



Cassia MQTT 旁路模式使用说明

V1.0

发布日期：2020 年 2 月 19 日

目录

| | |
|---------------------------------|----|
| 一、 MQTT 介绍 | 2 |
| 1、 名词解释 | 2 |
| 2、 MQTT 连接特性 | 3 |
| 3、 Cassia MQTT 旁路模式 | 3 |
| 二、 MQTT Server/Broker 的搭建 | 4 |
| 1、 基础准备 | 4 |
| 2、 编译安装 | 4 |
| 3、 启动服务器 | 5 |
| 三、 Cassia 路由器上的客户端配置 | 6 |
| 1、 在 AC 上配置 | 7 |
| 2、 在 AP 上配置 | 8 |
| 3、 数据测试 | 10 |
| 四、 使用场景 | 10 |
| 1、 无加密，无认证 | 10 |
| 2、 无加密，使用身份认证 | 11 |
| 3、 PSK 加密，用户验证 | 11 |
| 4、 证书加密，用户验证 | 13 |
| 五、 其他说明 | 14 |

一、MQTT 介绍

MQTT 协议 (Message Queuing Telemetry Transport)，翻译过来就是消息队列遥测传输协议，是 IBM 公司于 1999 年提出的，现在最新版本是 3.1.1。MQTT 是一个基于 TCP 的发布订阅协议，是为了解决内存极其有限的设备在带宽很低的网络情况下的通信问题，非常适合物联网通信。

1、名词解释

MQTT 协议中，分为三个角色：服务器\消息代理 (Server\Broker)、发布客户端 (Client)、订阅客户端 (Client)。

MQTT Client: 本文中简称客户端或 MQTT 客户端，根据分工不同，具有两个角色：

- 作为信息发布者的客户端用于信息的发布；
- 作为信息订阅者的客户端用于向 MQTT 服务器\消息代理订阅相关主题，获取发布者发布的信息。

MQTT Server/Broker: 本文中简称服务器\消息代理或 MQTT 服务器\消息代理，用于接收 MQTT 客户端发布的信息，并将相关信息推送至订阅信息的 MQTT 客户端。

Topic: 发布客户端向服务器\消息代理发布信息时的主题，用于订阅客户端订阅。

Cassia 路由器中安装了一个 MQTT Client，它可以把蓝牙路由器接收的蓝牙终端设备的广播数据，以特定的主题 (topic，例如温度) 发布至一个 MQTT Server 或者 MQTT Broker 中，其他 MQTT Client (安装在用户服务器、电脑、手机等) 可以从 MQTT Server 或者 MQTT Broker 订阅相关的 topic，获取蓝牙路由器发布的蓝牙终端设备的广播数据。有关的数据流程图请参见图 1。

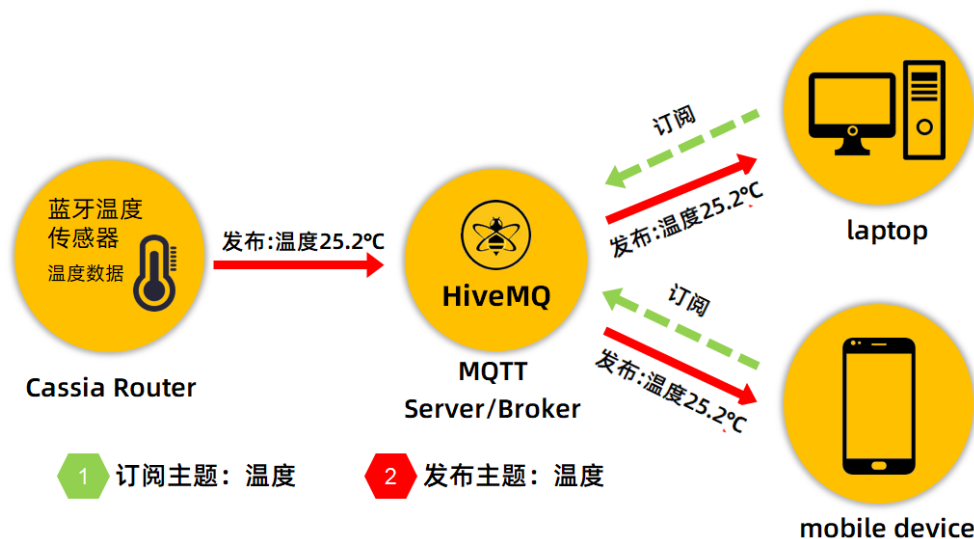


图 1: MQTT 数据流程图

注意: Cassia Router (蓝牙路由器) 中的 MQTT Client 只能作为发布者。上图中 HiveMQ 仅作为 MQTT Server/Broker 的举例，用户可选择其他 MQTT Server/Broker 软件。

在上图中，Cassia Router 采集到蓝牙温度传感器的温度数据后，其中的 MQTT 客户端以“温度”为主题(topic)发布至 MQTT 服务器\消息代理，用户的订阅客户端需要向 MQTT 服务器\消息代理订阅“温度”主题，服务器\消息代理会发送温度数据至订阅客户端。用户基于收到的数据进行展示平台的开发。

