涂鸦 BLE 通信协议_V3.1

修改记录：

版本	编写/修订说明	修订人	修订日期	备注
V2.0	协议升级	高永会	2019/04/10	新建
V2.1	增加和修改	高永会	2019/04/15	1、 完善广播数据结构描述。 2、 将分包数据结构包头中的加密标识移到数据中。 3、 增加分包数据描述。
V2.2	增加和修改	高永会	2019/04/28	1、 修改秘钥生成方式。 2、 重新定义 OTA 协议及指令。 3、 新增状态上报指令，优化状态查询指令。 4、 新增发送配网信息指令。 5、 优化分包数据结构。 6、 新增设备调试信息显示指令。 7、 优化设备获取时间指令。 8、 优化 bulkdata 数据传输流程和指令。
V2.3	增加和修改	高永会	2019/05/23	1、 增加秘钥 4 用于已绑定设备获取 srand。 2、 广播包中增加 company id。 3、 新增广播包中 PID 和 device id 加密说明。 4、 新增设备重置指令。 5、 新增设备异常解绑指令。 6、 新增带时间戳状态上报指令。 7、 新增两条获取实时时间指令。 8、 修改 OTA 升级文件信息指令响应内容。
V2.4	增加和修改	高永会	2019/06/01	1、 修改获取设备信息指令响应内容。 2、 优化通信加密流程图。 3、 修改秘钥生成方式。
V3.0	增加和修改	高永会	2019/07/20	1、 修改广播包数据格式（协议版本升级为 3）。 2、 修改查询设备信息指令响应格式（协议版本升级为 3）。 3、 修改发送配网信息指令码为 0x0021 。 4、 新增上报配网状态指令。 5、 新增数据透传指令。 6、 新增 ECDH 密钥协商指令。 7、 新增唤醒指令。 8、 修改 adv data flag。 9、 修改 scan response data 定义。

				10、 修改获取设备信息指令响应内容。
V3.1	增加和修改	高永会	2019/09/23	1、 删除 Bulkdata 通道 service 定义和相关传输指令。 2、 获取设备信息指令响应新增外部 MCU 相关版本号字段。 3、 OTA 指令新增支持 TYPE=1 外部 mcu 固件升级。 4、 完善 0x0012 ble 通道产测指令说明。 5、 删除原 0x8010 更新时间指令。

1 范围

本协议由涂鸦-嵌入式部门-BLE 组编制。

2 术语

变量	字符数	描述
product_id	8	产品标识符
device_id	16	设备标识符
auth_key	32	认证密钥
session_key	16	传输密钥
random	6	登入随机数
login_key	6	登入密钥

3 总体描述

本协议定义了一个与基础通信层无关的简单协议。

3.1 帧数据包结构

序号	长度	字段	说明
1	4	SN	发送方 SN
2			
3			
4			
5	4	ACK_SN	
6			
7			
8			
9	2	功能码	
10			
11	2	数据长度 len	
12			
13~(13+len)	len	数据	
13+len	2	CRC16	检验
14+len			

SN：app 和设备端各维护自己的 SN，每次新的连接初始值为 0，每发送一包指令依次滚动加 1，溢出复位清零。

ACK_SN：只有响应包的 ACK_SN 才有意义，代表是响应的哪条指令，不是响应包默认为 0。
功能码：操作命令码。
数据长度：本次传送数据的长度。
校验字：本协议采用 CRC16 校验，校验内容包括 SN、ACK_SN、功能码、数据长度、数据 4 个部分。

3.2 功能码定义

功能码			
序号	功能码	功能	备注
1	0x0000	查询设备信息	
2	0x0001	发起配对	
3	0x0002	命令下发	
4	0x0003	设备状态查询	
5	0x0004	设置密码	保留
6	0x0005	设备解绑	
7	0x0006	设备重置	
	0x0007	申请启动（APP）	原 bulkdata 通道传输协议
	0x0008	数据传输响应（APP）	
	0x0009	传输结束响应（APP）	
	0x000A	强制传输终止（APP）	
	0x000C	OTA 开始升级	
	0x000D	OTA 升级文件信息	
	0x000E	OTA 升级文件偏移请求	
	0x000F	OTA 数据	
	0x0010	OTA 升级结束	
	0x0012	BLE 通道产测指令	
	0x0013		
	0x0014	异常解绑	
	0x0020	ECDH 密钥协商	
	0x0021	发送配网信息	
	0x0022	上报配网状态	BLE->APP
	0x0023	数据透传指令	
	0x0024	唤醒指令	目前用于低功耗 ipc 设备
	0x8001	状态上报	
	0x8002	设备调试信息显示	
	0x8003	记录型状态数据上报	
	0x8011	设备获取实时时间 1	
	0x8012	设备获取实时时间 2	

