Bluetooth Communication Protocol V1.0

Content

The Bluetooth of the lock will communicate with IOS/Android application by this Bluetooth Communication Protocol. This document will explain the protocol according to the byte serial numbers. The fixed parts will be highlight in grey color, the variable parts in yellow color, optional or invalid in light blue color. Please note that all the byte date is presented in hex. Array index is started from 0, like the C Programming Language.

1 Broadcasting data

IO and Android application can receive the date and filter devices promptly via broadcasting data by lock Bluetooth. The Local Name value is allocated and managed by JimiLab, the default value is KKS Lock. Manufacturer value has included core data, its byte sequence definition is below:

0	1	2	3	4	5	6	7
ID[2]		MAC[6]					

The lock ID (the first two bits) are allocated and managed by JimiLab, default is 0102. This ID can fliter lock Bluetooth. The next 6 bits are the Bluetooth Mac address, the sequence method:
MAC[5]MAC[4]MAC[3]MAC[2]MAC[1]MAC[0]

The Service UUIDs have to containFEE7, UUID with the 16 bits.

The broadcasting data will display as below:

KKSLOCK
Local Name
<0102ffff c00004b2 06640001 10>
Manufacturer Data
FEE7
Service UUIDs

2 UUID

the Service UUID of the lock Bluetooth is FEE7.

The Characteristic UUIDs:

Eigenvalue	36F5	36F6
Property	Write	Read Notify

The lengths of 36F5 and 36F6 are fixed to 16 bits. IOS/Android applications will communicate with locks by 36F5 and 36F6.

3 Communication Protocol

To simplify the protocol, IOS and Android Application will be named as Host for short. The Host and Lock communicate by the basic communication frames. The communication frames is fixed to 16bits. Except command and valid data, the rest of parts can be filled with any other data.

The sender will be required to encrypt the communication frames before sending. The receiver has to decode the received data to communication frames. The encryption method is defined as AES-128, which is the common encryption method for Bluetooth communication.

AES-128 Encryption Demo code in JAVA:

```
public static byte[] Encrypt(byte[] sSrc, byte[] sKey){
    try{
        SecretKeySpec skeySpec = new SecretKeySpec(sKey, "AES");
        Cipher cipher = Cipher.getInstance("AES/ECB/NoPadding");
        cipher.init(Cipher.ENCRYPT_MODE, skeySpec);
        byte[] encrypted = cipher.doFinal(sSrc);
        return encrypted;
    }catch(Exception ex){
        return null;
    }
}
```

AES-128 Decoding Demo code in JAVA:

```
public static byte[] Decrypt(byte[] sSrc, byte[] sKey){
    try{
        SecretKeySpec skeySpec = new SecretKeySpec(sKey, "AES");
        Cipher cipher = Cipher.getInstance("AES/ECB/NoPadding");
        cipher.init(Cipher.DECRYPT_MODE, skeySpec);
        byte[] dncrypted = cipher.doFinal(sSrc);
        return dncrypted;
    }catch(Exception ex){
        return null;
    }
}
```

The key of AES-128, usually, is allocated and managed by JimiLab. Users can also change the key. If the key isn't changed successfully, the communication will fail. So double check before to change the key.

3.1 Get Access Token

When the Host and lock connected, the lock will generate an Access Token. The Token is with 4bits, dynamic, and only valid for the present connection. There will be a new token if the lock disconnect and then re-connect next time. Every connection will have the different token. The dynamic token can increase the security level of communication frames.

The procedures to get access token:

1. The AES-128 is fixed as:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
3A	60	43	2A	5C	01	21	1F	29	1E	0F	4E	0C	13	28	25

2. After the Host and lock connected, please prepare the communication frames as below:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
06	01	01	01	2D	1A	68	3D	48	27	1A	18	31	6E	47	1A

06 01 01 01 are fixed to get token. The bits highlighted in light blue color are the parts to wait for fill. They can be filled with any data. With AES_128 encryption method, we can get:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

28	BF	68	EB	5D	33	34	C8	61	37	62	E8	77	06	F9	FE

After the communication frames are encrypted, send them out. Then the lock will receive the command of getting Access Token. Then the lock will return the below info:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A5	6C	7D	75	48	DE	FF	EF	E7	AC	1E	A9	BC	CE	66	E6

With AES-128 decoding method, the data will be restored as below:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
06	02	07	5B	BA	71	E7	01	01	00	05	0B	86	20	18	0A

06 02 07 are fixed as Token Return identifier;

5B BA 71 E7 are the current 4-bit-token;

01 is for the chipset type, allocated and managed by JimiLab;

01 00 is for version number. Here, we can call it V1.0; Version number is defined by firmware engineers.

