蓝牙 Mesh 协议分析报告

阿里巴巴集团

版本号	发布/更新日期	更新人员（部门/职位）	重要变更内容
V1.0	2018 年 1 月 11 日	胡俊锋（崮德） 人工智能实验室-W 实验室	编辑并发布初版

目录

1.概念

一个普通的蓝牙 Device（设备），加入 Mesh 网络（Porvisioning 过程）后，成为 Node（节点）。每个 Node 可以包含多个 Element，一个 Element 对应一个 Unicast address（16bits，32767 个地址，bit15=0）；每个 Element 可以包含多个 Model（发送、接收和处理 Message），每个 Model 对应一个 Model ID（可以分 SIG Model ID 和 Vendor Model ID），类似这个 Model 的地址。其中，SIG Model ID 是 16bits 的，SIG 组织定义的专用 Model ID，SIG Model ID 参考例子如下表 1 所示，而 Vendor Model ID 是 32bits 的，由 16bits 的 Company ID 和 16bits 的 Vendor-assigned Model ID 组成。

表 1 SIG Model ID 参考例子

Model Name	SIG Model ID
Generic OnOff Server	0x1000
Generic OnOff Client	0x1001
Generic Level Server	0x1002
Generic Level Client	0x1003

下图 1 是 Mesh 网络分层结构，我们 Coding 的时候，一般操作其中的 Access Layer，也就是打包 Access Payload。Access Payload 的包结构分为两个字段：Opcode+Parameter。每个 Access Payload 可以最多是 32 个 Segment（12 字节），也即最多 384 个字节（包含 TransMIC），如果 TransMIC 是 4 字节，则有效载荷是 380 字节，可以有 3 种组合：1 字节的 Opcode（For Special Message）+379 字节的 Parametes；2 字节的 Opcode（For Standard Message）+378 字节的 Parameters；3 字节的 Opcode（For Vendor-Specific Message）+377 字节的 Parameters。当然，如果 Unsegment，则 Access Payload 最多可以有 11 字节。

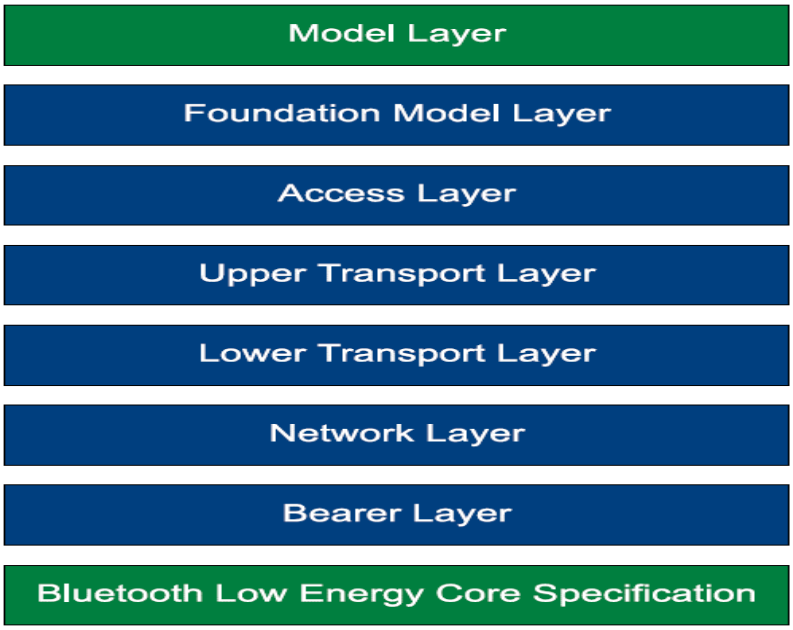

图 1 Mesh 网络分层结构

Mesh 网络是消息驱动的架构，每个 Model 处理一类 Messages，消息分 ACK 和非 ACK 消息，比如对应上表的 Generic OnOff Server 的 Model，需要处理以下表所示的 Messages。

表 2 Generic OnOff Server Model 处理的 Messages

Model Name	Message Name	Opcode
Generic OnOff Sever	Generic OnOff Get	0x82 0x01
	Generic OnOff Set	0x82 0x02
	Generic OnOff Set Unacknowledged	0x82 0x03
	Generic OnOff Status	0x82 0x04

