Bluetooth Security

Mohit Sethi
Ericsson, Finland
Aalto University, Finland

Bluetooth Security - Outline

- Part 1:
  - Bluetooth standard evolution
  - Bluetooth stack and protocols
- Part 2:
  - Pairing and Bonding
  - Privacy with Private addresses
- Part 3:
  - Mesh and secure joining

Bluetooth

- Developed by Ericsson in 1994
  - Named after Danish king Harald Blåtand Gormsen
- Standard specified by the Bluetooth SIG (Special Interest Group) together with Nokia, IBM, Intel, Toshiba etc.
- Major releases
  - Bluetooth 2.0 – 2004
  - Bluetooth 4.0 – 2010
  - Bluetooth 5.0 – 2016
  - Bluetooth Mesh profile – 2017

Bluetooth Standard Evolution

- Bluetooth 2.0 and 2.1 :
  - Lower power consumption and faster data transfer (≈ 3Mbit/s)
  - Secure Simple Pairing made pairing simpler and more secure
- Bluetooth 4.0 and 4.2:
  - Bluetooth Low Energy (BLE) aka Bluetooth Smart
  - Health and fitness trackers with longer battery life
  - IPv6 and improved Internet connectivity
  - Beacons and advertisements
  - Privacy enhancements with better protection against device tracking
- Bluetooth 5.0 – 2016
  - Faster and longer range (≈ 240 meters)
- Bluetooth Mesh profile – 2017
  - Mesh networking with 100s of devices
  - Can work with devices that support Bluetooth 4.2 and higher
  - Original Bluetooth from early 2000s defines piconets (1 master + 7 active slave devices). Most deployments were device-to-device!

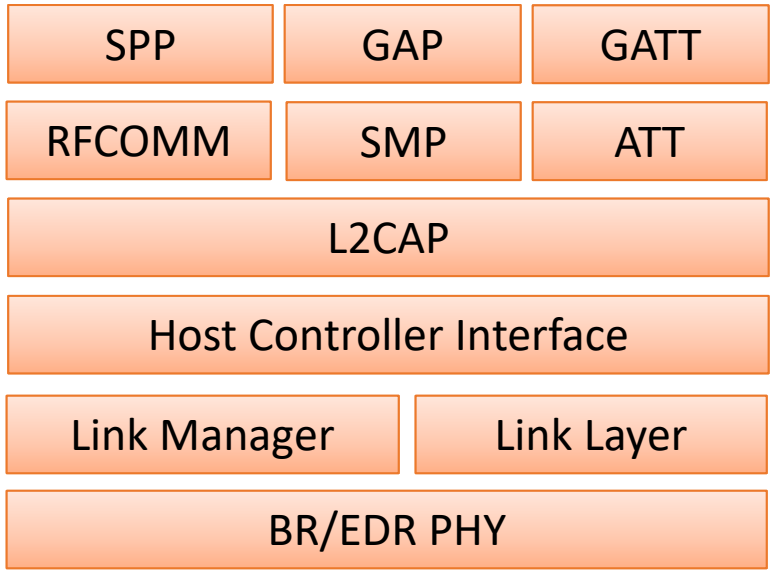
Bluetooth – Protocol Stack

- Bluetooth has two wireless technology systems:
  - Basic Rate (BR) : includes optional enhanced data rate (EDR) and Alternate Media Access Control (AMP) extensions
  - Low Energy (LE): low power, low cost, low data rates

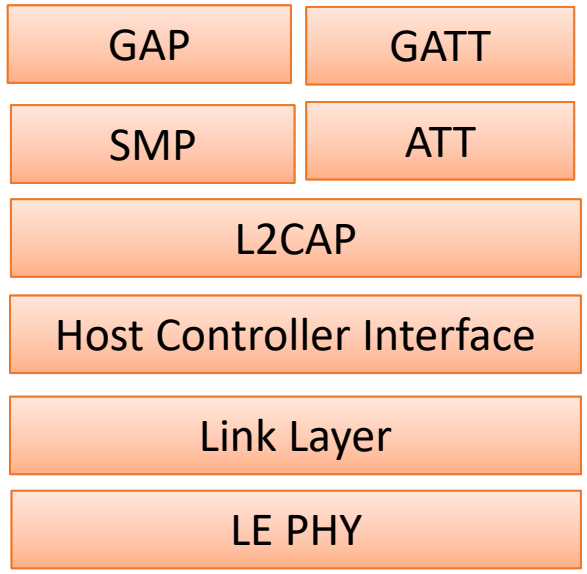
Bluetooth – Protocol Stack

– Bluetooth has two wireless technology systems:

- Basic Rate (BR) : includes optional enhanced data rate (EDR) and Alternate Media Access Control (AMP) extensions
- Low Energy (LE): low power, low cost, low data rates



BR/EDR protocol stack

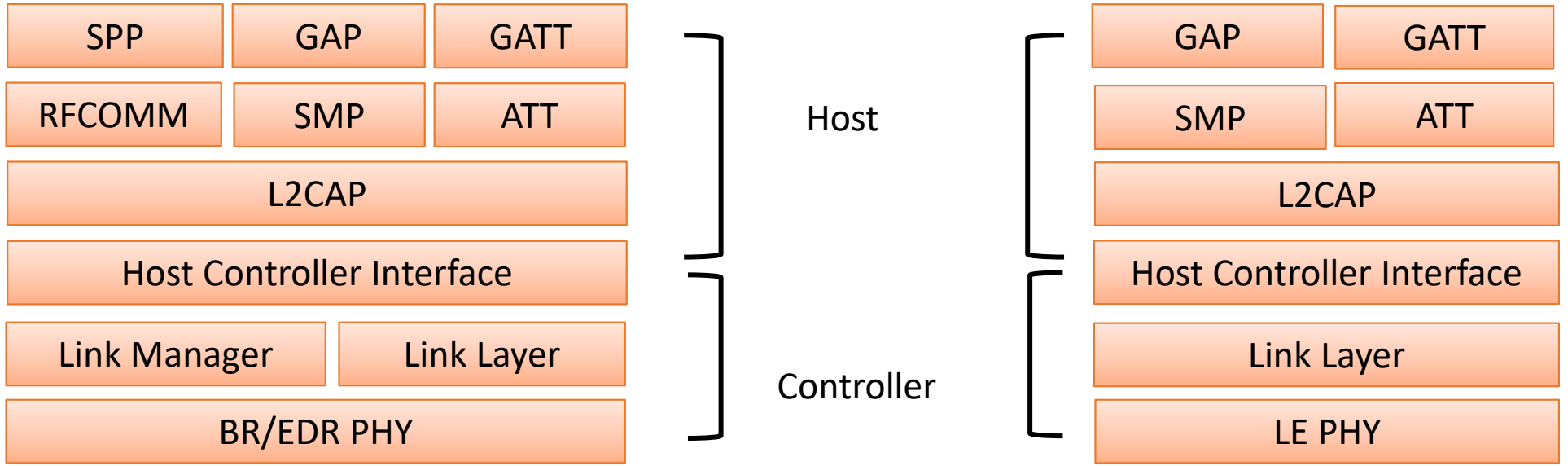


LE protocol stack

Bluetooth – Protocol Stack

– Bluetooth has two wireless technology systems:

