OpenVPN

1. 简介

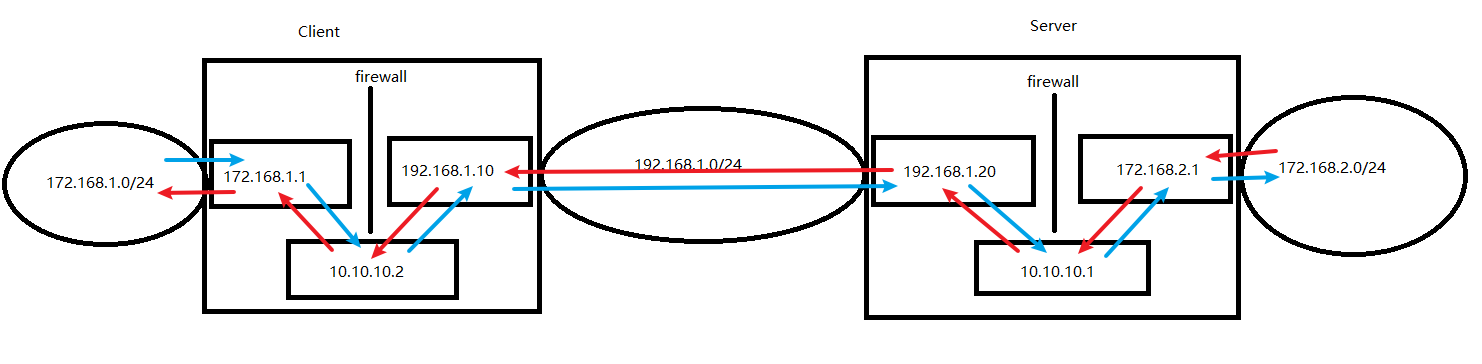
OpenVPN是一个开源的虚拟专用网络（VPN）软件，可以用于在不安全的网络（如互联网）上为用户提供安全的网络连接。它使用加密的数据通道来确保数据的安全性，并可以通过多种方式来构建虚拟网络，包括使用TCP或UDP协议，使用安全套接字层（SSL）或传输层安全（TLS）加密数据通道，或者使用混合模式，即同时使用多种方式。

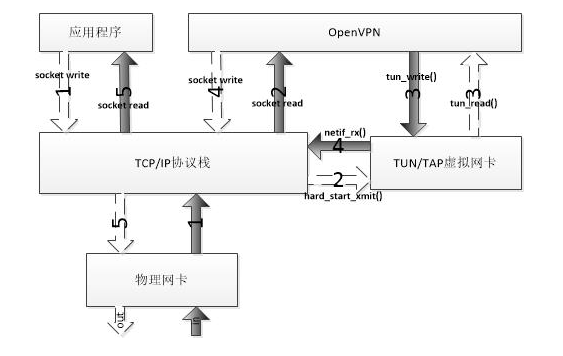
OpenVPN可以在多种平台上运行，包括Windows、MacOS、Linux、Android和iOS等。它可以与多种路由器和网络设备集成，并提供了丰富的配置选项，使用者可以根据自己的需要来自定义设置。

1. 基本原理

OpenVPN 的技术核心是虚拟网卡，其次是 SSL 协议实现。虚拟网卡是使用网络底层编程技术实现的一个驱动软件。安装此类程序后主机上会增加一个非真实的网卡（TAP 或 TUN），并可以像其它网卡一样进行配置。服务程序可以在应用层打开虚拟网卡，如果应用软件（如网络浏览器）向虚拟网卡发送数据，则服务程序可以读取到该数据。如果服务程序写合适的数据到虚拟网卡，应用软件也可以接收得到。

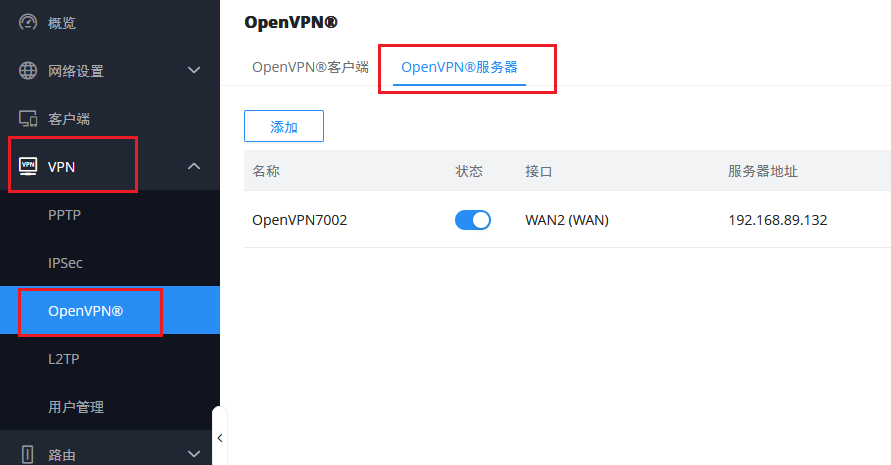
1. OpenVPN架构以及数据处理流程





1. GWN700x系列上的OpenVPN配置面板（C/S）

**服务端：**



**服务器的基本配置，如下图，标注红色\*号的是只必要配置项，不能为空。**



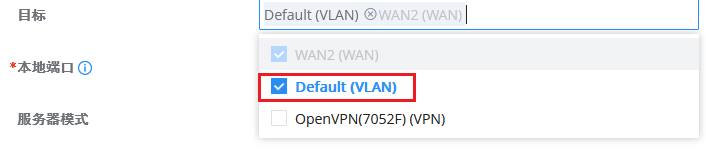
**名称：**自定义的服务器名，1-64位，目前支持所有字符。

**状态：**服务器的开关。

**协议：**OpenVPN使用的传输层封装协议，支持UDP/TCP。

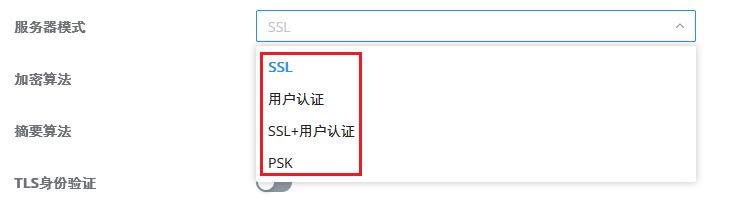
**接口：**选择当前连接公网的WAN口，因为VPN实际上使用的仍然是公网的线路资源，所以VPN的建立和流量传输都需要依靠公网接口。

**目标：**代表可以通过VPN访问的目的网络，默认只选中当前选择的WAN口，需要手动把需要通信的VLAN勾选。

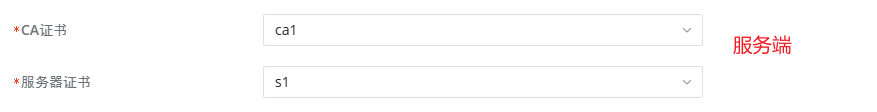


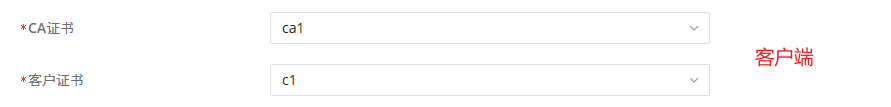
**本地端口：**服务器提供服务的端口号，默认是1194，修改范围是1-65535。

**服务器模式：**服务器采用的认证模式，支持SSL/用户认证/SSL+用户认证/PSK。

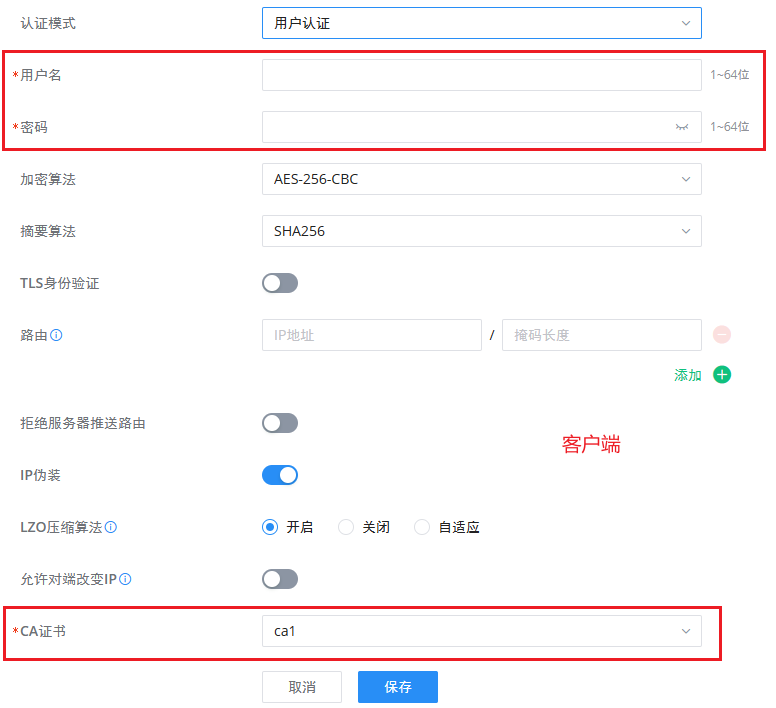


**SSL：**SSL（Security Socket Layer）协议是基于WEB应用的安全协议，它包括：服务器认证、客户认证（可选）、SSL链路上的数据完整性和SSL链路上的数据保密性。





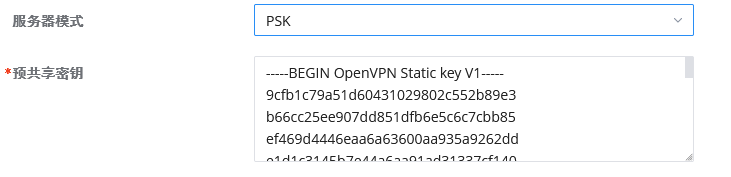
**用户认证：**客户端使用用户名和密码的验证方式，可以省略客户端证书，但是仍有一份服务器证书需要被用作加密。