另外 Messages 可以支持 Transactions（通过 Transaction Identifier 识别），在一个 Transaction 里面支持一系列 Messages，比如 Set，Recall 和 Clear 等。Transaction Identifier 可以识别这个消息是个新消息还是一个重发的之前的旧消息。

Generic OnOff Set 这个消息的包结构如下表所示：

表 3 Generic OnOff Set 包结构

Field	Size（Octets）	Notes
Opcode	1 or 2 or 3	0x8202
OnOff	1	
TID	1	Transaction Identifier
Transition Time	1	
Delay	1	

一个 Messages 只能对应一个 Model，如果需要处理两个相同的 Message，则需要设置两个不同的 Element 和 Model 来处理。如下图 2 所示，这个智能插排设备需要同时控制两个插座的开和关，因此需要处理两个相同的 Generic OnOff Set 的 Message，当该设备加入 Mesh 网络成为一个 Node 后，该 Node 需要设置两个 Element，获得两个 unicast address，并配置两个 Generic OnOff Server 的 Model，分别处理 Generic OnOff Set 的 Message（通过 Unicast address 区别）

关于所有 Messages 的 Opcode 定义，可以参考文档 1）的 4.3.4 和参考文档 2）的 7.1。

图 2 支持双插头的智能插排

Mesh 网络一共有 3 种 Mesh 专用广播包，如下图所示

0x29	«PB-ADV»	Mesh Profile Specification Section 5.2.1
0x2A	«Mesh Message»	Mesh Profile Specification Section 3.3.1
0x2B	«Mesh Beacon»	Mesh Profile Specification Section 3.9

图 3 Mesh 专用广播包

2. Server、Client 和 Control

Model 可以分为三种类型：Server、Client 和 Control。

Server Model：暴露自己的状态（states）给其他 Model 访问，比如一个灯，可以暴露开和关的状态给其他 Model 访问，这个灯就可以包含一个 Server Model。所有 Node 默认需要两个 Server Model：Configuration Server（0x0000）和 Health Server（0x0002）

Client Model：访问 Server Model 的状态，比如一个开关，可以发送 Message 获取灯的状态，或者设置灯的状态。

Server model 需要维护 States，而 Client Model 不需要维护 States。

Control Model：Server Model+Client Model。比如灯的控制器，即是 Client Model，需要访问传感器（环境光的亮度）的状态和控制灯的状态，又是 Server Model，需要接收手机发送过来的配置信息。

任何一个 Node 都可以包含以上三种类型的 Model。

上述提到的 States 的例子比如：Generic OnOff State，如下表所示：

表 4 Generic OnOff State 例子

Value	Description
0x00	Off
0x01	On
0x02-0xFF	Prohibited

3． Mesh Beacon

设备通过 Mesh Beacon 来广播信息，Mesh Beacon 分为两种：Unprovisioned Device Beacon 和 Secure Network Beacon。其中 Unprovisioned Device Beacon 是用来被 Provisioner 发现 device 用的，目前是 40ms 发一次，间隔 50ms，包结构如下图所示：

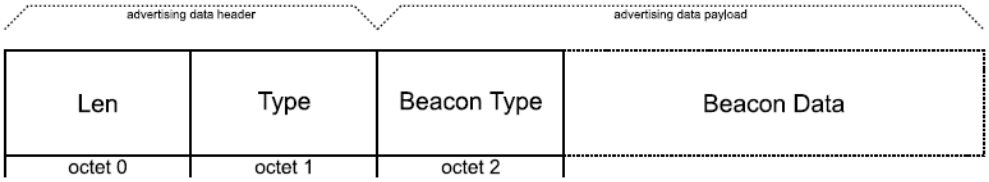

图 4 Unprovisioned Device Beacon 包结构

具体字段含义如下表所示：

表 5 Unprovisioned Device Beacon 包结构

Field	Size（Octets）	Notes
Len	1	长度
Type	1	0x2B
Beacon Type	1	0x00
Device UUID	16	通过时间随机产生，可以用来做设备商信息
OOB Information	2	bit0 Other bit1 Electronic / URI bit2 2D machine-readable code bit3 Bar code bit4 Near Field Communication (NFC) bit5 Number bit6 String bit7 Reserved for Future Use bit8 Reserved for Future Use bit9 Reserved for Future Use bit10 Reserved for Future Use bit11 On box bit12 Inside box bit13 On piece of paper bit14 Inside manual bit15 On device
URI Hash	4	Optional