- Basic Rate (BR) : includes optional enhanced data rate (EDR) and Alternate Media Access Control (AMP) extensions
- Low Energy (LE): low power, low cost, low data rates



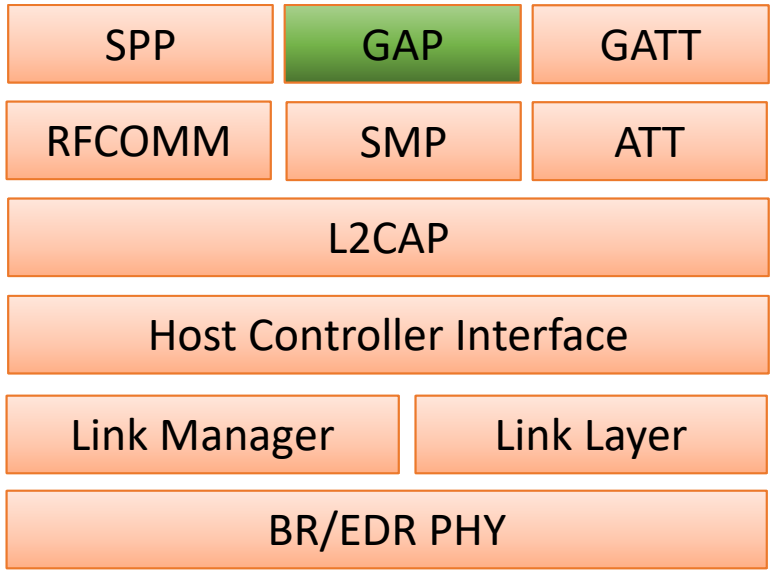
BR/EDR protocol stack

LE protocol stack

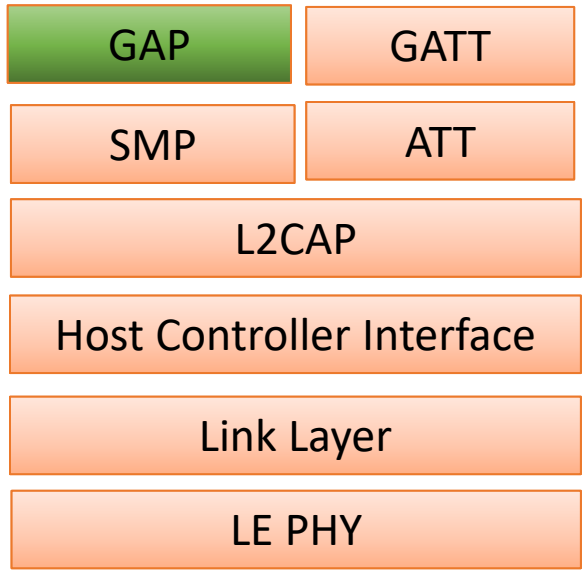
Bluetooth – Protocol Stack

– Bluetooth has two wireless technology systems:

- Basic Rate (BR) : includes optional enhanced data rate (EDR) and Alternate Media Access Control (AMP) extensions
- Low Energy (LE): low power, low cost, low data rates



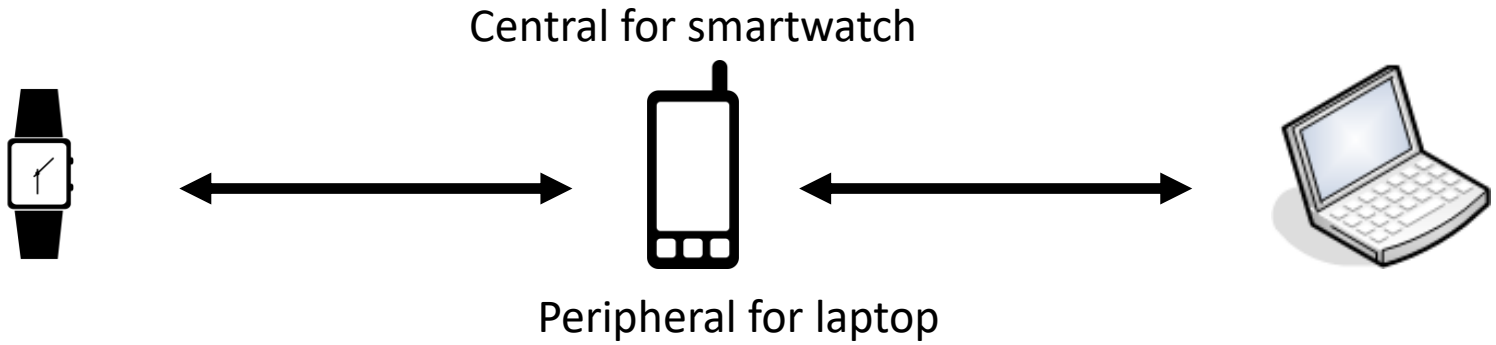
BR/EDR protocol stack



LE protocol stack

Bluetooth – GAP

- Generic Access Profile:
  - Base profile implemented by all Bluetooth devices
  - Defines device discovery, connection establishment, association models, security
- Roles:
  - Single role in BR/EDR – all devices can initiate or accept connections
  - Four roles in LE :
    - Broadcaster: Broadcast device advertises but does not accept connections
    - Observer: Observer listens to advertisements but does not initiate connection
    - Peripheral: Device advertises and accepts a single connection
    - Central: Initiator for all connections and can open multiple connections
  - Simultaneous multiple roles



Bluetooth – GAP

- GAP defines various modes a device can be in:
  - Discoverability modes
    - Non-discoverable/Discoverable/Limited discoverable/General discoverable
  - Connectability modes
    - Non-connectable
  - Bonding modes
    - Non-bondable/Bondable
  - Synchronizable modes
    - Non-synchronizable/Synchronizable
  - Periodic Advertising mode

