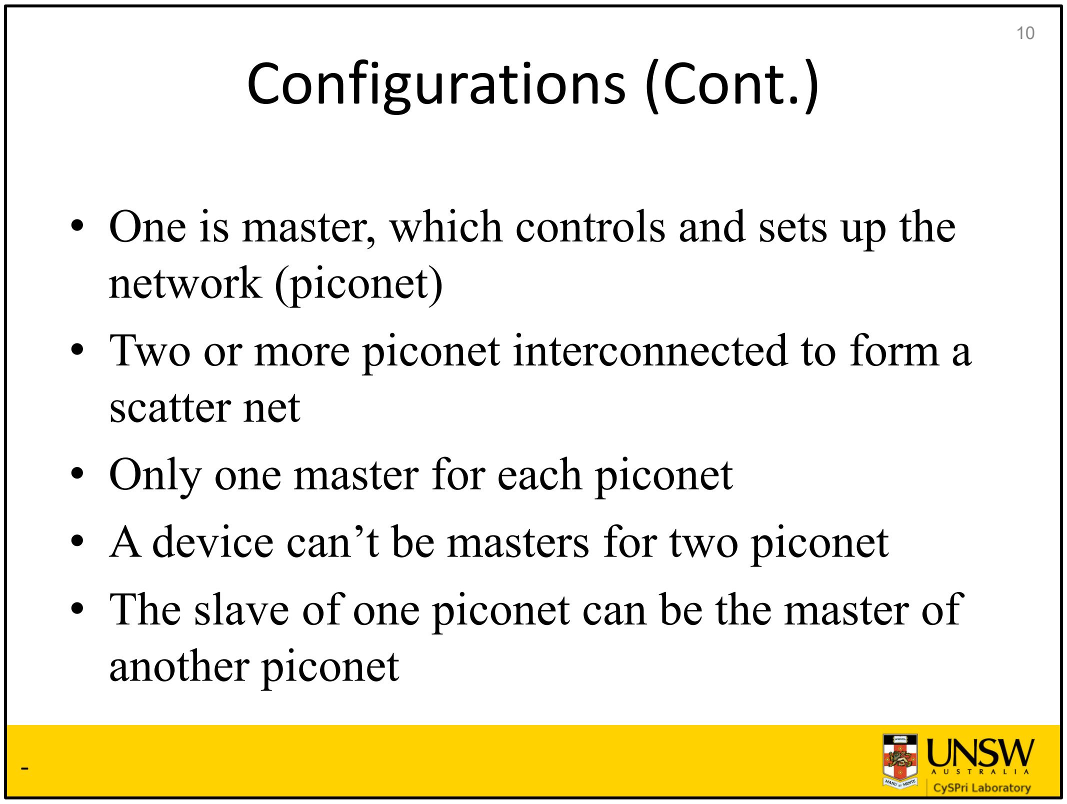
Bluetooth enabled devices that most of you have used, they can automatically locate each other.

When you use one Bluetooth enabled device it searches for other Bluetooth enabled device that are available to connect. These devices form a network in a very ad-hoc way.

If you go to your local shop (such as JB HiFi) to test the quality of a Bluetooth enabled speaker, you can use your Bluetooth enabled mobile device such as IPhone to connect to a Bluetooth speaker and then stream your music to the Bluetooth speaker to play your favorite songs.

The topology is formed on a temporary and random basis. You can thus connect up to 8 Bluetooth devices in a master slave relationship.

This Bluetooth network is called a Piconet. National Institute of Standards and Technology (NIST) defines this network as Wireless Personal Area Network (WPAN).



Configurations (Cont.)

- One is master, which controls and sets up the network (piconet)
- Two or more piconet interconnected to form a scatter net
- Only one master for each piconet
- A device can't be masters for two piconet
- The slave of one piconet can be the master of another piconet

-

UNSW AUSTRALIA CySPri Laboratory

In a Piconet, there can be only one master to control and setup the network.

And two or more Piconet can interconnect to form a Scatter net.

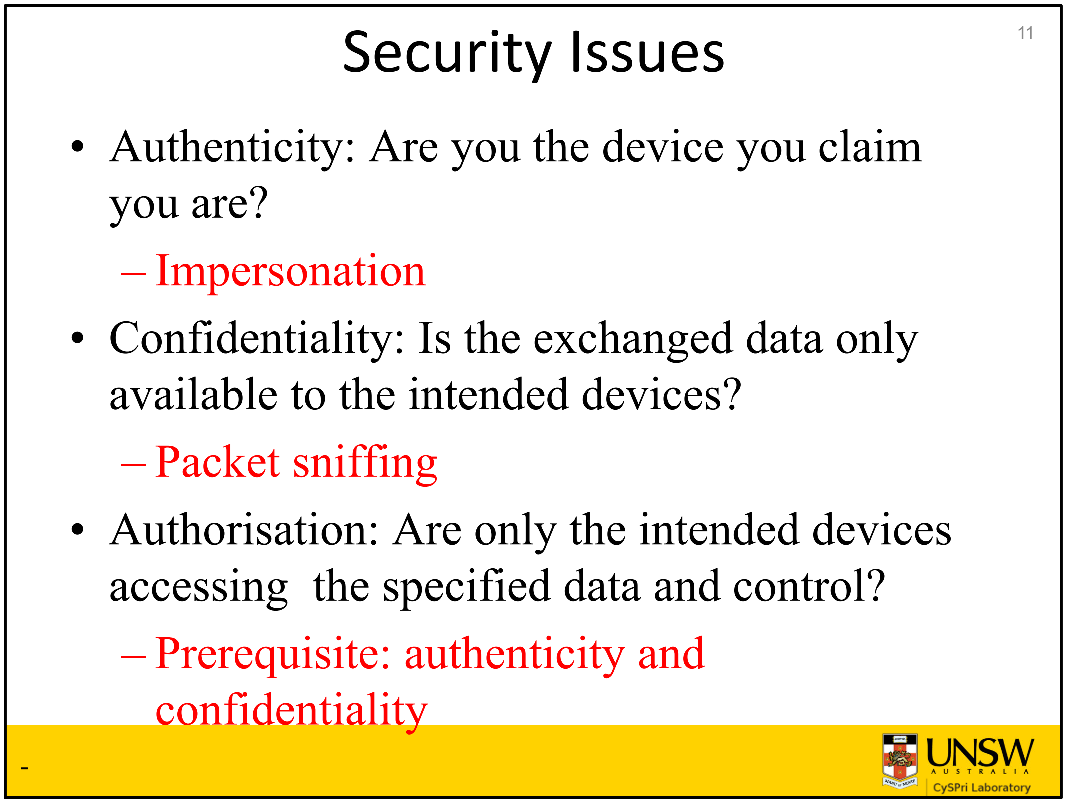
There is only one master allowed for each Piconet.

A device cannot be a master for two Piconet

However, a slave of one Piconet can be the master of another Piconet.

With newer standards, terminologies get a bit mixed up.

They also use Initiator/Responder or Client/Server etc at various laeyrs of protocol rather than the master/slave used for connection establishsment.