2、MQTT 连接特性

MQTT 协议基于 TCP / IP, 并且客户端和服务器\消息代理都需要具有 TCP / IP 协议栈。MQTT 连接本身始终在一个客户端和服务器\消息代理之间，一个客户端不能直接连接到另一个客户端。通过客户端向服务器\消息代理发送 CONNECT 消息来启动连接，服务器\消息代理使用 CONNACK 和状态码进行响应。MQTT 连接图请参见图 2。建立连接后，只要客户端不发送断开连接命令或连接意外终止，连接状态将会保持。

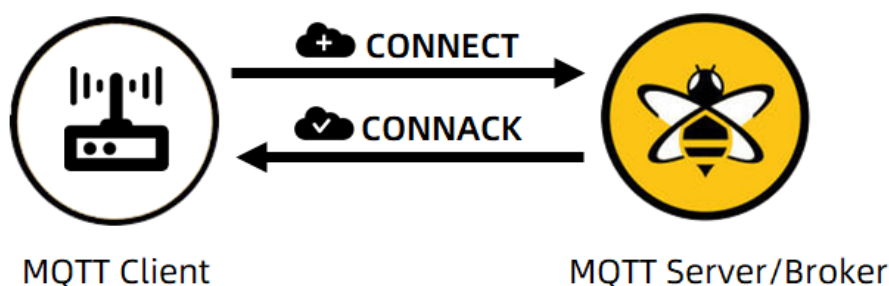


图 2: MQTT 连接图解

由于 MQTT 客户端始终启动连接，因此如果客户端和服务器\消息代理位于两个不同的网络中并且客户端位于 NAT 之后，则没有问题。典型的部署是将 Cassia 路由器放在防火墙后面的专用网络内，并将 MQTT 服务器\消息代理放在云中。

注意：请确保打开您的 MQTT 服务正在使用的 TCP 端口。默认情况下，MQTT 端口是 TCP 1883，而通过 SSL 的 MQTT 端口是 8883。您可以根据实际情况修改端口。

3、Cassia MQTT 旁路模式

Cassia 蓝牙路由器支持通过 MQTT 协议以旁路模式发送数据。即蓝牙路由器将采集到的数据通过 MQTT 协议发布到 MQTT 服务器/代理；控制消息（例如通过 AC 对蓝牙路由器进行配置更改等）仍通过 CAPWAP 发送到 AC。Cassia MQTT 旁路架构，请参见图 3。

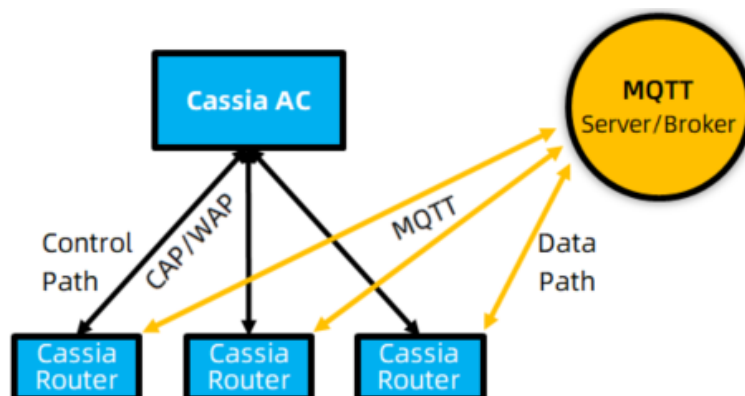


图 3: Cassia MQTT 旁路架构图

二、MQTT Server/Broker 的搭建

MQTT Server/Broker 为第三方软件，其安装说明不在本文档的说明范围内，本文档只对安装过程做简单描述，因软件更新或服务器配置（例如系统、端口等）等原因造成安装或测试不成功，请自行在网络上查询安装和配置说明。

1、基础准备

本文档使用 `mosquitto` 作为搭建 MQTT Server/Broker 的示例。请准备一台安装了 `linux` 系统的服务器（本地或云端），并确认 `1883` 端口（MQTT 协议的默认端口）为开启状态且未被占用，否则请选择其他端口。本文档使用的 `linux` 内核版本为 `Centos6.5`。

本文档以下指令均在 `linux` 系统中执行，请确保服务器可以正常访问互联网。

1.1 安装基础软件

分别安装以下三款基础软件

```
//安装软件
yum install gcc-c++
yum install cmake
yum install openssl-devel
```

1.2 下载 mosquitto 程序

```
//下载
wget http://mosquitto.org/files/source/mosquitto-1.4.4.tar.gz

//解压
tar -xzf mosquitto-1.4.4.tar.gz

//打开文件目录
cd mosquitto-1.4.4
```

注：下载地址为网络地址，如失效请自行寻找可用的官方版本地址下载。

2、编译安装

2.1 安装 c-ares

```
//下载
wget http://c-ares.haxx.se/download/c-ares-1.10.0.tar.gz

//解压
tar xvf c-ares-1.10.0.tar.gz

//打开文件目录
cd c-ares-1.10.0

//安装软件
./configure
make
make install
```

