ECEN 5823-001

Internet of Things Embedded Firmware

Lecture #22

08 November 2018

Agenda

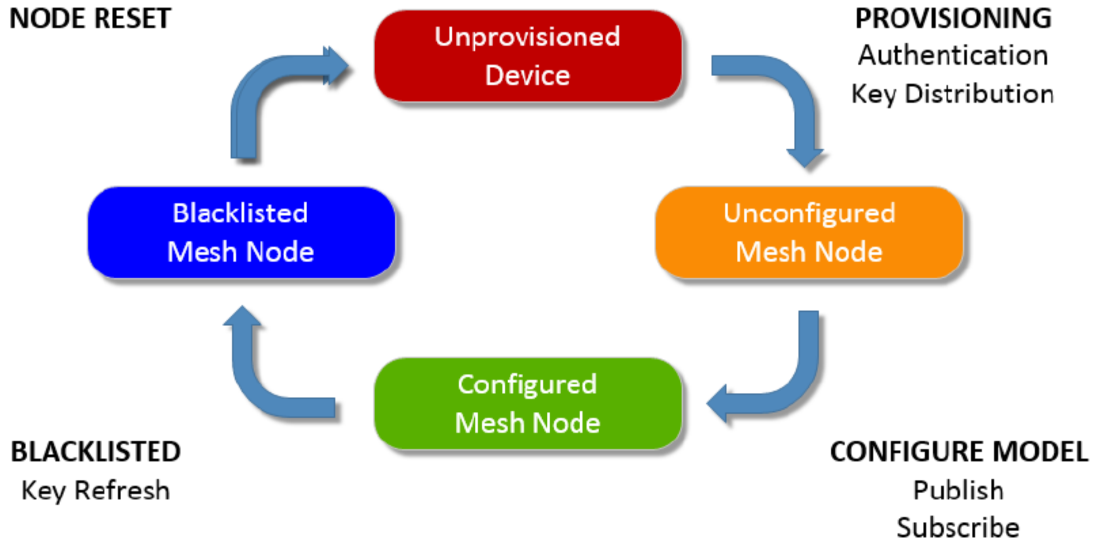
- Class Announcements
- Bluetooth Mesh

Class Announcements

- Quiz 10 due on Sunday, November 11th, at 11:59pm
- Course Project Proposal due Sunday, November 11th, at 11:59pm

Bluetooth Mesh - Lifecycle

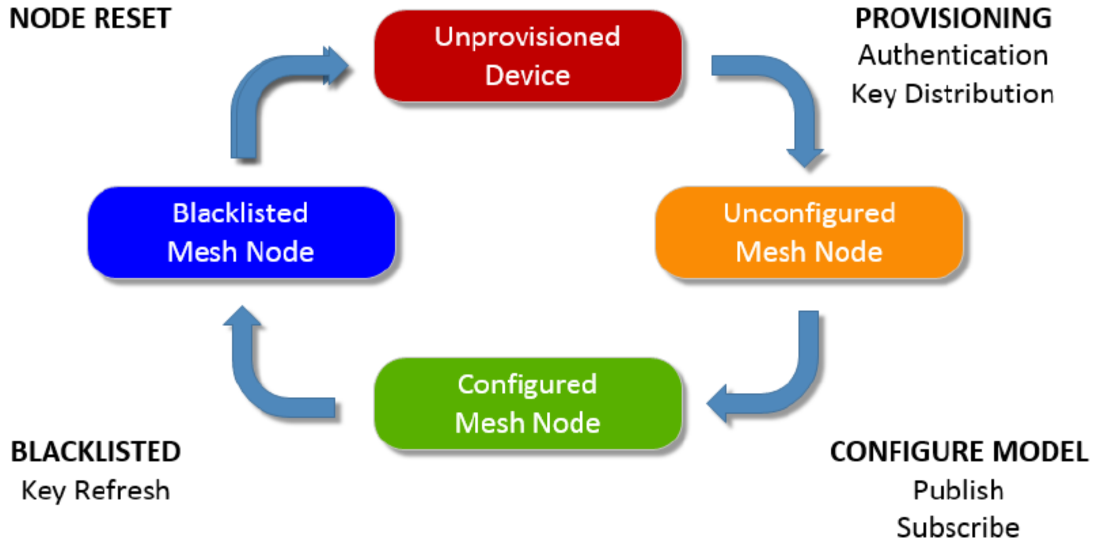
- First the Provisioner must detect an unprovisioned device and establish a provisioning bearer
- Then the Provisioner and the device use the Elliptic Curve Diffie-Hellman (ECDH) anonymous key agreement protocol to establish a shared secret
- After this the device needs to be authenticated using Out-of-Band (OOB) information
- Once authentication has succeeded both the Provisioner and the device together will generate a session key based on the shared secret
- The session key is then used to encrypt and authenticate provisioning data



Bluetooth mesh device lifecycle

Bluetooth Mesh - Lifecycle

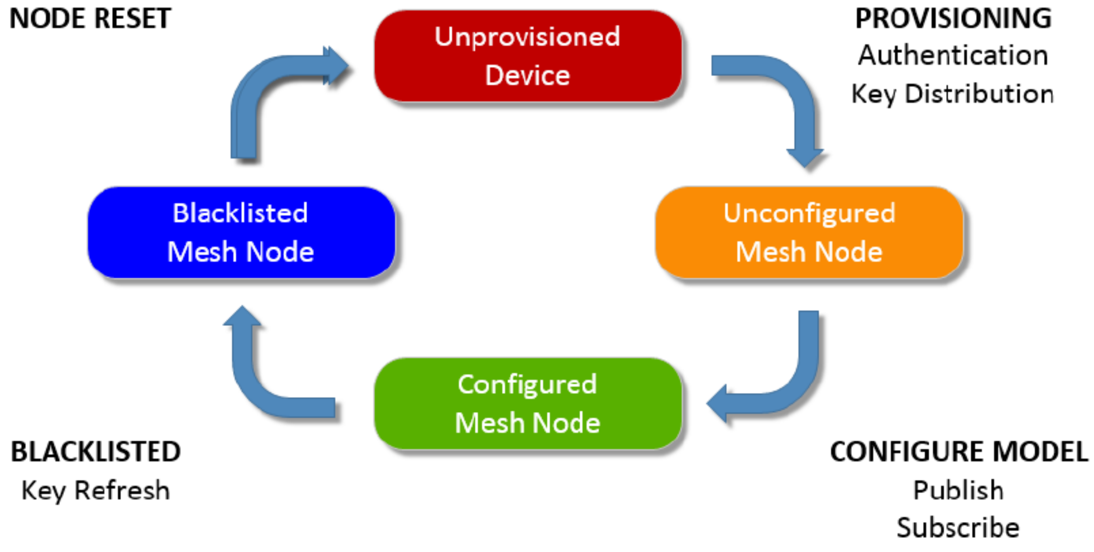
- All devices have a configuration block which includes information such as Company ID, Product ID and Model Information
- The Provisioner needs to read the data contained in the configuration block after which the Provisioner can proceed with the configuration according to the capabilities and Models present on the device
- These keys also need to be bound to the appropriate Network Keys
- Finally, the Provisioner needs to configure how the node publishes information and which information it needs to subscribe to