Security Issues

- Authenticity: Are you the device you claim you are?
  - Impersonation
- Confidentiality: Is the exchanged data only available to the intended devices?
  - Packet sniffing
- Authorisation: Are only the intended devices accessing the specified data and control?
  - Prerequisite: authenticity and confidentiality

-

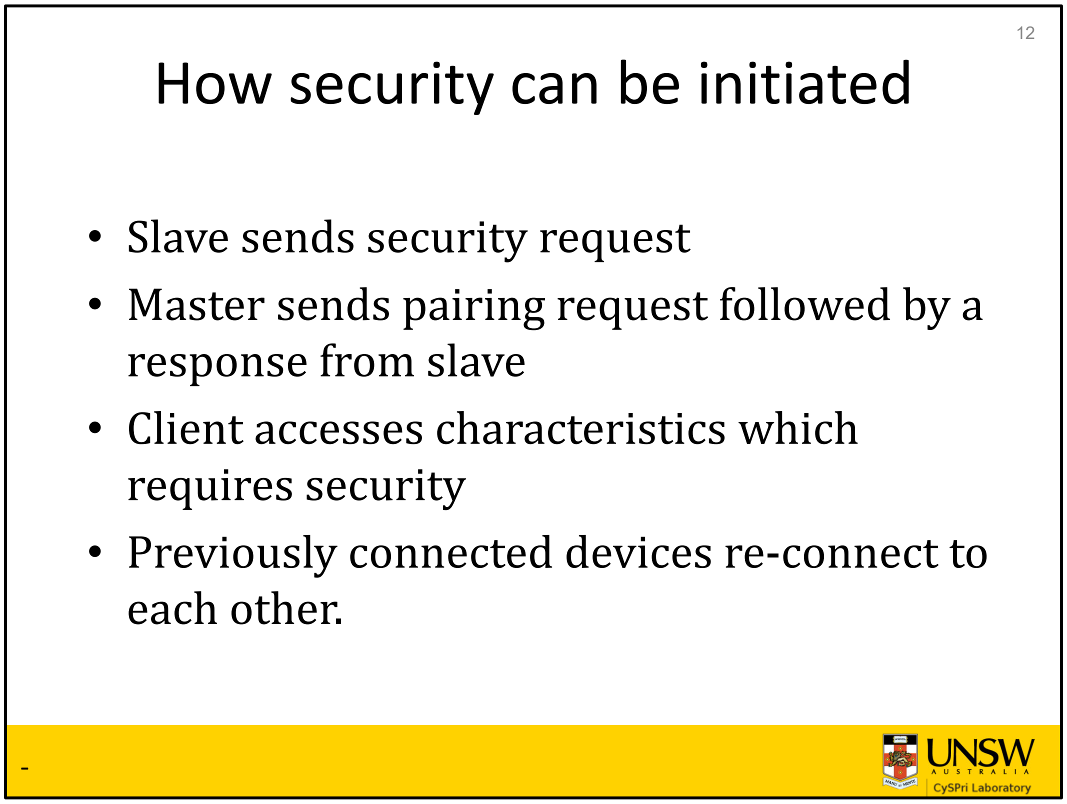
UNSW AUSTRALIA CySPri Laboratory

In Bluetooth communications these 3 are the most essential security issues that must be considered.

Authenticity means that: Any Bluetooth device must be able to authenticate themselves to prove/verify who they claim they are. An Impersonation attack is launched pretending to be someone else. If you cannot verify who you are. The protocol should be able to detect this to identify someone posing as another device.

For confidentiality, the data has to be only available to the intended device instead of some other device. Packet sniffing can be launched against this security feature to access data that is being transferred between two devices in an unauthorized manner.

And the third property is Authorization that means only the intended device has access to specified data and control. This combines the above two features. Prove your authenticity first and then exchange data in a secure manner.



How security can be initiated

- Slave sends security request
- Master sends pairing request followed by a response from slave
- Client accesses characteristics which requires security
- Previously connected devices re-connect to each other.

-

UNSW AUSTRALIA CySPri Laboratory

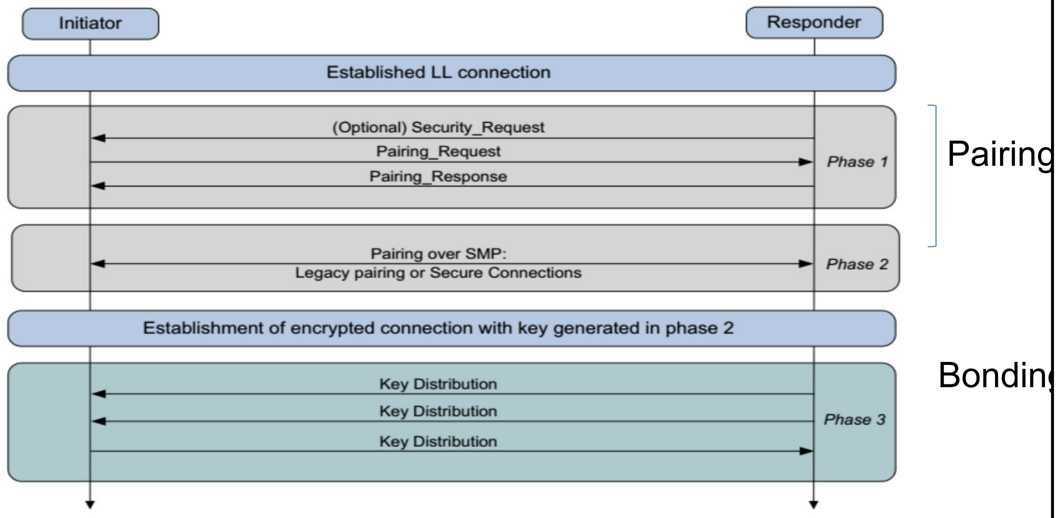
These are pretty self explanatory

The last point will come back again, this method is called bonding.

You can skip the explanation below as they are expanded further.

When two devices, which initially do not have security, wish to do something that requires security, the devices must first pair. This process is triggered by a Central device (e.g. a smartphone) that is attempting to access a data value (a “characteristic”) on a Peripheral device that requires authenticated access. Pairing involves authenticating the identity of two devices, encrypting the link using a short-term key (STK) and then distributing long-term keys (LTK) used for encryption. The LTK is saved for faster reconnection in the future, that is termed Bonding

Pairing/Key distribution


Initiator
Responder
Established LL connection
(Optional) Security_Request
Pairing_Request
Pairing_Response
Phase 1
Pairing
Pairing over SMP:
Legacy pairing or Secure Connections
Phase 2
Establishment of encrypted connection with key generated in phase 2
Key Distribution
Key Distribution
Key Distribution
Phase 3
Bondin

The pairing flowchart which applies to both legacy pairing and secure connections.

Image src: https://medium.com/rtone-iot-security/deep-dive-into-bluetooth-le-security-d2301d640bfc

-

(Fundamentally, these exchanges are not very different from what you have already learnt in many other protocols, details vary)

The role each device plays is defined in the Security Manager (SM) portion of the BLE stack. They are the Initiator (the central), and the Responder (the peripheral).

Phase One

two devices let each other know what they are capable of doing. eg. Feature exchange – where devices i/o capability, whether MiTM protection is needed, OOB flag is set…

-The values they are reading are Attribution Protocol (ATT) values. These live at layer 4 with L2CAP, and are typically not ever encrypted. -determine which pairing method is going to be used in phase two,

- ATT values will be different for a Bluetooth Smart vs a Bluetooth Smart Ready device. (again smart and smart-ready terms are for certain versio of standards, Smart-ready typicially phones/laptops, smart – other less capable devices)

Phase Two

-to generate a Short Term Key (STK).

-devices agreeing on a Temporary Key (TK) mixed with some random numbers which gives them the STK.

- STK itself is never transmitted between devices,   commonly known as LE legacy pairing.
- if the Secure Connection Only Mode is being used, a Long Term Key (LTK) is generated at this phase (instead of an STK), and this is known as LE Secure Connections.