3.3 数据编码

本协议数据项使用 Big-Endian 编码，即多个字节表示一个数据项时，先发送高字节。

4 蓝牙接口定义

4.1 服务接口 1-常用数据传输通道

UUID: 1910

包含 Write Characteristic 和 Notify Characteristic 两个子接口

子接口一：Write Characteristic(提供手机 APP 向 BLE 设备发送消息或取相应消息) UUID：2b11

属性：Write

数据类型：字节

数据长度：20（每包最大可传输数据）

子接口二：Notify Characteristic(提供给 BLE 设备向超级 APP 发送通知信息) UUID： 2b10

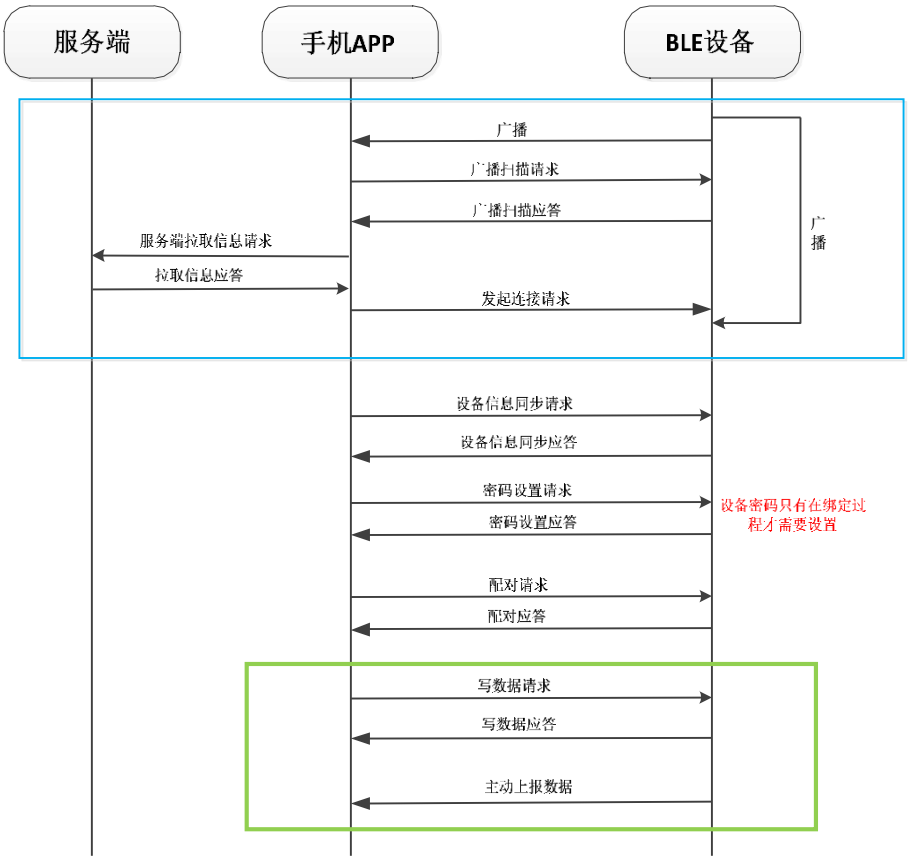
属性： Notify

数据类型：字节

数据长度：20（每包最大可传输数据）

5 设备通信流程

5.1 基本通信流程：



5.2 设备发现

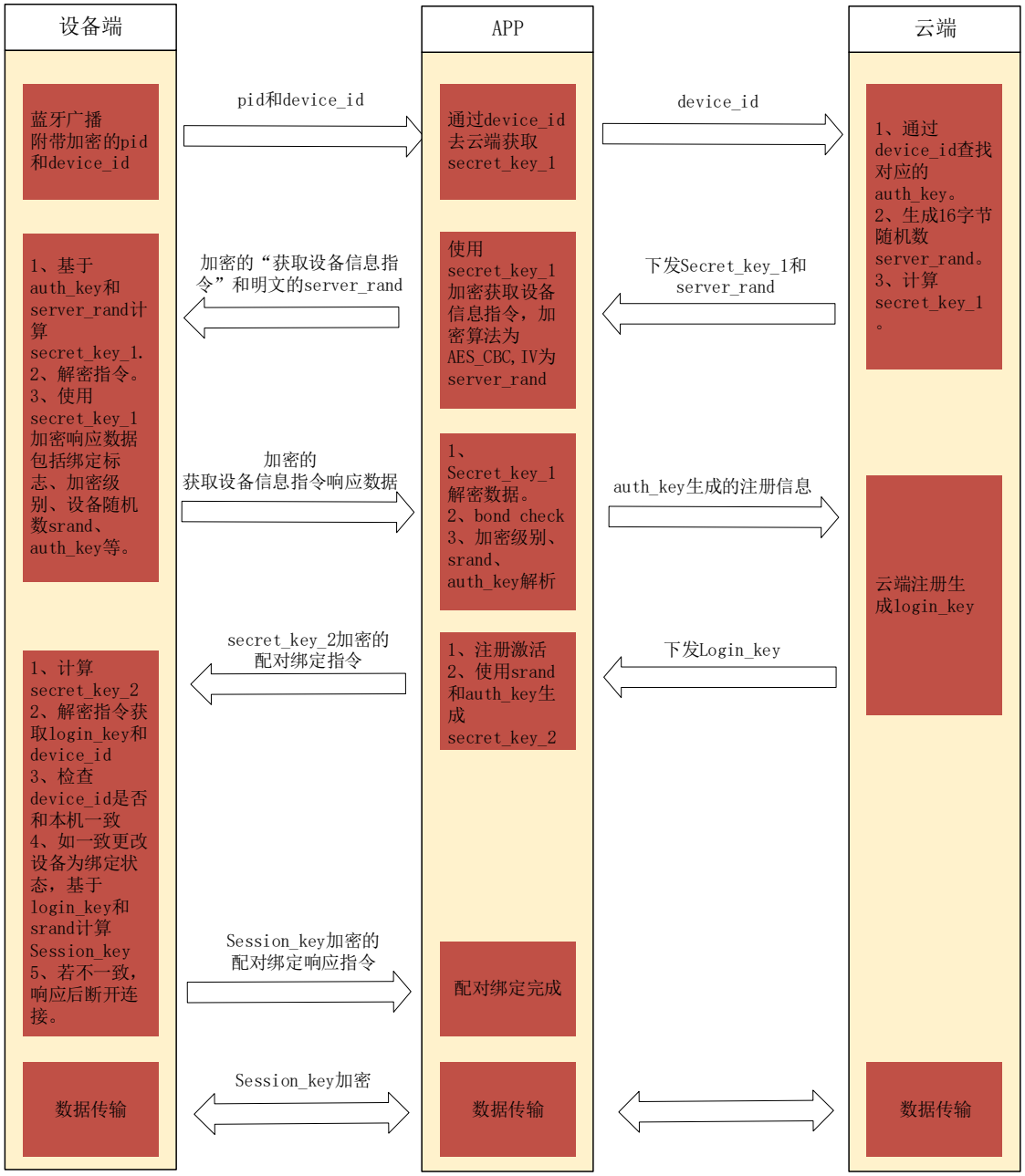
设备发现过程如下：BLE 设备在未连接状态下，会间断发送广播包。手机 APP 只对广播包中含涂鸦特定服务 UUID(A201)的蓝牙设备发送扫描请求处理，蓝牙设备在扫描广播应答包中发送设备的 product_id ， device_id 以及绑定标志和版本信息，手机APP 通过 device_id 从服务端拉取信息，通过product_id 从服务端拉取 UI 以及一些产品信息。