Bluetooth mesh device lifecycle

Bluetooth Mesh - Lifecycle

- If a node is removed from the mesh the keys of all remaining nodes need to be changed to prevent the possibility of a "trash-can attack"
- Nodes which may have been compromised can be blacklisted to prevent their use in the mesh
- Network Keys, Application Keys and all confidential data derived from them can be refreshed using a Key Refresh procedure



Bluetooth mesh device lifecycle

Bluetooth Mesh: States

- States are values representing conditions of node elements
  - When an element is exposing a state (value) representing the condition of the element that element is called a Server and implements the Server Model
  - When accessing a state of an node element the accessing node is called a Client, which implements the Client Model
- What are three different methods for controlling element states?
  - A state-changing message which is sent to a Server
  - An asynchronous event from the scheduler
  - A local event such as pressing a button

Bluetooth Mesh: Scenes

- Bluetooth Mesh stores states of elements as a Scene
  - A Scene register is a 17-element, zero-based, indexed array of 16-bit values
  - The “handle” of the state is the index into the scene array to state storage containers in which the associated state information is stored

Bluetooth Mesh: Initial, Present and Target States

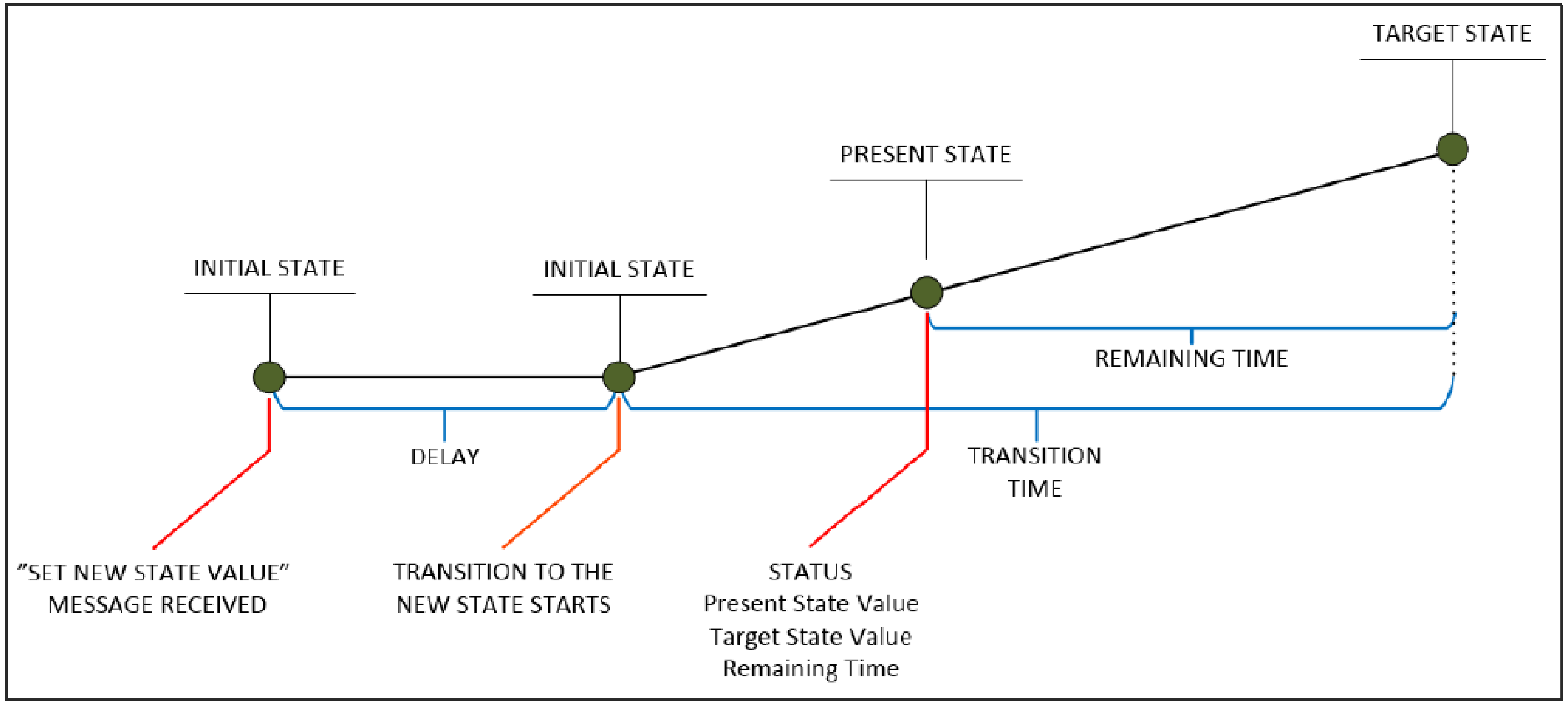
- States can change instantaneously or within a certain time period, this time period being called Transition Time
- The change begins from an Initial State ending to the Target State
- The Present State is the current state of an element.
- Bluetooth mesh defines a flexible method for defining how states can be changed using several different parameters and has a selection of specially optimized commands for various applications such as lighting control, sensors etc.