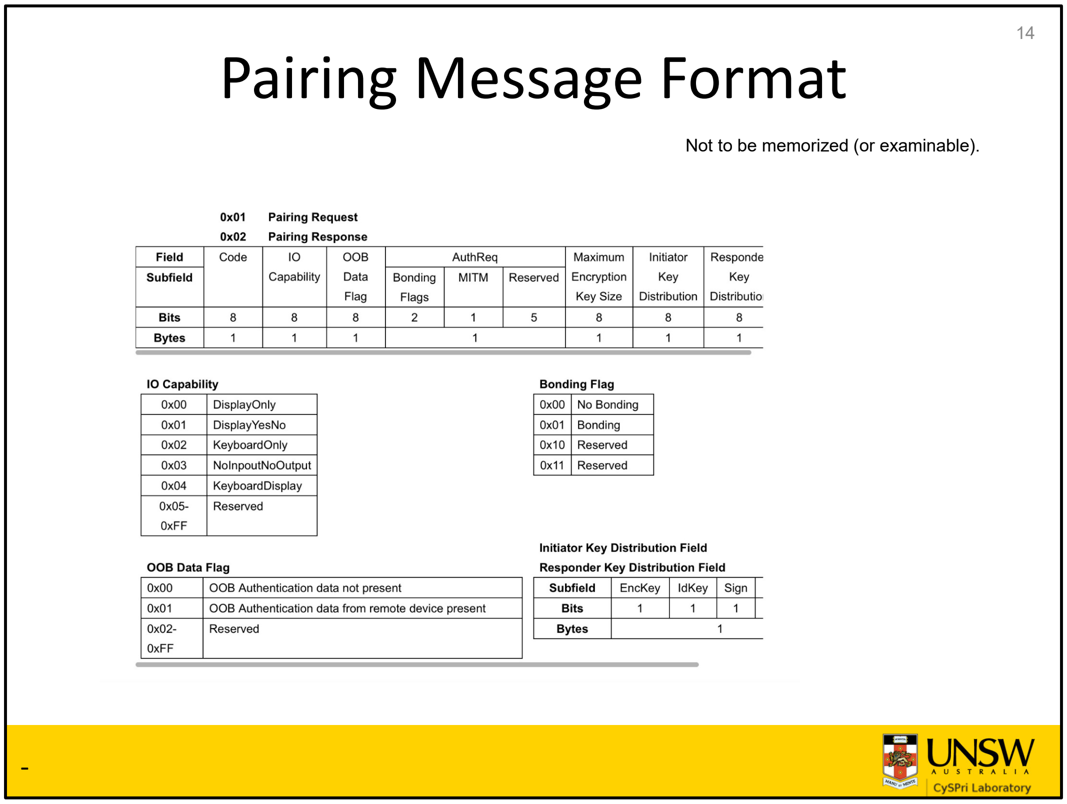
Phase Three

-the key from phase two is used to distribute the rest of the keys needed for communications.

If an LTK wasn't generated in phase two, one is generated in phase three.

Data like the Connection Signature Resolving Key (CSRK) for data signing, you already know digital signature

- the Identity Resolving Key (IRK) for private MAC address generation and lookup are generated in this phase. – we talk about this a bit later.



14

Pairing Message Format

Not to be memorized (or examinable).

	0x01	**Pairing Request**
	0x02	**Pairing Response**

Field	Code	IO	OOB	AuthReq			Maximum	Initiator	Responde
Subfield		Capability	Data Flag	Bonding Flags	MITM	Reserved	Encryption Key Size	Key Distribution	Key Distributio
Bits	8	8	8	2	1	5	8	8	8
Bytes	1	1	1	1			1	1	1

IO Capability

0x00	DisplayOnly
0x01	DisplayYesNo
0x02	KeyboardOnly
0x03	NoInpoutNoOutput
0x04	KeyboardDisplay
0x05-0xFF	Reserved

Bonding Flag

0x00	No Bonding
0x01	Bonding
0x10	Reserved
0x11	Reserved

OOB Data Flag

0x00	OOB Authentication data not present
0x01	OOB Authentication data from remote device present
0x02-0xFF	Reserved

Initiator Key Distribution Field
Responder Key Distribution Field

Subfield	EncKey	IdKey	Sign
Bits	1	1	1
Bytes	1		

-

UNSW AUSTRALIA CySPri Laboratory

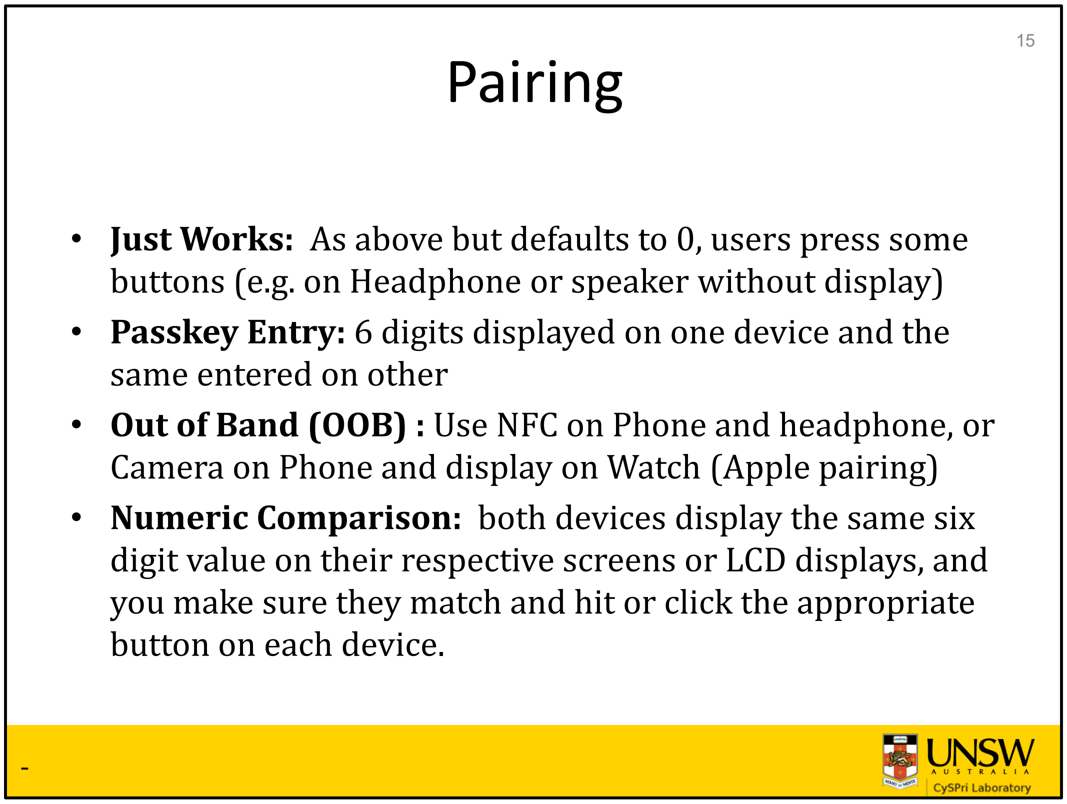
Not to be memorized (or examinable).

Just appreciate that the protocol exchange for request response has these fields as discussed in previous foil, phase-1.

The second step of the pairing procedure is the Authentication step which is performed based on the information exchanged in the previous step in the Pairing Request and Pairing Response messages.

In this step the Temporary Key is generated. There are 3 ways of generating the TK in BLE (described below): Just Works, OOB, and Passkey Entry. The priority of these methods is the following: if both devices have set the OOB flag than the OOB method is used regardless of the other flags in the Pairing Request and Response. If the OOB flag is not set on at least one of the devices then the MITM flag is checked. If both devices have not set the MITM flag then the just works method is chosen (IO capabilities are ignored), else the pairing algorithm is chosen based on the IO Capabilities.

The following table describes the relation between the TK and the pairing methods.



Pairing

- **Just Works:** As above but defaults to 0, users press some buttons (e.g. on Headphone or speaker without display)
- **Passkey Entry:** 6 digits displayed on one device and the same entered on other
- **Out of Band (OOB) :** Use NFC on Phone and headphone, or Camera on Phone and display on Watch (Apple pairing)
- **Numeric Comparison:** both devices display the same six digit value on their respective screens or LCD displays, and you make sure they match and hit or click the appropriate button on each device.

-

UNSW AUSTRALIA CySPri Laboratory

Just Works. Obviously, not all devices have a display, such as headphones or a speaker. Therefore, the Just Works method is probably the most popular one. Technically, it is the same as Numeric Comparison, but the six-digit value is set to all zeros. While Numeric Comparison requires some on-the-fly math if you are performing a MITM attack, there is no MITM protection with Just Works.