Bluetooth – Advertising

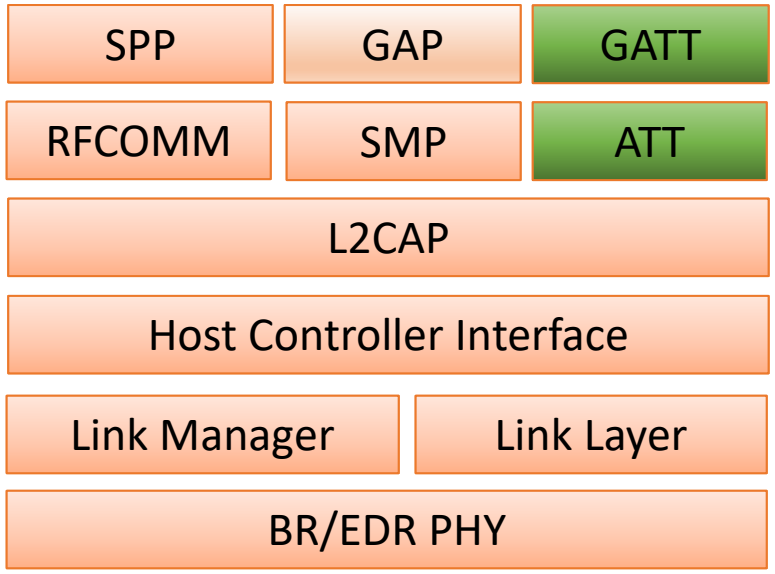
- Advertisements sent by broadcaster or peripheral
- 3 primary channels for advertisements chosen to avoid overlap with WiFi
- Advertisements can be: directed/undirected/connectable/non-connectable/scannable/non-scanable
- 31 bytes of data that includes:
  - Device name
  - Service UUID (Universally Unique Identifier)
- 2 popular standards that build on Bluetooth Advertising
  - Apple iBeacon
  - Google Eddystone
- Used for indoor positioning, asset tracking etc.

Bluetooth – Connections

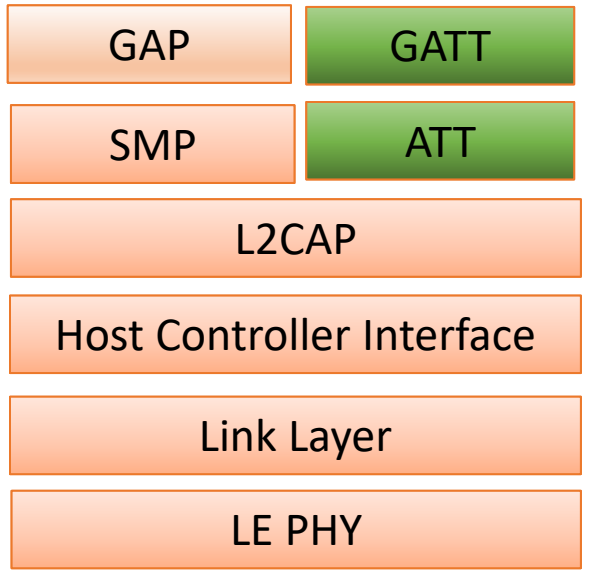
- Advertisements are unidirectional
- Connections enable bidirectional data transfer
- Several phases before connection establishment:
  - Inquiry and name discovery
  - Link establishment
- In LE: Peripheral -> Slave and Central -> Master
- In BR/EDR: initiating device is master and responding device is slave
  - Role switching is possible: initiating device wants to joining an existing piconet
- Connection request -> data exchange -> connection established
- If no existing link key for authentication and encryption, then pairing is necessary.

Bluetooth – GATT

- Generic Attribute (GATT) Profile
  - How is data formatted and exchanged between a client and server
  - Builds on ATT (Attribute Protocol)



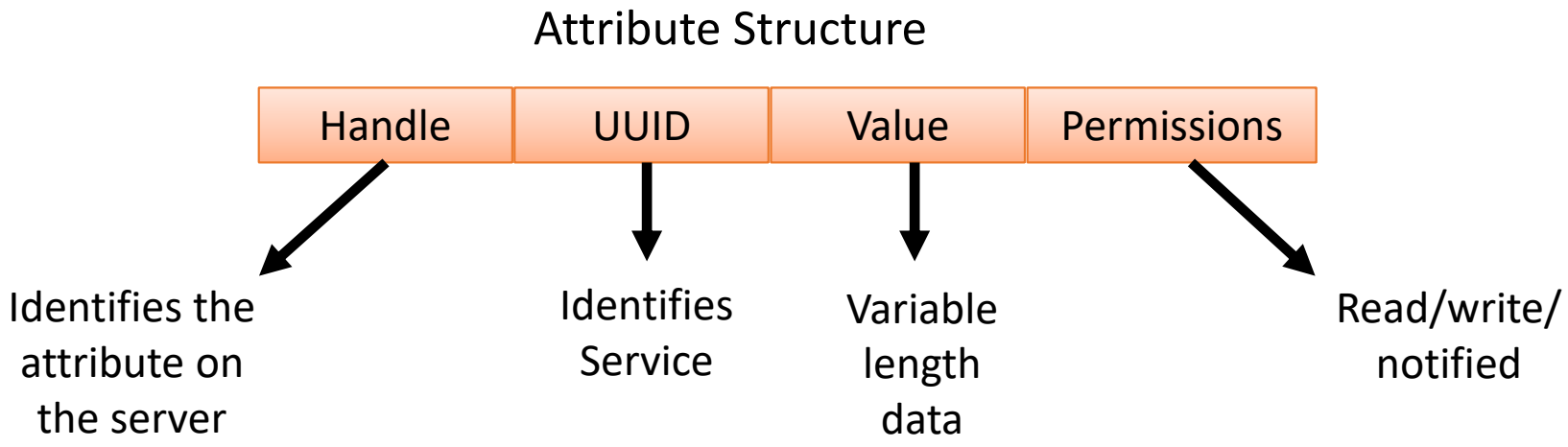
BR/EDR protocol stack



LE protocol stack

Bluetooth – GATT

- Attribute (ATT) protocol:
  - Defines how a server exposes data and clients read/query/commands
  - Data is structured as attributes
  - Client/server role independent of master/slave
  - Devices can be in both client and server role



Bluetooth – GATT

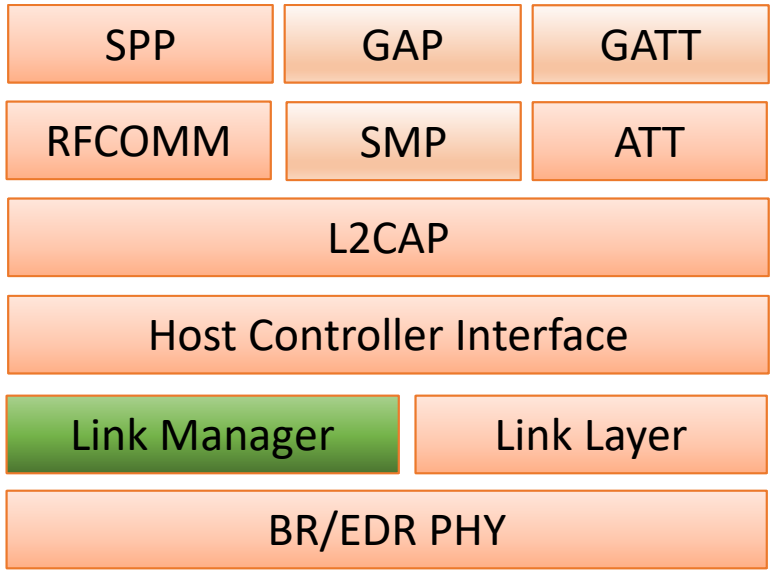
- A service is composed of attributes
  - Characteristic attributes: contain a value that can be read by the client.
  - Can include optional descriptor attributes that help define value it holds (format/unit)
- A profile is composed of services and defines client/server behavior
- Generic Attribute (GATT) profile:
  - defines how to use ATT for discovery, reading, writing, and obtaining indications
  - reference framework for other GATT-based profiles: SIG defined or custom