6 加密和配对

6.1 配对绑定流程

6.1.1 一般加密

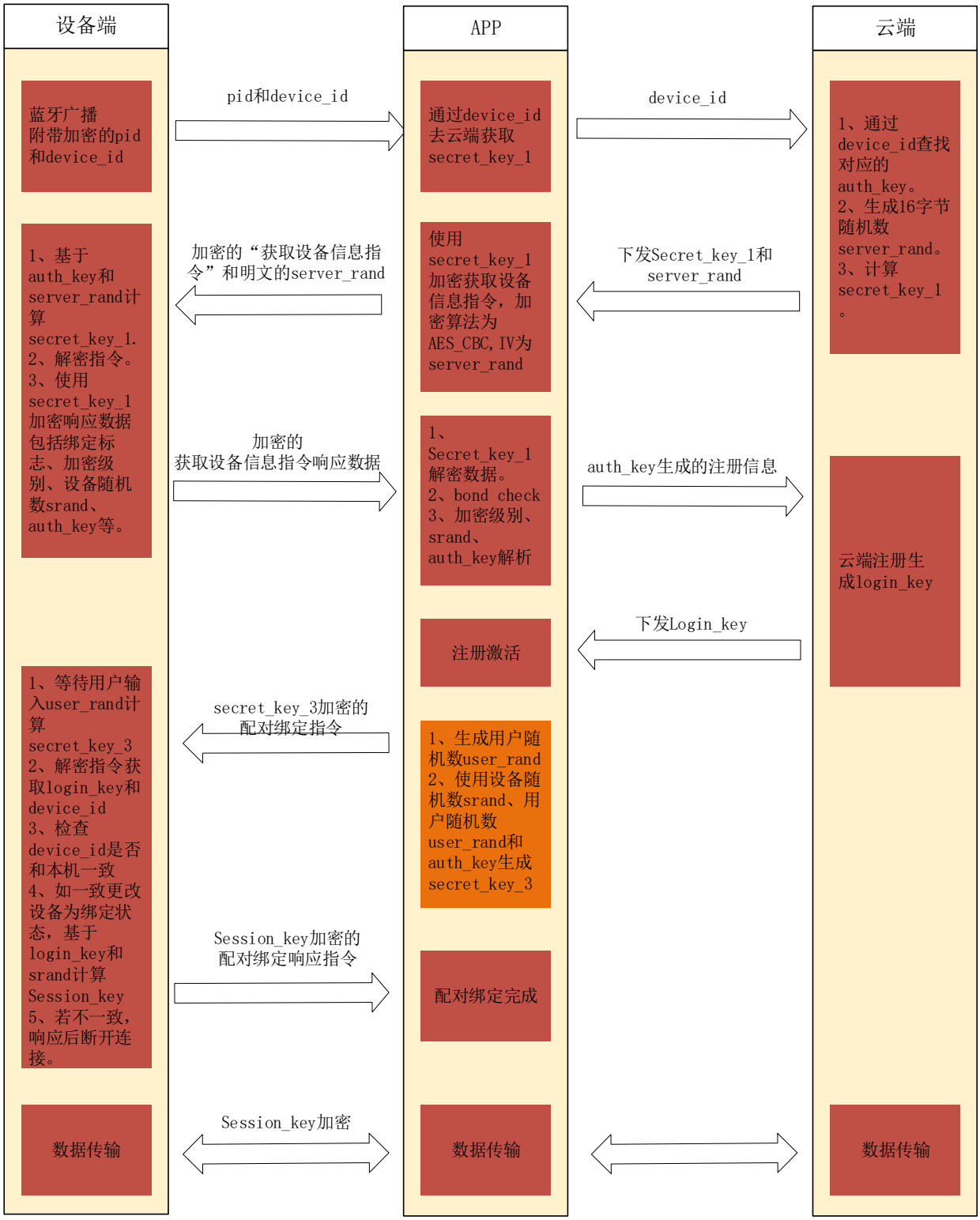
一般加密适用于对安全级别要求不高的设备。



1、secret_key_1 = AES_CBC(auth_key,server_rand)
2、secret_key_base = MD5(auth_key)
3、secret_key_2 = MD5(secret_key_base, srand)

6.1.2 高级加密

高级加密主要用于要求级别高的设备，比如蓝牙门锁，此种加密方式可以规避技术克隆设备进行攻击。



1、secret_key_1 = AES_CBC(auth_key,server_rand)
2、secret_key_base = MD5(auth_key)
3、secret_key_3 = MD5(secret_key_base,srand,user_rand)

6.1.3 加密安全

整个加密链路使用 device_uuid 与 auth_key 成对这种特质进行加密，避免了密码传输这一过程。

真正密钥使用的是 auth_key 和随机数的算法变化值，并不是原始值，保证了 auth_key 的安全。也避免了嵌入式需要存储 auth_key 的历史问题。

高加密方式，通过产生随机码，可以规避身份克隆攻击。克隆攻击流程大致为，产品已经绑定后，对设备进行克隆和重置配网，app 会对克隆设备重新绑定，绑定后关闭克隆设备，可直接与真实设备进行通信，完成了设备主人的身份变更，从而进行操作设备。

广播包攻击。由于绑定状态和加密等级存储在广播中，可以通过模拟广播进行更改绑定状态和加密等级从而进行攻击。协议中通过请求授权信息，根据信息中回复的内容，对广播中的信息进行二次 check，信息一致方可进行后续操作。

为了防止截取内容攻击，每次连接会有随机码加严。每次连接都会有不同的数据内容。

6.2 秘钥生成方式

秘钥 base：secret_key_base = MD5（auth_key)；

秘钥 1：secret_key_1 = AES_CBC(auth_key, server_rand),AES_CBC 的 IV 为全 0，key 为 auth_key 的前 16 个字节，server_rand 由云端生成，并在绑定时发送给设备，用于设备计算 secret_key_1。

秘钥 2：secret_key_2= MD5（secret_key_base,srand)，srand：设备每次连接产生的随机数。

秘钥 3：secret_key_3 = MD5(secret_key_base,srand,user_rand)，user_rand：用户随机数。

秘钥 4：secret_key_4 = MD5(login_key)。

秘钥 5：session_key = MD5(login_key,srand)，传输秘钥。

设备 ID（device_id）和授权码(auth_key)分别储存在设备端和云端，并且具有一一映射关系，在设备绑定前，设备会以广播的形式将设备 ID 的密文通知给 APP，APP 通过解密后的设备 ID 向云端请求 secret_key_1。

Srand：设备随机数，每次蓝牙连接时，app 通知设备产生一组随机数，app 使用 secret_key_1 加密的查询设备信息指令获取该 srand。

User_rand：用户随机数，该随机数由 APP 产生并显示在 app 界面上，让用户手动输入到设备上。

一般加密的设备通过 secret_key_2 加密 login_key 传输给设备。

高级加密设备通过 secret_key_3 加密 login_key 传输给设备。

对于已绑定的设备，每次新连接，app 通过 secret_key_4 加密“获取设备信息指令”获取 srand，之后所有的通信采用 session_key 加密。

6.3 加密算法

设备蓝牙广播包里的 device_id 采用 AES-CBC NoPadding 算法加密,加密秘钥为广播包里的 PID（加密或者非加密）做如下运算：KEY = MD5（PID），IV 等于 KEY；通信数据加密采用 AES-CBC NoPadding 。

6.4 配对绑定流程

6.4.1 一般加密设备绑定流程

1、手机 app 扫描设备广播包，并通过广播包中解密出来的 device_id 从服务器拉取对应的 secret_key_1，之后与设备建立连接。

2、app 发送通过 secret_key_1 加密的“获取设备信息”指令到设备。

3、设备根据自身绑定状态选用对应的解密方式解密指令，如果是已绑定状态选用 secret_key_4 解密指令，如果是未绑定状态选用 secret_key_1 解密指令，之后响应指令，响应指令采用 secret_key_1 或者 secret_key_4 加密，响应内容包含设备产生的 srand、设备版本号，协议版本号、设备安全等级以及设备绑定标志和 auth_key 等。

4、app 发起配对请求指令，配对请求指令中包含设备的 device_id 和 login_key，配对指令采用 secret_key_2 加密。

5、设备使用 secret_key_2 解密配对请求指令，首先判断其中的 device_id 是否和自身匹配，如果不匹配回复失败，并断开蓝牙连接，如果匹配回复绑定成功，响应指令采用 session_key 加密。

注：已绑定的设备通过 secret_key_4 加密发送“获取设备信息指令”来得到设备的 srand 。

6.4.2 高级加密设备绑定流程

1、手机 app 扫描设备广播包，并通过广播包中解密出来的 device_id 从服务器拉取对应的 secret_key_1，之后与设备建立连接。

2、app 发送通过 secret_key_1 加密的“获取设备信息”指令到设备。

3、设备根据自身绑定状态选用对应的解密方式解密指令，如果是已绑定状态选用 secret_key_4 解密指令，如果是未绑定状态选用 secret_key_1 解密指令，之后响应指令，响应指令采用 secret_key_4 或者 secret_key_1

加密，响应内容包含设备产生的 srand、设备版本号，协议版本号、auth_key，设备安全等级以及设备绑定标志等。

4、APP 根据设备响应包中的“设备安全等级标志为 1”来生成 6 位的用户随机码。

5、app 发起配对请求指令，配对请求指令中包含设备的 device_id 和 login_key，配对指令采用 secret_key_3 加密，发送成功后设置超时时间 30S。

6、设备根据数据包的加密标识判断为加密秘钥是 secret_key_3，此时设备等待用户输入 app 上产生的用户随机码，超时时间为 30S，收到用户的输入后生成 secret_key_3 解密配对请求指令，首先判断其中的 device_id 是否和自身匹配，如果不匹配回复失败，并断开蓝牙连接，如果匹配回复绑定成功，响应指令采用 session_key 加密。

