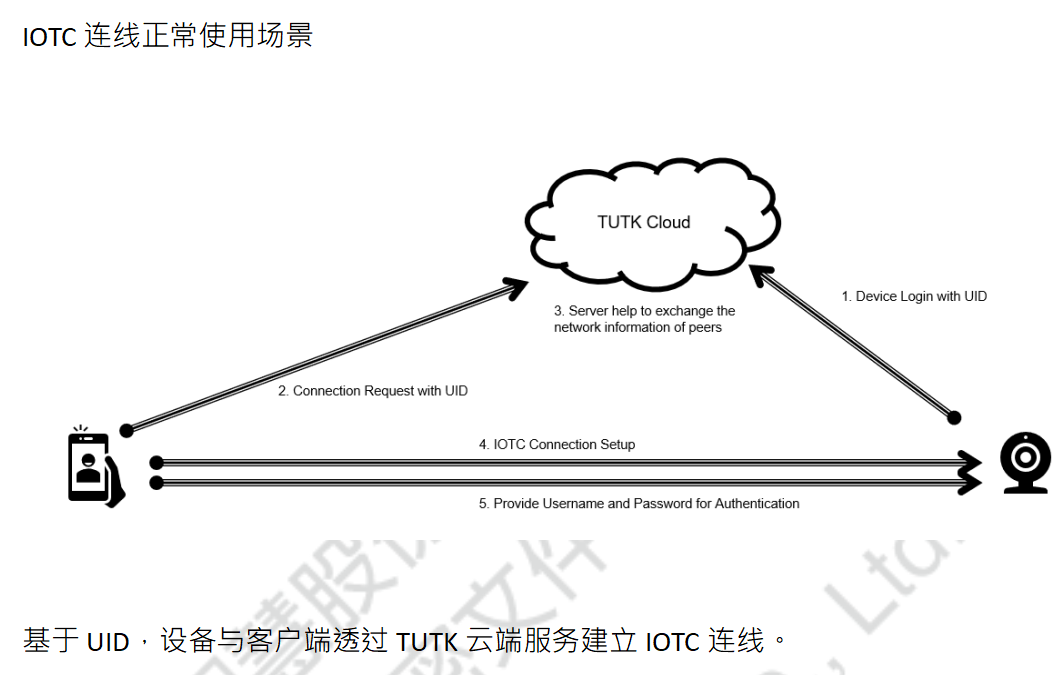
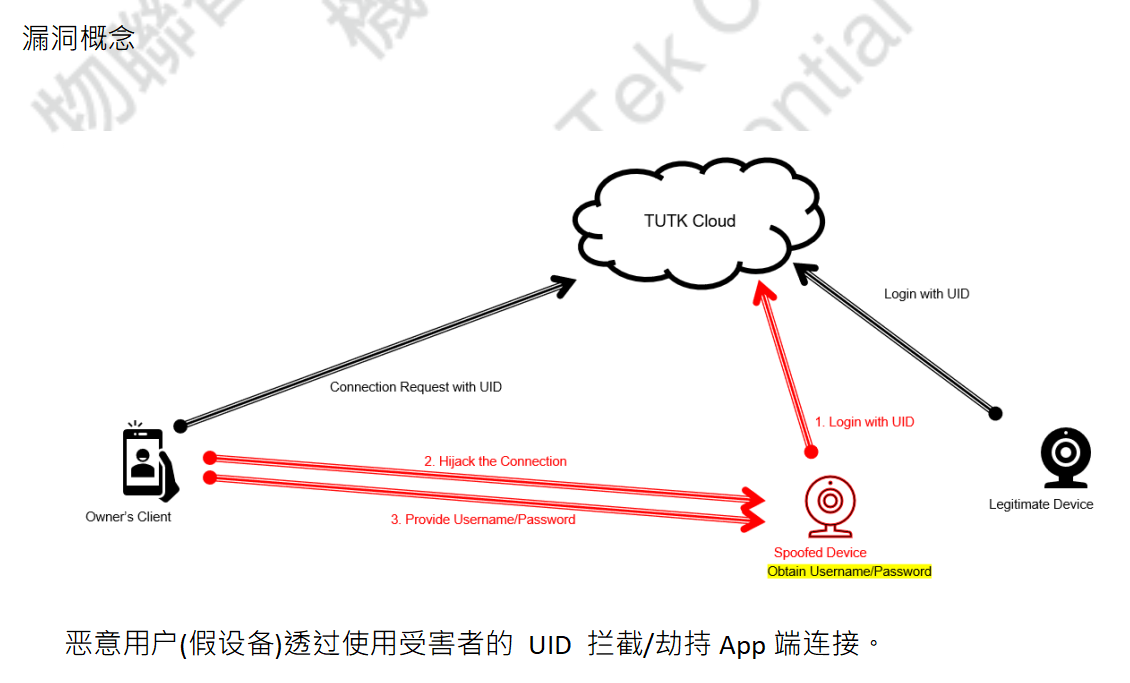