Secure Network Beacon 是被 Node 用来标识子网信息和安全状态。Relay and Friend nodes 必须发送 beacons 而其他类型 Nodes 可能发送 beacons。Beacons 的发送间隔计算公式：Beacon Interval = Observation Period * (Observed Number of Beacons + 1) / Expected Number of Beacons

Secure Network Beacon 包结构如下表所示：

表 6 Secure Network Beacon 包结构

Field	Size（Octets）	Notes
Len	1	长度
Type	1	0x2B
Beacon Type	1	0x01
Flags	1	Contains the Key Refresh Flag and IV Update Flag
Network ID	8	Contains the value of the Network ID
IV index	4	
Authentication Value	8	Authenticates security network beacon

4. Provision

Provisioning：Provisioner（比如天猫精灵）把一个 Device 加入一个 Mesh 网络成为一个 Node 的过程。

Provisioning Data：Network Key，current IV index 和 unicast address（对应每个 Element）。一个 Mesh 网络中可以有多个 Provisoner，为了简单，实际一般就一个 Provisioner。未来可以考虑设置灯为第二个 Provisioner，阿里来贡献这部分协议。

在 Provisioning 的过程中，每个设备都会提供自己的 Device UUID（128bits，可以通过当前时间随机产生，避免地址冲突）给 Provisioner 用来唯一标识自己（因为这个时候设备还没有被分配 16bits 的 unicast address）。Provisioning 承载在 PB-ADV 或者 PB-GATT（不支持 PB-ADV 的设备选择使用）之上。Provisioning 流程如下图所示。