2.2 安装 lib-uuid

```
yum install libuuid-devel
```

2.3 安装 libwebsockets

根目录下执行以下指令：

```
wget
```

```
https://github.com/warmcat/libwebsockets/archive/v1.3-chrome37-firefox30.tar.gz
```

```
tar zxvf v1.3-chrome37-firefox30.tar.gz
```

```
cd libwebsockets-1.3-chrome37-firefox30
```

```
mkdir build; cd build;
```

```
cmake .. -DLIB_SUFFIX=64
```

```
make install
```

2.4 开始安装 mosquito

返回到 mosquito 目录，执行以下指令：

```
make
```

```
make install
```

2.5 修正链接库路径

首先添加路径，返回根目录后，执行以下指令：

```
vi /etc/ld.so.conf.d/liblocal.conf
```

然后在打开的文件中，添加以下两行指令：

```
/usr/local/lib64
```

```
/usr/local/lib
```

添加完成后，在英文模式下，点击 ESC，然后输入 :wq 保存并退出。

然后输入一下命令刷新配置文件。

```
ldconfig
```

3、启动服务器

3.1 配置文件创建

返回根目录，执行以下三个指令，创建相关的配置文件。

```
mv /etc/mosquitto/mosquitto.conf.example /etc/mosquitto/mosquitto.conf
```

```
mv /etc/mosquitto/aclfile.example /etc/mosquitto/acl
```

```
mv /etc/mosquitto/pwfile.example /etc/mosquitto/pwfile
```

3.2 程序配置

根目录下，执行以下指令，修改配置文件：

```
vi /etc/mosquitto/mosquitto.conf
```

在打开的文件中，添加以下指令：

```
#服务进程的系统用户
```

```
user root
```

```
#服务绑定的端口号，可修改
port 1883
#允许的最大连接数，-1 表示没有限制
max_connections -1
#不允许匿名用户
allow_anonymous false
#配置用户密码文件
password_file /etc/mosquitto/pwfile
#配置 topic 和用户文件
acl_file /etc/mosquitto/acl
```

保存并退出。

3.3 添加用户

根目录下执行以下指令，用于添加用户 `cassiamqtt`:

```
mosquitto_passwd -c /etc/mosquitto/pwfile cassiamqtt
```

并根据提示设置密码。本文档中的用户仅为举例，用户可设置其他用户名。

3.4 设定权限

根目录下执行以下指令，用于配置用户权限:

```
vi /etc/mosquitto/acl
```

在打开的文件中的指定位置，填写以下内容:

在 `This only affects clients with username "roger"`. 下方添加

```
user cassiamqtt
topic write cassia/#
```

在 `This affects all clients` 下方添加

```
user cassiamqtt
topic read cassia/#
```

保存并退出。

为了方便描述，本文档中发布和订阅使用了同一个用户，实际使用中请分别添加用户，并在 `acl` 文件中分别设定权限。

3.5 启动服务器

为了让以上的配置生效，请重启服务器后进行后续的操作。

根目录下执行以下指令，开启 `mqtt` 服务器，该指令会按照 `mosquitto.conf` 的配置启动服务器。

```
cd /etc/mosquitto
mosquitto -c /etc/mosquitto/mosquitto.conf -d
```

三、Cassia 路由器上的客户端配置

Cassia 蓝牙路由器上的 MQTT 客户端，仅用作发布，即蓝牙路由器会将采集到的数据发布至指定的 MQTT 服务器消息代理中，订阅客户端通过订阅相关主题获取数据。

1、在 AC 上配置

AC 即桂花网的物联网控制器，如果您使用 AC 对蓝牙路由器进行管理，请将蓝牙路由器设置为 AC 模式，并在 AC 上线。AC 和蓝牙路由器相关的上线操作请参照其他相关文档。