Bluetooth Mesh: Initial, Present and Target States

- The user can utilize a default transition time or alternatively define the time by changing the appropriate field in the related message
- One can also add a Delay which defines the length of time between the receipt of the message and the actual start of the state transition
  - This is useful for synchronizing actions with multiple receivers

Bluetooth Mesh: Initial, Present and Target States

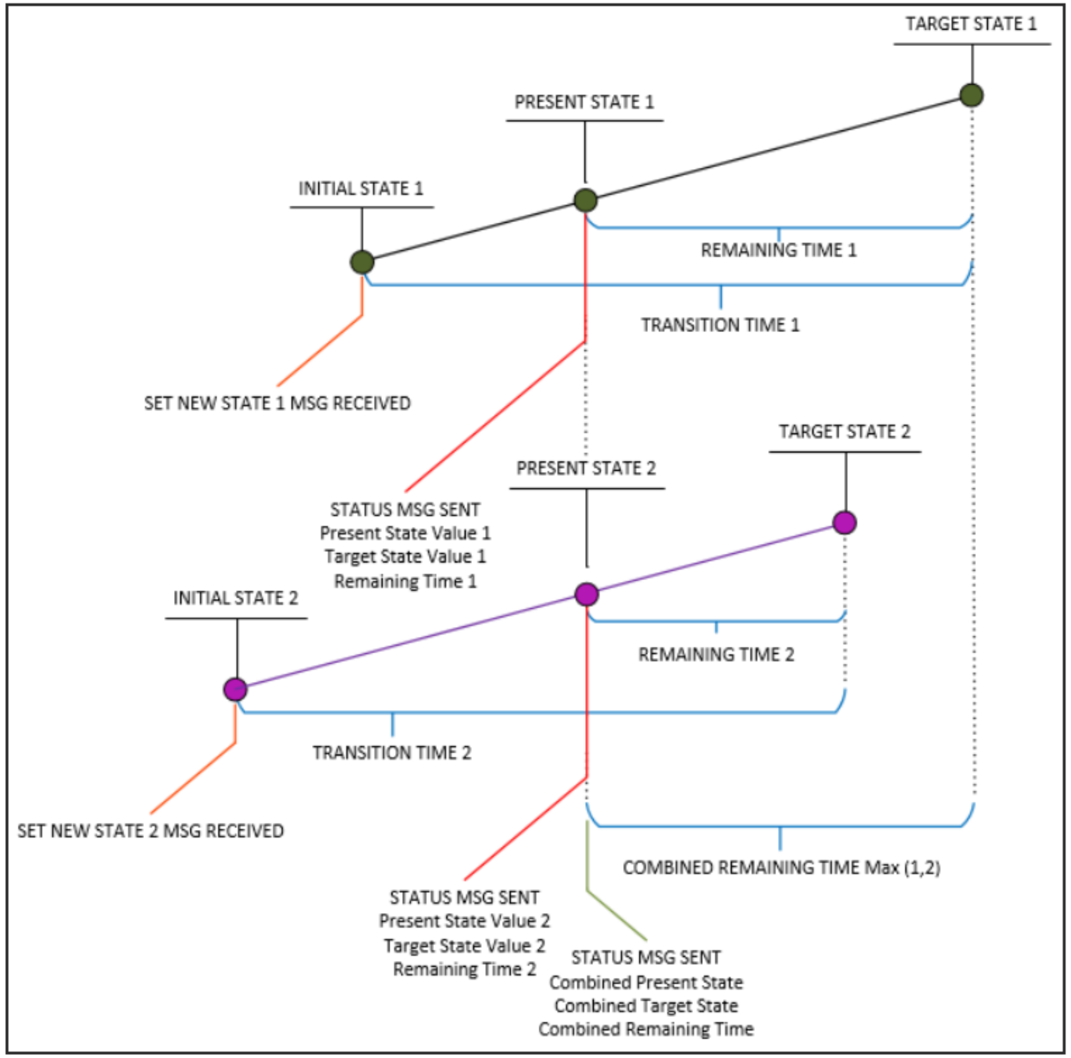
- Some states messages can be used to report the state of the transition, either the present state, or including both the present and target states together with the remaining time when the target state will be reached



State change using delay and transition time with related status info

Bluetooth Mesh: Initial, Present and Target States

- States can be one-dimensional or multi-dimensional
- For example, in light control applications one might be interested in controlling the hue, saturation and lightness



Bluetooth Mesh: Bound States

- States can also be bound together in which case a change in any of the states bound to each other will result in a change in the other states bound together
  - An example from lighting applications would be a case in which the light is dimmed to zero which will then cause the On/Off state to change to off-state
- It should be noted that bound states do not need to be from the same model nor do they have to be related to the same element
- Binding can be defined as bi-directional or uni-directional
  - This gives a lot of freedom for creative application development.

Bluetooth Mesh: Composite States

- To make the detection of changes in states as flexible as possible the user can group together multiple states
  - These kind of groups are called Composite States
  - As an example, consider the Light Lightness state, which is actually composed of three states name the Light Lightness Actual State, the Light Lightness Last State and the Light Lightness Default State
- If an indication of a change in any of the listed three states is required one can just refer to the Light Lightness state instead of referring to all three states separately.

Bluetooth Mesh: Scenes

- Bluetooth mesh includes a useful feature called Scene with which it is possible to recall a set of States for a group of Nodes
  - A practical example for a home application utilizing Scenes would be a light control by which a parent can control the lights in all of the children's bedrooms to the desired level
  - For example, in the evening the user could set all the lights in the bedrooms off except for a dimmed comfort light which slowly dims and finally switches off completely

Bluetooth Mesh: Messaging

- All communication in a Bluetooth mesh network is based on sending messages which operate on states
  - All states have a defined set of messages supported by a Server
  - Clients can use these same messages for requesting the value of state or to change the value of a state
- Information on states and/or the changing of states may be sent by a Server without any external request
- Messages consist of an opcode and related parameters
  - Single octet size messages are used with special messages when maximum payload size is required
  - Two octet messages are standard messages
  - Three octet messages are reserved for vendor-specific messages.

