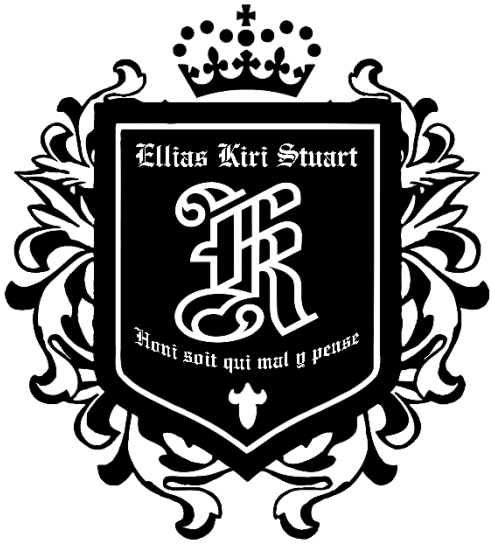


HongYanAsstSTM32

设备通信协议简明初步

Version: v1.0.0 alpha

Date: 2021/01/14

Platform: OneNET (China Mobile)

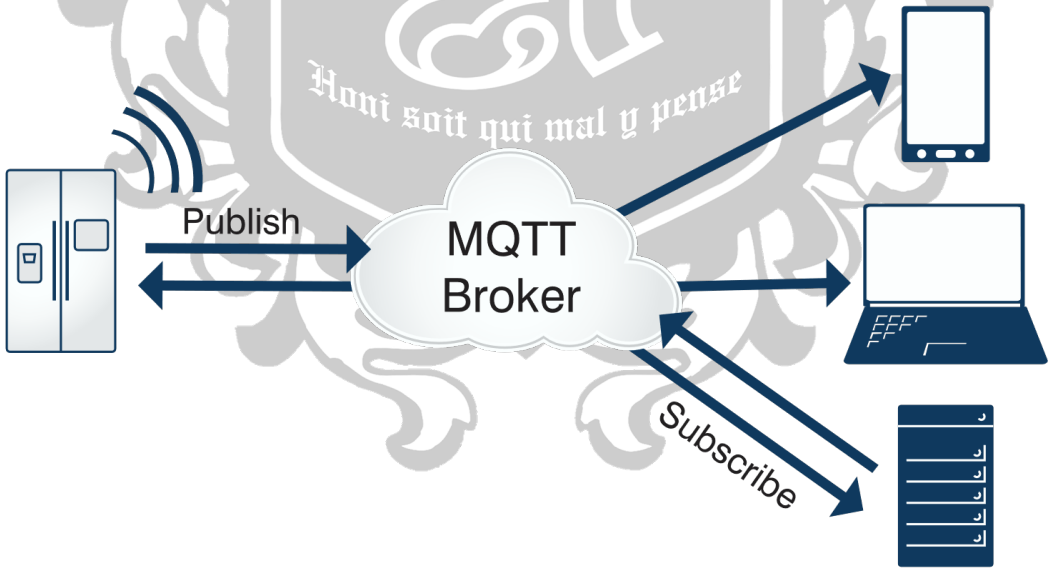
Scope: Developers

INDEX

1. MQTT 协议简介

1.1. 概述

MQTT 是机器对机器(M2M)/物联网(IoT)连接协议。它被设计为一个极其轻量级的发布/订阅消息传输协议。对于需要较小代码占用空间和/或网络带宽非常宝贵的远程连接非常有用，是专为受限设备和低带宽、高延迟或不可靠的网络而设计。这些原则也使该协议成为新兴的“机器到机器”(M2M)或物联网(IoT)世界的连接设备，以及带宽和电池功率非常高的移动应用的理想选择。例如，它已被用于通过卫星链路与代理通信的传感器、与医疗服务提供者的拨号连接，以及一系列家庭自动化和小型设备场景。它也是移动应用的理想选择，因为它体积小，功耗低，数据包最小，并且可以有效地将信息分配给一个或多个接收器。



1.2. 特点及应用

- 开放消息协议，简单易实现；

- 发布订阅模式，一对多消息发布；
- 基于 TCP/IP 网络连接，提供有序，无损，双向连接；
- 1 字节固定报头，2 字节心跳报文，最小化传输开销和协议交换，有效减少网络流量；
- 消息 QoS 支持，可靠传输保证。

MQTT 协议广泛应用于物联网、移动互联网、智能硬件、车联网、电力能源等领域。

- 物联网 M2M 通信，物联网大数据采集；
- Android 消息推送，WEB 消息推送；
- 移动即时消息，例如 Facebook Messenger；
- 智能硬件、智能家具、智能电器；
- 车联网通信，电动车站桩采集；
- 智慧城市、远程医疗、远程教育；
- 电力、石油与能源等行业市场。