6.5 解绑

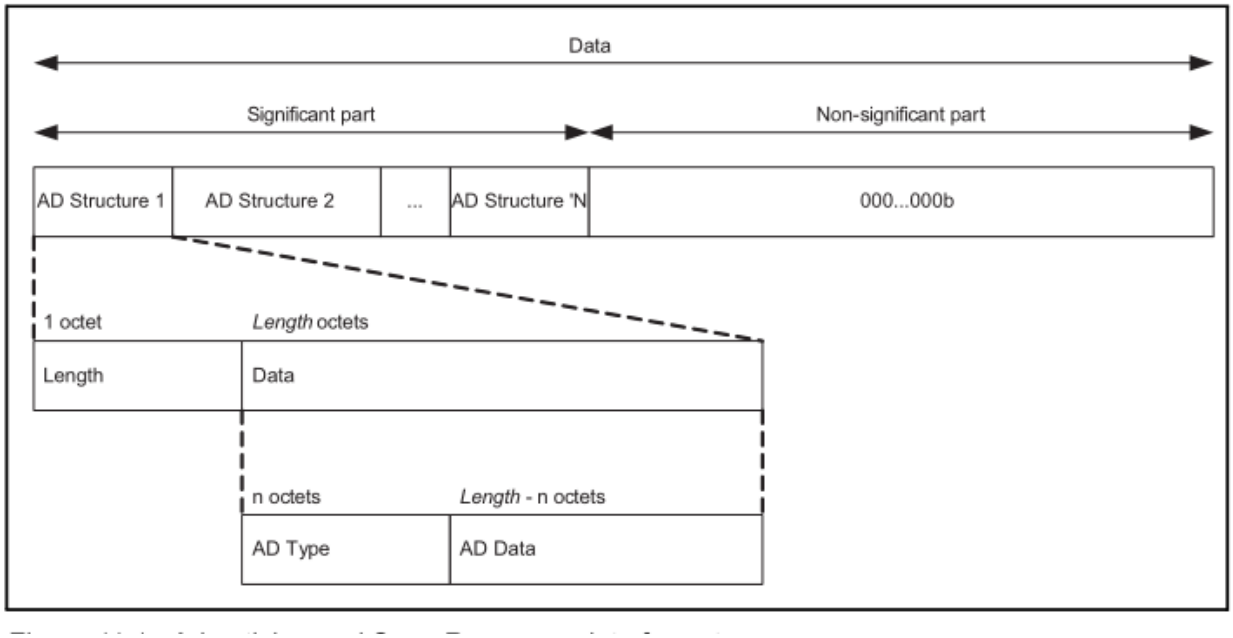
1、正常情况下 app 发送加密的解绑指令即可。

2、设备已被物理端重置解绑而 app 还在绑定中的的情况下，app 自动解绑之前的用户信息，并重新绑定流程。

3、设备在绑定状态，而 app 处于解绑状态：物理解绑或者 app 查询已存储在云端的信息，然后发送异常解绑指令解绑。

7 蓝牙广播格式

7.1 广播包的数据结构



7.2 广播包内容

广播数据段描述	类型	说明
设备 LE 物理连接标识	0x01	长度：0x02 类型：0x01 数据：0x06
Service UUID	0x02	长度：0x03 类型：0x02 数据：0xA201
Service Data （协议版本 3 才有该字段）	0x16	长度：0x0C 或者 0x14 类型：0x16 数据：0x01,0xA2,type(0-pid,1-product_key) PID 或者 product_key(8 或者 16 字节)
广播内容示例（8 字节 PID）:02 01 05 03 02 01 A2 0C 16 01 A2 00 00 00 00 00 00 00 00 00		

注：PID/PRODUCT_KEY：未绑定设备该字段存放明文 PID 或 PRODUCT KEY，已绑定设备该字段存放生成 device uuid 的加密 key 所需的元素 Elements，Elements 生成算法为 AES-ECB（device_id），AES-ECB key 由 login_key 的全部 6 个字节和 device_uuid 的前 10 个字节组成。

7.3 扫描响应包内容

<table>
<tr><td>广播数据段描述</td><td>类型</td><td>说明</td></tr>
<tr><td>Complete Local Name</td><td>0x09</td><td>长度：0x03
类型：0x09
数据：0x54，0x59</td></tr>
<tr><td>厂商自定义数据</td><td>0xff</td><td>长度：0x19
类型：0xff
数据 ：
COMPANY ID:0x07D0
FLAG：0x00
协议版本：0x03
加密方式：0x00
通信能力：0x0000
保留字段：0x00
ID 字段: 16 个字节或 6 字节</td></tr>
<tr><td colspan="3">广播内容示例（未绑定）：03 09 54 59 19 FF D0 07 00 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00</td></tr>
</table>

Bong flag：bit7：绑定标志（1 绑定），其他位保留。
协议版本：协议版本号（如果没有该标识，则默认版本号 1，目前是 3）。
加密方式：0x00-基于 auth key 和 device id 的加密；0x01-基于 ECC 算法加密；0x02-不加密透传通道。
通信能力(2 个字节大端格式，高字节在前)：

bit0-表示是否通过 BLE 注册，1 表示通过 BLE 注册绑定设备，0 表示其他通道。
Bit1-表示是否支持 MESH，1 表示支持，0 表示不支持。
Bit2-表示是否具有 wifi2.4G 通信能力，1 表示有，0 表示没有。
Bit3-表示是否具有 wifi5G 通信能力，1 表示有，0 表示没有。
Bit4-表示是否具有 zigbee 通信能力，1 表示有，0 表示没有。
Bit5-表示是否具有 NB 通信能力，1 表示有，0 表示没有。
其他位保留。

ID 字段：加密方式是 0x00，ID 字段存放 16 字节 DEVCIE UUID；如果是 0x01：该字段存放 6 字节 MAC 地址。
DEVCIE UUID：设备 UUID，16 字节，加密广播，加密算法为 AES-CBC,加密 key 由广播包中的 service data 字段中的 Elements 做 MD5 运算生成。
MAC 地址：明文。

8 分包数据编码

8.1 分包数据结构

由于蓝牙 4.0 每帧只能发送 20 字节数据，为了兼容 4.0 的蓝牙芯片，目前采用 20 字节一帧的方式通信，而每个指令包的数据长度很容易大于 20 个字节，故而采用分包的传输方式，目前限定大包总的长度不超过 256 字节。

分包结构：

包头				数据部分
分包序号（<=4B）	帧数据长度（<=4B）	协议版本标识（4bit）	保留（4bit）	
采用类似 MQTT 可变长度描述	采用类似 MQTT 可变长度描述，表示后续帧数据长度	协议版本号	保留	N

协议版本标识：1/2/3等。

当数据在分包处理传输时，帧数据长度、帧标识和加密方式只有是第一包时才有意义，其它包的情况下为空。

包序号以及包长度，单字节最大值为 01111111(127)，若第八位(最高位)为 1，表明还有后续字节存在。包序号或包长度最多只有 4 字节长度表示，所以最大四字节包序号或包长度： 0xff 0xff 0xff 0x7f(十进制:268435455 包或字节数)。

字节数	起始	末尾
1	0(00)	127(7f)
2	128(80 01)	16383(ff 7f)
3	16384(80 80 01)	2097151(ff ff 7f)
4	2097152(80 80 80 01)	268435455(ff ff ff 7f)

注：128 进制倒序表示法，例如 0x8001 = 0 + 1x(0x80) = 128； 0xff7f = 0x7f + 0x7f x 0x80 = 16383;