**SSL+用户认证：**既需要服务器和客户端证书，也需要用户名和密码进行验证。

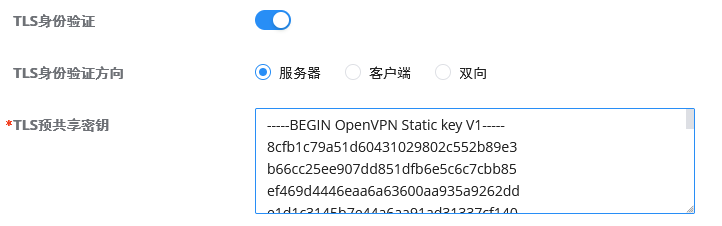


**PSK：**不需要CA，服务器证书和客户端证书，是基于点对点的放方式，使用PSK密钥进行认证。



**加密算法&摘要算法：**服务器与客户端之间数据交互采用的算法，默认为AES-256-CBC和SHA256。双端必须相同，才能成功连接server并互相通信。

**TLS身份验证：**一种加密的验证方式，TLS身份验证方向分为三种，服务器，客户端，双向。server的方向为服务器时，client的方向为客户端，才能连接server，server的方向为客户端时，client的方向为服务器，才能连接server，server的方向为双向时，client的方向为双向，才能连接server。默认关闭。



预共享密钥样板如下：

-----BEGIN OpenVPN Static key V1-----

8cfb1c79a51d60431029802c552b89e3

b66cc25ee907dd851dfb6e5c6c7cbb85

ef469d4446eaa6a63600aa935a9262dd

e1d1c3145b7e44a6aa91ad31337cf140

e37cfb45e83a66f3561ce314e303ba37

5fd29f932469e9e84e7de4bf6cefa254

1a34a772384b5f0fb5e0e7e23cd37ecc

5d6c2afc566f1cee2035a5a7d94dd78f

30992dee788c085946174f2bf574bfc1

5adee123accb7a00fa3367d635421fc2

ee5b75d8dce27dbedc5b924d9155dbb9

4fd498848a073ad239060dc331d2db13

a55a6a840e2a78a622190a2089249b62

c433f372743984f9d03d3147e32a69db

77ff2a51745533e2db88ec2e1496b5e7

056509f17de30dc8e5b87e4f56948788

-----END OpenVPN Static key V1-----

**允许重复的客户端证书:**允许多个客户端使用相同的客户端证书连接服务器。默认关闭。

**重定向网关：**与客户端建立连接后，会给客户端推送一条通过VPN接口的0.0.0.0的默认路由，也就是VPN客户端内网所有流量都走VPN。默认关闭。

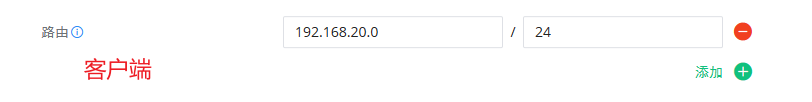




注意：目前我们GWN产品使用的openvpn topo是net30，该topo创建的路由表跟我们日常见的路由表逻辑有些不一样，可以自行研究。

**推送路由：**服务器可以推送指定的本地内网路由给客户端，下一跳地址为本地VPN接口。默认为空。作用与直接在客户端的路由配置项中填写服务器子网路由一样





注意：psk模式下服务器没有重定向网关和push路由的配置项

**LZO压缩算法：**使用LZO压缩传输的数据，必须与服务器端一致，不一致时，可以连接server,但是不能通信。默认开启。

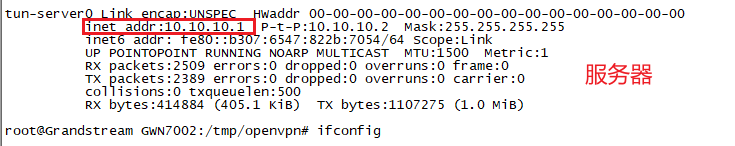
**允许对端改变IP：**用于对端地址经常变化的情况，允许对端IP改变后，也能顺利连接server。默认关闭。注意：开发说在我们的测试环境中表现不明显

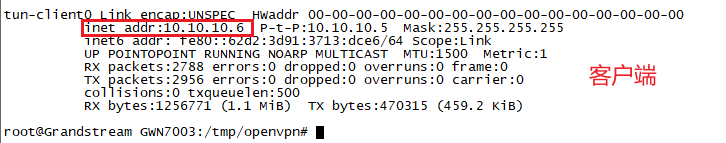
**CA证书：**CA证书，用于颁发服务器证书和客户端证书，客户端必须使用与服务器证书同一个CA证书颁发的客户端证书才能建立连接。

**服务器证书：**由CA证书颁发，用于数据加解密。

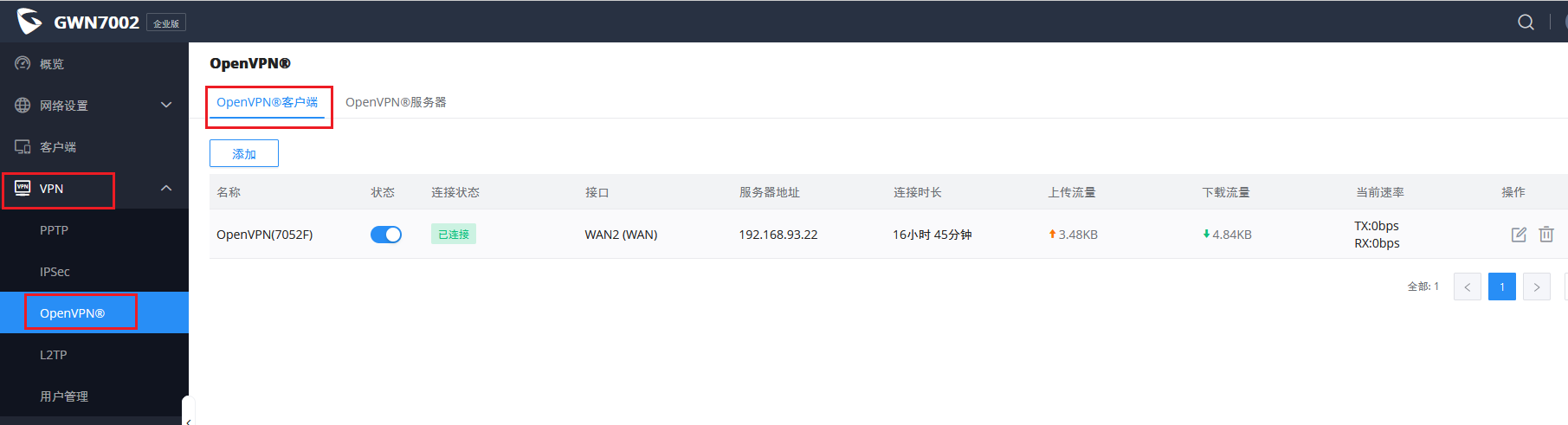
**IPV4隧道网络：**VPN两端使用的网络地址段，服务器默认使用第一个主机地址。







**客户端：**



**客户端的基本配置，如下图，标注红色\*号的是只必要配置项，不能为空。**



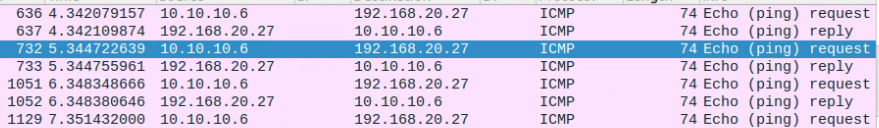
**远程OpenVPN服务器&&OpenVPN端口：**对端OpenVPN服务器的公网地址以及端口号。

**用户名&&密码：**当OpenVPN使用用户认证或者SSL+用户认证的时候，需要填写服务端分配的用户名和密码才能建立连接。

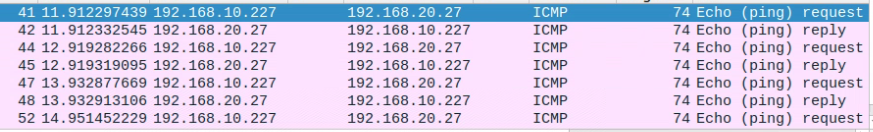
**路由：**客户端可以自己指定使用VPN接口的路由。

**拒绝服务器推送路由：**开启后服务器上配置的推送路由无法推送到客户端。默认关闭。

**IP伪装：**开启IP伪装后，对端收到的数据包的源地址为VPN的地址。

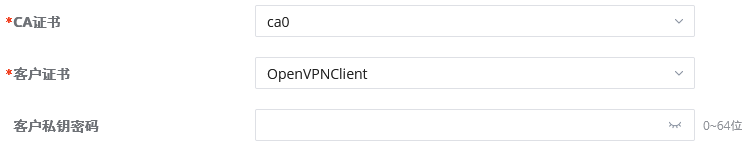


关闭的情况下为发送方的真实内网地址。默认开启。



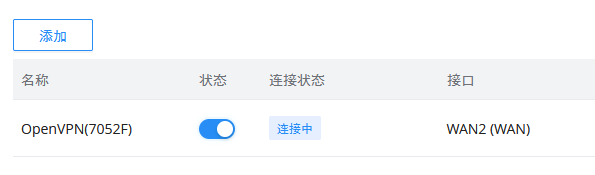
**客户证书：**由CA颁发的证书，必须使用与服务端证书必须是同一CA证书颁发的客户端证书才能与服务端建立连接。