进入 AC 的 Routers 选项卡，选中蓝牙路由器，点击右上角编辑按钮进入配置选项卡，然后点击旁路模块。参见图 4。

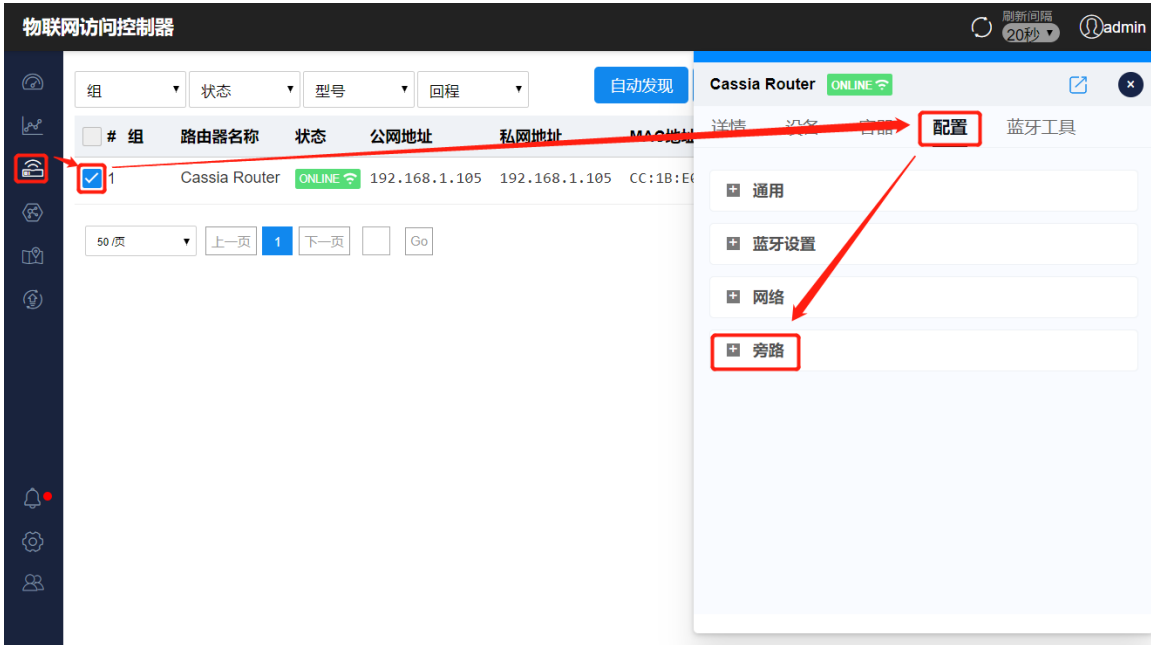


图 4：蓝牙路由器旁路配置

详细配置参照图 5：



图 5：蓝牙路由器 MQTT 配置

蓝牙路由器扫描参数设置：

- 选择扫描模式：**active** 主动扫描，可扫描到 **adData** 和 **scanData** 数据；**Passive** 被动扫描，只能扫描到 **adData** 数据；

- 扫描筛选条件：过滤蓝牙终端的广播包，可选择使用 **name**、**mac** 和 **uuid** 筛选；
- 选择 **MQTT** 作为旁路的数据传输协议；

MQTT 客户端的参数设置：

- **Host/Port**：配置 MQTT 服务器地址和端口（默认 1883）；
- 选择连接类型：长或短，默认为长；

短：消息发布后，客户端会立即断开与服务器或者代理的连接；

长：MQTT 客户端与服务器或代理保持连接状态。

注意：如果您经常发送消息，或使用证书对消息进行加密，建议您使用长连接以减少服务器资源消耗，否则，首选短连接。

- **Username/Password**：配置 MQTT 服务器设置的有发布(**write**)权限的账号密码；
- **Topic**：消息发布时的主题，用于用户的客户端订阅；
- **Qos**：服务质量（QoS）级别确定如何传递每个 MQTT 消息，必须指定一个。可用选项最多一次（0），至少一次（1）或恰好一次（2）；
- 加密模式：不加密，密钥加密或者证书加密。

2、在 AP 上配置

AP 即蓝牙路由器。如果您不使用 AC 对蓝牙路由器进行管理，可以在蓝牙路由器上开启 MQTT 客户端，请将蓝牙路由器调整为 **Standalone** 模式，然后在 **Services** 选项卡中进行 MQTT 客户端的配置。参见图 6。

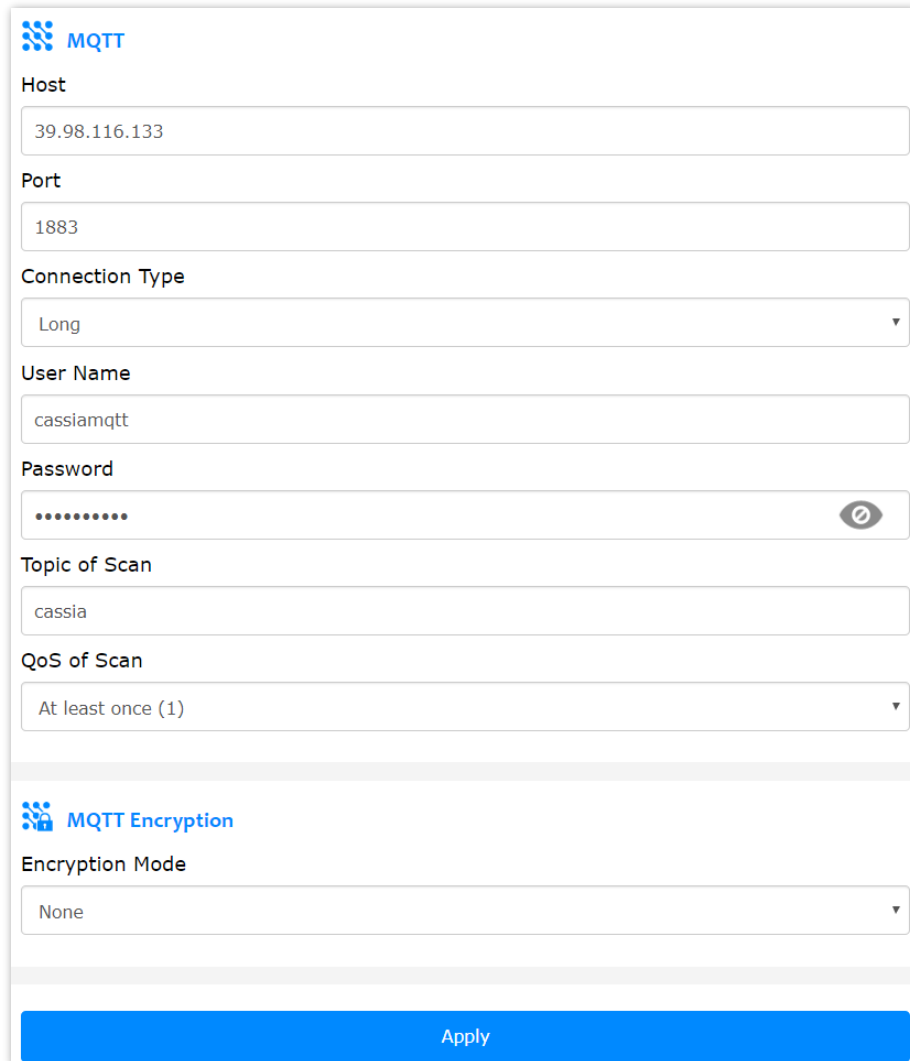
The image shows a configuration interface for MQTT service access. It includes a dropdown menu for 'Service Access' with 'MQTT' selected, a 'Scan mode' dropdown set to 'Active', and three empty text input fields for 'Name Filter', 'MAC Filter', and 'UUID Filter'.