8.2 数据部分

数据部分包括 1 个字节的加密标识，16 个字节的加密 IV 以及“3.1 节描述的帧数据”的明文或者密文。

1 字节	16 字节	N 字节
加密 FLAG	AES-CBC IV	帧数据的明文或者密文

加密 FLAG： 0- 未加密，明文；

1- secret_key_1；

2- secret_key_2；

3- secret_key_3；

4- secret_key_4；

5- session_key；

6- ECDH 协商的 key

AES-CBC IV：每次加密用随机数生成，明文传输，如果加密 FLAG 为 0，则没有 IV 域存在。

9 功能码描述

9.1 查询设备信息： 0x0000

命令帧：APP->BLE：

序号	长度	字段	说明
1	4	SN	X
2			
3			
4			
5	4	ACK_SN	0
6			
7			
8			
9	2	0x0000	功能码
10			
11	2	0x0000	数据长度
12			
13	2	CRC16	检验
14			

响应帧：BLE->APP

序号	长度	字段	说明
1	4	SN	Y
2			
3			
4			
5	4	ACK_SN	X
6			
7			
8			
9	2	0x0000	功能码
10			

11 12	2	0x0053	数据长度
……	0x0053	Data	见下表格式
	2	CRC16	检验

Data 格式：

1	2	3	4	5	6	7-12	13	14	15-46
固件版本号 1		协议版本号		Flag	Bond	Srand (6 个字节)	硬件版本 1		Auth_key(32 个字节)
47	48	49	50	51	52	53-54	55	56-77	
固件版本号 2			硬件版本号 2			通信能力	保留	Device virtual id(22 字节)	
78	79	80	81	82	83				
mcu 固件版本号			mcu 硬件版本号						

硬件版本号 2：例如 0x010000 代表 v1.1.0；硬件版本号 1：取硬件版本号 2 的高 2 位，例如 v1.1。
固件版本号 2：例如 0x010200 代表 v1.2.0；固件版本号 1：取固件版本号 2 的高 2 位，例如 V1.2。
MCU 固件版本号：例如 0x010000 代表 v1.1.0，没有外部 mcu 或者有外部 mcu 但是没获取到版本号，则填 0 。
MCU 硬件版本号：同 MCU 固件版本号。
该指令协议版本号：目前是 0x0301 (v3.1)。

Flag： bit2：0-使用固件版本号 1, 1-使用固件版本号 2。
bit1：0-一般加密设备 ， 1-高级加密设备。
bit0: 0-使用硬件版本号 1， 1-使用硬件版本号 2。

Bond：0-未绑定，1-已绑定。
Srand：设备当前有效随机数,6 字节。
通信能力(2 个字节大端格式，高字节在前)：
bit0-表示是否通过 BLE 注册，1 表示通过 BLE 注册绑定设备，0 表示其他通道。
Bit1-表示是否支持 MESH，1 表示支持，0 表示不支持。
Bit2-表示是否具有 wifi2.4G 通信能力，1 表示有，0 表示没有。
Bit3-表示是否具有 wifi5G 通信能力，1 表示有，0 表示没有。
Bit4-表示是否具有 zigbee 通信能力，1 表示有，0 表示没有。
Bit5-表示是否具有 NB 通信能力，1 表示有，0 表示没有。
其他位保留。
Device virtual id：22 字节，设备虚拟 id，设备如果没有此虚拟 ID，填 0 发送。

9.2 发起配对请求：0x0001

命令帧：APP->BLE：

序号	长度	字段	说明
1	4	SN	X
2			
3			
4			

5 6 7 8	4	ACK_SN	0
9 10	2	0x0001	功能码
11 12	2	0x002C	数据长度
	44 字节	Data	数据区
	2	CRC16	检验

Data 格式：

1-16	17-22	23-44
Device_id	Login_key	Device virtual id

Device virtual id：设备虚拟 ID，22 字节，设备每次被绑定时存储此虚拟 ID，被重置时删除此虚拟 ID。

响应帧：BLE->APP

序号	长度	字段	说明
1 2 3 4	4	SN	Y
5 6 7 8	4	ACK_SN	X
9 10	2	0x0001	功能码
11 12	2	0x0001	数据长度
……	1	Data	见下表格式
	2	CRC16	检验

Data 格式：

1
状态码

状态码：0-成功；1-失败；2-已经在绑定状态

9.3 命令下发：0x0002

命令帧：APP->BLE：

序号	长度	字段	说明
1	4	SN	X
2			
3			
4			
5	4	ACK_SN	0
6			
7			
8			
9	2	0x0002	功能码
10			
11	2	n	数据长度
12			
	n	Data	dp 组数据
	2	CRC16	检验

Data 格式：

Dp 数组 1 的数据				~	Dp 数组 n 的数据			
1	2	3	3~	~	n	n+1	n+2	n+3~
Dp_id	Dp_type	Dp_len	Dp_data	~	Dp_id	Dp_type	Dp_len	Dp_data

Dp_id： 1 个字节，在开发平台注册的 dp_id 序号。

Dp_type：1 个字节。

#define DT_RAW 0

#define DT_BOOL 1

#define DT_VALUE 2

#define DT_STRING 3

#define DT_ENUM 4

#define DT_BITMAP 5

Dp_len：1 个字节，数据长度，最大 255

Dp_data：数据，dp_len 个字节。

响应帧：BLE->APP

序号	长度	字段	说明
1	4	SN	Y
2			
3			
4			
5	4	ACK_SN	X
6			
7			
8			
9	2	0x0002	功能码
10			
11	2	0x0001	数据长度
12			
……	1	Data	见下表格式
	2	CRC16	检验

Data 格式：

1
状态码

状态码：0-成功；其他-失败

9.4 设备状态查询 0x0003

命令帧：APP->BLE：

序号	长度	字段	说明
1	4	SN	X
2			
3			
4			
5	4	ACK_SN	0
6			
7			
8			
9	2	0x0003	功能码
10			
11	2	n	数据长度
12			
	n	Data	Dp 点序号组合

	2	CRC16	检验

如果 n=0，代表查询所有 dp 点数据。

如果 n>0，代表查询 n 个 dp 点的数据，这 n 个 dp 点序号在 data 数据区。

响应帧：BLE->APP：

序号	长度	字段	说明
1	4	SN	Y
2			
3			
4			
5	4	ACK_SN	X
6			
7			
8			
9	2	0x0003	功能码
10			
11	2	0x0001	数据长度
12			
	1	data	
	2	CRC16	检验

Data 格式： 0-成功，1-dp 点个数超出注册的数量，2-其他错误。

状态数据通过“状态上报指令发送”。

9.5 设置登入密码 0x0004

暂不使用。

9.6 设备解绑 0x0005

命令帧：APP->BLE：

序号	长度	字段	说明
1	4	SN	X
2			
3			

4			
5	4	ACK_SN	0
6			
7			
8			
9	2	0x0005	功能码
10			
11	2	0x0000	数据长度
12			
13	2	CRC16	检验
14			

响应帧：BLE->APP

序号	长度	字段	说明
1	4	SN	Y
2			
3			
4			
5	4	ACK_SN	X
6			
7			
8			
9	2	0x0005	功能码
10			
11	2	0x0001	数据长度
12			
……	1	Data	见下表格式
	2	CRC16	检验

Data 格式：

1
状态码

状态码：0-成功；其他-失败

9.7 设备重置 0x0006

命令帧：APP->BLE：

序号	长度	字段	说明
1	4	SN	X
2			
3			
4			
5	4	ACK_SN	0
6			
7			
8			
9	2	0x0006	功能码
10			
11	2	0x0000	数据长度
12			
13	2	CRC16	检验
14			