Bluetooth Security - Outline

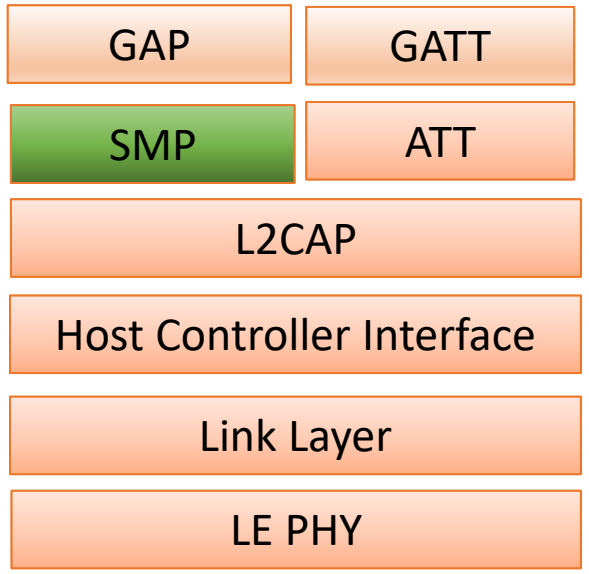
- Part 1:
  - Bluetooth standard evolution
  - Bluetooth stack and protocols
- Part 2:
  - Pairing and Bonding
  - Privacy with Private addresses
- Part 3:
  - Mesh and secure joining

Bluetooth – Pairing

- Pairing in BR/EDR vs. BLE:
  - Security Manager: defines protocols for managing pairing, authentication, and encryption



BR/EDR protocol stack



LE protocol stack

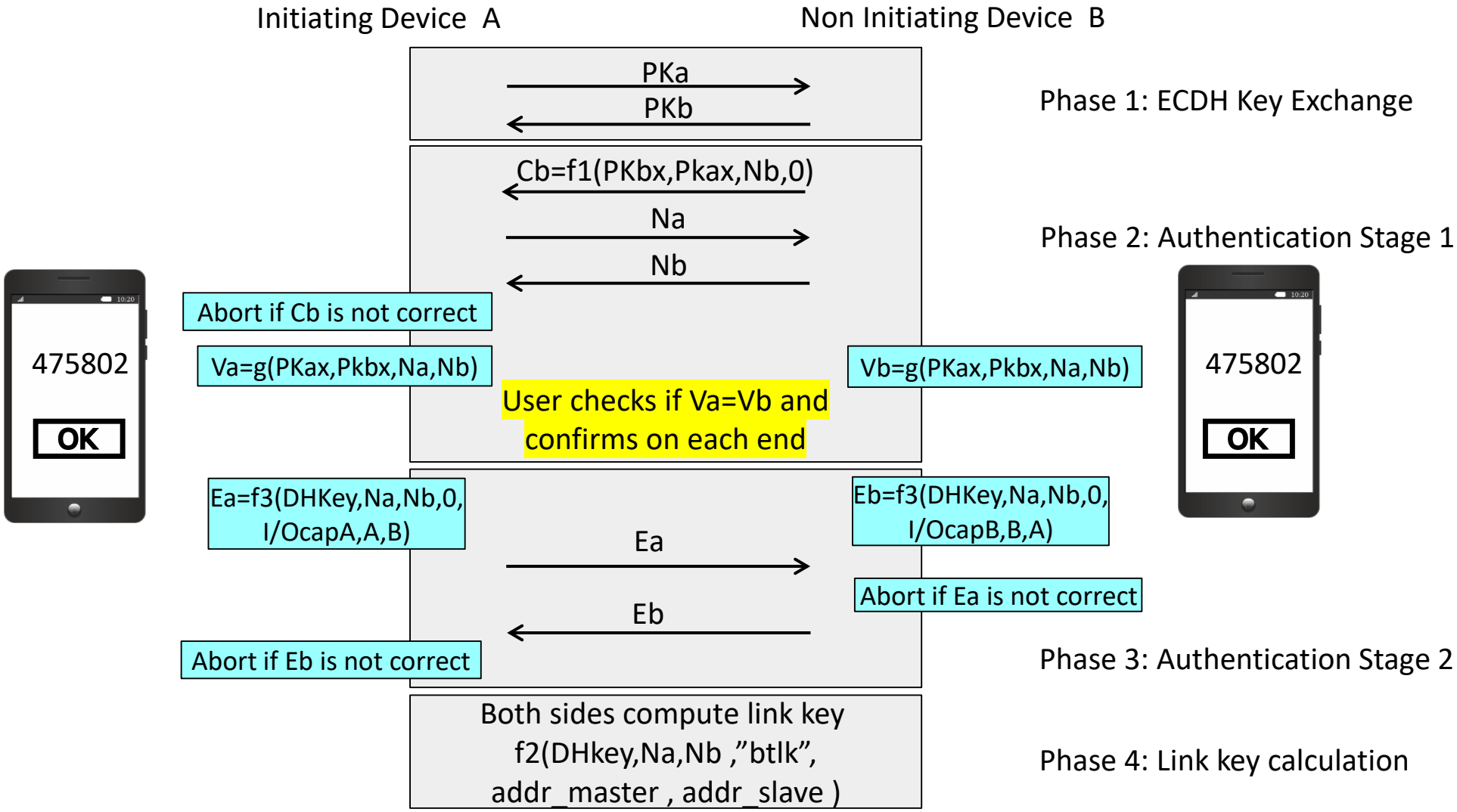
Bluetooth – Pairing

- Many versions and names
  - BR/EDR:
    - Version 2.1 – Secure Simple Pairing
    - Version 4.2 – Secure connections
  - LE:
    - Version 4.0/4.1 – called LE legacy pairing (based on SSP with modifications)
    - Version 4.2 – Secure connections
  - Most devices support old versions for interoperability => Susceptible to attacks