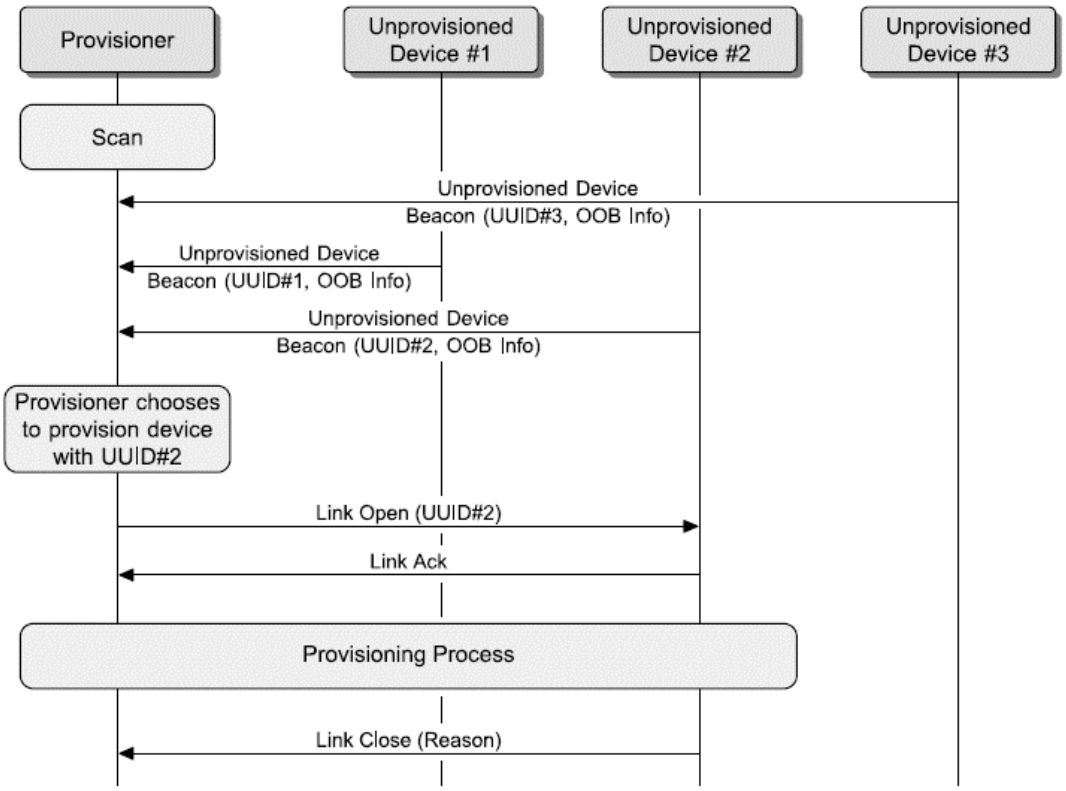

图 5 Provisioning Session

4.1 PB-ADV

蓝牙 BLE 广播包支持 31 字节，但是其中的 MTU 只有 24 字节，且需要设备 100%占空比的扫描广播信道。PB-ADV 包结构如下表所示：

表 7 PB-ADV 包结构

Field	Size（Octets）	Description
Length	1	长度
AD Type	1	0x29
Contents	Variable	PB-ADV PDU

上表中的 Contents 结构如下表所示：

表 8 PB-ADV PDU 结构

Field	Size（Octets）	Description
Link ID	4	在一次 Provisioning session 中保持恒定，随机产生避免冲突
Transition Number	1	标识每一个 Generic Provisioning PDU，如果

		Generic Provisioning PD 是分多个包，则多个包的 Transition Number 都相同。重发包，Transition Number 保持不变。
Generic Provisioning PDU	1-24	实际长度可以超过 24 字节，但是需要分包处理。

Generic Provisioning PDU 包结构如下表所示：

表 9 Generic Provisioning PDU 包结构

Field	Size（Octets）	Description
Generic Provisioning Control	1-17	
Generic Provisioning Payload	0-64	

Generic Provisioning Control 的第一个字节的 two least significant 表达含义如下表所示

表 10 Generic Provisioning Control 第一字节的 two least significant 含义

Value	Description
0b00	Transition Start
0b01	Transition Acknowledge
0b10	Transition Continuation
0b11	Provisioning Bearer Control

Generic Provisioning Payload 的包结构（Transition Start 包）如下表所示

表 11 Transition Start 包结构

Field	Size（bits）	Description
SegN	6	最后的 Segment Number，最多 64 个 Segment，如果有 7 个 Segment，则这个值为 0b000111
GPCF	2	0b00
Total Length	16	
FCS	8	Frame Check Sequence
Data		最少 1 字节

上表的 Data 结构如下表所示

表 12 Transition Start 包的 Data 结构

Field	Size（bits）	Description
Padding	2	0b00
Type	6	0x00 Provisioning Invite

		0x01 Provisioning Capabilities 0x02 Provisioning Start 0x03 Provisioning Public Key 0x04 Provisioning Input Complete 0x05 Provisioning Confirmation 0x06 Provisioning Random 0x07 Provisioning Data 0x08 Provisioning Complete 0x09 Provisioning Failed 0x0A-0xFF RFU
Parameters	Variable	

Device 端的 Public Key 可以选择使用 OOB 提供。

其中 Provisioning Data 包括 Network Key， NetKey Index，Key Refresh Flag，IV Update Flag，current value of the IV Index，and unicast address of the primary Element

Provisioning Bearer Control 包结构如下表所示：

表 13 Provisoning Bearer Control 包结构

Field	Size（bits）	Description
BearerOpcode	6	0b00 Link Open 0b01 Link ACK 0b10 Link Close 0b11 RFU
GPCF	2	0b11
Parameters	Variable	

整个 Provisioning 过程如下图所示，包含 5 个步骤：beaconing, invitation, exchanging public keys, authentication, and distribution of the provisioning data，其中 2a 和 2b 通过交换 Public Key（32 字节 Public Key X+32 字节 Public Key Y=64 字节），产生 ECDHSecret，公式如下：ECDHSecret = P-256(private key, peer public key) ；3a，3b 和 3c 是认证过程。