响应帧：BLE->APP

序号	长度	字段	说明
1	4	SN	Y
2			
3			
4			
5	4	ACK_SN	X
6			
7			
8			
9	2	0x0006	功能码
10			
11	2	0x0001	数据长度
12			
……	1	Data	见下表格式
	2	CRC16	检验

Data 格式：

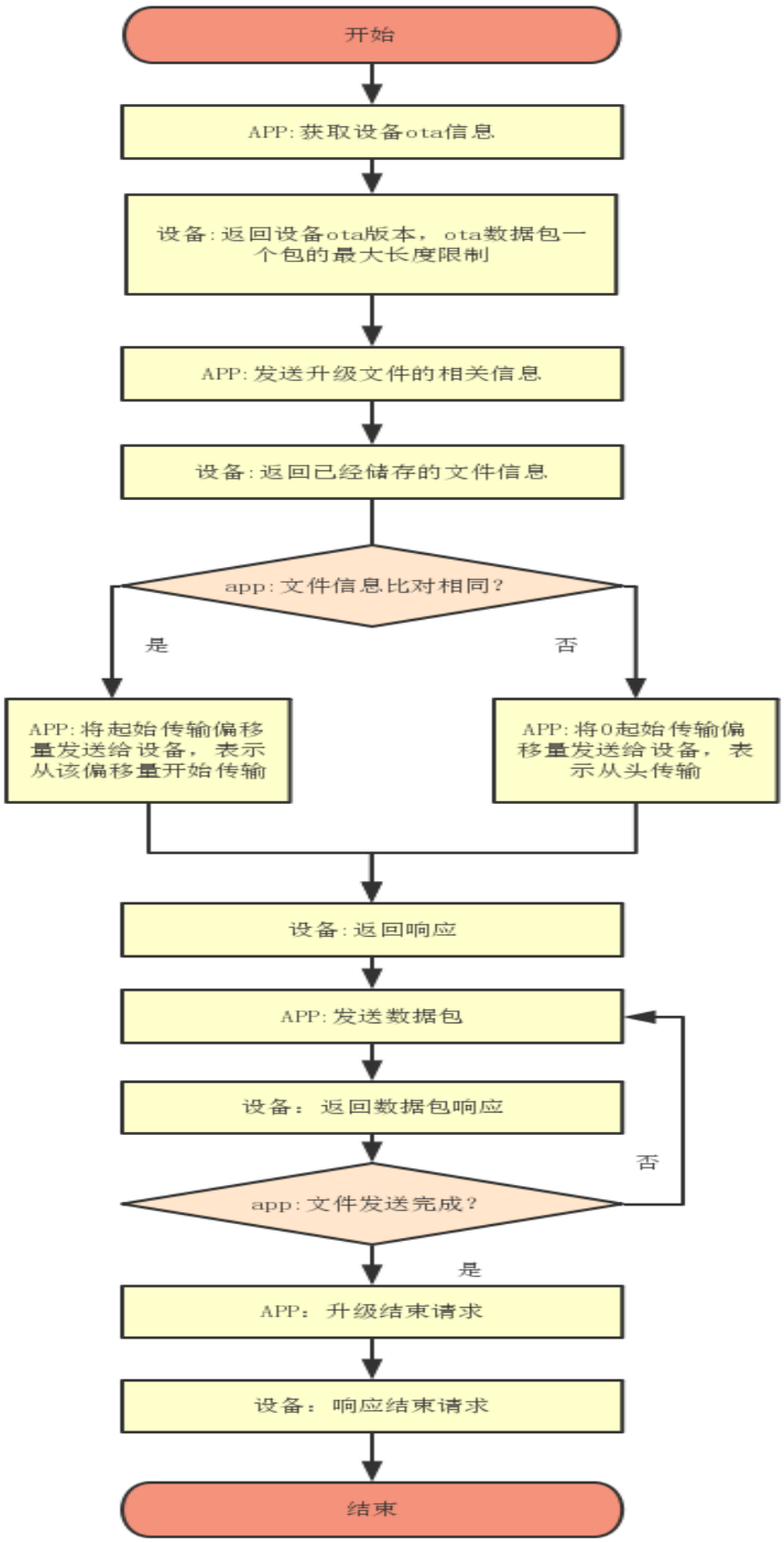
1

状态码

状态码：0-成功；其他-失败

设备收到该重置指令后，解绑设备并清除 login key 和虚拟 ID。

9.8 OTA 升级流程


开始
APP:获取设备ota信息
设备:返回设备ota版本，ota数据包一个包的最大长度限制
APP:发送升级文件的相关信息
设备:返回已经储存的文件信息
app:文件信息比对相同?
是
否
APP:将起始传输偏移量发送给设备，表示从该偏移量开始传输
APP:将0起始传输偏移量发送给设备，表示从头传输
设备:返回响应
APP:发送数据包
设备：返回数据包响应
否
app:文件发送完成?
是
APP：升级结束请求
设备：响应结束请求
结束

9.9 OTA 升级请求： 0x000C

命令帧：APP->BLE：

序号	长度	字段	说明
1	4	SN	X
2			
3			
4			
5	4	ACK_SN	0
6			
7			
8			
9	2	0x000C	功能码
10			
11	2	0x0001	数据长度
12			
13	1	TYPE	见下表
14	2	CRC16	检验
15			

TYPE：0 – BLE 升级固件，1 – 外部 mcu 升级固件，其他 – 保留。

响应帧：BLE->APP

序号	长度	字段	说明
1	4	SN	Y
2			
3			
4			
5	4	ACK_SN	X
6			
7			
8			
9	2	0x000C	功能码
10			
11	2	n	数据长度 2 个字节
12			
……	n	Data	见下表格式
	2	CRC16	检验

Data 格式：

1	2	3	4-n
Flag	OTA_Version	TYPE	Type_data

Flag：0x00-允许升级，0x01-拒绝升级。

OTA_Version：OTA 协议大版本，例如 0x03 代表 3.X 的协议版本。

TYPE：0 – BLE 升级固件，1 – 外部 mcu 升级固件，其他 – 保留。

Type_data：不同的 type 对应不同格式的 type_data，TYPE 为 0 和 1 对应的 type_data 格式如下所示：

1	2	3	4	5	6
Version 四字节				最大包长度	

Version：当前固件版本号，例如 0x00 01 00 02 代表版本为 V1.0.2。

最大包长度：设备允许的单包最大长度，单位字节。

9.10 OTA 升级文件信息： 0x000D

命令帧：APP->BLE：

序号	长度	字段	说明
1	4	SN	X
2			
3			
4			
5	4	ACK_SN	0
6			
7			
8			
9	2	0x000D	功能码
10			
11	2	0x0025	数据长度
12			
	37	DATA	升级文件信息
	2	CRC16	检验

DATA 格式：

1 字节	8 字节	4 字节	16 字节	4 字节	4 字节
TYPE	产品 PID	文件版本	文件 MD5	文件长度	CRC32

TYPE：0 – BLE 升级固件，1 – 外部 mcu 升级固件，其他 – 保留，不同的 TYPE，后面对应不同的数据格式，上表为 TYPE=0 和 1 对应的格式

文件版本：例如，0x00010002 代表版本为 V1.0.2。

响应帧：BLE->APP

序号	长度	字段	说明
1	4	SN	Y
2			
3			
4			
5	4	ACK_SN	X
6			
7			
8			
9	2	0x000D	功能码
10			
11	2	n	
12			
……	n	Data	见下表格式
	2	CRC16	检验