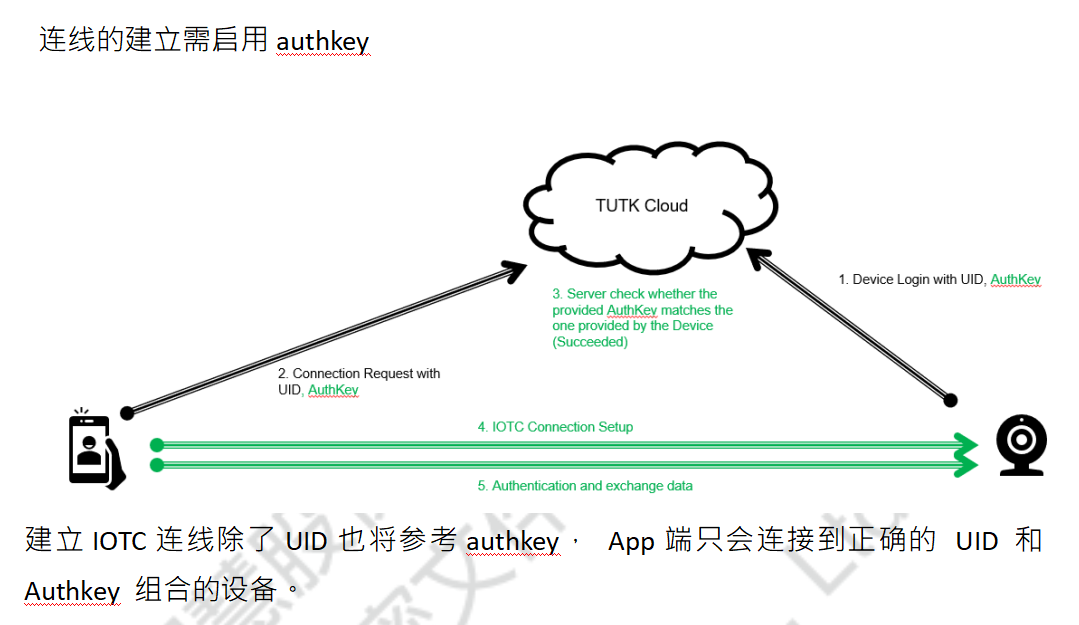
#### Authkey&DTLS升级说明文档V1.0