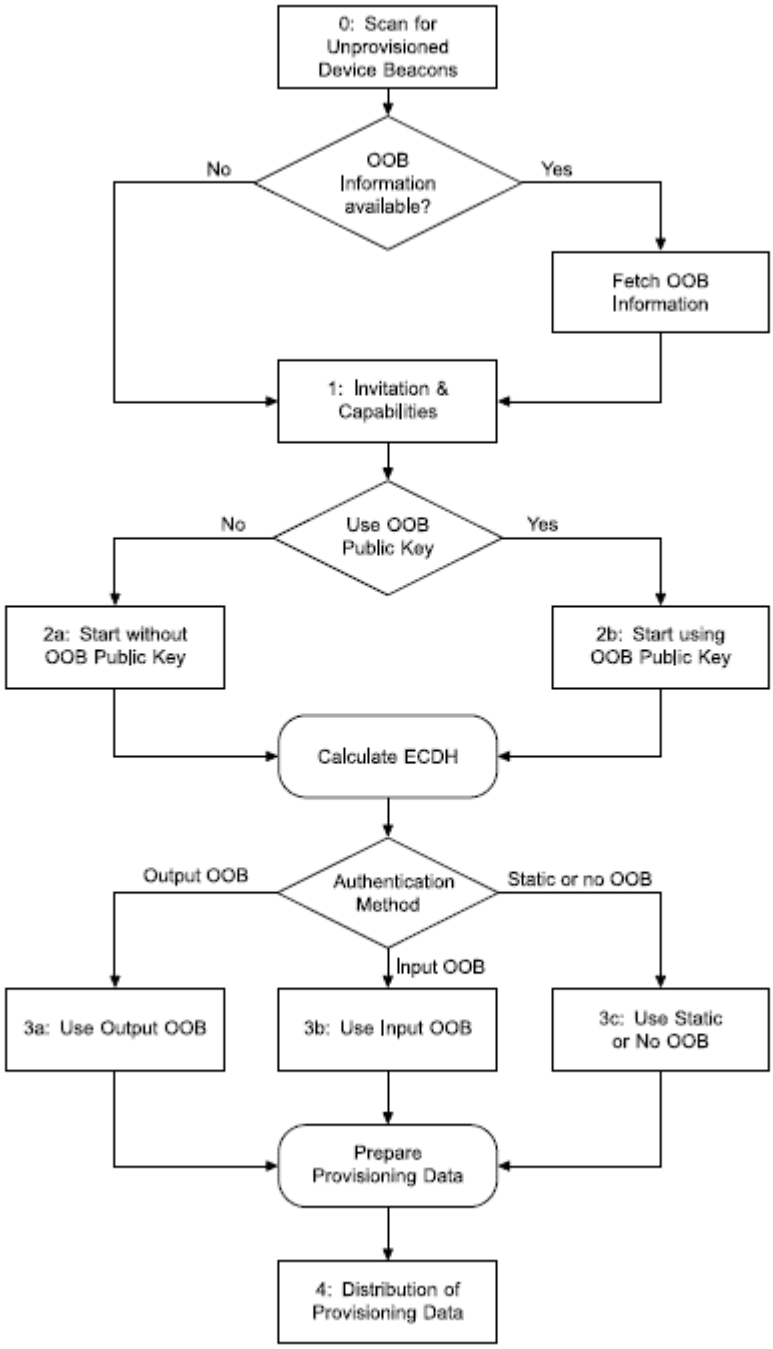

图 6 Provisioning 流程图

在 Provisioning 过程中，设备可以选择支持 Attention Timer 机制，来提醒用户该设备正在注册 Mesh 网络，不同设备可以有不同提醒方式，比如通过灯闪烁或者马达发出噪声来提醒。这样的好处是，如果有多个设备等待注册网络，则可以提醒用户当前是哪个设备在注册网络。

4.2 PB-GATT

不支持 PB-ADV 的设备，如蓝牙 4.0 手机，则需要通过和节点建立 GATT 连接，并通过 Mesh Provisioning Service 通信，通过 Proxy PDUs 重新封装 Provisioning PDUs。Mesh Provisioning Service 包含 Mesh Provisioning Data In characteristics 和 Mesh Provisioning Out characteristics。因此建立 GATT 连接之前需要 40ms 发一次，间隔 1 秒的广播包，包结构如下图所示：