Data 格式：

1 字节	1 字节	4 字节	4 字节	16 字节
TYPE	STATE	已储存文件长度	已储存文件 CRC32	已储存文件 MD5 （目前不使用）

TYPE：0 – BLE 升级固件，1 – 外部 mcu 升级固件，其他 – 保留，不同的 TYPE，后面对应不同的数据格式，上表为 TYPE=0 和 1 对应的格式。

STATE：

0x00：正常升级

0x01：产品 PID 不一致

0x02: 文件版本低于或者等于当前版本

0x03：文件大小超过范围。

其他：保留。

已储存文件信息：

说明：为了支持断点续传，这里会返回设备端已经储存的文件信息，APP 在收到后，首先根据设备返回的已储存文件长度计算新文件对应长度的 CRC32，然后和设备返回的 CRC32 对比，如果两者都吻合，那么在下面的文件起始传输请求中将起始传输偏移量改为该长度值，否则文件起始传输偏移量改为 0，表示从头开始传输。

9.11 OTA 升级文件偏移请求：0x000E

命令帧：APP->BLE：

序号	长度	字段	说明
1	4	SN	X
2			
3			
4			
5	4	ACK_SN	0
6			
7			
8			
9	2	0x000E	功能码
10			
11	2	0x0005	数据长度
12			
	5	TYPE+Offset	TYPE+偏移量
	2	CRC16	检验

TYPE：0 – BLE 升级固件，1 – 外部 mcu 升级固件，其他 – 保留
offset：文件起始传输偏移量，四字节。

响应帧：BLE->APP

序号	长度	字段	说明
1	4	SN	Y
2			
3			
4			
5	4	ACK_SN	X
6			
7			
8			
9	2	0x000E	功能码
10			

11 12	2	0x0005	
……	5	TYPE+offset	
	2	CRC16	检验

TYPE：0 – BLE 升级固件，1 – 外部 mcu 升级固件，其他 – 保留

offset：设备要求的起始传输文件偏移量。

说明：实际文件传输的偏移地址应该以设备端要求的为准，且设备端要求的地址会小于等于 APP 端给出的偏移。

9.12 OTA 升级数据： 0x000F

命令帧：APP->BLE：

序号	长度	字段	说明
1 2 3 4	4	SN	X
5 6 7 8	4	ACK_SN	0
9 10	2	0x000F	功能码
11 12	2	7+n	数据长度
	7+n	DATA	1字节type+2字节包号+2字节长度+2字节CRC16+当前包数据
	2	CRC16	检验

DATA 格式：

1 字节	2 字节	2 字节	2 字节	n 字节
TYPE	包号	当前包数据长度 n	当前包数据 CRC16	当前包数据

TYPE：0 – BLE 升级固件信息，1 – 外部 mcu 升级固件信息，其他 – 保留。

响应帧：BLE->APP

序号	长度	字段	说明
1	4	SN	Y
2			
3			
4			
5	4	ACK_SN	X
6			
7			
8			
9	2	0x000F	功能码
10			
11	2	0x0002	
12			
……	2	Data	见下表格式
	2	CRC16	检验

Data 格式：

1	2
TYPE	STATE

TYPE：0 – BLE 升级固件信息，1 – 外部 mcu 升级固件，其他 – 保留。

STATE：

0x00：成功

0x01：包号异常

0x02：长度不一致。

0x03：crc 检验失败

0x04：其它

9.13 OTA 升级结束：0x0010

命令帧：APP->BLE：

序号	长度	字段	说明
1	4	SN	X
2			
3			
4			

5	4	ACK_SN	0
6			
7			
8			
9	2	0x0010	功能码
10			
11	2	0x0001	数据长度
12			
13	1	TYPE	
14	2	CRC16	校验
15			

TYPE：0 – BLE 升级固件信息，1 – 外部 mcu 升级固件，其他 – 保留。

响应帧：BLE->APP：

序号	长度	字段	说明
1	4	SN	Y
2			
3			
4			
5	4	ACK_SN	X
6			
7			
8			
9	2	0x0010	功能码
10			
11	2	0x0002	
12			
……	2	Data	见下表格式
	2	CRC16	检验

Data 格式：

1	2
TYPE	STATE

TYPE：0 – BLE 固件信息，1 – 外部 mcu 升级固件，其他 – 保留。

STATE：

0x00：成功

0x01：数据总长度错误

0x02: 数据总 CRC 检验失败

0x03:其它

9.14 BLE 通道产测指令：0x0012

数据帧：APP<->BLE

序号	长度	字段	说明
1	4	SN	Y
2			
3			
4			
5	4	ACK_SN	X
6			
7			
8			
9	2	0x0012	功能码
10			
11	2	n	
12			
……	n	Data	见下表格式
	2	CRC16	检验

Data 格式：

N 个字节
55AA 完整产测指令或者响应

9.15 保留：0x0013

9.16 设备异常解绑 0x0014

命令帧：APP->BLE：

序号	长度	字段	说明
1	4	SN	X
2			
3			
4			
5	4	ACK_SN	0
6			
7			
8			
9	2	0x0014	功能码
10			
11	2	0x0000	数据长度
12			
13	2	CRC16	检验
14			

响应帧：BLE->APP

序号	长度	字段	说明
1	4	SN	Y
2			
3			
4			
5	4	ACK_SN	X
6			
7			
8			
9	2	0x0014	功能码
10			
11	2	0x0001	数据长度
12			
……	1	Data	见下表格式
	2	CRC16	检验

Data 格式：

1
状态码

状态码：0-成功；其他-失败。

9.17 ECDH 密钥协商：0x0020

命令帧：APP->BLE：

序号	长度	字段	说明
1	4	SN	X
2			
3			
4			
5	4	ACK_SN	0
6			
7			
8			
9	2	0x0020	功能码
10			
11	2	n	数据长度
12			
	n	DATA	数据
	2	CRC16	检验

DATA 格式：

1	64 or n
CURVE TYPE	PUBLIC KEY

CURVE TYPE：加密曲线类型

0 - secp160r1

1 - secp192r1

2 - secp224r1

3 - secp256r1

4 - secp256k1

PUBLIC KEY：公钥。

响应帧：BLE->APP

序号	长度	字段	说明
1	4	SN	Y
2			
3			
4			
5	4	ACK_SN	X
6			
7			
8			

9 10	2	0x0020	功能码
11 12	2	n	数据长度
……	n	DATA	见下表格式
	2	CRC16	检验

DATA：同请求帧。

9.18 发送配网信息：0x0021

命令帧：APP->BLE：

序号	长度	字段	说明
1 2 3 4	4	SN	X
5 6 7 8	4	ACK_SN	0
9 10	2	0x0021	功能码
11 12	2	n	数据长度
	n	DATA	字符串
	2	CRC16	检验

DATA 格式：如下所示的字符串

```
{
   “mac”:”aabbccddeeff”
   “ssid”:”tuya_test”,
   “passwd”:”12345678”,
   “token”:”AYaawossptx3ma”
}
```

采用 json 无格式字符串，包含结束符。

响应帧：BLE->APP

序号	长度	字段	说明
1	4	SN	Y
2			
3			
4			
5	4	ACK_SN	X
6			
7			
8			
9	2	0x0021	功能码
10			
11	2	0x0001	数据长度
12			
……	1	Data	见下表格式
	2	CRC16	检验