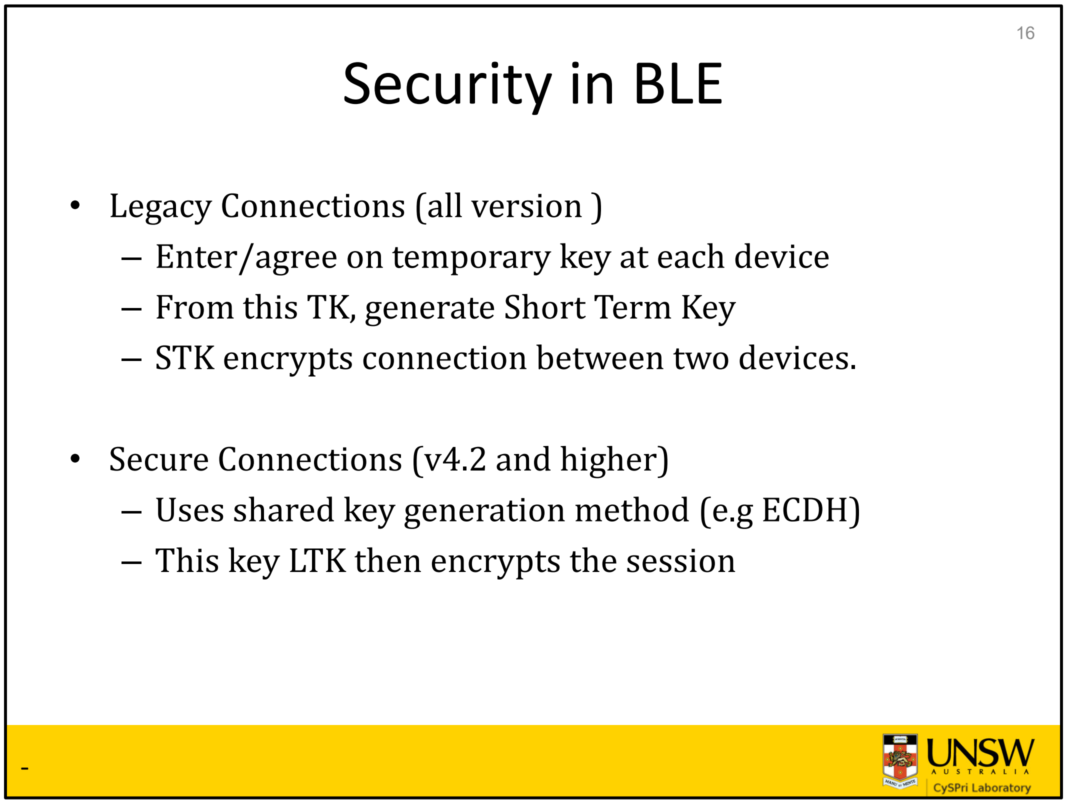
Passkey Entry. With Passkey Entry, a six-digit value is displayed on one device, and this is entered into the other device.

Out Of Band (OOB). A communication method outside of the Bluetooth communication channel is not used, but the information is still secured. The Apple Watch is a good example of this workflow. During the Apple Watch pairing method, a swirling display of dots is displayed on the watch face, and you point the pairing iPhone's camera at the watch face while (according to Apple) bits of information are transmitted via this process. Another example is using Near Field Communication (NFC) between NFC-capable headphones and a pairing phone.

For key exchange in Bluetooth, there are currently above pairing. However, the Bluetooth core specifications states that none of these paring methods provide protection against a passive eavesdropper.

Numeric Comparison. Basically, both devices display the same six digit value on their respective screens or LCD displays, and you make sure they match and hit or click the appropriate button on each device. This is not to prevent a man-inthe-middle (MITM) attack, mainly because it doesn't, but rather to identify the devices to each other.

"None of these key pairing methods provide protection against a passive eavesdropper" – Bluetooth Core Spec



Security in BLE

- Legacy Connections (all version)
  - Enter/agree on temporary key at each device
  - From this TK, generate Short Term Key
  - STK encrypts connection between two devices.

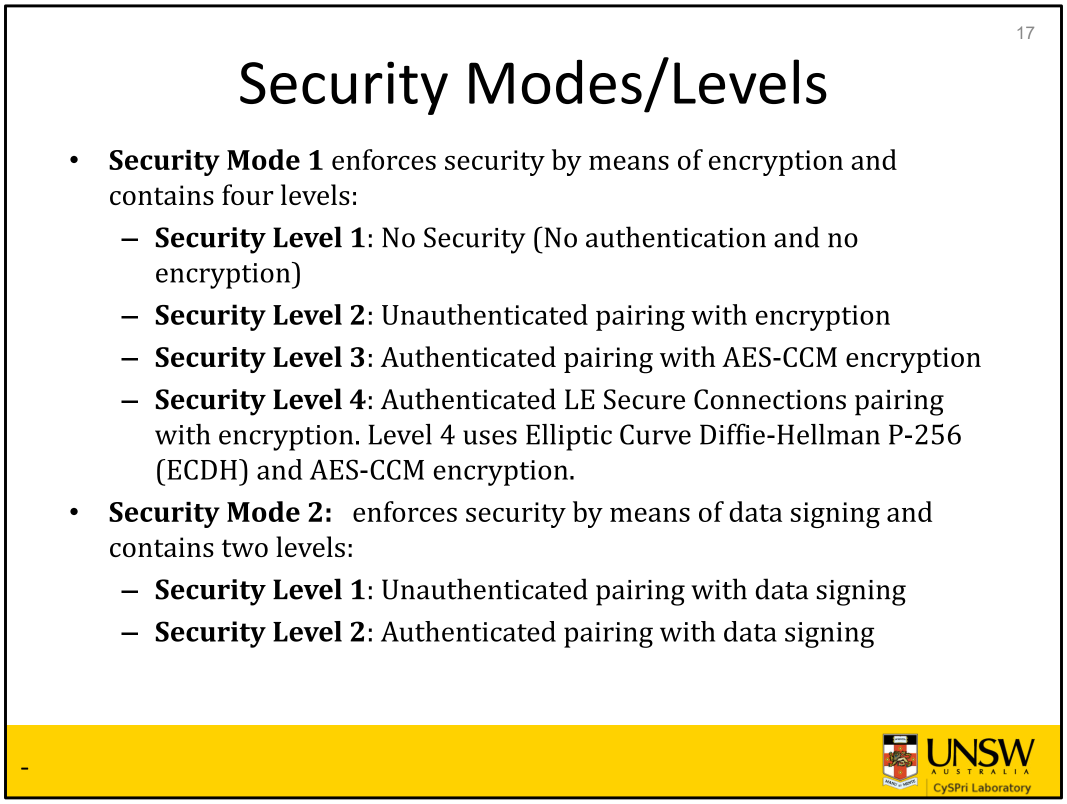
- Secure Connections (v4.2 and higher)
  - Uses shared key generation method (e.g ECDH)
  - This key LTK then encrypts the session

-

UNSW AUSTRALIA CySPri Laboratory

Self explanatory. Also read explanation on different phases.

Later options of using session key using Elliptic curve DH is more secure. Again details of certificate/key generation etc is similar to what you have seen earlier.



Security Modes/Levels

- **Security Mode 1** enforces security by means of encryption and contains four levels:
  - **Security Level 1**: No Security (No authentication and no encryption)
  - **Security Level 2**: Unauthenticated pairing with encryption
  - **Security Level 3**: Authenticated pairing with AES-CCM encryption
  - **Security Level 4**: Authenticated LE Secure Connections pairing with encryption. Level 4 uses Elliptic Curve Diffie-Hellman P-256 (ECDH) and AES-CCM encryption.
- **Security Mode 2:** enforces security by means of data signing and contains two levels:
  - **Security Level 1**: Unauthenticated pairing with data signing
  - **Security Level 2**: Authenticated pairing with data signing

-

UNSW AUSTRALIA CySPri Laboratory

JUST SKIM THOUGH THIS. There is too much details, no need to memorise any of this.