图 6：蓝牙路由器扫描配置

蓝牙路由器扫描参数设置：

- 选择 **MQTT** 作为旁路的数据传输协议；
- 选择扫描模式：**active** 主动扫描，可扫描到 **adData** 和 **scanData** 数据；**Passive** 被动扫描，只能扫描到 **adData** 数据；

- 扫描筛选条件：过滤蓝牙终端的广播包，可选择使用 name、mac 和 uuid 筛选；MQTT 客户端的参数配置，参见图 7。



The image shows a web-based configuration form for MQTT. It is divided into two main sections: 'MQTT' and 'MQTT Encryption'. The 'MQTT' section includes fields for Host (39.98.116.133), Port (1883), Connection Type (Long), User Name (cassiamqtt), Password (masked with dots), Topic of Scan (cassia), and QoS of Scan (At least once (1)). The 'MQTT Encryption' section has an Encryption Mode dropdown set to None. A blue 'Apply' button is at the bottom.

图 7：蓝牙路由器 MQTT 客户端配置

MQTT 客户端的设置：

- **Host/Port:** 配置 MQTT 服务器地址和端口（默认 1883）；
- 选择连接类型：长或短，默认为长；
短：消息发布后，客户端会立即断开与服务器或者代理的连接；
长：MQTT 客户端与服务器或代理保持连接状态。
注意：如果您经常发送消息，或使用证书对消息进行加密，建议您使用长连接以减少开销。否则，首选短连接。
- **Username/Password:** 配置 MQTT 服务器设置的有发布(write)权限的账号密码；
- **Topic:** 消息发布时的主题，用于用户的客户端订阅；

- **Qos:** 服务质量 (QoS) 级别确定如何传递每个 MQTT 消息, 必须指定一个。可用选项最多一次 (0), 至少一次 (1) 或恰好一次 (2);
- **加密模式:** 不加密, 密钥加密或者证书加密。

3、数据测试

按照第二章节, 服务器完成搭建后, 启动 MQTT 服务器。

按照本章节的配置, 完成蓝牙路由器的 MQTT 客户端配置后, 蓝牙路由器即可向服务器发布数据。

用户启动一个 MQTT 客户端, 输入以下指令订阅 **cassia** 主题, 用户的客户端即可接收到蓝牙路由器发布的 **cassia** 主题的内容。

```
mosquitto_sub -h 39.98.116.133 -t cassia -u cassiamqtt -P cassiamqtt
```

-h 参数为 MQTT 服务器的地址;

-t 参数为用户客户端订阅的主题;

-u 参数为 MQTT 设置的具有订阅 (read) 权限的用户的用户名;