Bluetooth – Pairing

- Exchange I/O capabilities – decides association model:
  - Just works – protection only from passive attacker
  - Numeric Comparison – short 6-digit confirmation values show
  - Out-of-band – message sent over NFC for example
  - Passkey entry – user enters passkey into two devices being paired
- Phases:
  - Exchange of ECDH public keys
  - Authentication stage 1 and 2
  - Link-key calculation

Bluetooth – Pairing with Numeric Comparison

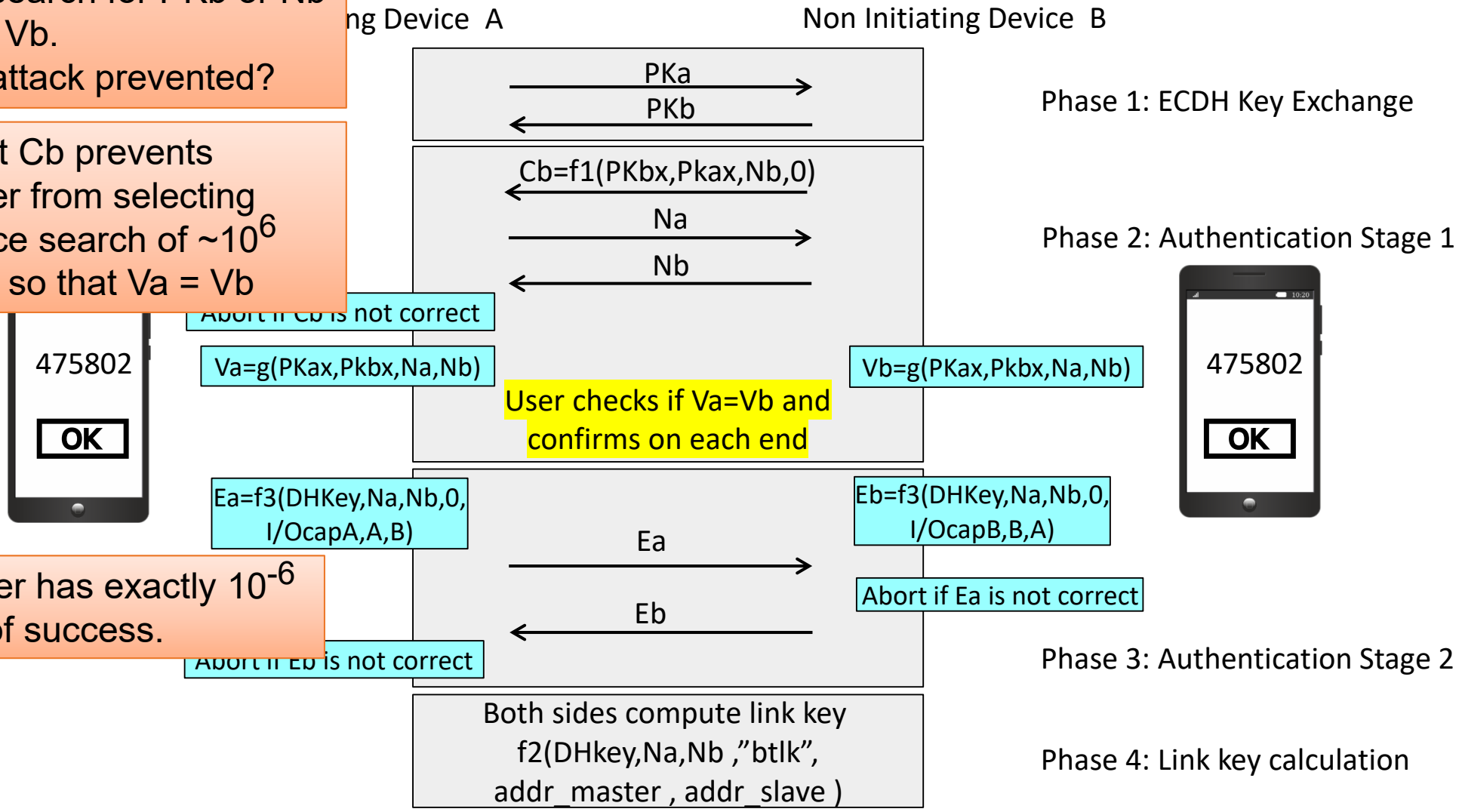


Pairing with Numeric Comparison

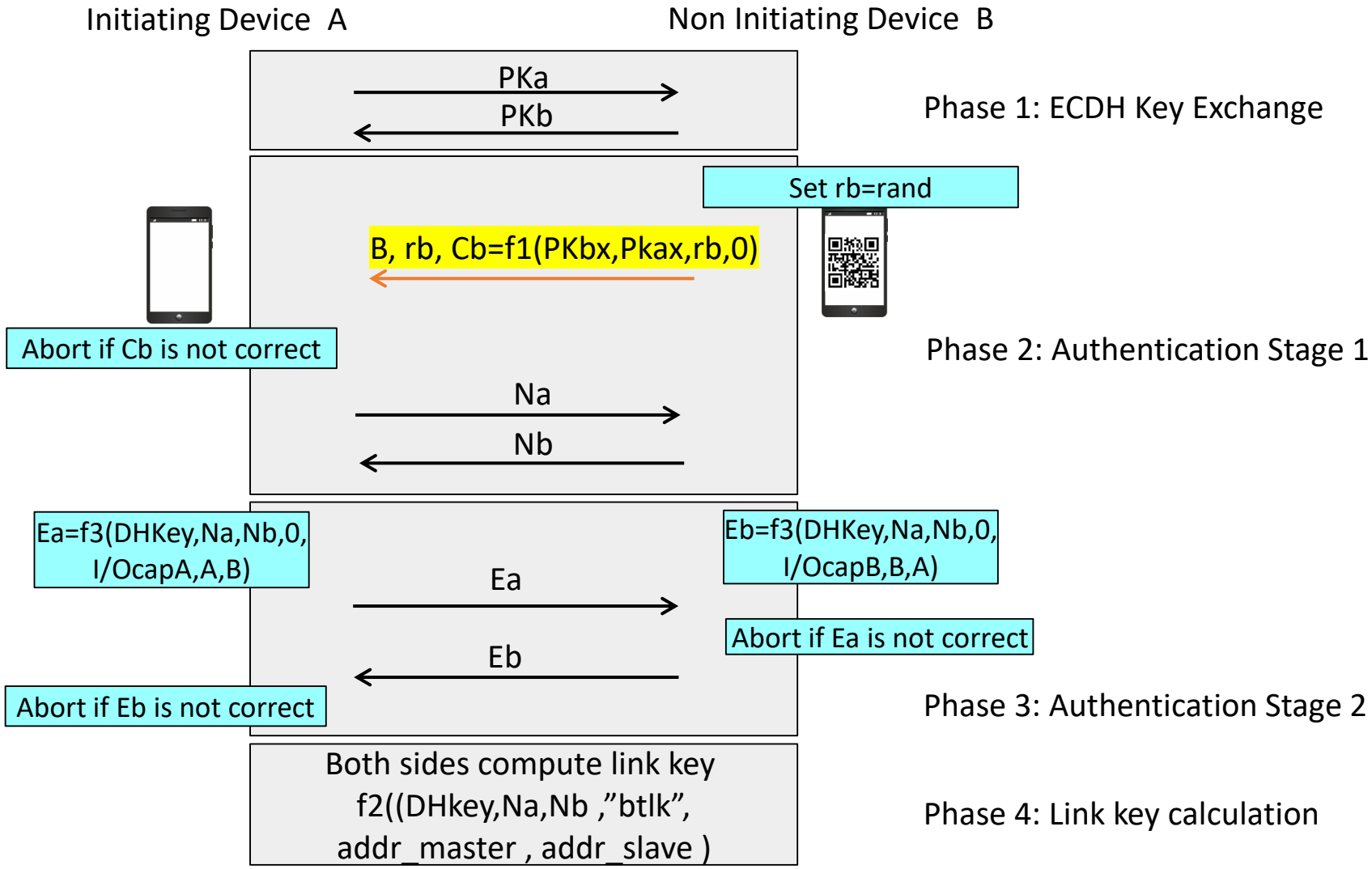
MitM attack will fail if Va ≠ Vb.
But 6-digit hash is too short, and attacker could do a brute-force search for PKb or Nb so that Va = Vb.
How is this attack prevented?