Data 格式：

1
STATE

state：0 – 接收成功，1 -接收失败。

9.19 上报配网状态：0x0022

命令帧：BLE->APP：

序号	长度	字段	说明
1	4	SN	X
2			
3			
4			
5	4	ACK_SN	0
6			
7			
8			
9	2	0x0022	功能码
10			
11	2	0x0002	数据长度
12			

13-14	2	result	结果码
15	2	CRC16	检验
16			

Result：两字节结果码，蓝牙透传，值由 wifi 端定义。

响应帧：APP->BLE

序号	长度	字段	说明
1	4	SN	Y
2			
3			
4			
5	4	ACK_SN	X
6			
7			
8			
9	2	0x0022	功能码
10			
11	2	0x0001	数据长度
12			
13	1	Status	见下表格式
14	2	CRC16	检验
15			

Data 格式：

1
Status

Status：0 – 接收成功，1 -接收失败。

9.20 数据透传：0x0023

命令帧：APP<->BLE：

序号	长度	字段	说明
1	4	SN	X
2			
3			
4			
5	4	ACK_SN	0
6			
7			
8			

9 10	2	0x0023	功能码
11 12	2	n	数据长度
	n	DATA	透传数据
	2	CRC16	检验

DATA ：透传数据，不解析。

响应帧：无。

9.21 唤醒指令：0x0024（目前仅用于 IPC）

命令帧：APP<->BLE：

序号	长度	字段	说明
1 2 3 4	4	SN	X
5 6 7 8	4	ACK_SN	0
9 10	2	0x0024	功能码
11 12	2	0	数据长度
13 14	2	CRC16	检验

响应帧：无。

9.22 状态上报：0x8001

命令帧：BLE->APP：

序号	长度	字段	说明
1 2 3 4	4	SN	Y

5 6 7 8	4	ACK_SN	X
9 10	2	0x8001	功能码
11 12	2	n	数据长度
	n	Data	dp 组数据
	2	CRC16	检验

Data 格式：

Dp 数组 1 的数据				~	Dp 数组 n 的数据			
1	2	3	4~	~	n	n+1	n+2	n+3~
Dp_id	Dp_type	Dp_len	Dp_data	~	Dp_id	Dp_type	Dp_len	Dp_data

Dp_id： 1 个字节，在开发平台注册的 dp_id 序号。

Dp_type：1 个字节。

#define DT_RAW 0

#define DT_BOOL 1

#define DT_VALUE 2

#define DT_STRING 3

#define DT_ENUM 4

#define DT_BITMAP 5

Dp_len：1 个字节，数据长度，最大 255

Dp_data：数据，dp_len 个字节。

响应帧：APP->BLE：

序号	长度	字段	说明
1 2 3 4	4	SN	Y
5 6 7 8	4	ACK_SN	X
9 10	2	0x8001	功能码

11 12	2	0x0001	
……	1	Data	见下表格式
	2	CRC16	检验

Data 格式：

1
STATE

STATE：

0x00：成功

0x01：失败

9.23 设备调试信息显示：0x8002

命令帧：BLE->APP：

序号	长度	字段	说明
1 2 3 4	4	SN	X
5 6 7 8	4	ACK_SN	0
9 10	2	0x8002	功能码
11 12	2	n	数据长度
	n	Data	string
	2	CRC16	检验

Data 格式：string 字符串，例如“hello from tuya ble”。

响应帧：APP->BLE：

序号	长度	字段	说明
1	4	SN	Y
2			
3			
4			
5	4	ACK_SN	X
6			
7			
8			
9	2	0x8002	功能码
10			
11	2	0x0001	
12			
……	1	Data	见下表格式
	2	CRC16	检验

Data 格式：

1
STATE

STATE：

0x00：成功显示 。

0x01：失败。

9.24 带时间戳状态数据上报：0x8003

命令帧：BLE->APP：

序号	长度	字段	说明
1	4	SN	Y
2			
3			
4			
5	4	ACK_SN	X
6			
7			
8			
9	2	0x8003	功能码
10			
11	2	n	数据长度
12			

	n	Data	dp 组数据
	2	CRC16	检验

Data 格式：

时间戳		Dp 数组 1 的数据				~	Dp 数组 n 的数据			
1	n	1	2	3	3~	~	n	n+1	n+2	n+3~
timeType	time	Dp_id	Dp_type	Dp_len	Dp_data	~	Dp_id	Dp_type	Dp_len	Dp_data

TimeType：0 表示后面传输的数据是 13 字节 ms 级时间字符串，例如”1553932355000”，13 字节 ms 级，2019/3/30 15:52:35

1 表示后面 time 是 4 字节 unix 秒级时间戳，大端格式传输。

Dp_id： 1 个字节，在开发平台注册的 dp_id 序号。

Dp_type：1 个字节。

#define DT_RAW 0

#define DT_BOOL 1

#define DT_VALUE 2

#define DT_STRING 3

#define DT_ENUM 4

#define DT_BITMAP 5

Dp_len：1 个字节，数据长度，最大 255

Dp_data：数据，dp_len 个字节。

响应帧：APP->BLE：

序号	长度	字段	说明
1	4	SN	Y
2			
3			
4			
5	4	ACK_SN	X
6			
7			
8			
9	2	0x8003	功能码
10			
11	2	0x0001	
12			
……	1	Data	见下表格式

	2	CRC16	检验

Data 格式：

1
STATE

STATE：

0x00：成功

0x01：失败

9.25 设备获取实时时间 1：0x8011

命令帧：BLE->APP：

序号	长度	字段	说明
1	4	SN	X
2			
3			
4			
5	4	ACK_SN	0
6			
7			
8			
9	2	0x8011	功能码
10			
11	2	0x0000	数据长度
12			
13	2	CRC16	校验
14			

响应帧：APP->BLE：

序号	长度	字段	说明
1	4	SN	Y
2			
3			
4			
5	4	ACK_SN	X
6			
7			
8			

9 10	2	0x8011	功能码
11 12	2	15	
……	15	Data	见下表格式
	2	CRC16	检验

Data 格式：

13 字节字符串	2 字节时区（有符号型）
Unix ms 级时间	Time_zone

时区：实际时区的 100 倍，例如北京东八区为 8x100=800，西 7.5 区为-750；

注：如果 app 从云端获取不到时间，app 需要回复全 0 数据（二进制 0）。

9.26 设备获取实时时间 2：0x8012

命令帧：BLE->APP：

序号	长度	字段	说明
1 2 3 4	4	SN	X
5 6 7 8	4	ACK_SN	0
9 10	2	0x8012	功能码
11 12	2	0x0000	数据长度
13 14	2	CRC16	校验

响应帧：APP->BLE：

序号	长度	字段	说明
1 2 3 4	4	SN	Y

5 6 7 8	4	ACK_SN	X
9 10	2	0x8012	功能码
11 12	2	9	9 字节
……	9	Data	见下表格式
	2	CRC16	检验

Data 格式：

1 字节	1 字节	1 字节	1 字节	1 字节	1 字节	1 字节	2 字节（有符号型）	
年	月	日	时	分	秒	星期	时区高	时区低

注：如 2019 年 4 月 28 日 12:23:25 星期二， 应按如下顺序发送：0x13,0x04,0x1C,0x0C,0x17,0x19,0x02。

星期：0-6 , 0 代表星期日。

时区：实际时区的 100 倍，例如北京东八区为 8x100=800，西 7.5 区为-750；

注：如果 app 从云端获取不到时间，app 需要回复全 0 数据（二进制 0）。

附录：