Basically, devices start with lowest level and then can ramp up based on application needs and capabilities. Connection operates at a specific Security mode. Within each mode are several security levels. The required security mode/level of a connection may change from time to time, there are procedures to increase that level

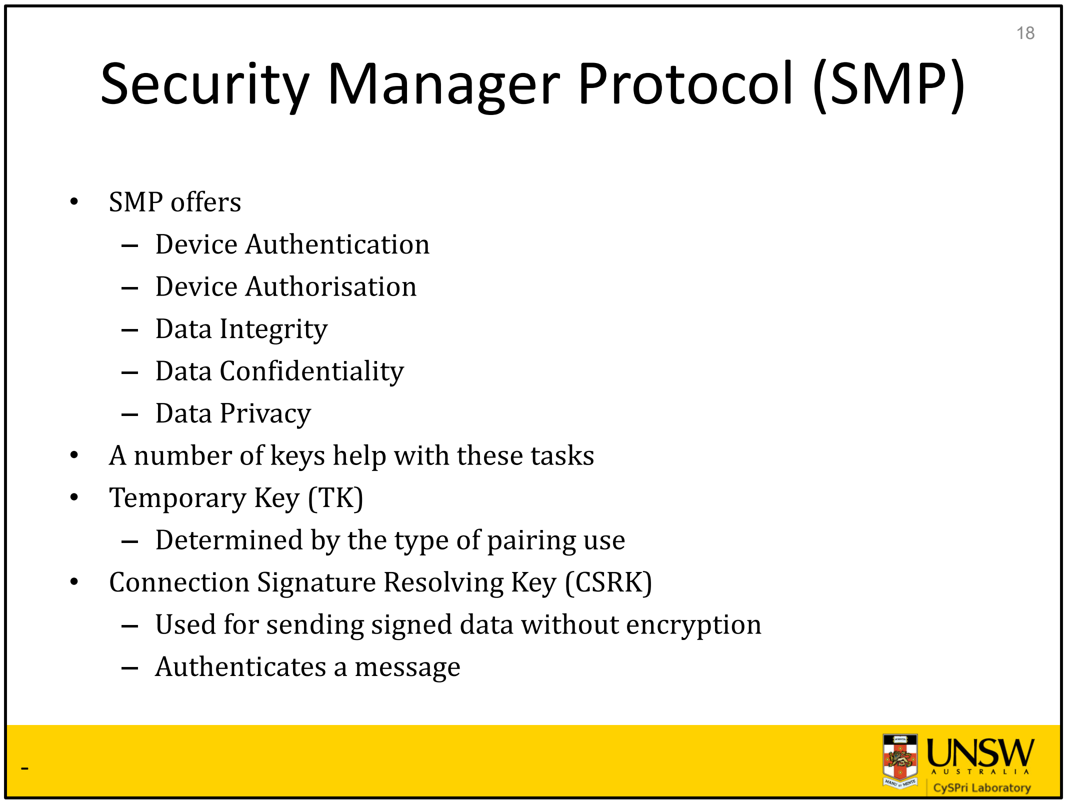
The new security level of the connection is based on the method of pairing performed and this is selected based on the I/O capabilities of each device. The security level of any subsequent reconnections is based on the level achieved during the initial pairing. Standards has a nice table of device capabilities and how the intersection gives you appropriate choice.

The role each device plays is defined in the Security Manager (SM) portion of the BLE stack. They are:

Initiator: Always corresponds to the Link Layer Master and therefore the GAP central.

Responder: Always corresponds to the Link Layer Slave and therefore the GAP peripheral.

Each connection starts its lifetime in Security mode 1, Level 1, and can later be upgraded to any security level by means of an authentication procedure discussed below.



Security Manager Protocol (SMP)

- SMP offers
  - Device Authentication
  - Device Authorisation
  - Data Integrity
  - Data Confidentiality
  - Data Privacy
- A number of keys help with these tasks
- Temporary Key (TK)
  - Determined by the type of pairing use
- Connection Signature Resolving Key (CSRK)
  - Used for sending signed data without encryption
  - Authenticates a message

-

UNSW AUSTRALIA CySPri Laboratory

Self explanatory.

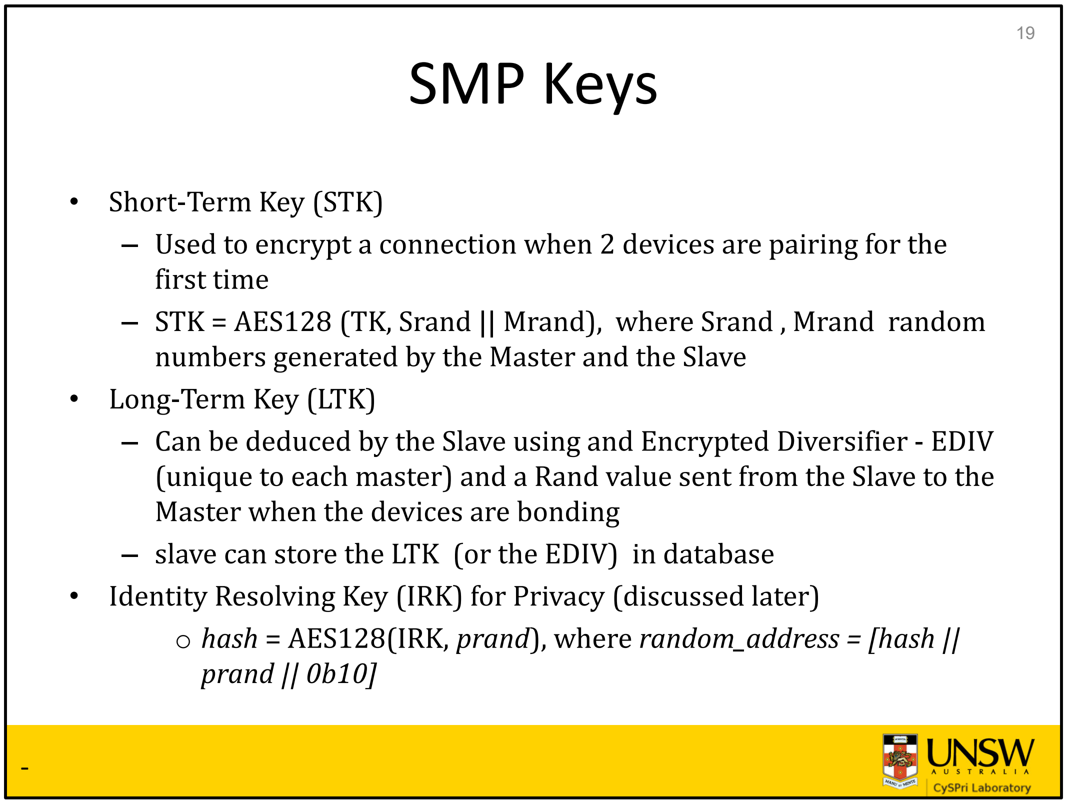
Important to understand purpose of each key (again this is not very different from what you have learnt earlier)

What services and keys are used for the communication between two devices are established during the SMP Pairing procedure which is performed by the SMP and set up by the Application according to its needs.

The third phase of the pairing procedure is the distribution of the keys. The keys are distributed using specific SMP packets. The keys are encrypted with the STK and once stored in a security database must have the same properties (authentication, MITM protection) as the STK. The keys which can be distributed are: (LTK, EDIV and Rand), IRK, CSRK.

The distributed EDIV and Rand values are transmitted in clear text by the master device to the slave device during encrypted session setup.

The BD_ADDR of devices is also distributed in this phase.



SMP Keys

- Short-Term Key (STK)
  - Used to encrypt a connection when 2 devices are pairing for the first time
  - STK = AES128 (TK, Srand || Mrand), where Srand , Mrand random numbers generated by the Master and the Slave
- Long-Term Key (LTK)
  - Can be deduced by the Slave using and Encrypted Diversifier - EDIV (unique to each master) and a Rand value sent from the Slave to the Master when the devices are bonding
  - slave can store the LTK (or the EDIV) in database
- Identity Resolving Key (IRK) for Privacy (discussed later)
  - *hash* = AES128(IRK, *prand*), where *random_address = [hash || prand || 0b10]*

