# TLS对接示例说明

## 背景介绍

互联网的通信安全，建立在SSL/TLS协议之上。

不使用SSL/TLS的HTTP通信，就是不加密的通信。所有信息明文传播，带来了三大风险。

（1） **窃听风险**（eavesdropping）：第三方可以获知通信内容。

（2） **篡改风险**（tampering）：第三方可以修改通信内容。

（3） **冒充风险**（pretending）：第三方可以冒充他人身份参与通信。

SSL/TLS协议是为了解决这三大风险而设计的，希望达到：

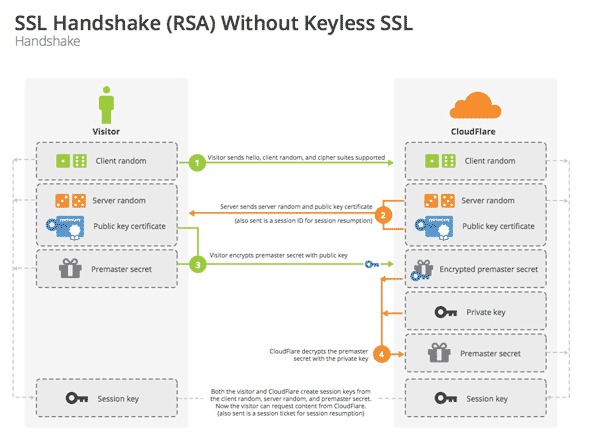
（1） 所有信息都是**加密传播**，第三方无法窃听。

（2） 具有**校验机制**，一旦被篡改，通信双方会立刻发现。

（3） 配备**身份证书**，防止身份被冒充。

## TLS协议

### TLS协议图解



### TLS握手协议

  Client                                               Server  
  
  
      ClientHello                  -------->  
                                                      ServerHello  
                                                     Certificate\*  
                                               ServerKeyExchange\*  
                                              CertificateRequest\*  
                                   <--------      ServerHelloDone  
      Certificate\*  
      ClientKeyExchange  
      CertificateVerify\*  
      [ChangeCipherSpec]  
      Finished                     -------->  
                                               [ChangeCipherSpec]  
                                   <--------             Finished  
      Application Data             <------->     Application Data

（\*代表可选消息）

1. 客户端发起TLS握手请求
2. 服务端ServerHello阶段

a、（服务器端发送消息）

服务端发送自己的证书 （绝大部分情况都需要）

服务端发送密钥交换消息

服务端发送客户端证书请求 （在双向认证场景下需要）

b、（客户端对ServerHello消息的处理）

客户端校验服务端证书

将服务端密钥放到会话缓存中

保存客户端证书请求中的证书类型、可信任CA等，供客户端发送证书时使用

3、客户端响应及Finish阶段

a、（客户端发送消息）

客户端发送自己的证书 （在双向认证场景下需要）

客户端发送密钥交换消息

客户端发送验证证书请求 （在双向认证场景下才需要）

客户端将生成的密钥设置为会话密钥 [ChangeCipherSpec]

客户端发送ClientFinish

b、（服务端对客户端消息的处理）

服务端验证客户证书

将客户端密钥保存到会话缓存中

1. 服务端Finish阶段

服务端将生成的密钥设置为会话密钥 [ChangeCipherSpec]

服务端发送Finish消息，握手完成。

### 证书校验

证书的合法性校验包含了

1、检查证书是否符合X.509格式标准

2、检查证书的有效期是否过期

3、校验对端证书是否是由合法根CA签发

4、检查证书是否被吊销

5、校验证书主体通用名称 （Common Name）

## IOT系统中的TLS对接

在与IOT系统对接时，需要特别注意证书校验的第3和第5点。

### 信任IOT自认证CA证书

目前IOT系统的证书默认是IOT自认证的CA证书发布的设备证书，此证书不属于公认的权威CA机构发布的证书，不是公认的有效证书。因此客户端在对IOT证书进行校验时，必须先获取到IOT的CA证书，并且把IOT的CA证书加入到客户端信任列表中，然后对IOT设备证书的校验才能成功。

商用环境上，局点使用向权威CA机构申请的商用证书，则不存在此问题。客户端不需要进行特殊处理。

### 关闭证书域名校验

目前IOT系统的默认证书域名是[www.smarthome-demo.com/www.iotplatform-demo.com](http://www.smarthome-demo.com/www.iotplatform-demo.com)。在联调测试或者演示环境中，一般不会为IOT申请域名和使用域名进行访问。访问时直接使用的是IP地址，而TLS在校验的时候，一般会校验访问服务器的url中的域名信息与服务器证书中的域名信息是否匹配，因此客户端校验失败。所以客户端在访问IOT系统时，需要关闭域名的校验。

商用环境上，局点会申请域名，并且访问时会使用域名进行访问，只要局点的域名和局点申请的证书中的域名匹配，则不存在此问题。客户端不需要进行特殊处理。

### 双向认证场景

双向认证场景下，客户端不但要校验服务端的证书，也需要提供自己的证书以供服务端进行校验。IOT系统默认的信任证书包含了IOT的CA证书，因此Smarthome在接入IOT platform时验证不有问题（Smarthome使用的证书也是IOT CA签发的）。

此外，为保证第三方NA接入IOT校验不会有问题，IOT Platform还包含了权威机构baltimore digicert entrust geotrust globalsign thawte verisign的CA证书，如果第三方NA的设备证书是由上述权威机构或者IOT CA签发的，IOT对NA的证书校验就能通过。如果第三方NA没有使用这些默认信任的CA签发的证书，那么在与IOT Platform进行对接的时候，需要提供第三方NA设备证书的签发CA的证书，并通知IOT Platform维护人员将此CA加入到IOT Platform的信任列表中。

## 对接示例代码

为方便第三方NA使用HTTPS协议接入到IOT Platform，避免对接过程出现证书校验的问题，本文提供了JAVA中三种方式发送HTTPS消息的代码示例，并在示例代码中为 信任IOT自认证CA证书、关闭证书域名校验、使用双向认证场景功能给出了具体的实现。

代码中有比较详细的注释，因此不在文档中再另外解释了

### 基于Apache的HttpClient



### 基于JDK的 HttpsURLConnection



### 基于Spring的 RestTemplate