**客户私钥密码：**客户端私钥的密码，如过Key没有加密则不需要填。针对一些设置了私钥密码的证书。当前GWN作为服务器时，客户端连接不需要该配置。



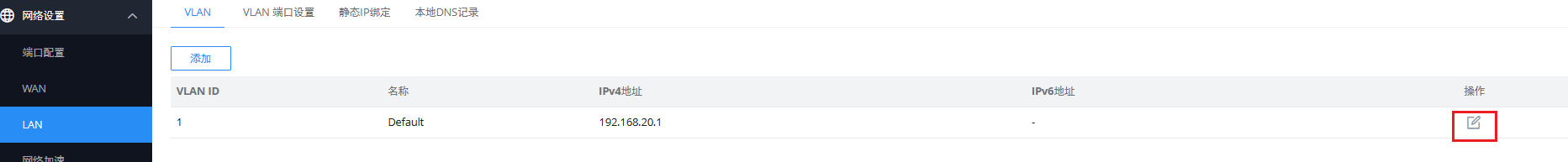
客户端连接的三种状态：未连接，连接中，已连接。







注意：VPN建立后要两端相通，除了VPN配置时添加路由还需要在各自的LAN配置的地址，选择相应的目标，允许流量的转发比如LAN转发到VPN，VPN转发到LAN。





**后台查看相关配置文件：/tmp/openvpn**

客户端配置文件：client.conf

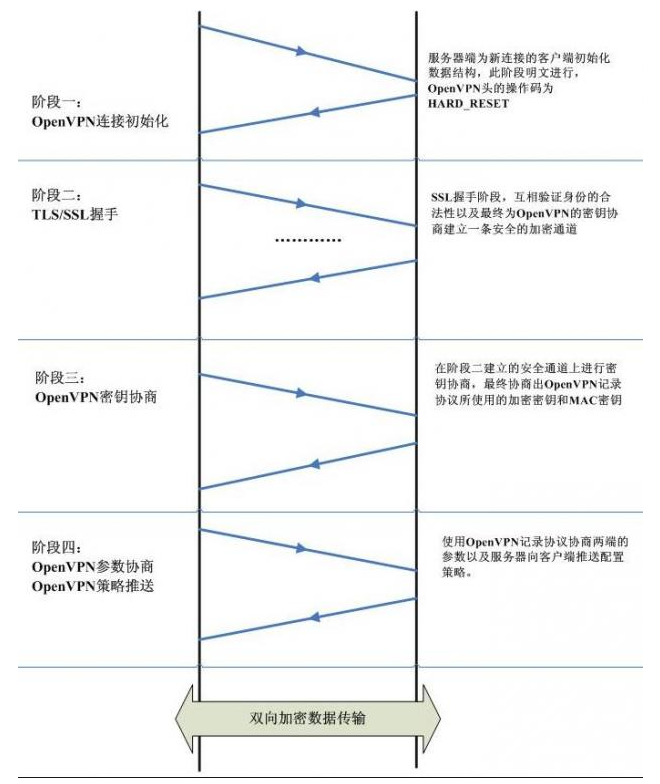


服务端配置文件：server.conf

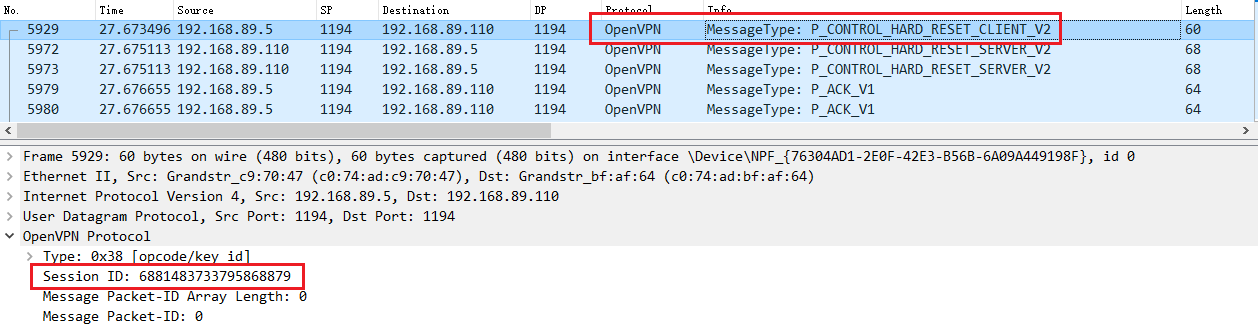