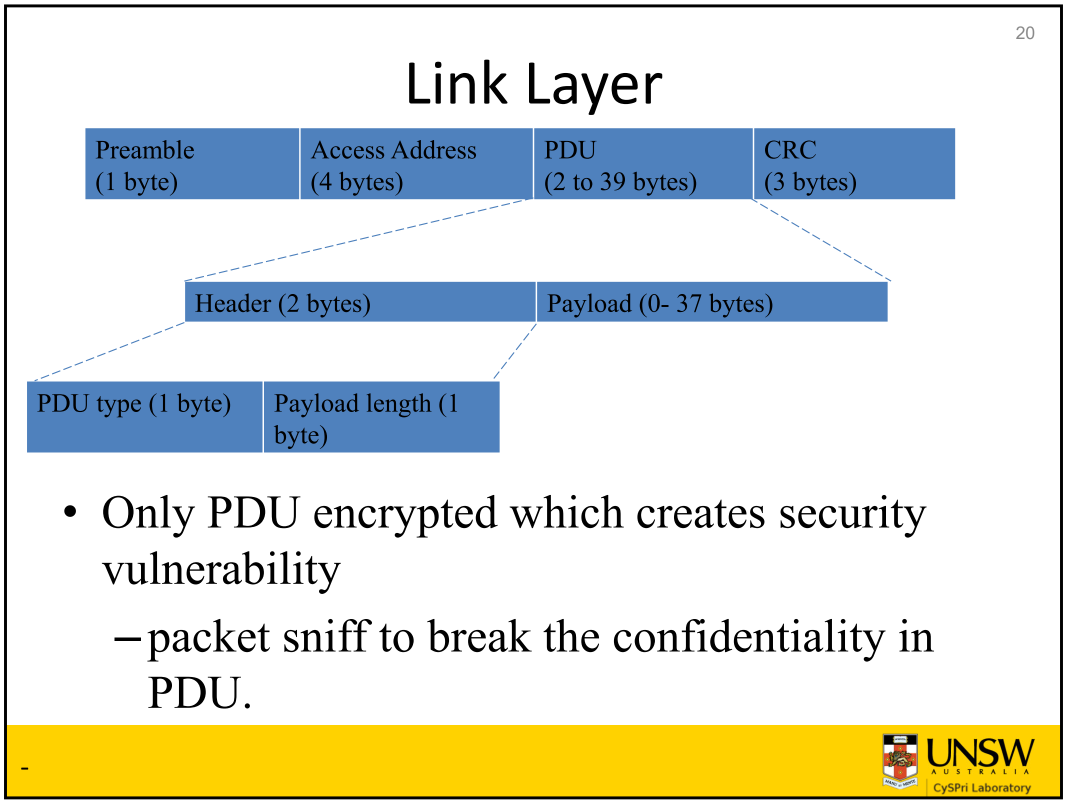
-

UNSW AUSTRALIA CySPri Laboratory

encrypted diversifier

slave can store the LTK (or the EDIV) in database

If EDIV stored, then it can derive LTK as needed without further exchange with master. Makes things quicker.



Link Layer

- Only PDU encrypted which creates security vulnerability
  - packet sniff to break the confidentiality in PDU.

-

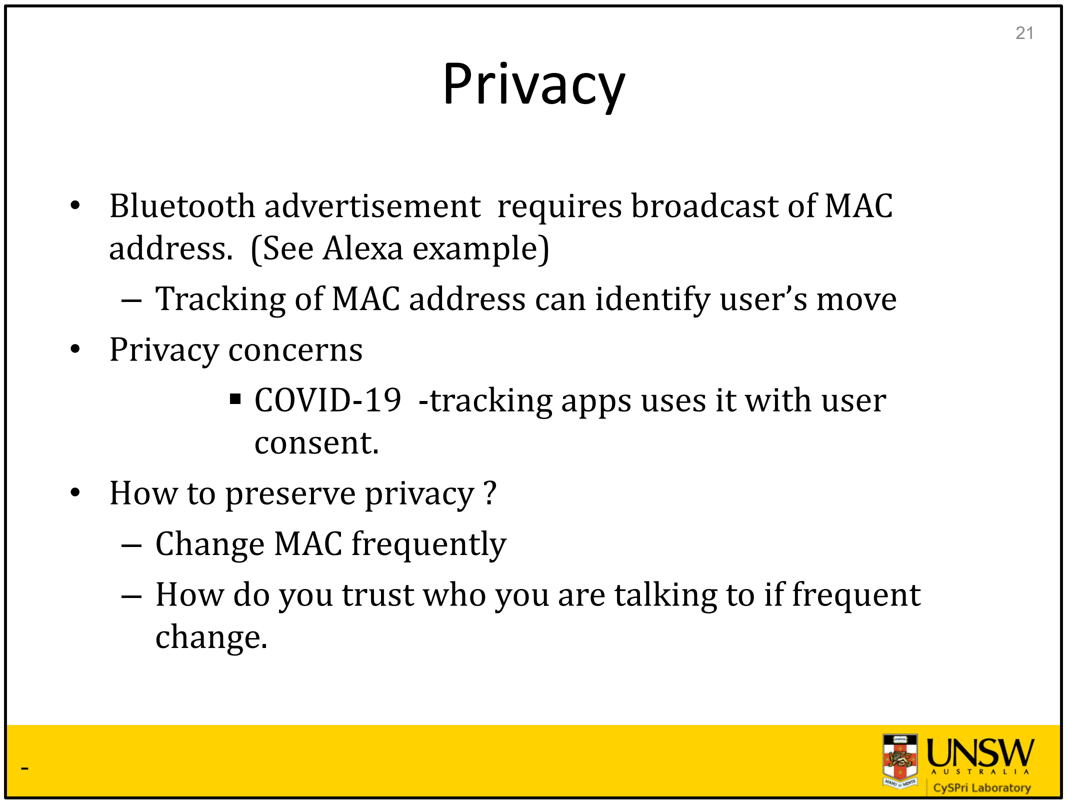
From networking perspective, we need to under the link layer protocol to be able to do security analysis. Again, no need to memorise any of this.

Preamble: 8-bit sequence (01010101), it is for receiving radio to synchronize the transmission.
Access Address: it is a 32-bit address for recipient to pickup the intended Bluetooth packets.
PDU = Protocol Data Unit
CRC = Cyclic Redundancy Check

In each Bluetooth packet, you have 4 parts. 1 byte Preamble, 4 bytes Access address, (2-39 bytes) PDU and 3 bytes CRC.
CRC provides the integrity check whether the packet has experienced any bit corruption.

In actual PDU, there is a 2 byte header and up to 37 bytes of payload.

Note that only the PDU gets encrypted using the Session key (STK or LTK) that we discussed earlier, rest three (Preamble, Access address and CRC) are sent in plaintext. Packet sniffing can be used to break the confidentiality.



Privacy

- Bluetooth advertisement requires broadcast of MAC address. (See Alexa example)
  - Tracking of MAC address can identify user's move
- Privacy concerns
  - COVID-19 -tracking apps uses it with user consent.
- How to preserve privacy ?
  - Change MAC frequently
  - How do you trust who you are talking to if frequent change.