Bluetooth Mesh: Messaging

- Messages are either acknowledged or unacknowledged
  - Acknowledged messages require a response
  - Unacknowledged messages do not require a response
- Set, Clear, Recall, and Store messages can be either acknowledged or unacknowledged
  - Even though their semantics are the same their opcodes are different

Bluetooth Mesh: Messaging

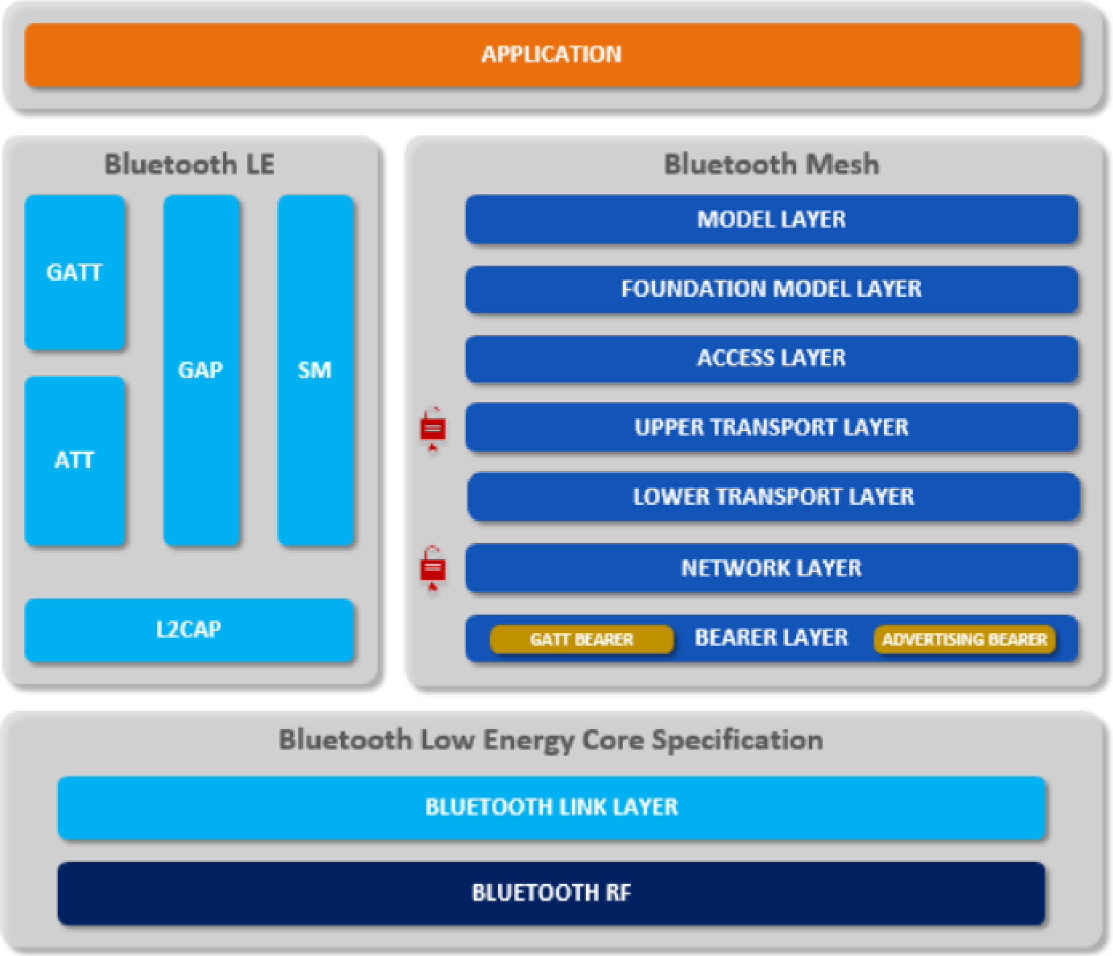
- Broadcast messages to all nodes listening at any time are based on the Advertising Bearer using BLE non-connectable advertising
- Messages are sent using broadcast channels
- Point-to-point messages are based on the GATT Bearer using BLE connections and standard GATT service
- Any node may support either or both bearers

Bluetooth Mesh: Segmentation & Reassembly

- The Transport Layer defines the total message size
  - Segmentation and Reassembly (SAR) can be used
  - In an optimal case only a single segment is used
  - With Segmentation and Reassembly a maximum message of 32 segments which translates to a maximum message of 384 octets
- With a single octet opcode this means that 379 octets are available for parameters, and with two or three octets 378 or 377 octets
- Bluetooth mesh messages are sent inside models using opcodes and element addresses

Bluetooth Mesh: Segmentation & Reassembly

- With large PDU's, Protocol Data Unit, exceeding the maximum size the Lower Transport Layer of the transmitting node takes the PDU from the Upper Transport Layer and segments it by splitting the PDU into several Transport PDU's
- When the receiving node receives these PDU's the Lower Transport Layer reassembles the original PDU and then passes it upward to the Upper Transport Layer in the receiving node



Bluetooth mesh architecture (right) with Bluetooth LE architecture (left)

Bluetooth Mesh: Segmentation & Reassembly

- Segments are identified by using a Segment Offset Number (SegO) where the first segment has the value 0, the second has the value 1 and so on
- The offset of the last segment needs to be also defined with the Last Segment Offset Number (SegN).
- For example, if the message consists of four segments the value to use for SegN would be 3
- In addition each message is identified by using a Sequence Authentication (SeqAuth) value which is used to encrypt or authenticate the access message
- The normal procedure would be to use the Sequence Number (SEQ) of segment number 0

Bluetooth Mesh: Segmentation & Reassembly

- There are four different message types defined for segmentation and reassembly

	MESSAGE TYPE	
SEGMENTATION TYPE	Control	Access
Unsegmented	Unsegmented Control Message	Unsegmented Access Message
Segmented	Segmented Control Message	Segmented Access Message

Bluetooth Mesh: Segmentation & Reassembly

- With segmented messages Bluetooth mesh defines a Segment Acknowledgement procedure which enables retransmission of segments which have not been received during the first try
- The mechanism allows indication of missed segments to make the source send only missed segments
- In case the receiving node is receiving messages on behalf of a defined Low Power node in a Friendship setup an OBO (On Behalf Of) acknowledgement message is used
- The Friend node which buffers the received message replies with the OBO on behalf of the Low Power node

Bluetooth Mesh: Addressing

- IoT use cases require that the user is able:
  - to control a single node
  - a group of nodes
  - Or, all of the nodes simultaneously
- This is provided in Bluetooth mesh by using:
  - unicast
  - virtual and group addressing
  - A special value representing an unassigned address not used in messages is also available
  - An element is an addressable entity