Commitment Cb prevents MitM attacker from selecting (by bure-force search of ~10^6 values) PKb so that Va = Vb

MitM attacker has exactly 10^{-6} probability of success.



Bluetooth – Pairing with OOB



Bluetooth – Bonding and LMP authentication

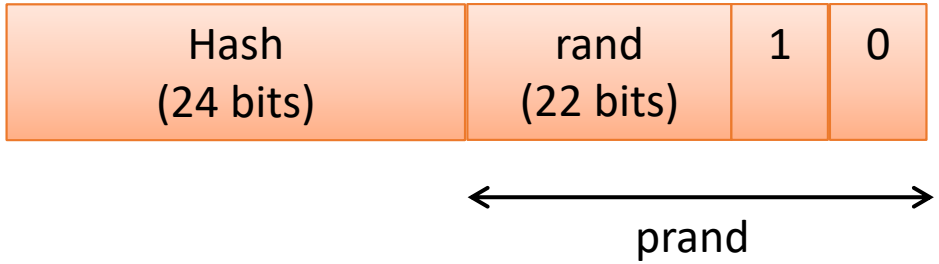
- Pairing results in generation of link-key
- Bonding stores a LTK after pairing for establishing future connections without pairing
  - Bonding in LE also distributes Identity Resolving Key (IRK) and Connection Signature Resolving Key (CSRK)
- LMP authentication – mutual authentication to confirm that both have same link key
  - Secure authentication: exchange random numbers, compute hash with link-key and random numbers, send SRES (expected response). If SRES match with locally computed values, link-key authenticated and fresh keys generated

Bluetooth LE - Privacy

- 4 types of address in LE
  - Public address: Fixed, global (registration with IEEE), never changes
  - Random addresses:
    - Static address: Can change at bootup but static during runtime
    - Resolvable private address: Optional. Changes periodically (≈ 15 min): generated using IRK and a random number. Can be resolved by other devices which have bonded earlier
    - Non-resolvable private address: Optional. Also changes periodically. No one else can resolve such addresses. Used for privacy in beacons or Covid-19 tracing

Bluetooth LE - Privacy

- Resolvable private address:
  - Generation: hash = ah(IRK, prand) concatenated with prand



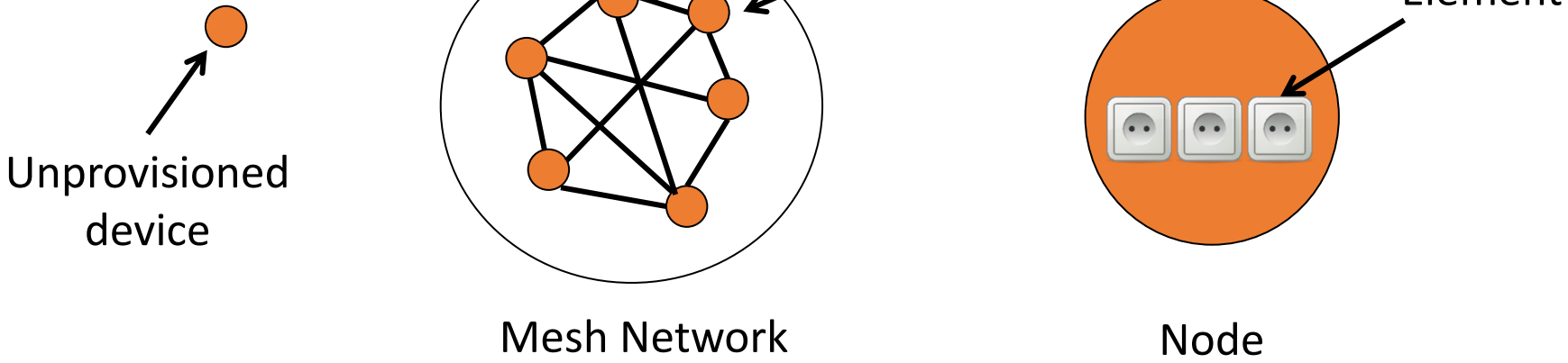
  - Resolution: Receiver uses the prand with all IRKs in its database to lookup the peer device

Bluetooth Security - Outline

- Part 1:
  - Bluetooth standard evolution
  - Bluetooth stack and protocols
- Part 2:
  - Pairing and Bonding
  - Privacy with Private addresses
- Part 3:
  - Mesh and secure joining

Bluetooth Mesh

- Added in 2017 to support many-to-many topology
  - Builds on top LE and is a profile
  - Utilize advertising: no connections are setup
  - Low-power battery-operated nodes can be supported with Proxy



Bluetooth Mesh

- State: value representing the condition of an element
- Properties: add context to state
- Messages sent in a mesh network with managed flooding
  - Unacknowledged : sent when no response required
  - Acknowledged : message is acknowledged with a response (containing data)
  - get: get state of a peer element
  - set: set state of a peer element
  - status: sent in response to a get/acknowledged set or time intervals