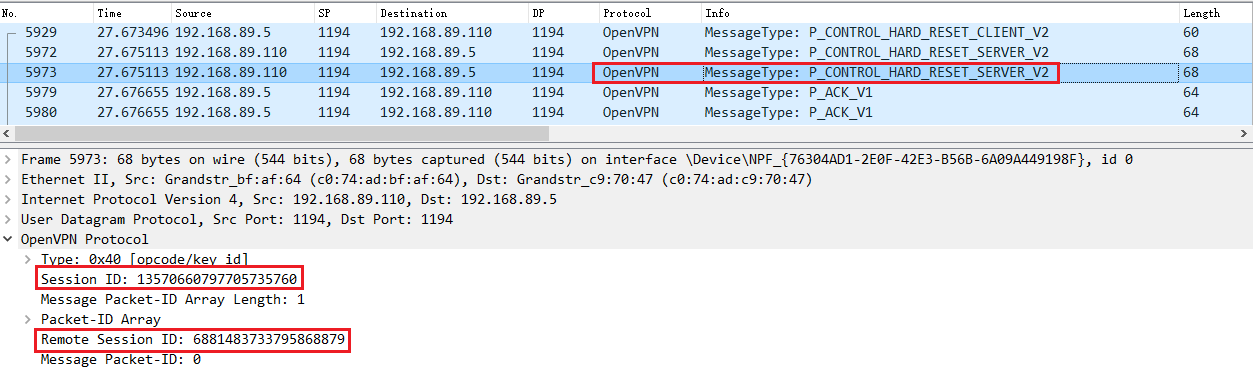
注意：配置文件还有一些其他配置，文档中显示的只是正在使用的

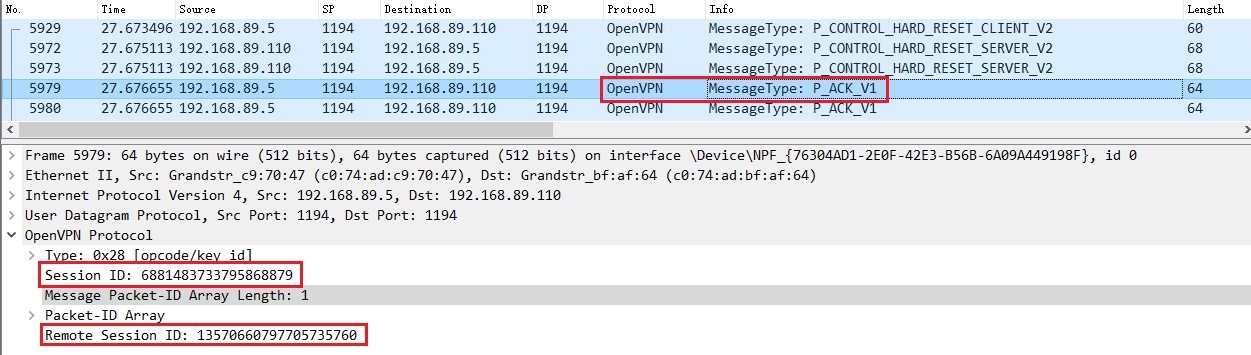
1. OpenVPN连接过程



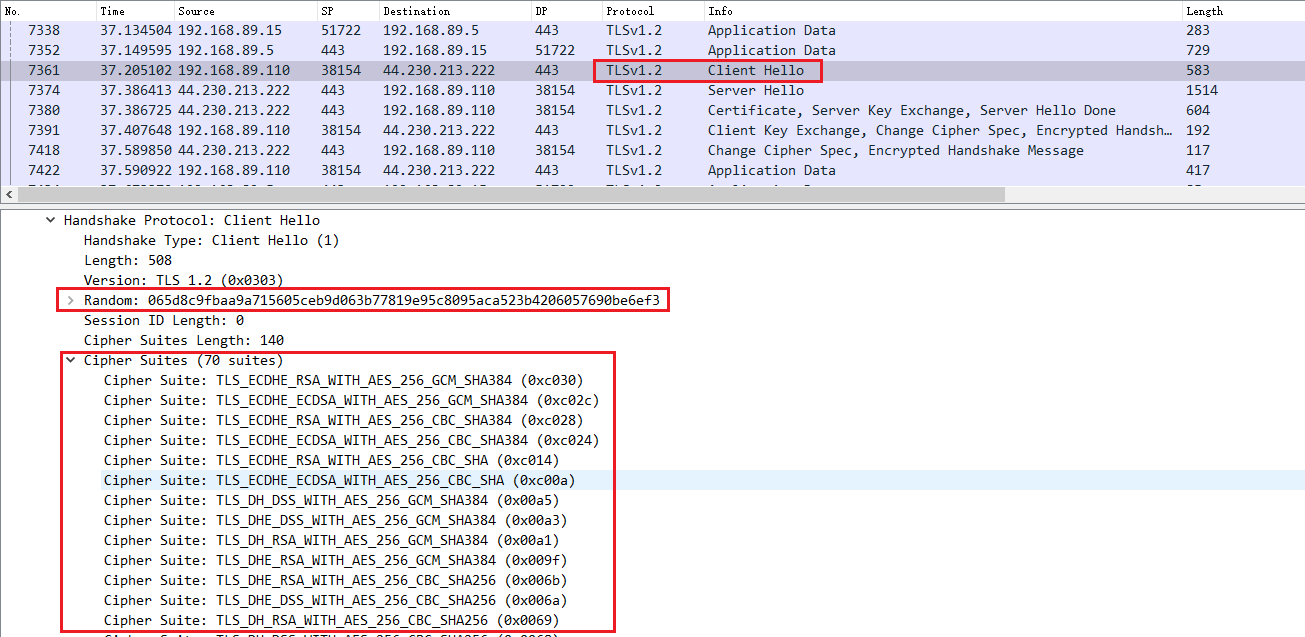
1. OpenVPN初始化：初始化的过程会形成一个会话，根据会话ID来交互数据



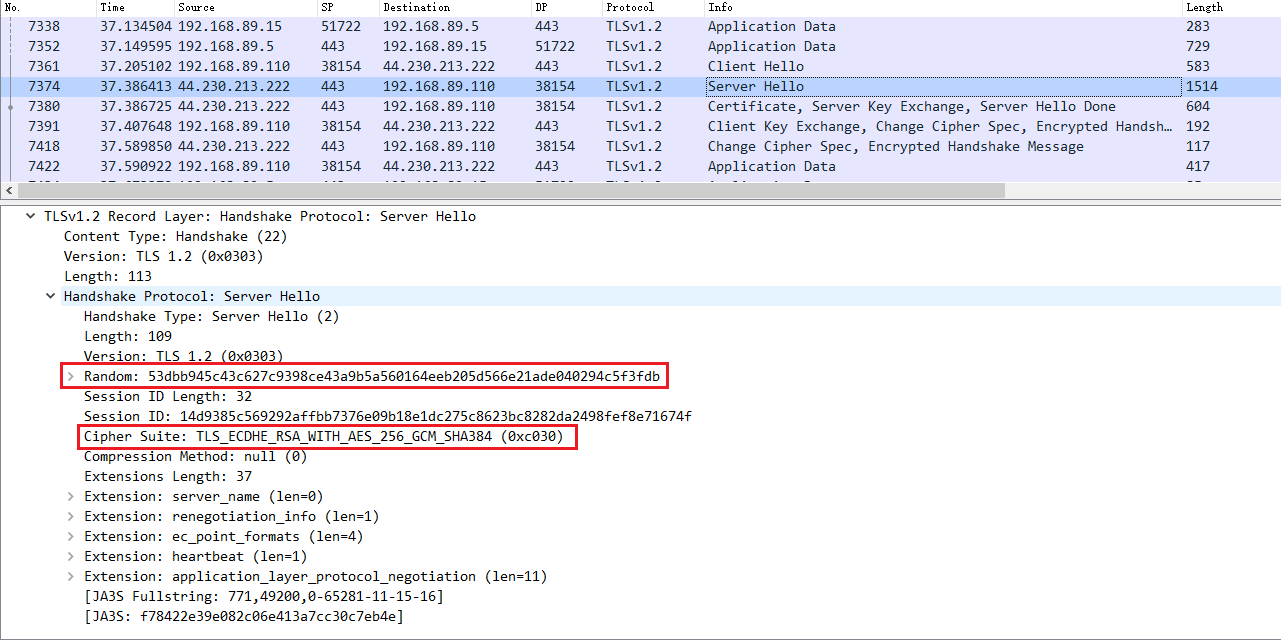




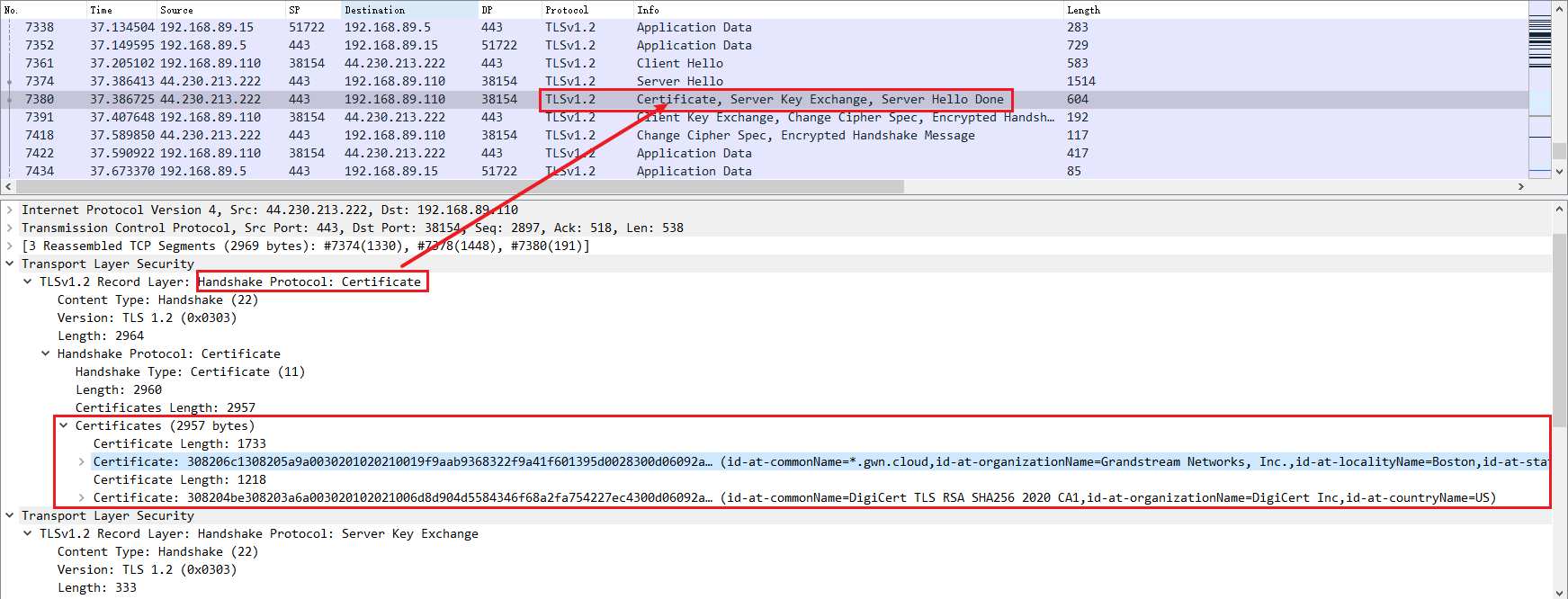
1. TLS握手（由于OpenVPN会加密握手过程，所以用https的握手过程作为参考）
2. 发送客户端随机数Random
3. 发送客户端支持的所有加密套件