-P 参数为 MQTT 设置的具有订阅 (read) 权限的用户的密码, 请注意 P 为大写。

```
root@iZ8vbluijyv4z7u2n71a2Z ~]# cd /etc/mosquitto/
[root@iZ8vbluijyv4z7u2n71a2Z mosquitto]# mosquitto_sub -h 39.98.116.133 -t cassia -u cassiamqtt -P cassiamqtt
[{"name": "(unknown)", "evtType": 0, "rssi": -67, "adData": "0201061AFF4C00021574278BDAB64445208F0C720EAF0599350100ED66C5", "bdaddrs": [{"bdaddr": "98:04:ED:D6:50:A3", "bdaddrType": "public"}]}]
[{"name": "(unknown)", "evtType": 0, "rssi": -79, "adData": "02011A0AFF4C0010050318BF4AB1", "bdaddrs": [{"bdaddr": "5C:32:2A:B4:0C:EC", "bdaddrType": "random"}]}, {"name": "(unknown)", "evtType": 0, "rssi": -58, "adData": "0201061AFF4C00021574278BDAB64445208F0C720EAF0599350100ED66C5", "bdaddrs": [{"bdaddr": "98:04:ED:D6:50:A3", "bdaddrType": "public"}]}]
[{"name": "(unknown)", "evtType": 0, "rssi": -58, "adData": "0201061AFF4C00021574278BDAB64445208F0C720EAF0599350100ED66C5", "bdaddrs": [{"bdaddr": "98:04:ED:D6:50:A3", "bdaddrType": "public"}]}]
[{"name": "HSE96722001CEC", "evtType": 0, "rssi": -72, "adData": "0201060F094853453936373232303031434543020AFE", "bdaddrs": [{"bdaddr": "E9:67:22:00:1C:EC", "bdaddrType": "public"}]}]
[{"name": "HSE967220019BC", "evtType": 0, "rssi": -74, "adData": "0201060F094853453936373232303031394243020AFE", "bdaddrs": [{"bdaddr": "E9:67:22:00:19:BC", "bdaddrType": "public"}]}]
[{"name": "(unknown)", "evtType": 0, "rssi": -80, "adData": "02011A0AFF4C0010050318BF4AB1", "bdaddrs": [{"bdaddr": "5C:32:2A:B4:0C:EC", "bdaddrType": "random"}]}, {"name": "(unknown)", "scanData": "", "evtType": 4, "rssi": -80, "bdaddrs": [{"bdaddr": "5C:32:2A:B4:0C:EC", "bdaddrType": "random"}]}]
```

四、使用场景

1、无加密, 无认证

此种方式非常不安全, 基于安全考虑, 不推荐使用此种方式。若需要使用无加密无认证的方式, 请在第二章节的服务器搭建过程中注意以下问题:

- **程序配置:**

```
#不允许匿名用户
allow_anonymous false
```

修改为

```
#不允许匿名用户
allow_anonymous true
```

删除以下两行:

```
#配置用户密码文件
password_file /etc/mosquitto/pwfile
```

```
#配置topic和用户
acl_file /etc/mosquitto/acl
```

- 添加用户和设定权限请略过，不进行任何操作。
- 在第三章节的蓝牙路由器客户端配置中，不填写用户名和密码。
- 用户客户端订阅时，省略掉 `-u` 和 `-P` 参数。即通过以下指令订阅。
`mosquitto_sub -h 39.98.116.133 -t cassia`

2、无加密，使用身份认证

本文中描述的服务器搭建过程和客户端配置过程，即为“无加密，使用身份认证”的方式。网络上可以找到的大多数文档，一般也采用这种方式。请参照上述文档或自行寻找文档进行服务器的搭建。

MQTT 服务器仅接收经过身份验证的用户发送的数据。并且只有经过身份认证的用户客户端可以订阅主题，获取数据。这可以在一定程度上保护我们的 MQTT 服务器。

关键点：

- 程序配置中，不允许匿名用户登录。

```
#不允许匿名用户
allow_anonymous false
```

- 添加用户，并且为用户设定读或者写的权限。在蓝牙路由器发布消息和用户客户端订阅时，必须填写服务器配置的对应权限的用户名和密码。
- 为了方便描述，本文中发布信息 and 订阅使用了同一个用户名密码，实际使用中请分别添加用户，并在 `acl` 文件中分别设定权限。

```
# This only affects clients with username "roger".
user roger
topic foo/bar
user cassiamqtt
topic write cassia/# 写权限用户，并对应发布主题

# This affects all clients.
pattern write $SYS/broker/connection/%c/state
user cassiamqtt
topic read cassia/# 读权限用户，并对应订阅主题
```

3、PSK 加密，用户验证

我们知道了如何使用用户验证，但是传输的信息为明文状态，仍然不够安全，本小节我们将介绍对文件进行加密的方式，进一步提升 MQTT 服务器的安全性。消息加密可以和用户验证同时启用，我们建议同时开启。

启用 `psk` 加密方式，需要对服务器和客户端做出相应的修改，如果您已经按照本文档完成了服务器的搭建，请在搭建过程中的操作的基础上，增加如下操作：

服务器配置：

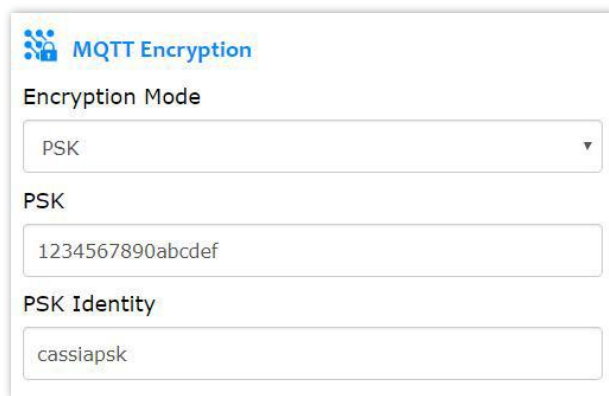
- 根目录下执行以下操作，生成 `pskfile` 文件：

```
mv /etc/mosquitto/pskfile.example /etc/mosquitto/pskfile
```