1.3. MQTT 协议基本原理

1.3.1. MQTT 协议实现方式

实现 MQTT 协议需要客户端和服务器端通讯完成，在通讯过程中，MQTT 协议中有三种身份：发布者（Publish）、代理（Broker）（服务器）、订阅者（Subscribe）。其中，消息的发布者和订阅者都是客户端，消息代理是服务器，消息发布者可以同时是订阅者。

MQTT 传输的消息分为：主题（Topic）和负载（payload）两部分：

1. Topic，可以理解为消息的类型，订阅者订阅（Subscribe）后，就会收到该主题的消息内容（payload）；

2. payload，可以理解为消息的内容，是指订阅者具体要使用的内容。

1.3.2. 网络传输与应用消息

MQTT 会构建底层网络传输：它将建立客户端到服务器的连接，提供两者之间的一个有序的、无损的、基于字节流的双向传输。

当应用数据通过 MQTT 网络发送时，MQTT 会把与之相关的服务质量（QoS）和主题名（Topic）相关连。

1.3.3. MQTT 客户端

一个使用 MQTT 协议的应用程序或者设备，它总是建立到服务器的网络连接。客户端可以：

1. 发布其他客户端可能会订阅的信息；
2. 订阅其它客户端发布的消息；
3. 退订或删除应用程序的消息；
4. 断开与服务器连接。

1.3.4. MQTT 服务器

MQTT 服务器以称为"消息代理"（Broker），可以是一个应用程序或一台设备。它是位于消息发布者和订阅者之间，它可以：

1. 接受来自客户的网络连接；
2. 接受客户发布的应用信息；
3. 处理来自客户端的订阅和退订请求；
4. 向订阅的客户转发应用程序消息。

1.3.5. MQTT 协议中的订阅、主题、会话

1. 订阅（Subscription）

订阅包含主题筛选器（Topic Filter）和最大服务质量（QoS）。订阅会与一个会话（Session）关联。一个会话可以包含多个订阅。每一个会话中的每个订阅都有一个不同的主题筛选器。

2. 会话（Session）

每个客户端与服务器建立连接后就是一个会话，客户端和服务器之间有状态交互。会话存在于一个网络之间，也可能在客户端和服务器之间跨越多个连续的网络连接。

3. 主题名（Topic Name）

连接到一个应用程序消息的标签，该标签与服务器的订阅相匹配。服务器会将消息发送给订阅所匹配标签的每个客户端。

4. 主题筛选器（Topic Filter）

一个主题名通配符筛选器，在订阅表达式中使用，表示订阅所匹配到的多个主题。

5. 负载（Payload）

消息订阅者所具体接收的内容。

1.3.6. MQTT 协议中的方法

MQTT 协议中定义了一些方法（也被称为动作），用于表示对确定资源所进行操作。这个资源可以代表预先存在的数据或动态生成数据，这取决于服务器的实现。通常来说，资源指服务器上的文件或输出。主要方法有：

1. Connect。等待与服务器建立连接。

2. Disconnect。等待 MQTT 客户端完成所做的工作，并与服务器断开 TCP/IP 会话。

3. Subscribe。等待完成订阅。

4. UnSubscribe。等待服务器取消客户端的一个或多个 topics 订阅。

5. Publish。MQTT 客户端发送消息请求，发送完成后返回应用程序线程。

1.3.7. MQTT 官方文档(MQTT 3.1.1)

https://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.pdf

1.4. MQTT 协议在本系统中的应用

在本系统中通过订阅 OneNET 所指定的"$dp"主题并完成数据流的上传。

2. 数据流及应用声明

在线数据流总共包含五个字段，接下来将对其格式和意义进行声明。

2.1. 数据流基本格式

ID	名称	单位	数据格式	数据范围
1	La	°(度)	六位浮点数	-90~+90
2	Lo	°(度)	六位浮点数	-180~+180
3	S	kmph	两位浮点数	-inf~+inf
4	A	-	一位浮点数	0~50
5	F	-	布尔值	0/1

2.2. 数据流基本含义及约定

1. La

顾名思义，"La"为"Latitude"的缩写，表示设备当前在 WG-84 坐标系下的纬度。数据正值表示北纬，负值表示南纬。

2. Lo

顾名思义，"Lo"为"Longitude"的缩写，表示设备当前在 WG-84 坐标系下的经度。

数据正值表示东经，负值表示西经。

3. S

顾名思义，"S"为"Speed"的缩写，表示设备当前的 GPS 速度。

4. A

顾名思义，"A"为"Accuracy"的缩写，表示设备当前的位置精度因子 PDOP。注意：在某些情况下该值可能会越界，但越界通常不会超过 100 且这种情况通常被判定为"GPS NO SIGNAL"。