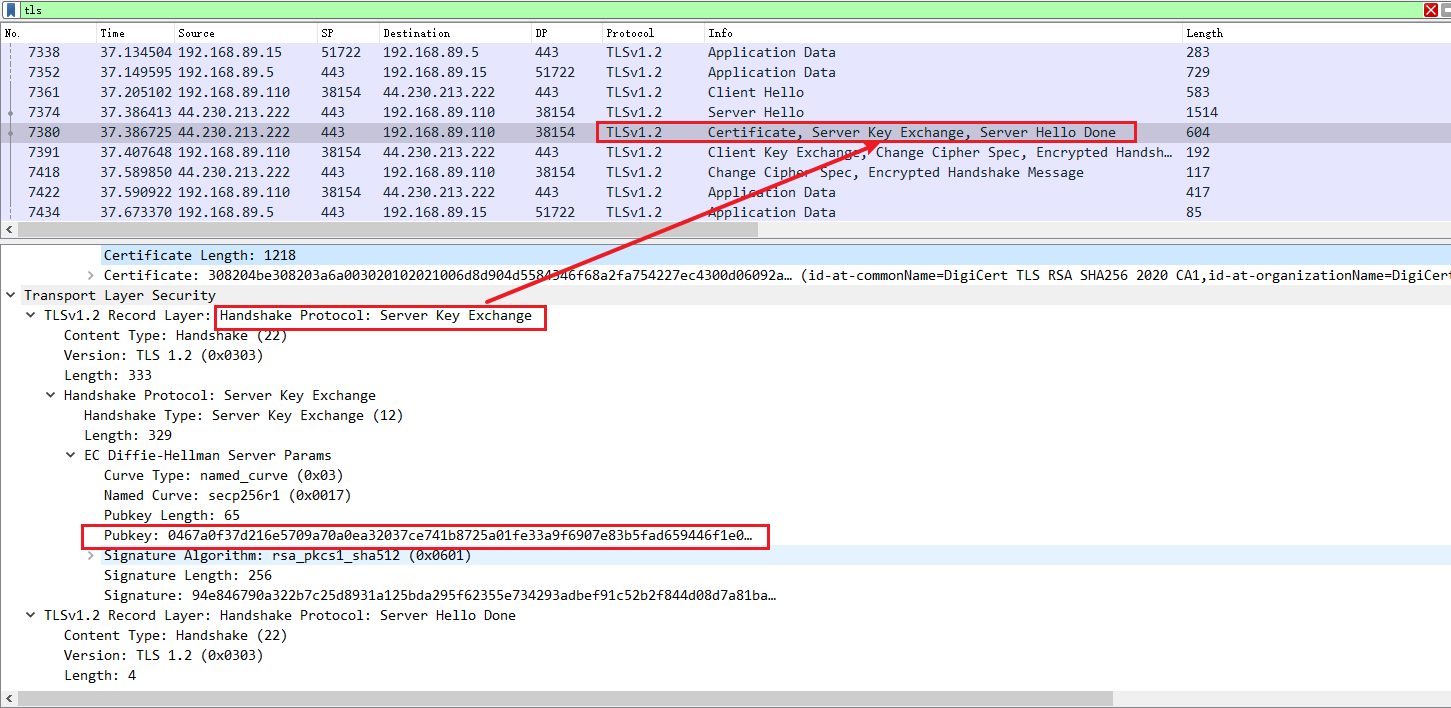


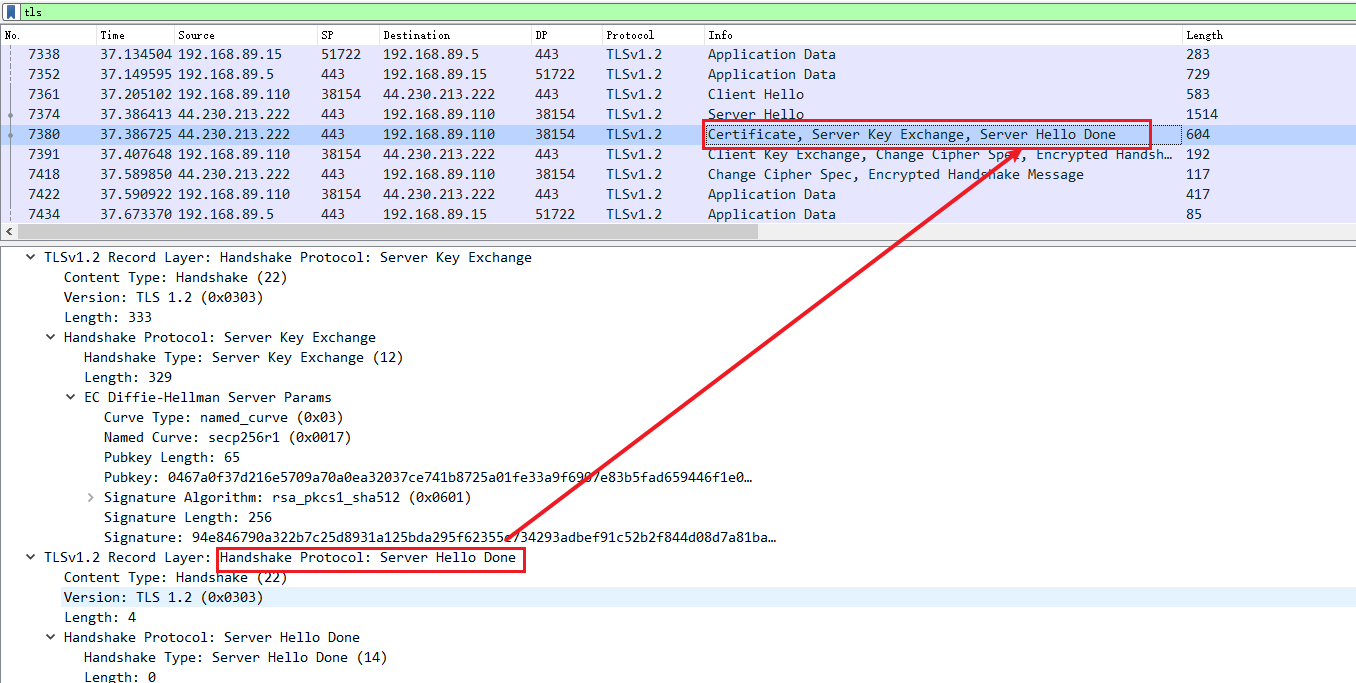
1. 服务端回复服务端随机数Random
2. 服务端回复选择使用的加密套件



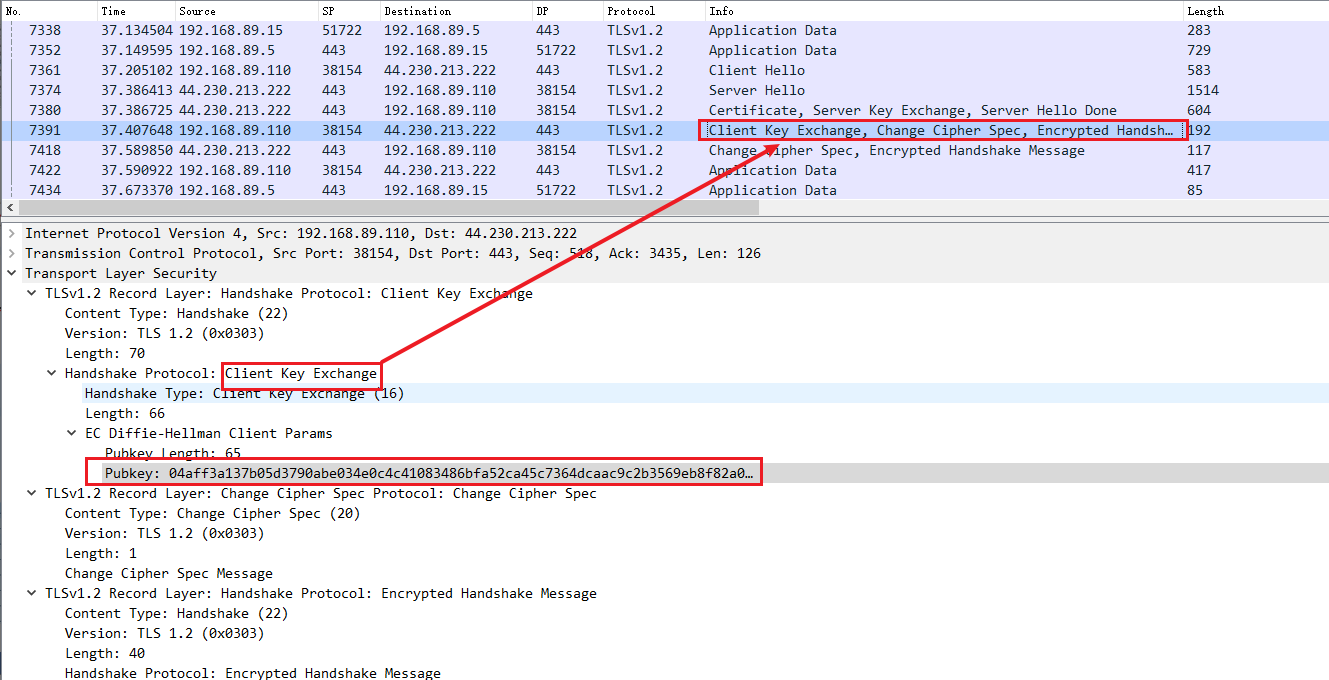
1. 服务端发送证书信息、服务端证书的公钥给客户端，公钥用于数据加密，Server Hello Done表示服务端的信息已经发送完毕