Bluetooth Mesh

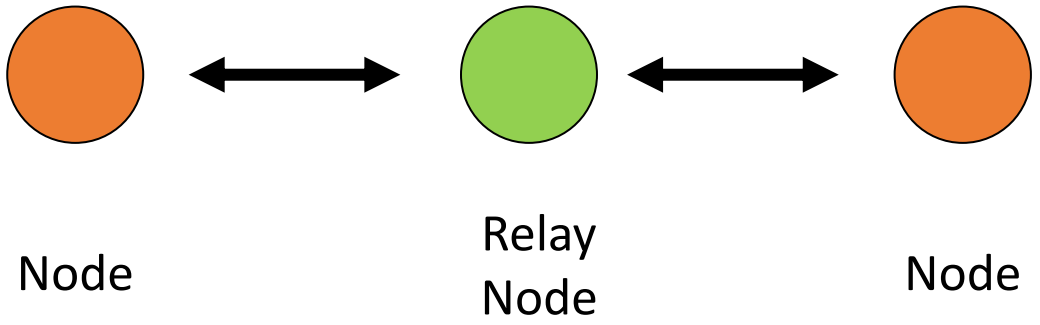
- Address: messages sent in a mesh must have source and destination
  - Unicast address: message sent to a particular node
  - Group address: identifies a group of nodes (SIG fixed or dynamically). Several group addresses within a mesh network
  - Virtual address: multicast address for multiple elements on one or more nodes
- Bluetooth mesh uses publish/subscribe paradigm

Bluetooth Mesh

- Model: defines functionality. Applications are defined with models instead of profiles:
  - Server model: states, state transitions, messages which element can send or receive
  - Client model: defines messages such as get/set/status sent to server
  - Control model: contain client and server model and defines interactions with other devices containing client/server models
- Scene: collection of states. One action to set multiple states of different nodes

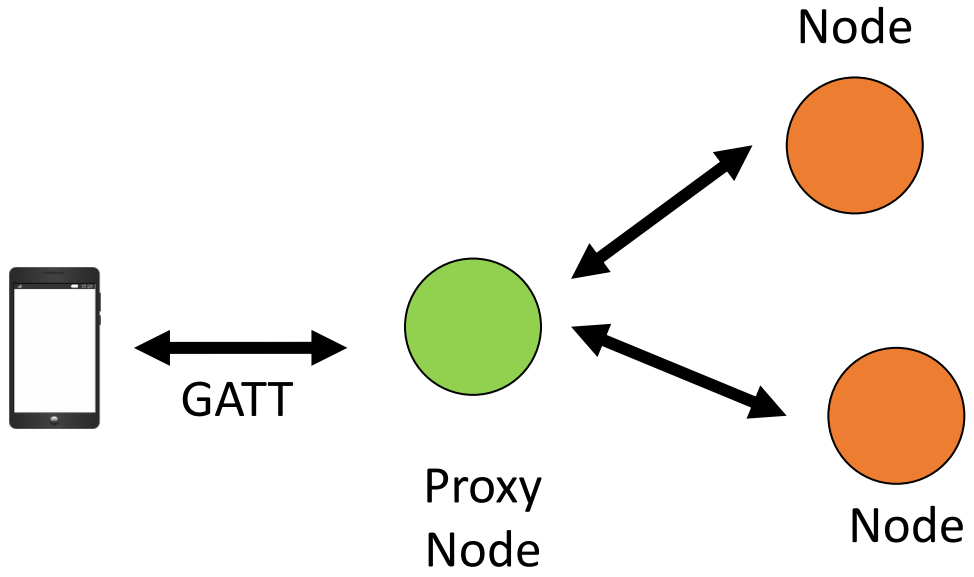
Bluetooth Mesh

- Nodes can support optional functionality:
  - Relay: able to forward messages that are broadcast. Necessary for messages to traverse the network. Time to live (TTL) controls relaying behavior.



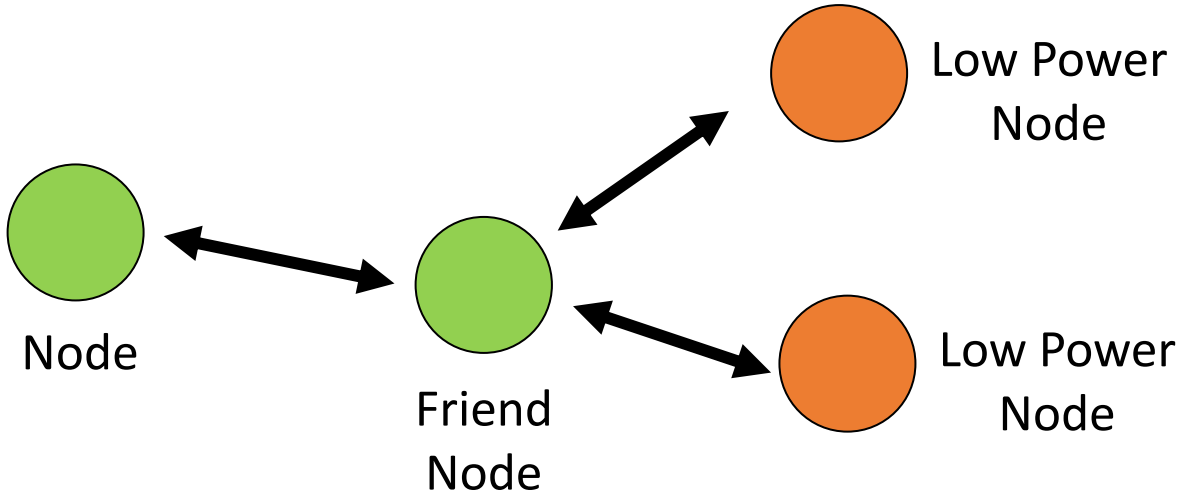
Bluetooth Mesh

- Nodes can support optional functionality:
  - Proxy: exposes a proxy service with GATT. Allows non-mesh device to interact with the mesh network



Bluetooth Mesh

- Nodes can support optional functionality:
  - Low-power: resource-constrained. Sleep most of the time and send data on wakeup. Polls friend nodes on wakeup
  - Friend: not resource-constrained. Caches messages for low-power nodes