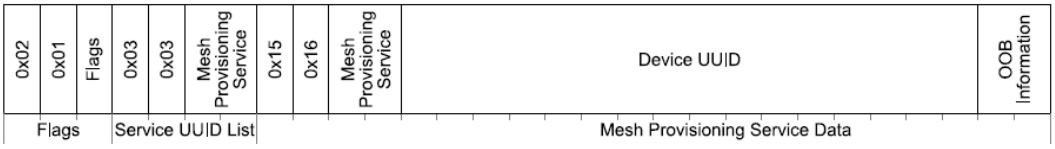

图 7 PB-GATT 的广播包格式

具体字段含义如下表所示：

表 14 Mesh Beacon 广播包各个字段含义

Field	Size（Octets）	Notes
Flags	1	长度，0x02 表示后面还有 2 个字节
	1	类型，0x01 表示类型是 Flags（Assigned Number）
	1	Flags 的值，0x02 表示 LE General Discoverable Mode
Service UUID List	1	长度，0x03 表示后面还有 3 字节
	1	类型，0x03 表示 Complete List of 16-bit Service Class UUIDs
	2	Mesh Provisioning Service，0x1827
	1	长度，0x15 表示后面还有 21 字节
	1	类型，0x16 表示 Service Data - 16-bit UUID
	2	Mesh Provisioning Service，0x1827
Device UUID	16	设备 UUID
OOB Information	2	bit0 Other bit1 Electronic / URI bit2 2D machine-readable code bit3 Bar code bit4 Near Field Communication (NFC) bit5 Number bit6 String bit7 Reserved for Future Use bit8 Reserved for Future Use bit9 Reserved for Future Use bit10 Reserved for Future Use bit11 On box bit12 Inside box bit13 On piece of paper bit14 Inside manual bit15 On device

GATT 连接使用了 Mesh Provisioning Service，并且是 Primary Service，Service UUID 为 0x1827，对应的 Characteristics 使用了 Mesh Provisioning Data In（0x2ADB）和 Mesh Provisioning Data Out（0x2ADC）。

5．NetKey，AppKey 和 DevKey（Configuration Client）和 Session Key

在通信阶段使用 NetKey，AppKey 和 DevKey。

Network Key：用来在 Network Layer 加密通信数据。NetKey 是 16 字节。NetKey 使用随机数产生方式，避免冲突。NetKey 属于 Configuration Client 维护的 NetKey List（12bits NetKey Index），可以多达 4096 个 NetKey。可以通过 Config NetKey Add 的 Message 来给 Node 分配 NetKey。

Application Key：用来在 Upper Transport Layer 加密通信数据。在 Provisioning 结束后，进入 Configuration 过程，添加 AppKey 并且把 AppKey 和具体的 Model 绑定。一个 Model 可以有 251 个 AppKey。APPKey 是 16 字节。AppKey 使用随机数产生方式，避免冲突。APPKey 属于 Configuration Client 维护的 APPKey List（12bits AppKey Index），可以多达 4096 个 AppKey。可以通过 Config AppKey Add 的 Message 来给 Node 分配 AppKey。

NetKey Index（12bits）+AppKey Index（12bits）=24bits，使用 3 个字节表达。

Device Key：Configuration Client 通过这个密钥来管理 Node，同时发布更新的网络信息和 Application Key。公式：DevKey = k1(ECDHSecret, ProvisioningSalt, “prdk”)