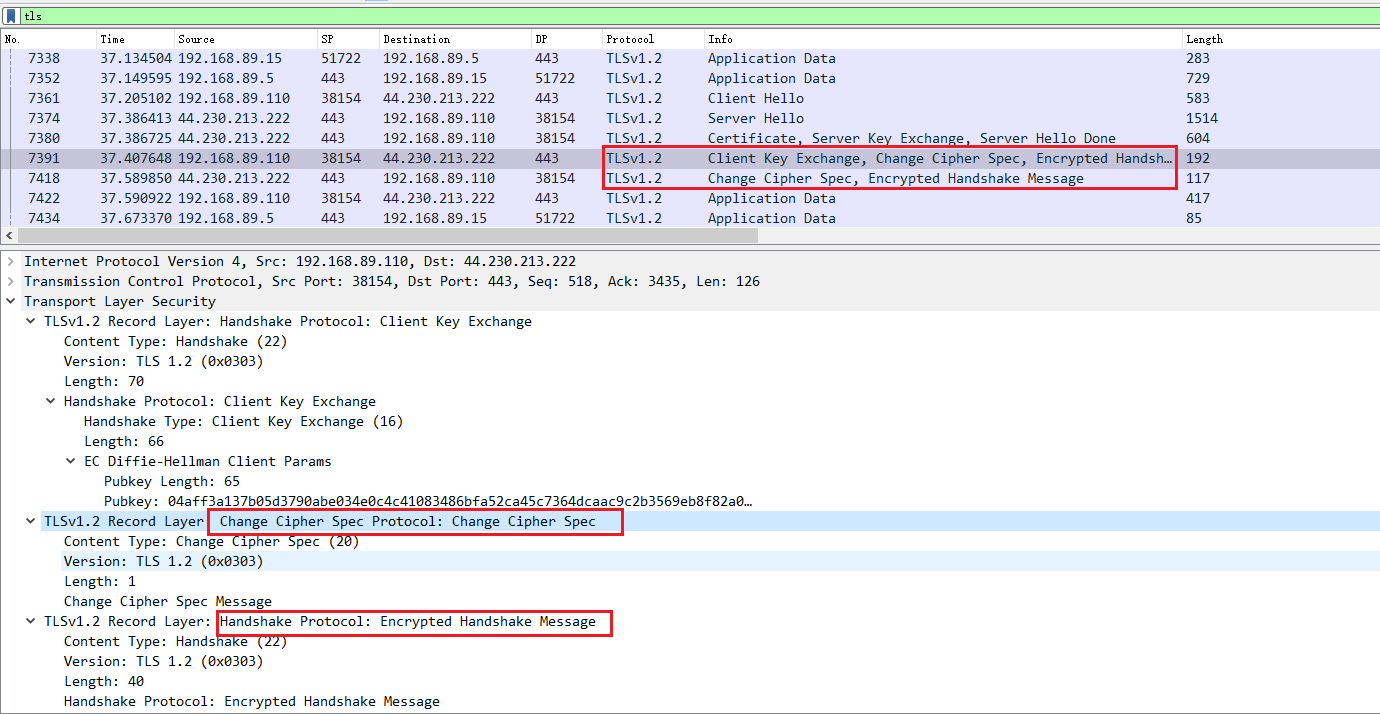




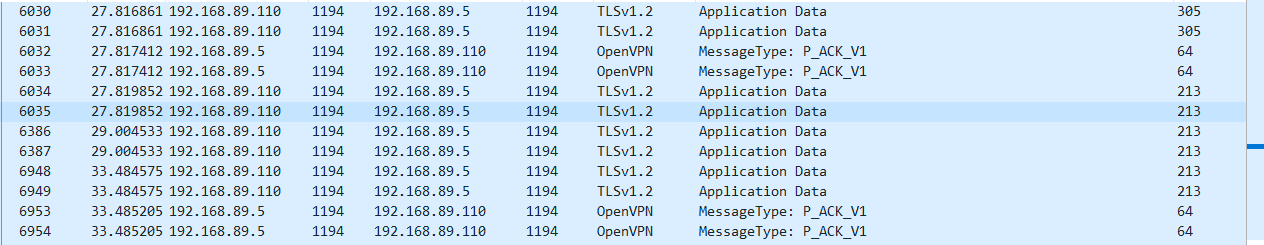
1. 客户端发送客户端证书公钥给服务端，用于数据加密



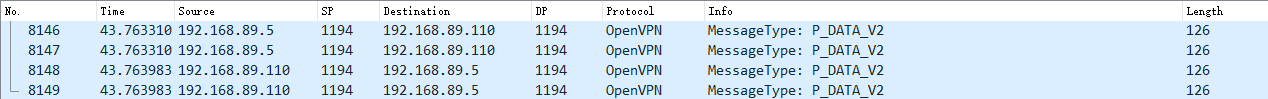
1. 最后双方都协商切换成加密模式，并且宣告TLS握手结束



1. 配置加密推送



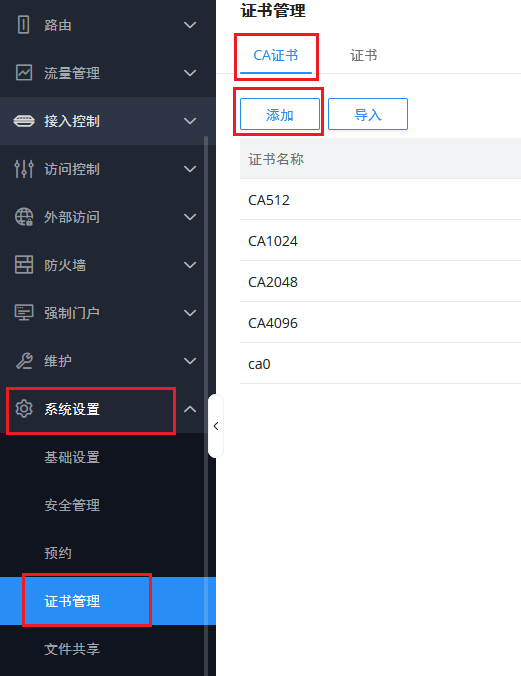
1. 数据加密传输

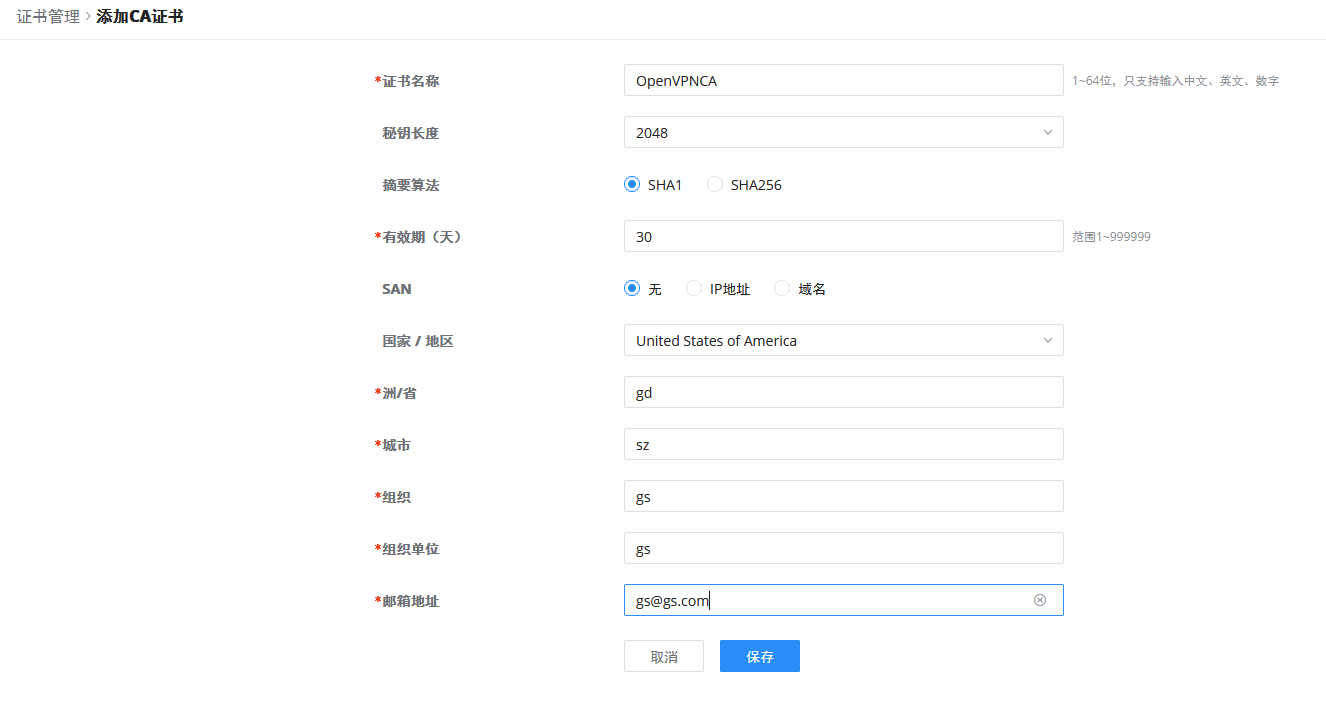