##### 为什么要升级Authkey和DTLS？

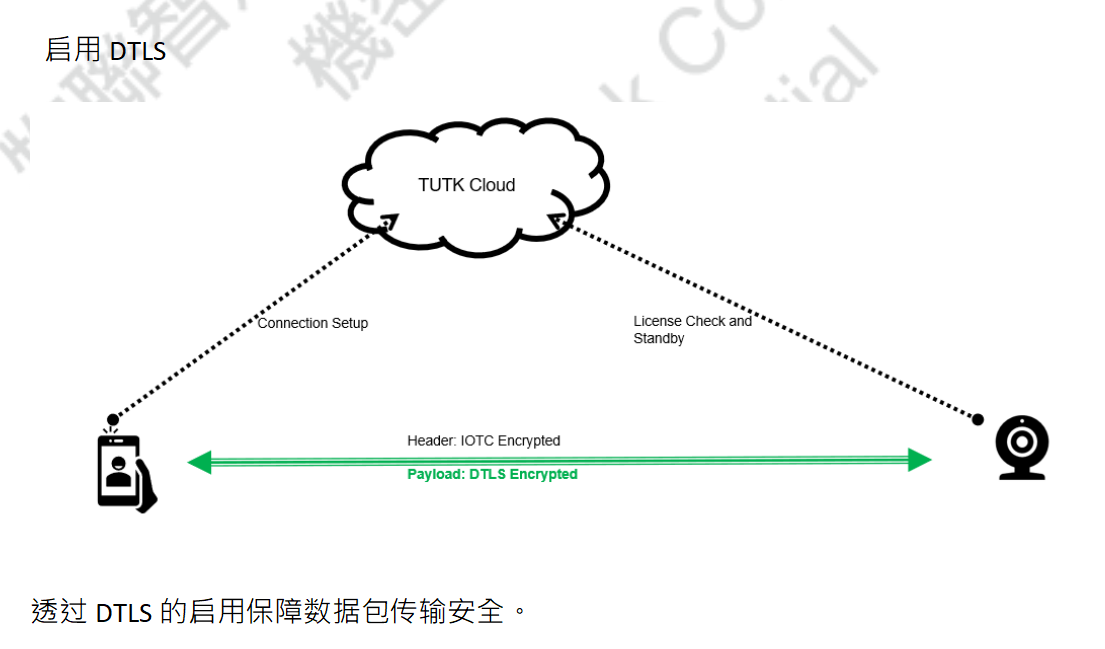




* **Authkey：IOTC层连线的密码，在没有开启DTLS的情况下，可以解决设备欺诈的问题。**



* **DTLS：更安全的加密方式，开启后，数据传输更安全，以及可以解决设备冒用的问题。**



##### 二、SDK版本要求？

客户端和设备端：需要升级到3.1.10以上的版本（3.1.10.x，3.3.x，3.4.x）。具体版本以跟FAE确认的版本为准。

P2P服务器版本：如果设备使用了3.4.x版本的SDK，则P2P服务器需要使用3.3.x以上版本（P2PServer\_3.3.x,P2PServer\_3.4.x），具体版本以跟FAE确认的版本为准。

##### 三、如何打开Authkey？

APP端：使用IOTC\_Connect\_ByUIDEx去连线设备。

设备端： 使用IOTC\_Device\_LoginEx去登录服务器。

备注：authkey是个8位的ascii字符，不能有’\0’结束符。所以在某些场景下做authkey数据的拷贝（如strcpy）的时候，需要注意。

如何更新服务器上面的authkey？

调用IOTC\_Device\_Update\_Authkey（your\_new\_authkey）

##### 四、如何打开DTLS？

APP端：使用avClientStartEx建立AV通道，参数security\_mode = AV\_SECURITY\_DTLS 。

设备端： 使用avServStartEx建立AV通道，参数security\_mode = AV\_SECURITY\_DTLS 。

如果参数security\_mode = AV\_SECURITY\_SIMPLE ，SDK则使用TUTK私有加密方式对数据进行加密，**需注意的是两端必须使用相同的加密方式才能保证数据正常传输。**

##### 兼容性设计：

**如果使用了DTLS的话，可以将authkey作为安全上的一种补充，也可以选择不用authkey，这个取决于产品设计者在安全性和用户体验上怎么取舍，因为引入authkey，在兼容性设计上可能会变得复杂，从而可能导致用户体验下降。如果不需要考虑兼容性，则无此问题。**

###### Authkey的兼容性设计：

* 1. 如果有使用DTLS，可以不使用authkey。**如果不使用authkey，则可以跳过1.2部分。**
  2. 如果仍然想使用Authkey（来获得更好的安全性），且之前有出旧版本的设备，可以根据以下不同的情景做兼容性设计：
     1. APP端可以知道设备有没有使用authkey，这种情况下，

如果设备有使用，则用IOTC\_Connect\_ByUIDEx去连接设备。

否则，使用IOTC\_Connect\_ByUID\_Parallel去连线设备。

* + 1. APP无法知道设备有没有使用authkey，则这种情况，SDK建议升级到3.4.3以上版本，服务器建议升级到3.4.3以上版本。这种版本组合，支持APP用默认authkey去连接non-authkey设备（即旧设备）和authkey为8个0的设备。

APP和设备都使用”00000000”作为初始的authkey，APP在第一次连线设备后，可以从IOTC\_Session\_Check\_Ex拿到设备是否使用authkey，在本地或者设备管理服务器记录下这个flag，下次连线时的做法参考1.2.1，做不同的处理。

如果设备支持authkey，第一次添加成功后需要修改掉这个默认的authkey。设备端修改后，需要调用IOTC\_Device\_Update\_Authkey（your\_new\_authkey）更新本地和服务器的authkey。

**备注：**

**以下两种情况，IOTC\_Connect\_ByUID\_Parallel或者IOTC\_Connect\_ByUIDEx连线会超时，并且不会拿到特别的错误码：**

**（1）一端使用了authkey，一端没使用authkey。**

**（2）APP使用了错误的authkey。**

如果使用authkey，也需要考虑用户体验的问题，即用户最好不要感知authkey的存在，可以将用户输入密码作为av password，然后对这个密码做md5加密，取其中8位作为authkey。后面修改av password的时候，authkey也会跟着修改，实现动态的authkey。