Configuration message 通过这个 Key 来加密信息。

一个节点可以有一个 Device Key，多个 AppKey，多个 NetKey。

注意：Provisioner 的 Device Key 用来和其他 Provisioner 通信用，并且通过 OOB 来提供这个 Key。

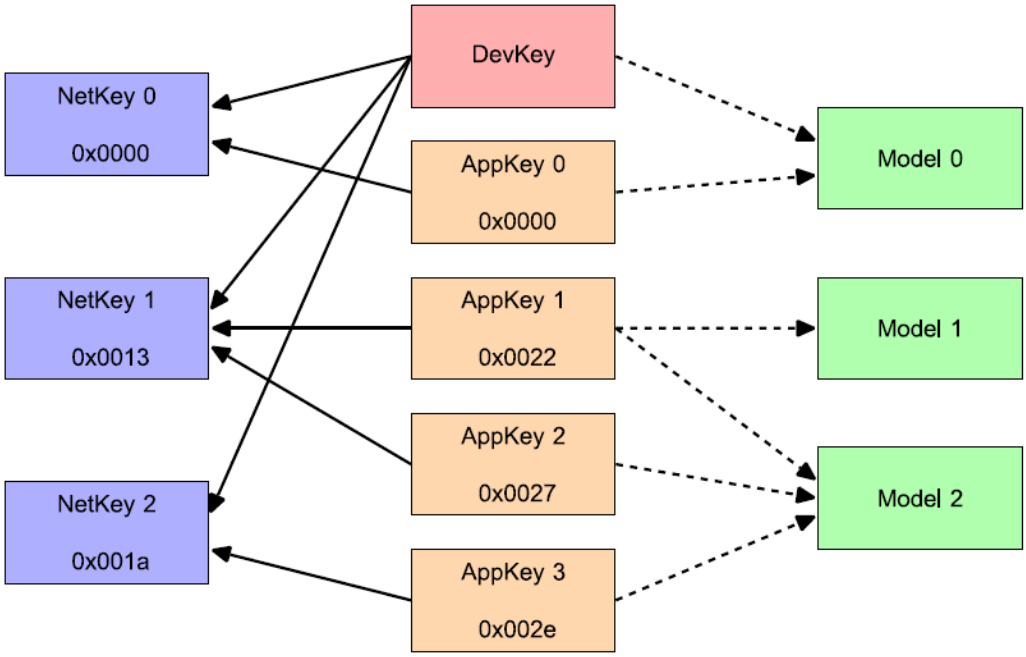

图 8 NetKey 和 APPKey

Session Key：Provisioning 过程使用到这个 Key，在 Provisioning 的交换公开密钥获得 ECDHSecret 以及认证通过后，通过 k1 算法获得 Session Key。公式：SessionKey = k1(ECDHSecret, ProvisioningSalt, “prsk”)。然后，Provisioner 和 Device 用这个 Session Key 交换 Provisioning Data。

6. Mesh 网络创建过程

Provisioner（比如 Smart Phone）主动或者被动扫描上述的 Unprovisioned Device Beacon，并在 Provisioning 过程提供 Network Key， NetKey Index，Key Refresh Flag，IV Update Flag，current value of the IV Index，and unicast address of the primary Element（其他 Element 地址自动递增）。其中 IV Index 设置为 0x00000000。在 Provisioning 过程结束，Configuration Client（比如 Smart Phone）给 Node 提供 AppKey，Publish 和 Subscribe 地址。

7. Key 刷新过程

Configuration Client model 通过 Config NetKey Update message 和 Config AppKey Update message 刷新 NetKey 或 APPKey，比如用来防止垃圾桶攻击。Key 刷新过程和 IV 更新过程互不影响，独立运作。分为 3 个步骤：

1、 发布新的 Key 给所有的 Nodes，这些 Nodes 继续使用旧的 Key 发送消息，以及用新的和旧的 Key 接收消息。
2、 发布一个 Secure Network Beacon 通知网络，新的 Key 已经发给了所有的 Nodes，这些