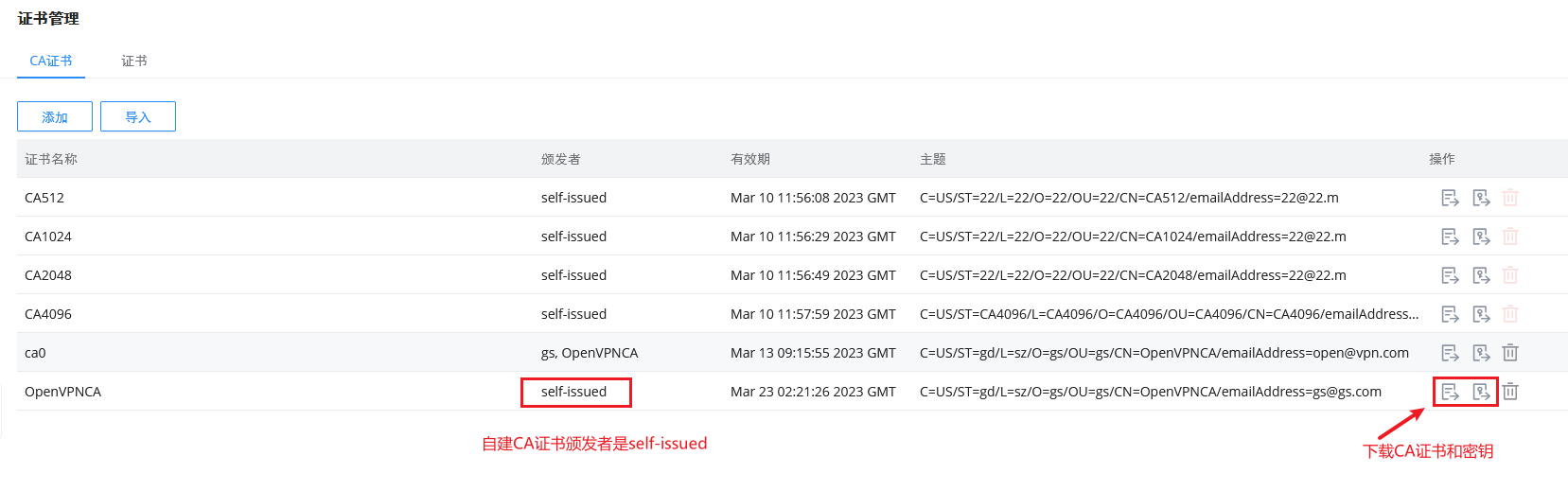
1. 配置案例

SSL认证模式

1. **创建CA证书（也可以在添加服务器时在配置页面创建）**

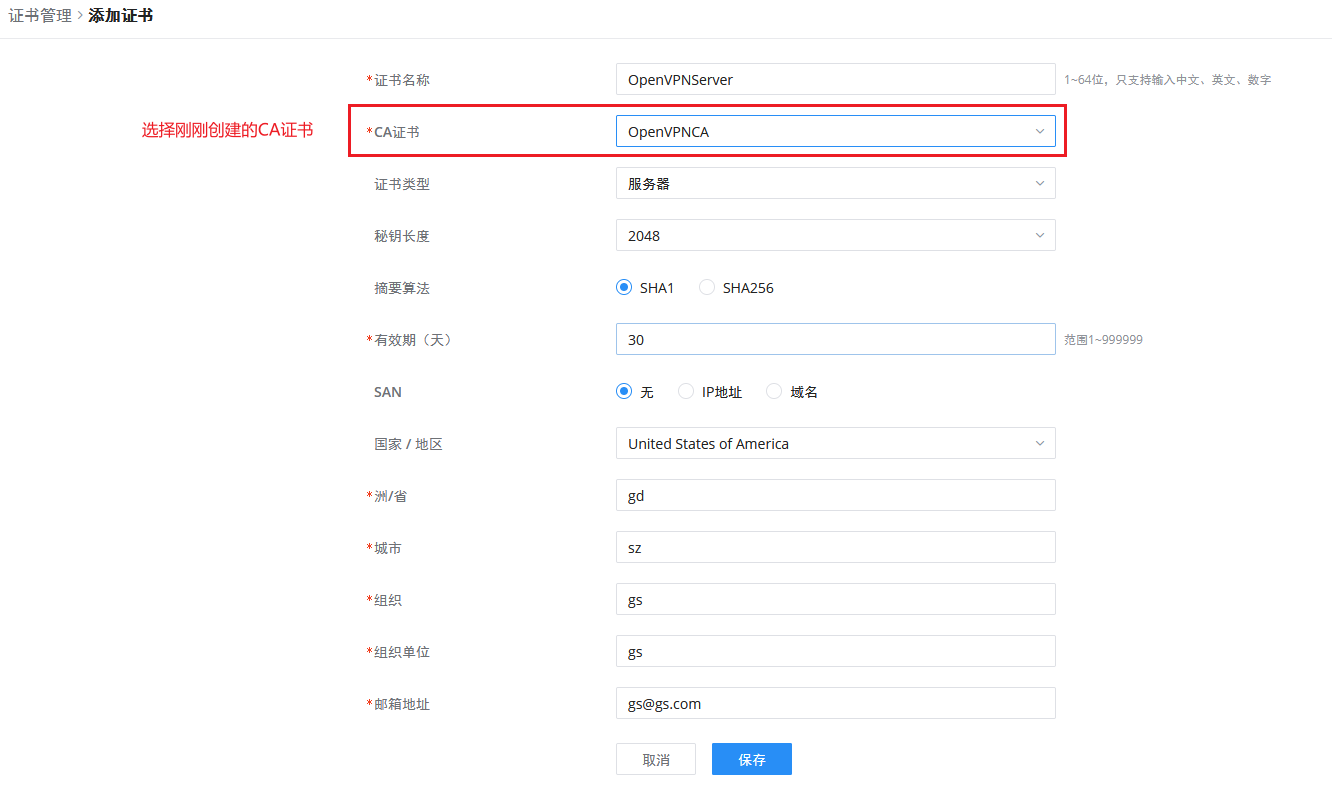




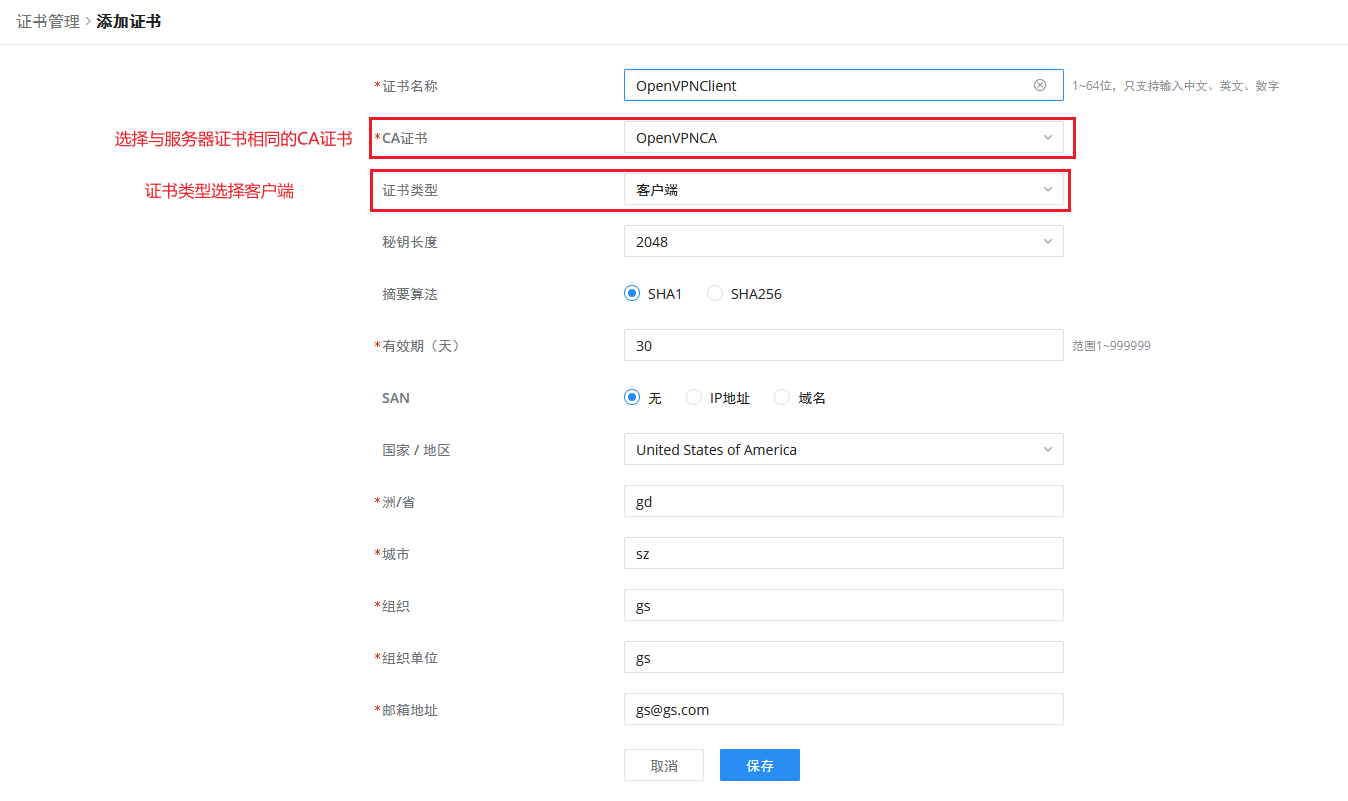


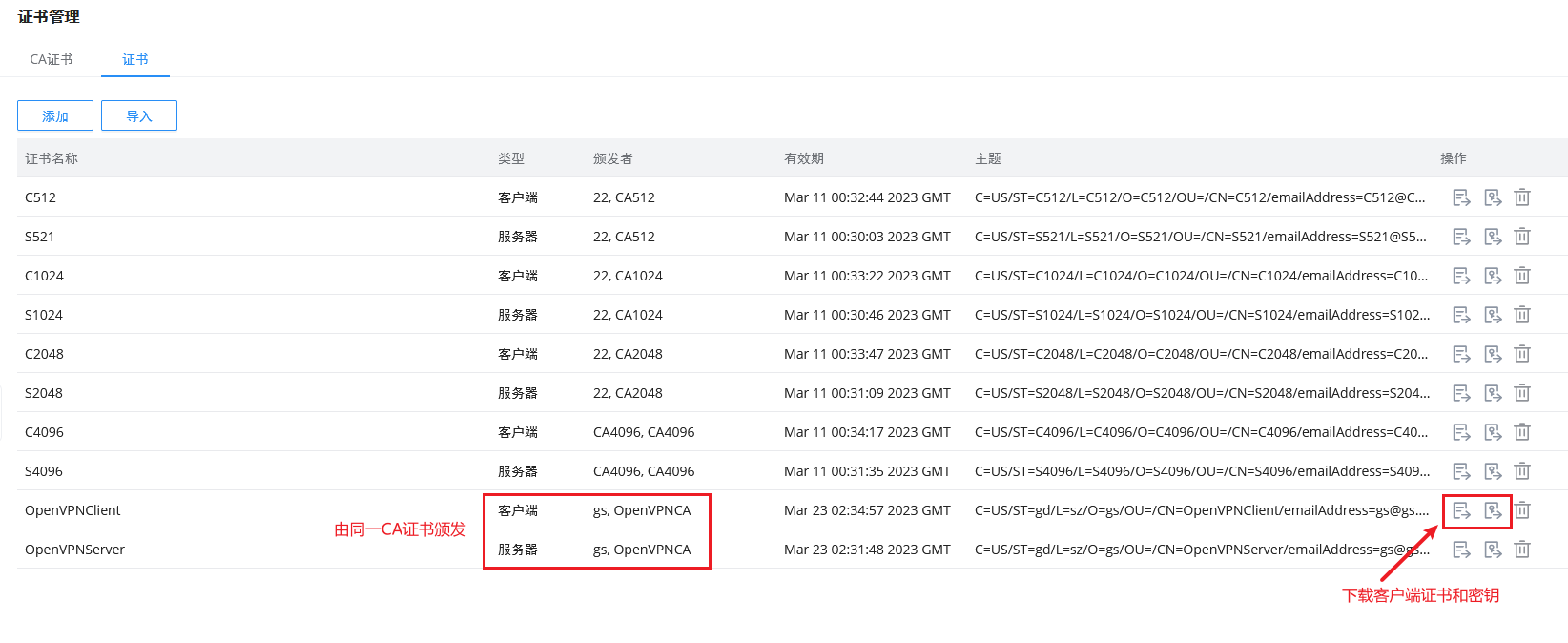
1. **创建服务器证书**





1. **创建客户端证书**



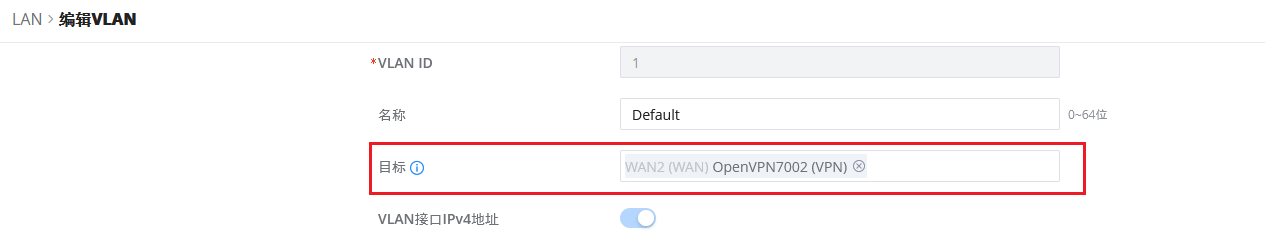