- 根目录下执行以下指令,编辑 pskfile 内容,按照 identity:key 的格式设置 psk 密钥:
cd /etc/mosquitto
vi pskfile
- 在打开的文件中,输入 identity:key (自定义,其中 kdy 必须为 16 进制),然后保存退出:
cassiapsk:1234567890abcdef
- 根目录下,执行以下指令,修改配置文件,用于设置服务器为 psk 验证:
vi /etc/mosquitto/mosquitto.conf
- 在打开的文件中,添加如下指令,然后保存退出:
#配置 psk 验证
psk_hint true
psk_file /etc/mosquitto/pskfile
- 根目录下,执行如下指令,开启 MQTT 服务器,我们将会看到如下错误信息:

```
1581665763: New connection from 115.34.104.133 on port 1883.  
1581665763: OpenSSL Error: error:140760FC:SSL routines:SSL23_GET_CLIENT_HELLO:unknown protocol  
1581665763: Socket error on client <unknown>, disconnecting.  
1581665763: OpenSSL Error: error:140760FC:SSL routines:SSL23_GET_CLIENT_HELLO:unknown protocol  
1581665763: Socket error on client <unknown>, disconnecting.  
1581665763: OpenSSL Error: error:140760FC:SSL routines:SSL23_GET_CLIENT_HELLO:unknown protocol  
1581665763: Socket error on client <unknown>, disconnecting.
```

这是因为我们的客户端(发布)为配置 psk 验证导致,按照下图进行蓝牙路由器 MQTT 客户端的配置。



The image shows a configuration window titled "MQTT Encryption". It contains three input fields: "Encryption Mode" with a dropdown menu set to "PSK", "PSK" with the text "1234567890abcdef", and "PSK Identity" with the text "cassiapsk".

蓝牙路由器配置完成后(请注意用户名密码也需要配置),我们将会在服务器上看到如下信息,至此 psk 验证配置成功。

```
1581665866: New connection from 115.34.104.133 on port 1883.  
1581665866: New connection from 115.34.104.133 on port 1883.  
1581665866: New client connected from 115.34.104.133 as cc:lb:e0:e1:00:c4-3 (cl, k60, u'cassiamqtt').  
1581665866: New client connected from 115.34.104.133 as cc:lb:e0:e1:00:c4-4 (cl, k60, u'cassiamqtt').  
1581665866: New client connected from 115.34.104.133 as cc:lb:e0:e1:00:c4-0 (cl, k60, u'cassiamqtt').  
1581665866: New client connected from 115.34.104.133 as cc:lb:e0:e1:00:c4-2 (cl, k60, u'cassiamqtt').  
1581665866: New client connected from 115.34.104.133 as cc:lb:e0:e1:00:c4-1 (cl, k60, u'cassiamqtt').
```

另外,如果我们看到如下的报错信息,说明蓝牙路由器上配置的 psk 信息错误,请检查后重新配置。

```
1581665828: New connection from 115.34.104.133 on port 1883.
1581665828: OpenSSL Error: error:1408F119:SSL routines:SSL3_GET_RECORD:decryption failed or bad record mac
1581665828: Socket error on client <unknown>, disconnecting.
1581665828: OpenSSL Error: error:1408F119:SSL routines:SSL3_GET_RECORD:decryption failed or bad record mac
1581665828: Socket error on client <unknown>, disconnecting.
```

4、证书加密，用户验证

除了 **psk** 加密以外，您还可以选择使用证书加密。证书加密比 **psk** 加密更加安全，但是配置也更加复杂。**psk** 加密和证书加密不能同时使用。

如果要配置证书加密，您首先需要向 **CA** 申请证书或通过自签名生成证书。这是您启用证书加密的前提条件！本文档不对如何申请或生成证书作出描述。

您需要对 **MQTT** 服务器作出以下修改：

- 将您的证书文件放置在服务器的 `/etc/mosquito/certs/`目录下：
 - `ca.crt`: 授权证书是已签署的服务器证书
 - `server.crt` : **CA** 为 **server** 端签发的证书文件
 - `server.key` : **server** 端证书使用的 `key` 文件
- 修改 `mosquitto.conf` 文件，增加以下内容，因 **psk** 验证和证书验证只能同时启用一个，请将 **psk** 验证的相关内容注释掉：
 - `cafile /etc/mosquito/certs/ca.crt`
 - `certfile /etc/mosquito/certs/server.crt`
 - `keyfile /etc/mosquito/certs/server.key`

参照下图：

```
#配置psk验证
#psk_hint true
#psk_file /etc/mosquitto/pskfile
#配置证书验证
cafile /etc/mosquito/certs/ca.crt
certfile /etc/mosquito/certs/server.crt
keyfile /etc/mosquito/certs/server.key
```

- 重启相关服务，或直接重启服务器，重新运行 `mosquitto`。

蓝牙路由器的客户端，需要作出如下更改：

- 修改加密模式（**Encryption Mode**）为证书（**Certificate**）
- 填写 **CA** 证书（**CA Certificate**，获取方法见下文）
- 选择要求客户端提供证书，如果选择是，用户订阅必须上传您的客户端证书。推荐选择否。
- 证书内容获取方法：