Bluetooth Mesh - Provisioning

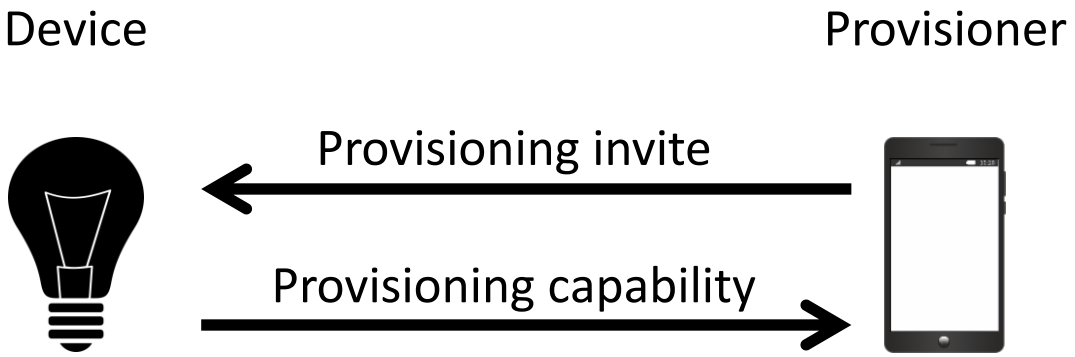
- Provisioning: adding a new device to the mesh network
- Provisioner: smartphone for provisioning new devices
  - Beaconing: unprovisioned device sends advertisements to indicate that it is waiting to be provisioned

Unprovisioned device

Provisioner

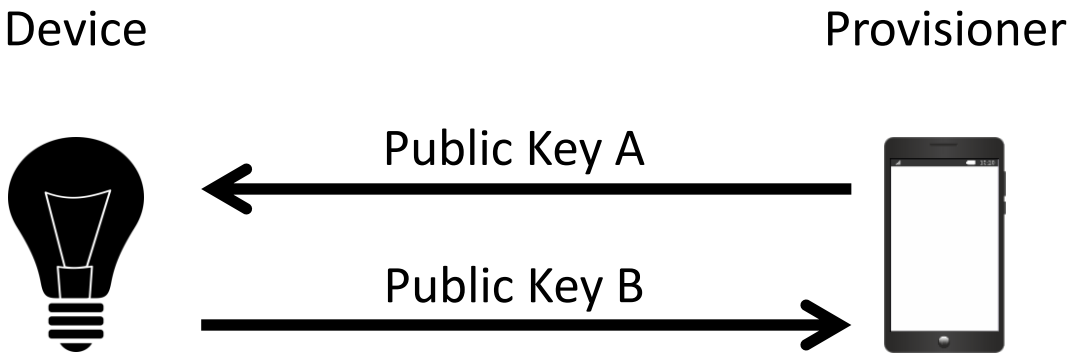
Bluetooth Mesh - Provisioning

- Provisioning: adding a new device to the mesh network
- Provisioner: smartphone for provisioning new devices
  - Invitation: provisioner discovers new device via beacon and sends an invitation. New device responds with provisioning capabilities (including elements, security algorithms, I/O capability etc.)



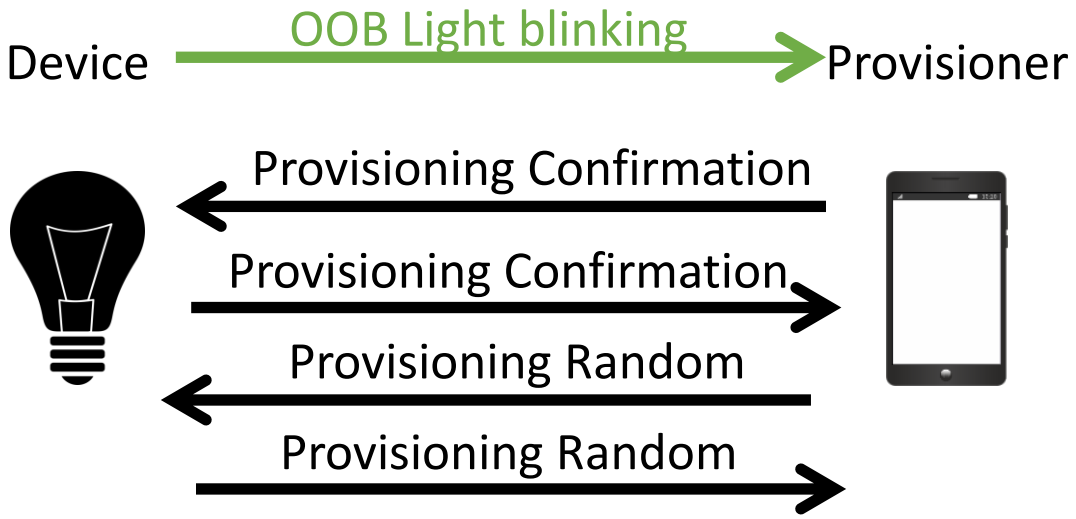
Bluetooth Mesh - Provisioning

- Provisioning: adding a new device to the mesh network
- Provisioner: smartphone for provisioning new devices
  - Public key exchange: ECDH key exchange with fresh keys (or static for device, i.e. printed on a sticker)



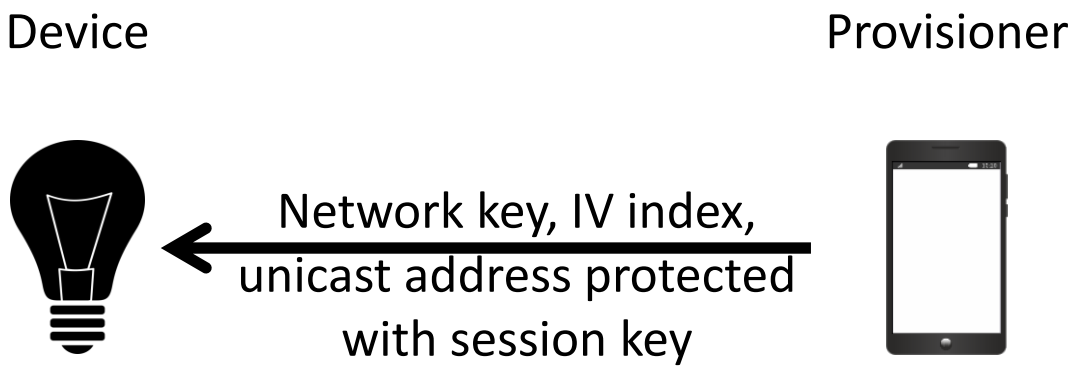
Bluetooth Mesh - Provisioning

- Provisioning: adding a new device to the mesh network
- Provisioner: smartphone for provisioning new devices
  - Authentication: Device or Provisioner generate and show a random number (as blinking LED, audio etc.) that is input on the other side. Both send commitments with random number and reveal random numbers after. Generate session key.



Bluetooth Mesh - Provisioning

- Provisioning: adding a new device to the mesh network
- Provisioner: smartphone for provisioning new devices
  - Distribution of provisioning data : Provisioner sends data: network key, IV index, unicast address assigned etc.



Bluetooth Mesh - Security

- Key Separation:
  - Application Key: shared by a subset of nodes in the mesh network, e.g., light bulbs and switches
  - Device Key: shared between device and provisioner for sending network information
  - Network Key: shared by all nodes in the mesh network. Used for deriving network encryption key and privacy key

- Privacy key derived from network key
- Network header including source address obfuscated with this key