-

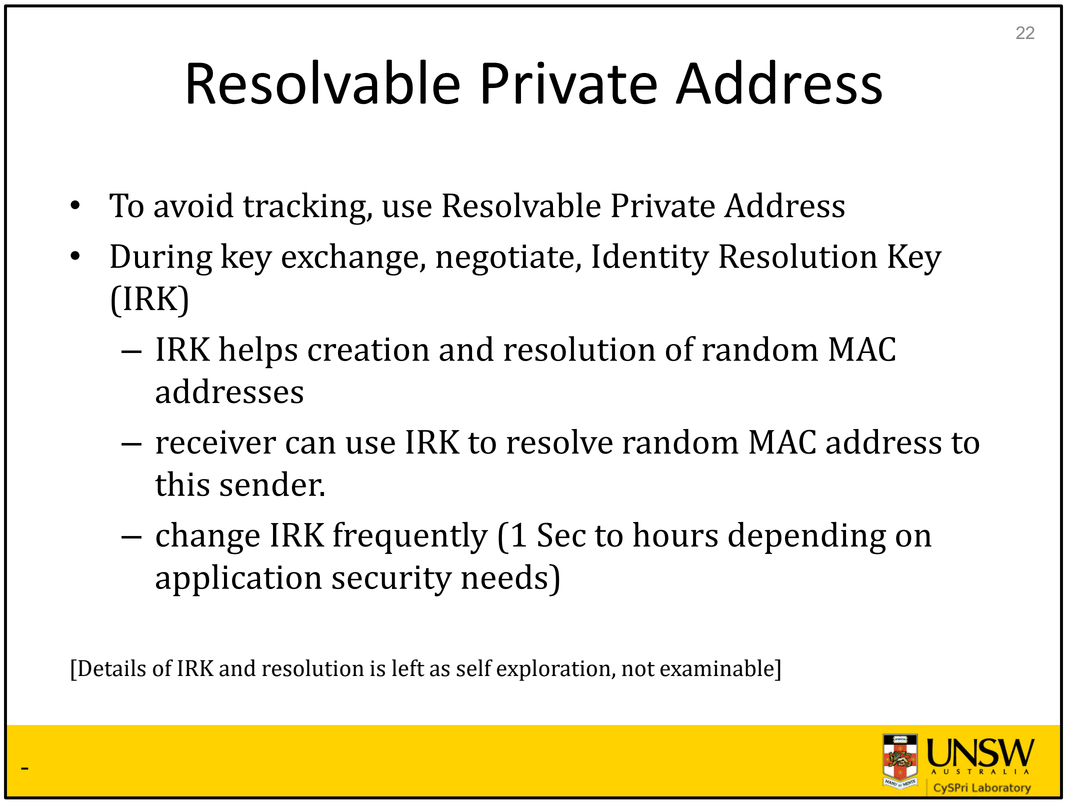
UNSW AUSTRALIA CySPri Laboratory

A quick detour considering that we are in the middle of COVID.

How privacy plays a role (both positive and negative)

Singapore COVID app

https://www.businessinsider.com.au/singapore-coronavirus-app-tracking-testing-no-shutdown-how-it-works-2020-3?r=US&IR=T



Resolvable Private Address

- To avoid tracking, use Resolvable Private Address
- During key exchange, negotiate, Identity Resolution Key (IRK)
  - IRK helps creation and resolution of random MAC addresses
  - receiver can use IRK to resolve random MAC address to this sender.
  - change IRK frequently (1 Sec to hours depending on application security needs)

[Details of IRK and resolution is left as self exploration, not examinable]

-

UNSW AUSTRALIA CySPri Laboratory

Public

48 bit LAN MAC address from IEEE, can identify chip vendor

(possibly security problems as can guess what model had what security)

IEEE address also gives away chip vendor /generation etc, which can expose security vulnerability if older standard used.

Rather than using the standard 48 bit IEEE allocated MAC address, they suggest using Resolvable Private Address.

- Random: follow rules set by SIG

Static: doesn't change except for power-cycling

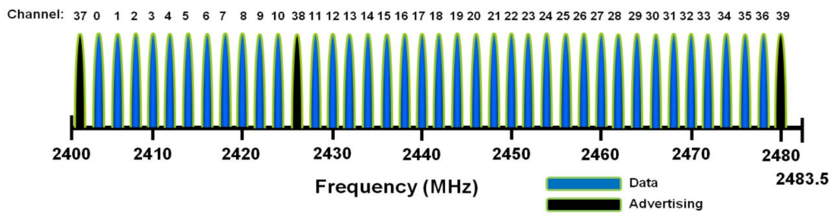
Private: a) resolvable, can change eg every 15 min

b) unresolvable – can't identify device

Physical Layer

Frequency Bands

The radio uses the 2.4 GHz Industrial, Scientific, and Medical (ISM) band to communicate and divides this band into 40 channels on 2 MHz spacing from 2.4000 GHz to 2.4835 GHz, starting at 2402 MHz:



- Physical layer: channels for hopping (40 available channels in 2.4Ghz)
  - Advertising: 3 channels
  - Data: 37 channels

-

Back to channel hopping:

In Bluetooth low energy communications, there are 40 available channels in 2.4GhZ range in total for the communicating parties to use.

Out of these 40, 3 channels are for advertisement and rest 37 are for data transfer. Each channel is of 2 Mhz width.

There are two phases of communication; First is advertising.

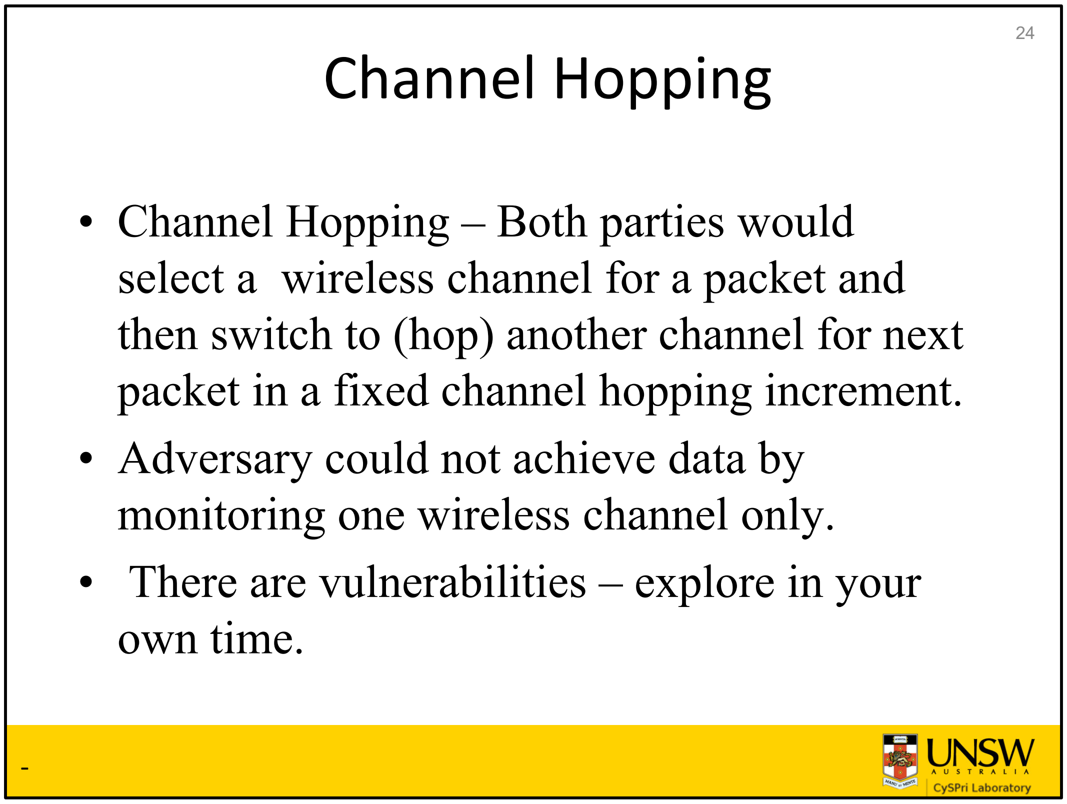
Advertising phase is required for device discovery (broadcast and connection establishment) so that devices can contact each other.

There are 3 channels available for advertising phase.

Next phase is the data transfer (after connection establishment) that uses the rest of the 37 data channels.

In latest standard, there is some change in how any channel can be used for advertising. You can explore this yourself – not examinable.

Broadcasting data – aka advertising (ch 37, 38,, 39) spread out though (not consective in frequency)



Channel Hopping

- Channel Hopping – Both parties would select a wireless channel for a packet and then switch to (hop) another channel for next packet in a fixed channel hopping increment.
- Adversary could not achieve data by monitoring one wireless channel only.
- There are vulnerabilities – explore in your own time.

-

UNSW AUSTRALIA CySPri Laboratory

Another mechanism to secure wireless networks is called Channel Hopping.

In Channel Hopping, we have multiple channels available within a given frequency for wireless devices to use. Devices select the same channel frequency to communicate.

In Bluetooth Smart protocol, the Bluetooth Special Interest Group (SIG) has specified channel hopping to increase the security.