5. F

顾名思义，"F"为"Flag"的缩写，用来标注当前是否发生了事故。如果发生了事故，则该值为 1，否则为 0。

2.3. 触发器初步

2.3.1. 关联设备

添加触发器时可以选择关联设备。关联设备有两种类型：关联一个产品下的全部设备，或在创建触发器时关联一个设备，在创建触发器之后选择性地增添其他设备。

2.3.2. 触发数据流

每个触发器尽可以选择一个数据流，在本系统中选择 "F" 字段为触发数据流，以实现对事故状态的实时监控。

2.3.3. 触发条件

触发条件由一个逻辑运算符及常量组成，当值为真时触发告警。

项目	参数数量	含义
>,>=,<,<=,==	1	易得
INOUT	2	当数值进入或离开区间时触发
FROZEN	1	给定时间内未上报数据时触发
CHANGE	0	当数值变化时触发
LIVE	1	给定时间内上报数据时触发

2.3.4. 接收信息方式

OneNET 提供邮箱和 URL 两种收信方式，邮件每天限额 20 封。以下为邮箱告警样例：


```
1. title:
2. 【OneNET】您的 senpai 设备的 JK 触发器在 1919-08-10 11:45:14 被触发。详情：konnsui 数据流、==类型、触发值 1
3. content:
4. 触发器信息
5.
6. 触发器 id: 1919810
7. 触发器名: JK
8. 类型: ==
9. 阈值: 1
10.
11.触发数据
12.
13.设备 id: 192608171
14.设备名: senpai
15.数据流: konnsui
16.触发时间: 1919-08-10T11:45:14.708
17.触发值: 1
```

3. API 鉴权

3.1. 概述

为提高 API 访问安全性，OneNET API 的鉴权参数作为 header 参数存在。OneNET 支持普通以及安全两种鉴权认证方式，对比如下表：

项目	普通鉴权	安全鉴权
核心密钥	apiKey	accessKey
HEADER 参数名	"api-key"	"Authorization"
HEADER 参数值	apiKey（直接传输密钥）	由参数组构成的 token，不含密钥
访问时间控制	不支持	支持（由参数组中参数控制访问时效）
自定义权限	不支持	支持
核心密钥更新	不支持	支持
HTTPS 协议	支持	支持
安全性	较低	较高

普通鉴权方式通过 apiKey 作为鉴权密钥，apiKey 分为两个访问层级：产品级（Master）与设备级。

项目	产品级	设备级
数量	产品下唯一	产品下可以有多个
权限范围	产品下所有资源的操作，包括：设备、数据流、数据点、触发器、文件、命令以及设备 apiKey	部分设备的全量操作，包括 设备详情，设备数据流，设备数据点
自定义权限	不支持	支持设备级。需要用户进行 apiKey 与设备的关联操作，一旦关联则具备该设备的最大权限

3.2. 安全鉴权

安全鉴权方式以 accessKey 为核心密钥，用户需要使用核心密钥通过签名算法计算签名，与其他参数共同组成 token，然后将 token 作为请求 Header 参数进行鉴权。

安全鉴权方式通过避免核心密钥在网络上直接传输，增加认证参数时效控制，增加密钥权限粒度控制（即将到来）等方式提高鉴权安全性，最大限度保证访问安全。

3.2.1. 安全鉴权策略

安全鉴权策略主要通过如下方式保证访问安全：

- 避免核心密钥网络中直接传输，从而避免核心密钥在传输中泄露；
- 通过包含由**非可逆算法**生成的签名的 token 来进行身份认证，即使 token 被窃取，攻击者也无法通过 token 反向获得核心密钥；
- 鉴权参数 token 具有用户自定义的过期时间属性，可从时间维度降低被攻击/仿冒的风险。