Bluetooth Mesh: Addressing

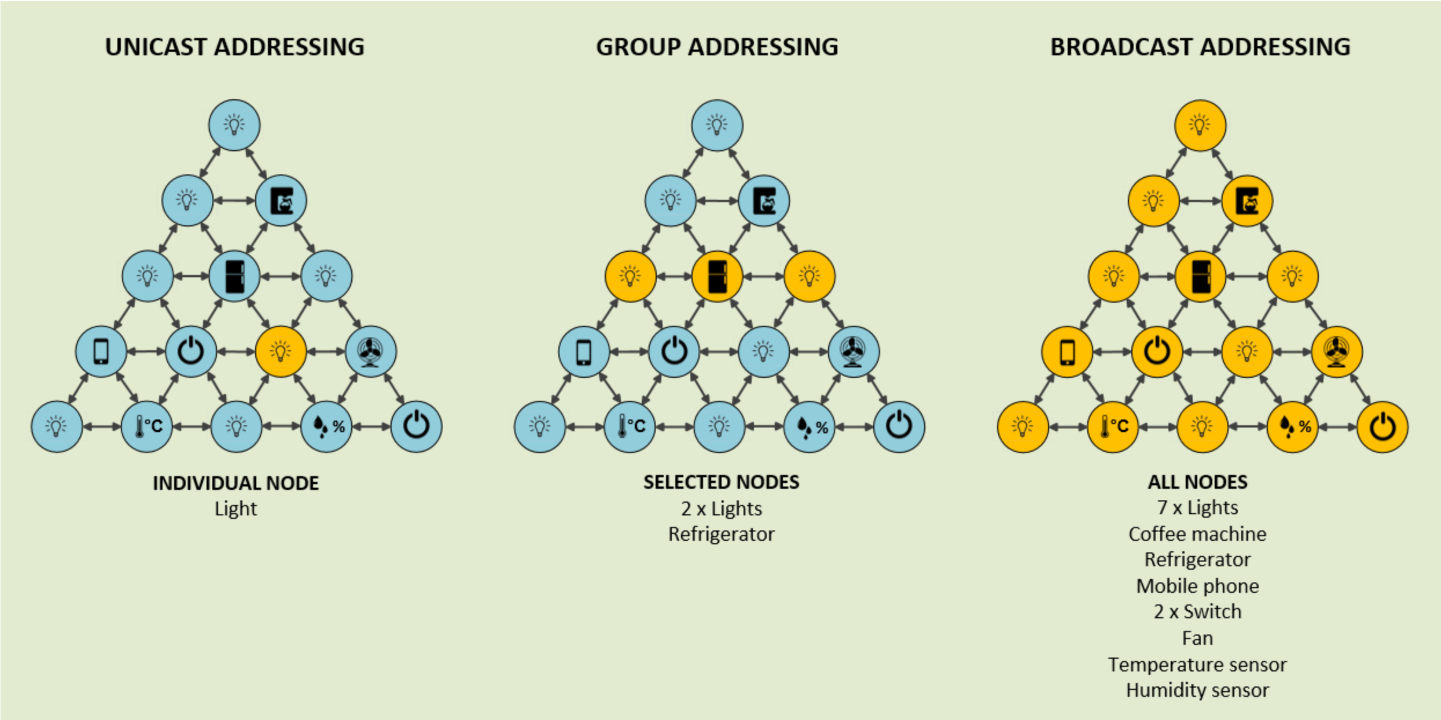
- An element is an addressable entity within a node and each node has at least one element called the Primary Element
- In addition to the Primary Element, a node may have one or more additional elements
  - This characteristic is stable throughout the lifetime of the node

Bluetooth Mesh: Addressing

- Unicast Addresses are allocated to Elements and they represent a single Element in a Node
  - A Bluetooth mesh network allows a maximum of 32767 unicast addresses
- A virtual address is a hash of a 128-bit Label UUID
  - Resulting in a very large amount of virtual addresses even if they are represented as 14-bit quantities in the network PDU's
- Group addresses are also used for multicasting and may represent several Elements in one or more nodes
  - A mesh network offers 16384 group addresses.
  - The Bluetooth mesh standard defines a set of fixed group addresses which can be used to address a subset of all primary Elements of nodes of certain functionality
  - The rest are dynamically assigned group addresses and the user has 256 fixed group addresses and 16128 dynamically assignable groups

Bluetooth Mesh: Addressing

- Since each node in the mesh may contain one or more elements the Provisioner allocates each element with a unique unicast address
- In addition to the unicast addresses for each element the Provisioner also allocates group addresses
- important to note that Bluetooth mesh allows grouping of all kinds of devices to allow flexible grouping of mesh devices



Bluetooth Mesh: Security

- Bluetooth SIG has taken security concerns seriously and made security a mandatory feature of Bluetooth mesh
- All traffic is encrypted and sending of unencrypted messages is prohibited
- This fact alone makes the security model used in Bluetooth mesh stronger compared to BR/EDR and LE

Bluetooth Mesh: Encrypted Layers and Multiple Keys

- Achieving a high level of security is based on encryption and authentication both at network and transport layers
- Message Integrity Check (MIC) is applied to traffic of both layers
- All mesh messages are encrypted and authenticated using two different keys which are protected by using AES-128
  - Network traffic (Network Key)
  - Application data traffic (Application Key)
- Application specific keys provide effective isolation of applications and application data

Bluetooth Mesh: Encrypted Layers and Multiple Keys

- Authentication and confidentiality of all data in Bluetooth mesh is provided by using three different keys:
  - Device Key
  - Network Key
  - And, Application Key
- There are also some other keys derived from these three main keys which as a group are called Derived Keys

Bluetooth Mesh: Device Keys

- Each individual node of a mesh network has its own unique Device Key
- The value of the Device Key is calculated with the shared secret provided by the Elliptic Curve Diffie-Hellman key agreement between the two devices and is never transmitted over the air
- In theory only the Provisioner has knowledge of the Device Key except if the Device Key has been handed over to another device for configuration purposes