如果有OTA，则设备在OTA后，需要考虑ota的实作方法，是否需要恢复为初始的authkey。

###### DTLS的兼容性设计：

2.1如果之前的设备都是DTLS的设备，则不需要再做兼容性设计，可忽略后面的内容。

2.2如果之前的设备，有NON-DTLS的设备，则可以按照以下方式做兼容性设计：

* （1）先检查AVClientStartOutConfig 结构体中，是否有[security\_mode](F:/mysharefiles/TUTK_Kalay_Platform_SDK_3.4.3.0-alpha2-0-g719f43f_openssl_shared/UserManual/APIRefGuide/struct_a_v_client_start_out_config.html" \l "a0a65729e56bab4337a402f386f44e994)这个输出参数，有的话，可以在第一次创建通道（即avClientStartEx）时，AVClientStartInConfig填入参数 [security\_mode](F:/mysharefiles/TUTK_Kalay_Platform_SDK_3.4.3.0-alpha2-0-g719f43f_openssl_shared/UserManual/APIRefGuide/struct_a_v_client_start_out_config.html" \l "a0a65729e56bab4337a402f386f44e994) = AV\_SECURITY\_AUTO（即由SDK自动选择加密方式），成功后，将[security\_mode](F:/mysharefiles/TUTK_Kalay_Platform_SDK_3.4.3.0-alpha2-0-g719f43f_openssl_shared/UserManual/APIRefGuide/struct_a_v_client_start_out_config.html" \l "a0a65729e56bab4337a402f386f44e994)记录下来，然后在后面根据这个值带入AVClientStartInConfig 做连线。否则，参考下面步骤做设计。
* （2）在avClientStartEx中，先用[security\_mode](F:/mysharefiles/TUTK_Kalay_Platform_SDK_3.4.3.0-alpha2-0-g719f43f_openssl_shared/UserManual/APIRefGuide/struct_a_v_client_start_out_config.html" \l "a0a65729e56bab4337a402f386f44e994) = AV\_SECURITY\_DTLS，如果成功，则记录下[security\_mode](F:/mysharefiles/TUTK_Kalay_Platform_SDK_3.4.3.0-alpha2-0-g719f43f_openssl_shared/UserManual/APIRefGuide/struct_a_v_client_start_out_config.html" \l "a0a65729e56bab4337a402f386f44e994)，并在后续的调用中，沿用此种方式。如果不成功，则继续使用[security\_mode](F:/mysharefiles/TUTK_Kalay_Platform_SDK_3.4.3.0-alpha2-0-g719f43f_openssl_shared/UserManual/APIRefGuide/struct_a_v_client_start_out_config.html" \l "a0a65729e56bab4337a402f386f44e994) = AV\_SECURITY\_SIMPLE 去做avClientStartEx，如果成功，记录下[security\_mode](F:/mysharefiles/TUTK_Kalay_Platform_SDK_3.4.3.0-alpha2-0-g719f43f_openssl_shared/UserManual/APIRefGuide/struct_a_v_client_start_out_config.html" \l "a0a65729e56bab4337a402f386f44e994)，并在后续的调用中，沿用此种方式。
* 如果上面两种方式都失败，则下次连线，重新做两种方式的尝试。

备注：如果使用了authkey，那么DTLS也可以不记录[security\_mode](F:/mysharefiles/TUTK_Kalay_Platform_SDK_3.4.3.0-alpha2-0-g719f43f_openssl_shared/UserManual/APIRefGuide/struct_a_v_client_start_out_config.html" \l "a0a65729e56bab4337a402f386f44e994)，而是填入[security\_mode](F:/mysharefiles/TUTK_Kalay_Platform_SDK_3.4.3.0-alpha2-0-g719f43f_openssl_shared/UserManual/APIRefGuide/struct_a_v_client_start_out_config.html" \l "a0a65729e56bab4337a402f386f44e994) = AV\_SECURITY\_AUTO由SDK自己选择加密方式，但是，有可能导致创建通道的时间变长。

如果没有使用authkey，且第一次使用avClientStartEx(mode: AV\_SECURITY\_AUTO)去建立通道，在获取到设备实际使用的加密方式后，后续不能再用AV\_SECURITY\_AUTO，否则可能导致设备被冒用的情况存在。