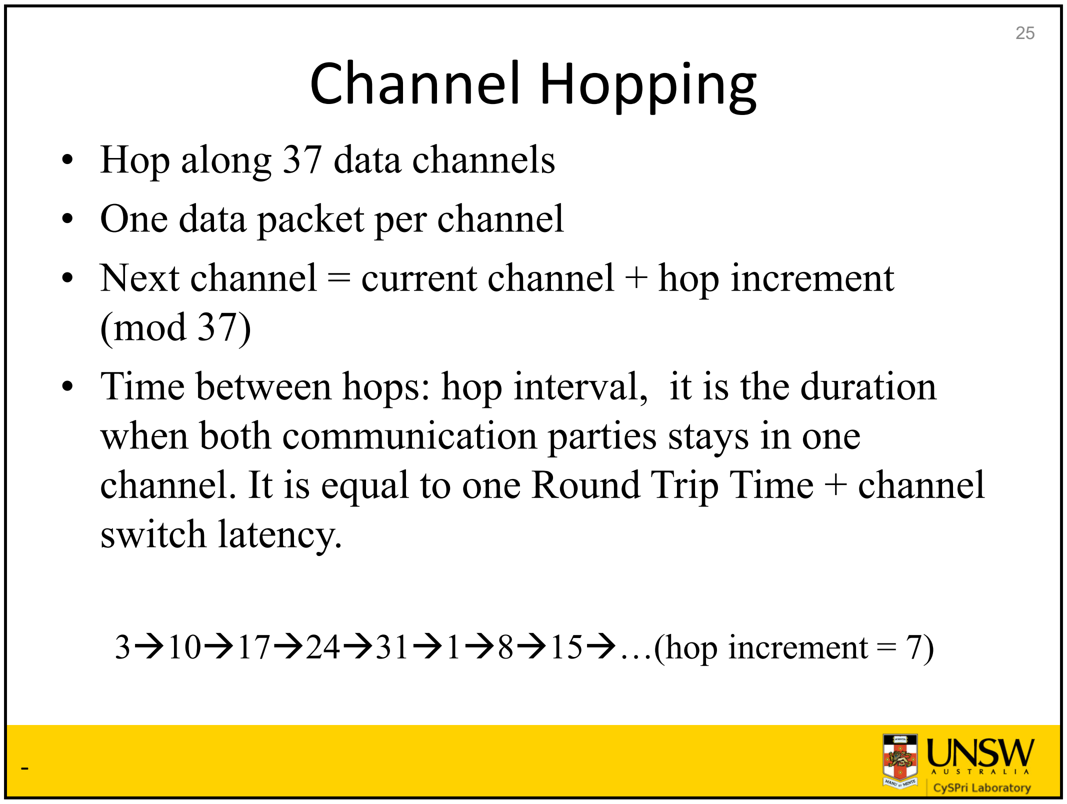
The reason to do this is that by doing channel hopping to different channels for each data packet, the attacker is not able to fully capture

the whole conversation by monitoring a single channel.

The SIG expected that in order for the attacker to intercept all data packets/conversation, the attacker needs to monitor all channels which increases the cost for the attacker.

This channel hopping basically works with fixed channel hopping increments and

we also have vulnerability in this channel hopping mechanism that we will study in the later slides.



Channel Hopping

- Hop along 37 data channels
- One data packet per channel
- Next channel = current channel + hop increment (mod 37)
- Time between hops: hop interval, it is the duration when both communication parties stays in one channel. It is equal to one Round Trip Time + channel switch latency.

3→10→17→24→31→1→8→15→…(hop increment = 7)

-

UNSW AUSTRALIA CySPri Laboratory

One main observation is that you can hop along 37 channels and transmit one data packet per channel in a round robin fashion.

If you go beyond 37, you wrap around and select a channel starting from lowest number (1 if the channel increment is 1).
The next channel is the current channel + hop increment modulo of 37

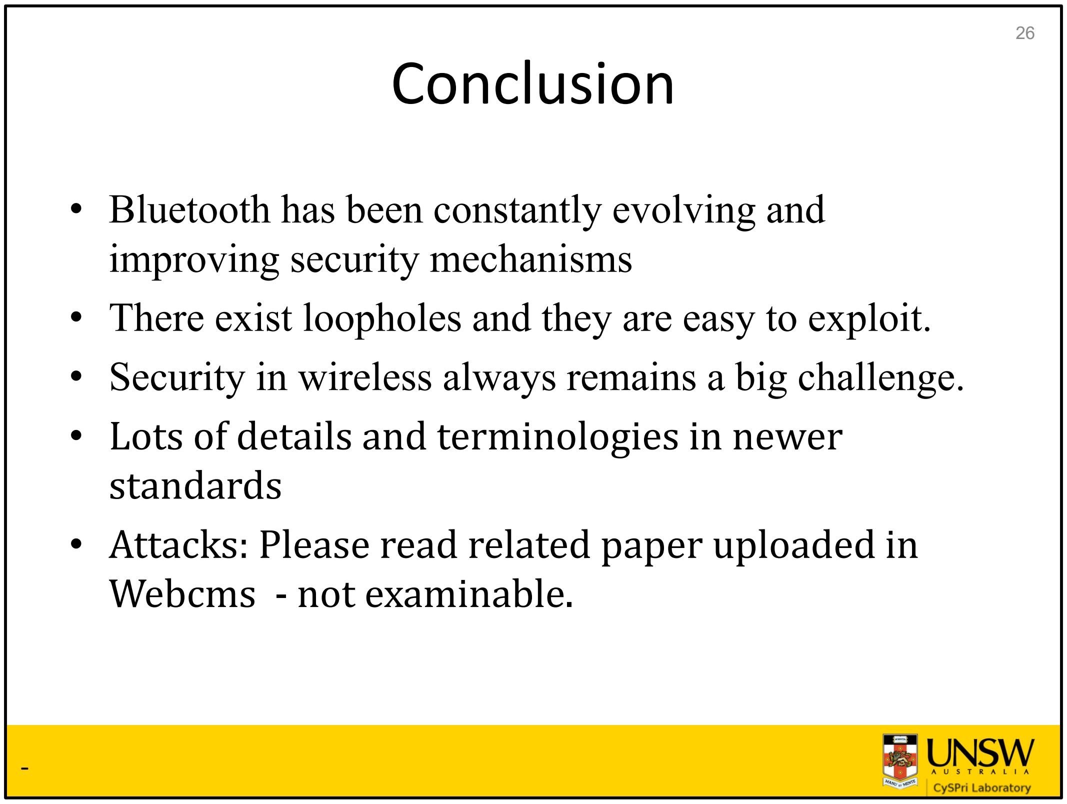
Hop increment is the number of channels skipped while selecting the next channel. You use mod 37 in order to always select a channel value that is <=37.

Hop interval is the time between each hop or the time both devices spend in one channel. = RTT + channel switching latency. Not always the same otherwise it would be very easy to guess.

Communicating parties negotiate Hop increment and Hop interval in connection establishment phase.

Example with hop increment of 7. After you hit 31, add 7 = 38%37 is 1. and so on.

37 is a prime. It is important.



Conclusion

- Bluetooth has been constantly evolving and improving security mechanisms
- There exist loopholes and they are easy to exploit.
- Security in wireless always remains a big challenge.
- Lots of details and terminologies in newer standards
- Attacks: Please read related paper uploaded in Webcms - not examinable.

-

UNSW AUSTRALIA CySPri Laboratory

Bluetooth have some security mechanisms but it is not secure at all.

There exist loopholes everywhere and they are easy to exploit.

So security in Bluetooth is yet to be improved

Unfortunately not many effective methods are in use.

References

- Security Analysis of Bluetooth Technology:https://courses.csail.mit.edu/6.857/2018/project/Filizzola-Fraser-Samsonau-Bluetooth.pdf
- https://community.nxp.com/thread/332191
- HackTool comparison:

  https://duo.com/decipher/bluetooth-hacking-tools-comparison

Deep dive into BLE Security:

https://medium.com/rtone-iot-security/deep-dive-into-bluetooth-le-security-d2301d640bfc

- Good intro to application profile setup and exchange of data (not security specific but good to have this background if interested in this area)
  - https://www.youtube.com/watch?v=pLgnHuGI69s&feature=youtu.be

  - You can play with https://github.com/securing/gattacker on Raspberry Pi. Unfortunately we couldn't give hands on project due to COVID.
- How a malicious app can take advantage of previously established trust (Usenix)
- https://www.youtube.com/watch?v=C1TNuxSCz9Y
- Some newer protocosl have been hard to find, online tutorials are usually very inconsistent, I had tried to read and make sense from multiple sources. If you spot any issues, please report to me.

-

UNSW AUSTRALIA CySPri Laboratory