Bluetooth Mesh: Device Keys

- The Device Key is used to authenticate and encrypt communications between the Configuration Client and a single node
- Other nodes have no way of interpreting communication between the said two devices
- The Device Key is the most critical key in the mesh network since the Device Key is used to:
  - Read information about the node
  - Enable configuring how the node will communicate with other
  - The distribution of security credentials

Bluetooth Mesh: Network Key

- Mesh networking flexibility is provided by the fact that any node can be configured to be in one or multiple Mesh Networks
  - Requiring the ability to use one or multiple Network Keys simultaneously
- Similarly, a Mesh device can be in one or more subnet through the definition of Subnet Network Keys
- Subnets are groups of nodes able to communicate with each other at the Network Layer
- Alternatively a node may belong to two or more totally separate mesh networks
  - A good example of such a case would be a mobile phone which could be part of a home mesh network, an office mesh network and a vehicle mesh network

Bluetooth Mesh: Network Key

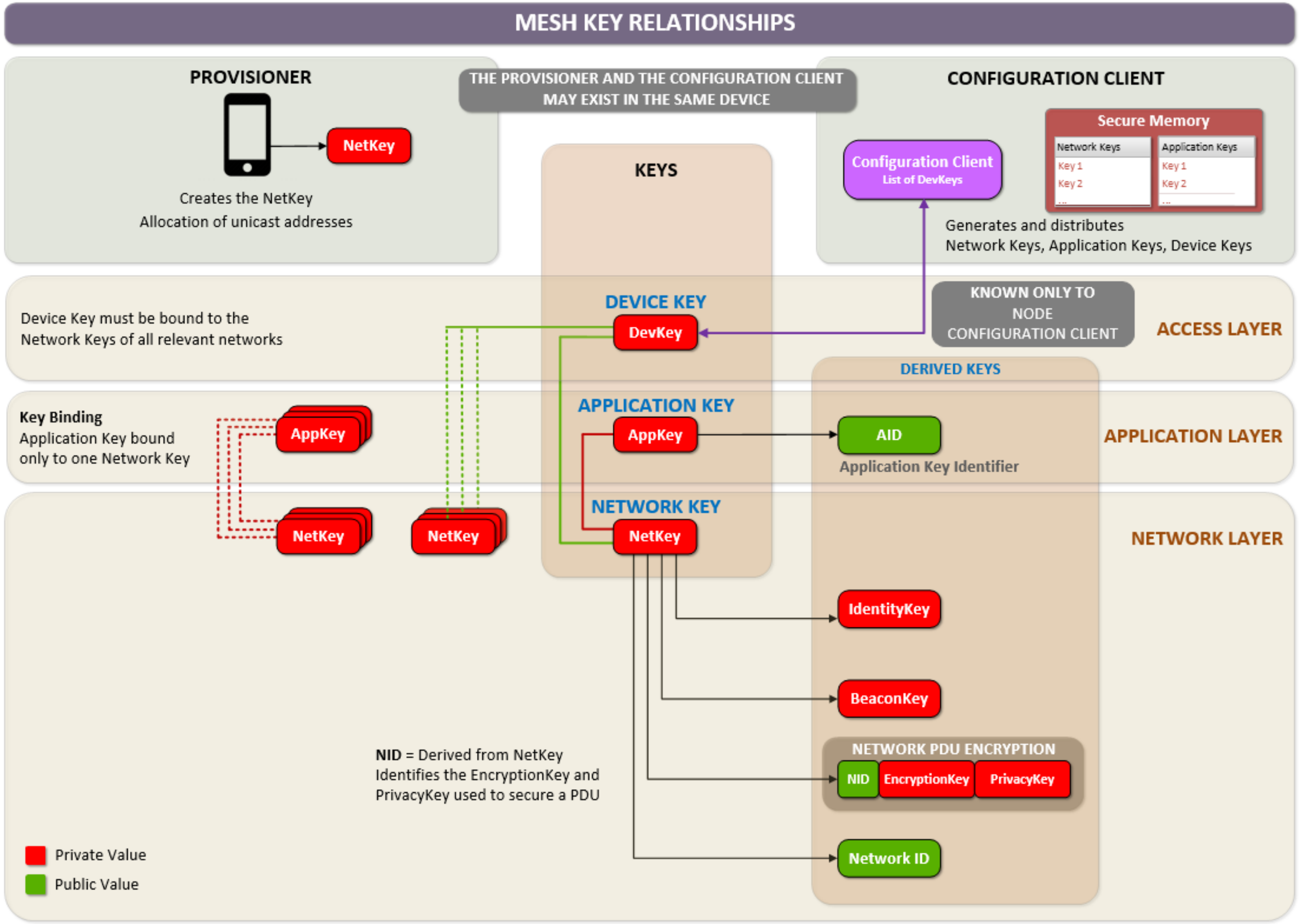
- The maximum number of different Network Keys in any installation is 4096 which makes it possible to create secure partitions inside a network when required
  - There is no direct use for the Network Key
  - It is used to derive encryption and other keys using AES-CMAC hashing

Bluetooth Mesh: Application Key

- A mesh network can contain one or more application domains each with a unique Application Key
- Nodes which are configured with a particular Application Key can receive and transmit messages related to the said application while messages originating from nodes part of different application domains are just relayed
- The above solution allows compartmentalization of applications which increases security
  - As an example consider a home with a wireless doorbell and an electrically controllable door lock
  - Having separate application domains for these two devices increases security since a compromised doorbell would still not make it possible to hack the door lock
- One can think of application domains as virtual security areas which belong to a larger security area provided by the mesh network itself