The rest of parts are invalid filled data.

3.2 Unlock

The Host send the below communication frames:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
05	01	06	PWD[6]						TOKEN[4]				FILL[3]		

PWD is the unlock passwords. The default passwords: 30
30 30 30 30 30

The lock return the communication frames:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
05	02	01	RET	FILL[12]											

RET is for returning status: 00 unlocked successfully; 01 failed unlock

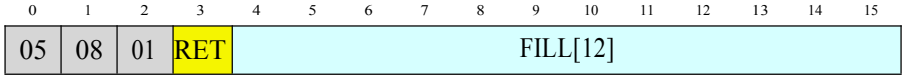
3.3 Lock

Lock manually. The lock will upload lock status to the host. Generally speaking, Lock command issued by host has no practical function. But in some cases, the lock command can be used to reset and repair.

The Host send the below communication frames:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
05	0C	01	0 1	TO KE N[4]				FILL[12]							

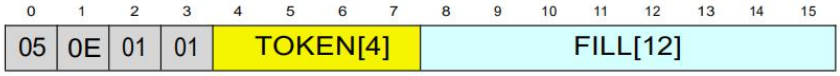
The lock return the communication frames:



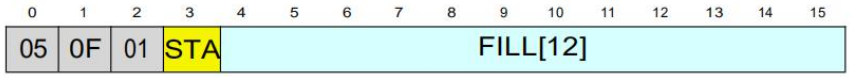
RET is for returning status: 00 unlocked successfully; 01 failed unlock

3.4 Lock status query

The Host send the below communication frames:

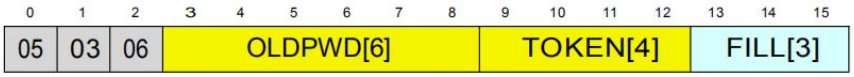


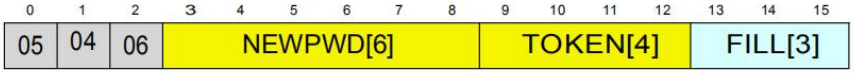
The lock return the communication frames:



3.5 Change password

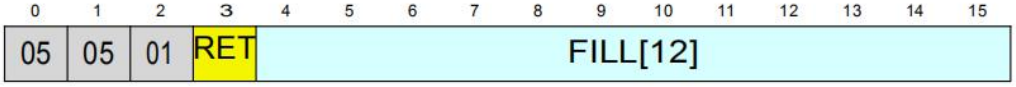
The Host send the below 2 communication frames successively:





OLDPWD means old password while NEWPWD means new password.

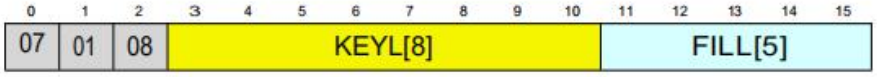
The lock return the communication frames:

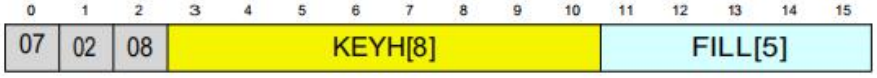


RET is for returning status: 00 unlocked successfully; 01 failed unlock

3.6 Modify password

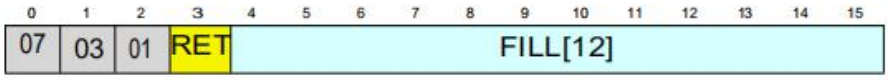
Think twice before modifying the key as it is risky. The Host send the below 2 communication frames successively:





KEYL is the first 8 bytes of the new key while the KEYH is the last 8 bytes of it.Bytes of key are sorted by little endian.

The lock return the communication frames:



RET is for returning status: 00 unlocked successfully; 01 failed unlock

Useful command:

BCM,READPASSWORD	Read password
BCM,PASSWORD, NEW PASSWORD	Change password
BCM,READAESKEY	Check key
BCM,AES_KEY,NEW KEY	Change key