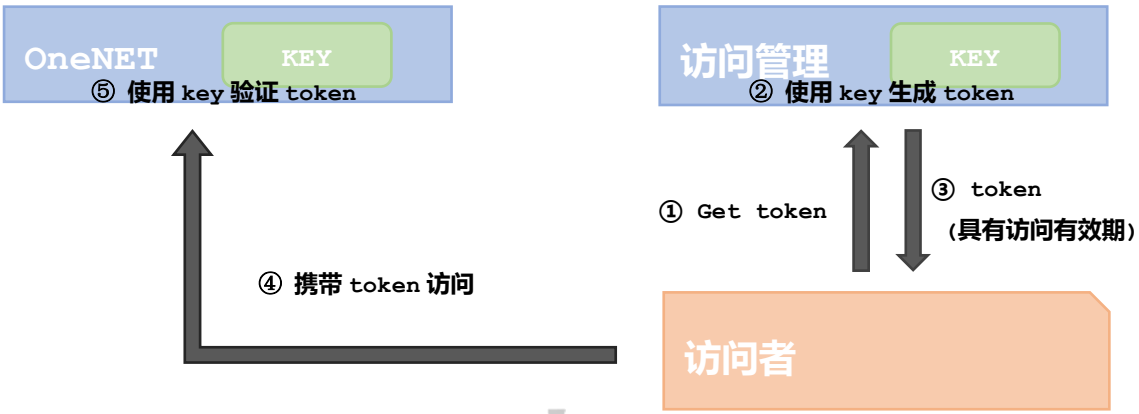
3.2.2. 常见安全方案

3.2.2.1. 方案 1

访问者（可以为**应用**或者**设备**）固化访问密钥于软件中，在需要进行服务访问时，通过密钥计算零时 token，通过临时 token 进行服务访问认证。

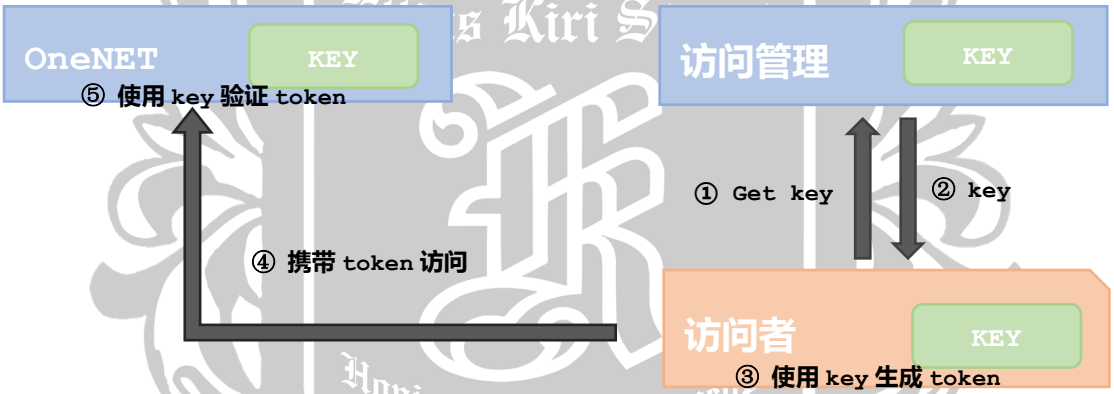
3.2.2.2. 方案 2

访问者首先通过访问管理者获取临时访问 token，访问管理者可根据需要自定义该 token 的访问有效期（即过期时间），访问者获取该 token 后方才能访问 OneNET。



3.2.2.3. 方案 3

访问管理者直接将密钥授权给访问者（例如，直接为设备烧写 key），访问者通过密钥生成 token 进行访问。



3.2.3. token 算法

token 由多个参数构成，如下表：

名称	类型	是否必须	参数说明
VERSION	string	Y	参数组版本号，日期格式，目前仅支持"2018-10-31"
RES	string	Y	访问资源 resource 格式为：父资源类/父资源 ID/子资源类/子资源 ID 见 res 使用场景说明
ET	int	Y	访问过期时间 expirationTime，unix 时间 当一次访问参数中的 et 时间小于当前时间时，平台会认为访问参数过期从而拒绝该访问
METHOD	string	Y	签名方法 signatureMethod 支持 md5、sha1、sha256
SIGN	string	Y	签名结果字符串 signature

3.2.3.1. res 使用场景说明

场景	RES 参数格式	说明
API 访问	products/{pid}	
设备连接	products/{pid}/devices/{device_name}	需使用设备级密钥

3.2.3.2. sign 签名算法

```
1.  sign = base64(hmac_<method>(base64decode(accessKey), utf-8(StringForSignature)))
```

其中：

- accessKey 为 OneNET 为独立资源（例如，产品）分配的唯一访问密钥，其作为签名算法参数之一参与签名计算，为保证访问安全，请妥善保管；
- accessKey 参与计算前应先进行 base64decode 操作；
- 用于计算签名的字符串 StringForSignature 的组成顺序**按照参数名称进行字符串排序**，以'/n'作为参数分隔，当前版本中按照如下顺序进行排序：et、method、res、version。

3.2.3.3. 参数编码

token 中 key=value 的形式的 value 部分需要经过 **URL 编码**，需要进行编码的特殊符号如下：

序号	符号	编码
1	+	%2B
2	space	%20
3	/	%2F
4	?	%3F
5	%	%25
6	#	%23
7	&	%26
8	=	%3D

4. API 使用

4.1. SDK

OneNET 为 JavaScript、PHP、Java、C#、C、Android、IOS 提供了 SDK，可以在 OneNET 的 GitHub 仓库 https://github.com/cm-heclouds?tab=repositories 中找到并下载。

4.2. 测试用权限发放

项目	数据
DEVICE NAME	ORIGIN
DEVICE ID	653000696
API ADDRESS	http://api.heclouds.com/devices/653000696
APIKEY NAME	test
APIKEY	A6Y9Go15M3q3grJ4Vj5ZKFyHIws=