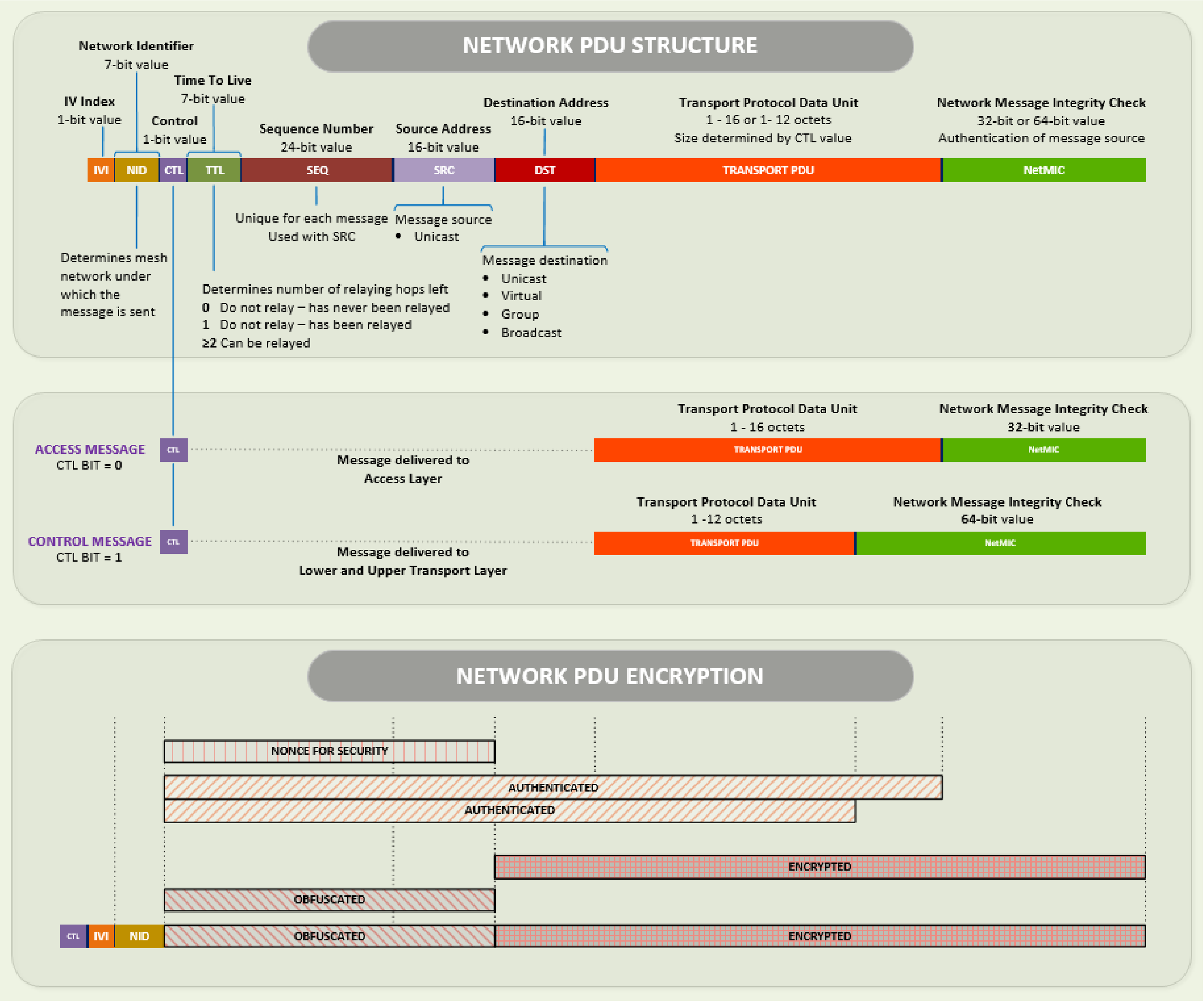
Bluetooth Mesh: Derived Keys

- Due to the nature of the Device Key and the fact that it is not used directly requires that we have other keys
- These are established from the Device Key using a Key Derivation Function (KDF) as defined by the Bluetooth SIG
- The idea here is to protect the Device Key even though a Derived Key is compromised for any reason
- The relationship of Derived Keys with the three basic keys is shown in the figure on next page.



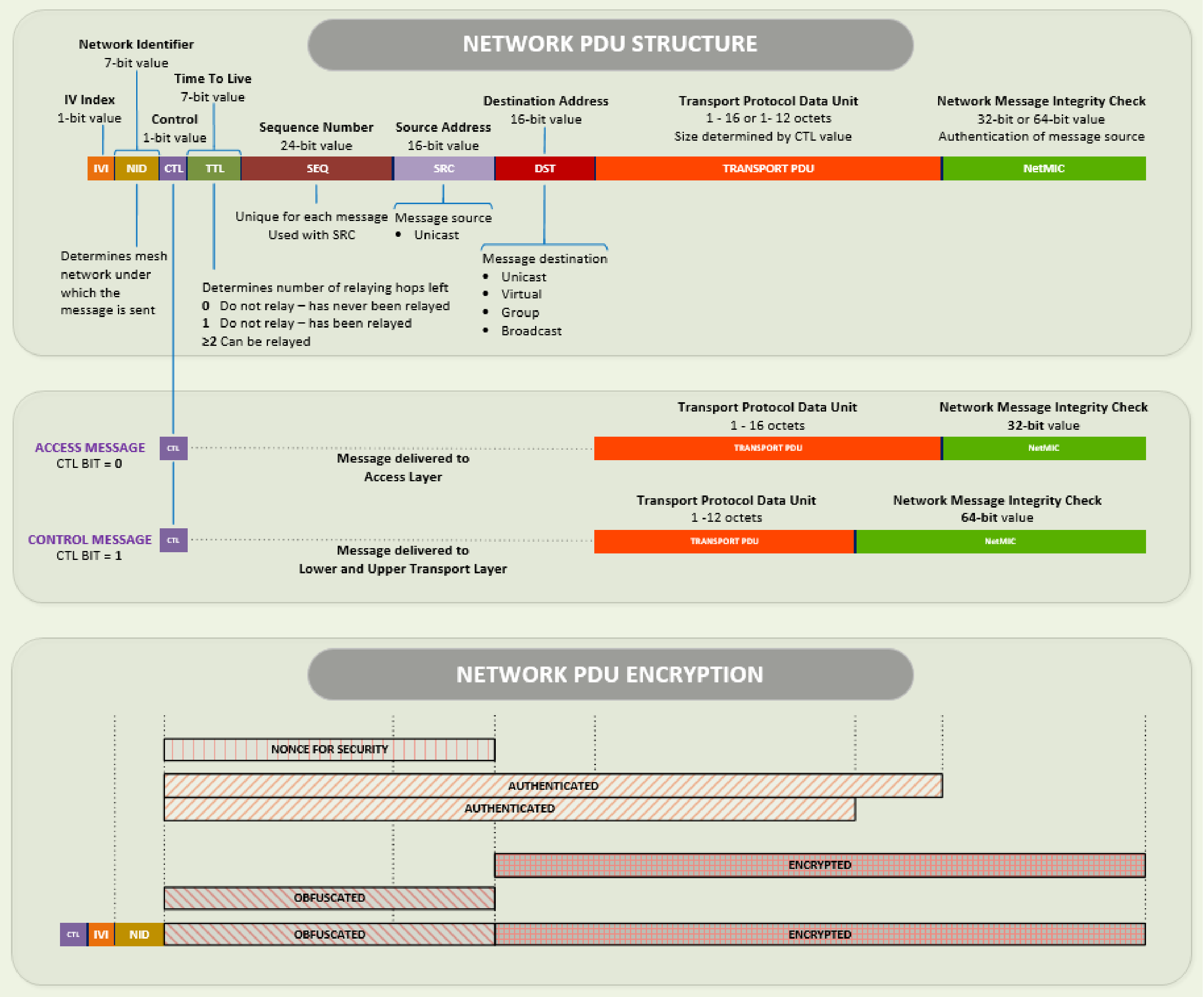
Bluetooth Mesh: Network PDU Obfuscation and Encryption