Nodes 使用新的 Key 发送消息，但是仍然可以使用新的和旧的 Key 接收消息。

3、 发布一个 Secure Network Beacon 通知网络，所有 Nodes 必须回收旧的 Key，以后都通过新的 Key 发送和接收消息。

更新状态如下图所示：

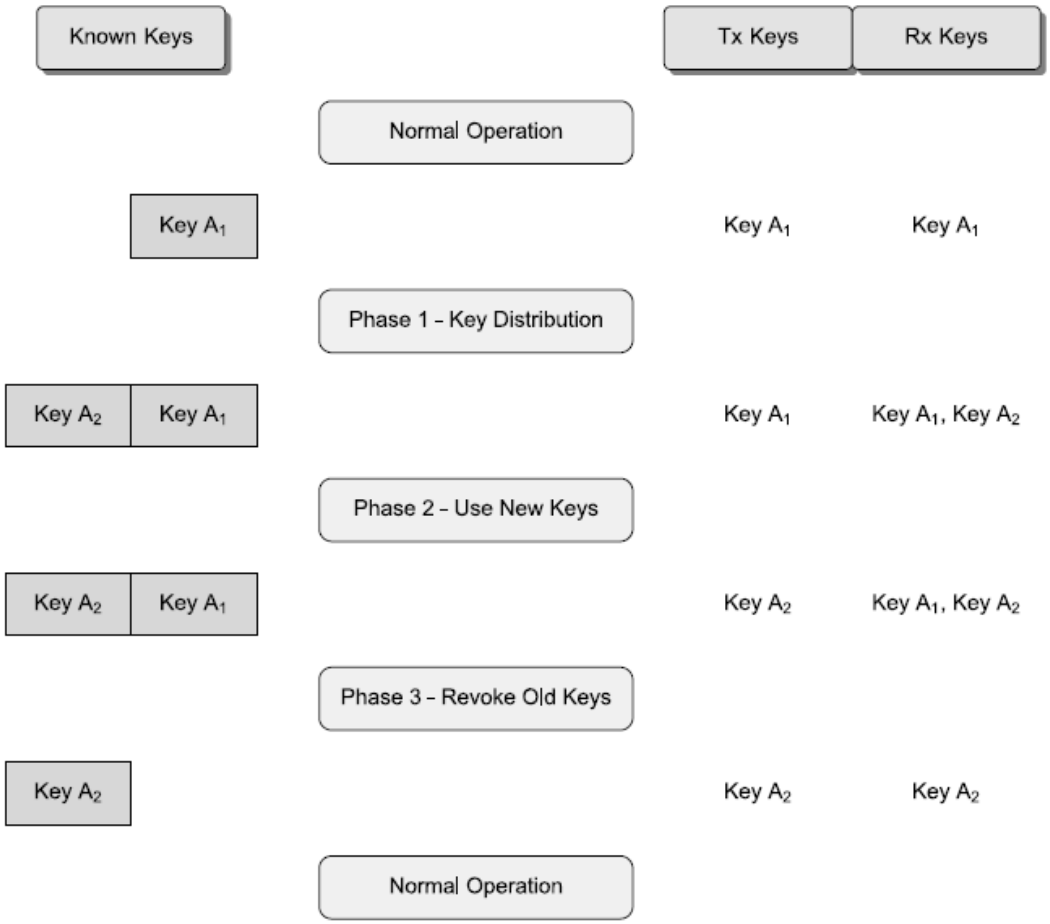

图 9 Key 刷新流程

8. Sequence number

Sequence Number：Network PDU 的 SEQ 位, 24bits，主要是防止 Replay Attacks.它是配合 IV Index 使用的。当 Sequence Number 耗尽后，需要更新 IV Index 来重新使用之前用过的 Sequence Number，因为 Sequence Number 又从零开始循坏。

9. Node 移除流程

通过把 node 放入 blacklist 来移除该 node，并把它排除在 Key 刷新流程之外。当这个

Node 被成功移除后，它的 unicast address 可以被重新使用。当然必须在 IV Index 更新后再使用这个地址，以保证 Sequence Number 可以被重新使用。

10.Proxy， Relay， Friends 和 Low Power

Mesh 网络的节点可以支持四种角色：Proxy，Relay，Friends 和 Low Power。Proxy 必须是在 GATT 连接下起作用，比如一个支持 GATT 但是不能广播 Mesh Message AD Type 的节点，可以和一个即支持 GATT 连接，又支持广播的节点（Proxy）建立 GATT 连接，然后基于 Proxy 协议交换信息。

11. Address

Mesh 网络地址分为 4 种：unassigned address、unicast address、virtual address 和 group address。地址长度是 16bits，其中 Unicast address 是用来表达一个 node 的某一个 Element，一共有 32767 个地址。

表 15 Mesh 地址类型

Values	Address Type
0b0000000000000000	Unassigned address
0b0xxxxxxxxxxxxxx（不含 0b0000000000000000）	Unicast address
0b10xxxxxxxxxxxxx	Virtual address
0b11xxxxxxxxxxxxx	Group address

其中 group address 如下表所示，有 256 个固定地址 0xFF00-0xFFFF，有 16128 个动态地址（0xC000-0xFEFF）

表 16 Secure Network Beacon 包结构

Values	Fixed Group Address Name
0xFF00-0xFFFB	RFU
0xFFFC	All-proxies
0xFFFD	All-friends
0xFFFE	All-relays
0xFFFF	All-nodes
0xC000-0xFEFF	For other usage

12. Example

- 添加 NetKey，NetKeyIndex 到 Provisioner

- 添加 APPKey，AppKeyIndex，NetKeyIndex 到 Provisioner
- Provisioning 过程中获得 Device 的 Device UUID
- 根据 Device UUID 设置该 Device 的 NetKey 和 Unicast address
- Provisioning 结束后 Configuration Device，比如 AppKey 和绑定到 Model。

参考文档：

1） Bluetooth Mesh Profile specification, Version 1.0 or later

2） Bluetooth Mesh Model specification