MQTT 服务器根目录下，执行以下指令，获取证书，并将证书复制到蓝牙路由器的 **MQTT** 客户端相关的配置中。

```
cat /etc/mosquitto/certs/ca.crt
```

您可以获取到证书的内容：

```
[root@i28vbb1iujyv4z7u2n71a2z ~]# cat /etc/mosquito/certs/ca.crt
-----BEGIN CERTIFICATE-----
dhAfB3vez1KaHHILO5roanSj0kOIHrWmtPuG/Kcu91eWfdJtEkhs10Og3qLJhp6rMn29Kuffefp8/5qsniwF90iJDRSP6FoawcWZOS44ZwZnUD9c62Vt9LO0jBjQ18tC
1TZjN09Yx4x6iCVR9kqQYmmMvTt1v61o4t3Sueg2PQ9pNBp0ieOXMM80KduY7uCK0Pp5KbCvY1xvFu9H6lcNrgAW5q/jXLatDK7M8oL6aT
wCFsrXBwIDAQABAoIBADUJ16D5H07Dp5TzU3St5yQP3P1Rk/k96E503x5F5srV7TzOM+duEGLSHgibucAvLd5/Z/DnHargPdIbOh3JsrCz9imMUENxvakA5OMH/Q4NL
b+1tlerYA4MDKlM4lycZRG3QNAYWds6KcNVOvdOU2GR3ANF+6Z8T5LXzktg7cP6nRd350GngUx6H10adXqcxSEY23dL2HZEHW4RlXremNOWEsRc2Q/vJwad2Akd
SulBvoF+rQF+kejXxL2qPTbvDO7uix3MpoPDbpBLzf+SF4Mr3oPr/3uLgKc8p/aKra1HCFYr3fpuv6GHHxU2Iha6GQ4CyGjZdmQjxgECgYEA+7rMo4nkI439mLf47D
z/DBxUCTkMZ38hfI6DwRkIoeNz/1ygpVj4eA/Di+PCEY/B6A83fdGg+ugyP2qkG5uzMT0828u6eQoBMGHCJm5AGTMr4gPTZnDdxSvYbGzIz9usJnIVKLA+mh+8tnhum
XYuWCs8ondxOLELNIUf1PBaOEcGyEA+GFT81IQvbgEmaPtPQUjm8GrEdAc4ocM0EknmTt6tm9XXT318tI1oItNsAUTMjIUN0NyeriY3GevVgETDVLtiKof2QqdV92j0C
RxsG143NDgFoBtM8XmRdg82RP0stVw9RyPZiAB/fi2IMzJenLiblIRDtEK9rDrI5Touk8CgYEA4XnqfrmuXaJ7emWfU4s3MFPXegNddv7KsdS9cyLHNqqtZjJDupcmwh
2ont5AMcD8gwomOu3dcGC4Pg/ozUH9gPEHb7UxILM4/TmT5Wnr6cxgAwIugA1gpEREAc34pur1rM2yQQ0fKaybuU8r/1fMjfvONdkyUsty1Gsl7DN9VEECgYEA4htq
G+ts6CAXFRMHZ1YKHgfgmYvKGU5JZBcu7oRzTxW93dZv+/HIQrZNL/Fi+u4PyeW5g7wrRotVulMwTp/jaUcURrLvi6U6nCWjyKfGtqBq1CTsp0NjA4/RfucPe4P
b3evhkyqgEIZ9IoiuYKlGj/JzbxAXNxMaam9MCgYAi79ozPjRopNks5wpRI73G+2DChTM84JndNeER4jVik0S94m+tQcfdJLk2LrvZpWeeMMrjGq3QjQjFDT2Hqv2gF+BipR9rHWP3pv+SjC5Bq7
7gNZ+jFTqE4KXjO79x9OngM0m/+TMqXCf8ePvTUPk0/H+erIKLC2PK8UPLasvRdQ==
```

蓝牙路由器 MQTT 客户端配置如下：

The screenshot shows the MQTT Encryption configuration page. The 'Encryption Mode' dropdown is set to 'Certificate'. The 'CA Certificate' field contains the following text: `dhAfB3vez1KaHHILO5roanSj0kOIHrWmtPuG/Kcu91eWfdJtEkhs10Og3qLJhp6rMn29Kuffefp8/5qsniwF90iJDRSP6FoawcWZOS44ZwZnUD9c62Vt9LO0jBjQ18tC1TZjN09Yx4x6iCVR9kqQYmmMvTt1v61o4t3Sueg2PQ9pNBp0ieOXMM80KduY7uCK0Pp5KbCvY1xvFu9H6lcNrgAW5q/jXLatDK7M8oL6aT...`. The 'MQTT Require Client Certificate' dropdown is set to 'No'. A blue 'Apply' button is located at the bottom of the form.

五、其他说明

桂花网的蓝牙路由器作为 MQTT 协议中的发布者，为用户提供了额外的数据接收方式，相关的 MQTT 服务器或者代理不在我们的提供范围和技术支持范围内，需要用户自行购买或搭建。

如果您在搭建服务器或者使用过程中遇到了问题，您也可以联系我们，我们会力所能及的为您提供相关的技术支持服务。