- The only plaintext parts of the Bluetooth mesh PDU are the IVI Index (IVI) and the Network Identifier (NID) at the start of the PDU package
- The rest of the PDU is obfuscated using AES-ECB or encrypted using AES-CCM
- This is to make passive snooping of the network structure next to impossible since the header, TTL, sequence number, source address etc. are not sent as plain text



Bluetooth Mesh: Network PDU Obfuscation and Encryption

- Nonce data is generated from sequence numbers and other network header information while message integrity is protected by a 32-bit or 64-bit MIC
- To keep track of individual messages each message has a unique 24-bit Sequence Number
- Replay protection is achieved by using message sequence numbering which the nodes keep track of based on source address



Bluetooth Mesh: Key Management

- The network, device keys, addresses, human friendly names related to nodes and groups are all saved into a Provisioning Database
- This solution enables the use of more than one Provisioner which is useful especially in large mesh networks
- Individual databases must then be compiled into a single database at a suitable time

Bluetooth Mesh: Key Refresh and Blacklisting

- To ensure security a Bluetooth mesh network needs to have its keys refreshed on a regular basis
- Whenever a node is removed from a Bluetooth mesh network intentionally, or should a network node become compromised for any reason, the keys must be to be refreshed!
- Bluetooth mesh provides a Key Refresh procedure
    - The Provisioner/Configuration client can utilize this feature to decrease the likelihood of security problems by performing a key refresh of all known acceptable devices on a regular basis such as once a week

Bluetooth Mesh: Minimizing Security Threats

- Bluetooth SIG has designed the security features of Bluetooth mesh to provide protection against many different types of security threats and attack methods.

Attack type	Defence method
Replay	Increasing message sequence number SEQ IV Index values within messages from a given element must always be equal to or greater than the last valid message from that element.
Bit Flipping	64-bit Message Integrity Check 32-bit Network layer / 32-bit Application layer
Eavesdropping	Encryption based on AES-CCM Encryption at Application layer Encryption at Network layer
Man-in-the Middle	Out-of-band authentication
Brute Force	Refresh of Network Key Refresh of Application Key
Physically Insecure Device	Complete separation between the network, subnetworks and application layer security
Trash-Can	Blacklisting and key refresh
Guest Access	Separate network and application keys for guests without key refresh option Limited lifetime
Privacy	Privacy built in into foundation of mesh nodes Obfuscation of identifying information on mesh payload Bluetooth address inclusion in advertising packets no longer required

Bluetooth Mesh: Practical Security Guidelines

- Bluetooth mesh provides various means to maximize security
- Applications should be configured with individual Application Keys which will limit problems if a node is compromised
- Pairwise Application Keys can be configured between nodes which need to exchange data only between each other
- Device Keys represent the most critical information in the network
  - If a mobile phone is used as a Provisioner, the Device Keys as well as Network Keys and Application Keys should be stored using secure storage API's available through the operating system of the mobile device
  - A lost mobile phone would then not present risks regarding the security of the mesh network
- For embedded devices security credentials should be stored in secure memory elements

Bluetooth Mesh: Practical Security Guidelines

- In device provisioning using out-of-band input or output the user should use at the minimum a 6-octet numeric or alphanumeric value but the longer the value the stronger the security
  - Alternatively a device may use other non-Bluetooth data transfer methods available
- When devices are removed from the mesh the Provisioner should initiate a Key Refresh procedure throughout the whole mesh network for both the network and the Application Key
- The removed device should also be blacklisted
- There is also a need to refresh the above mentioned keys periodically to help maintain the strength of the obfuscation
- Bluetooth SIG recommends this to be done every 14 days

Bluetooth Mesh: Practical Security Guidelines

- In similar fashion a periodic refresh of the IV Index keeps the nonces applied to encryption and authentication fresh
  - Bluetooth SIG recommends the same 14 day period for IV Index refreshing
- Programming fixed Network Keys during device manufacturing is strongly discouraged since refreshing as recommended above would not be possible
- The Provisioning Database contains very sensitive information regarding the safety of a mesh network special attention to the transfer of the database is needed
  - Authentication using at the minimum 128-bit MIC is recommended
  - In addition using OOB methods will ensure that only intended devices can utilize the database

Bluetooth Mesh: Low Power

- Low sensors may be in deep sleep most of the time while gathering and saving data with the wireless radio circuitry switched off to save battery power
- How can the user be sure that messages addressed to dormant nodes will eventually reach the node?
  - In Bluetooth mesh this is managed by using the so called Friendship feature
  - A node which supports the Low Power feature must have the feature enabled
  - And, establish a friendship relationship with a Friendship node

Bluetooth Mesh: Low Power

- The Low Power feature reduces receiver duty cycles by allowing the Friend node to store messages intended to the Low Power node
- When the Low Power node wakes up it polls the Friend node for any messages stored on its behalf and can receive them "batch style"
  - How is this Low Power and Frienship node different in architecture compared to Bluetooth Low Energy?
- The Friendship node will need to be of the "Always on" type so that is can receive messages addressed to the Low Power node at any time