1. **添加OpenVPN服务器**

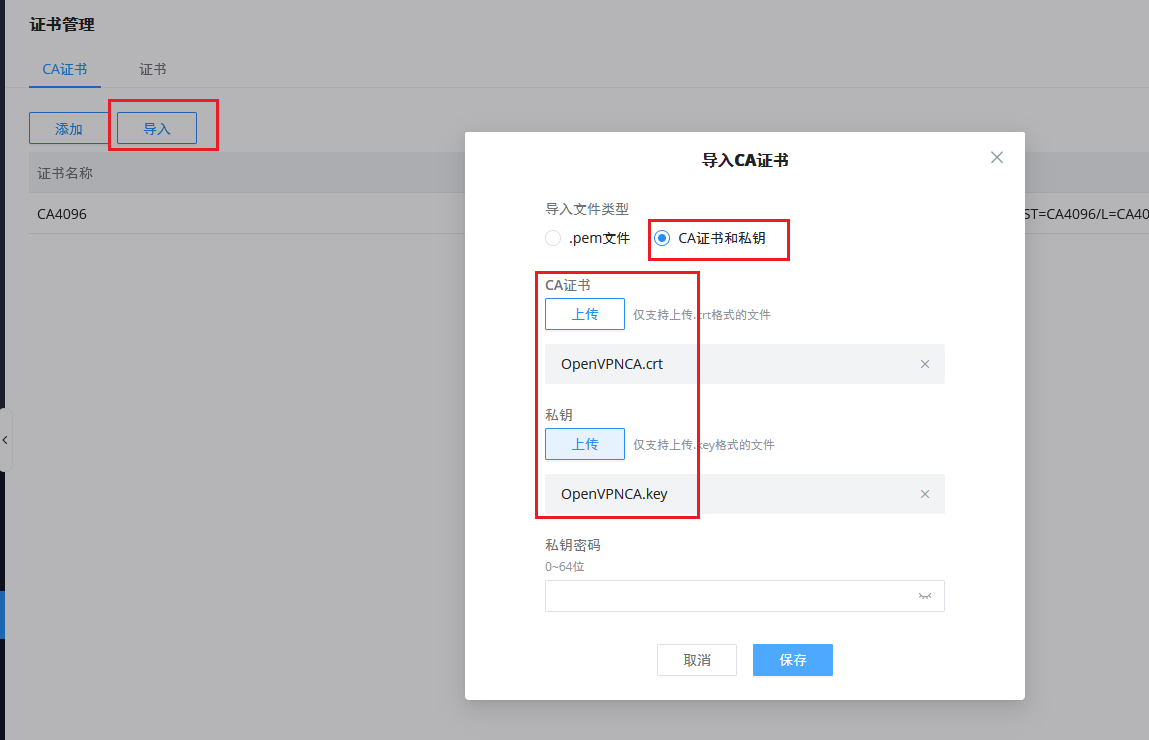


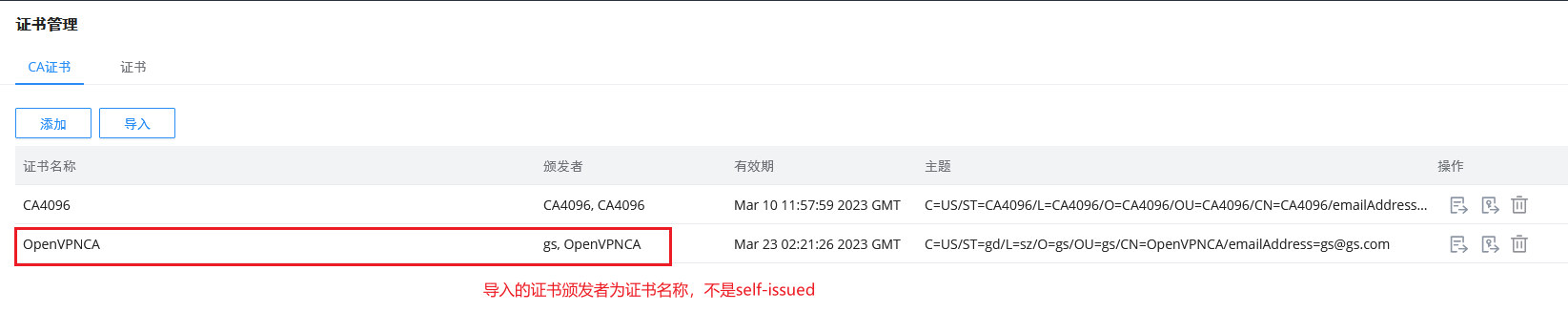


1. **配置LAN**

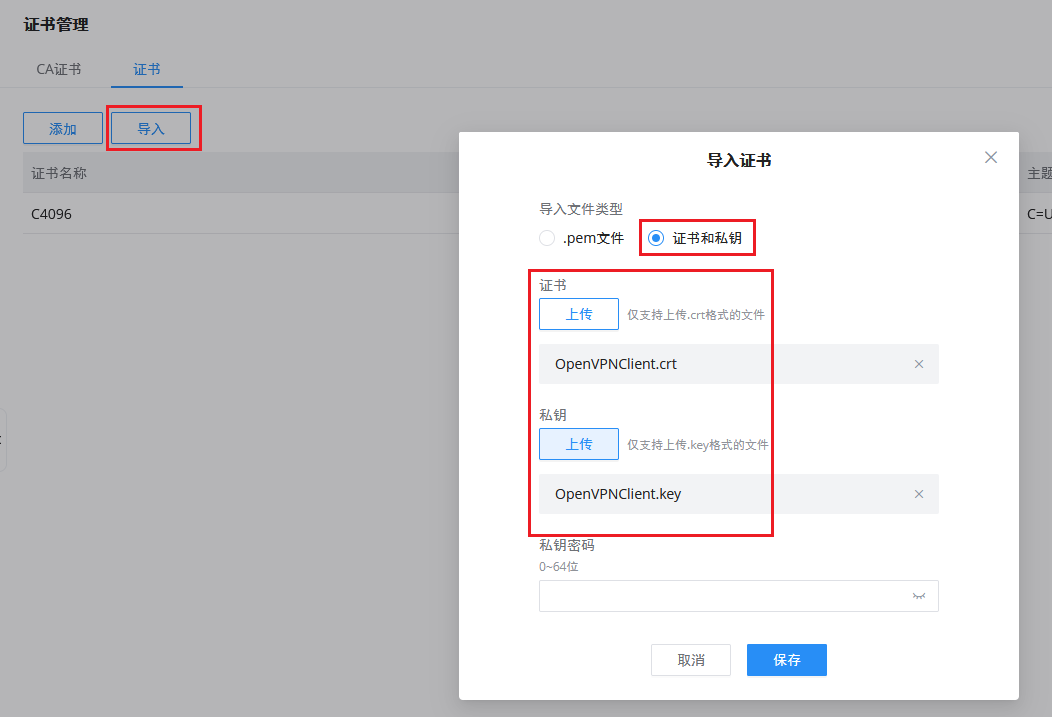


1. **客户端上传CA证书**





1. **客户端上传客户端证书**





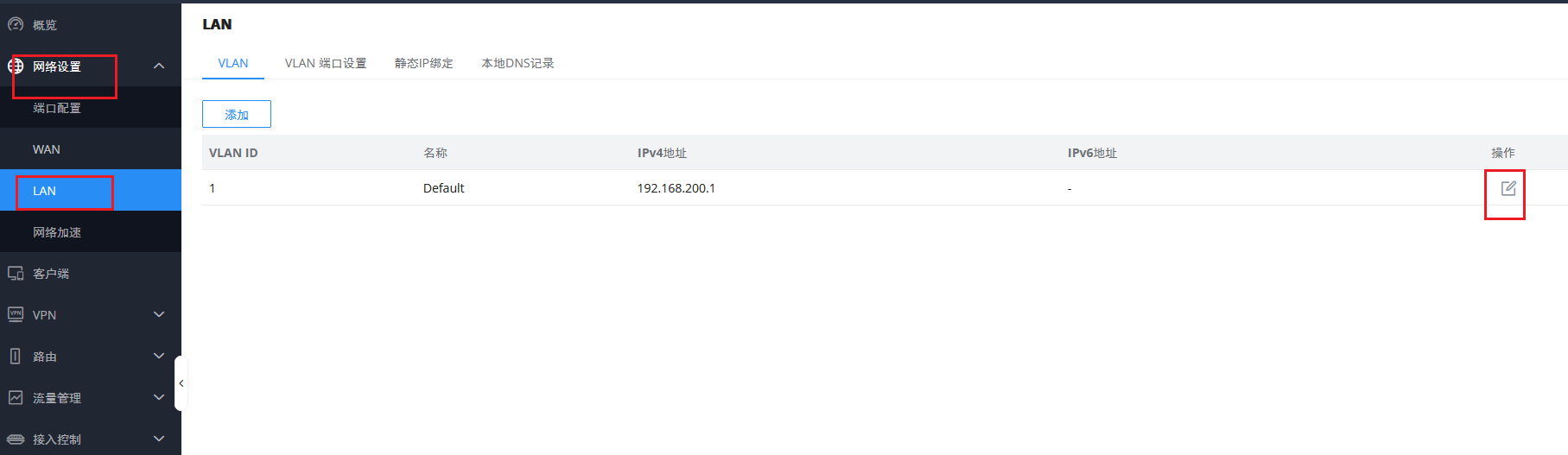
1. **添加OpenVPN客户端并进行连接**

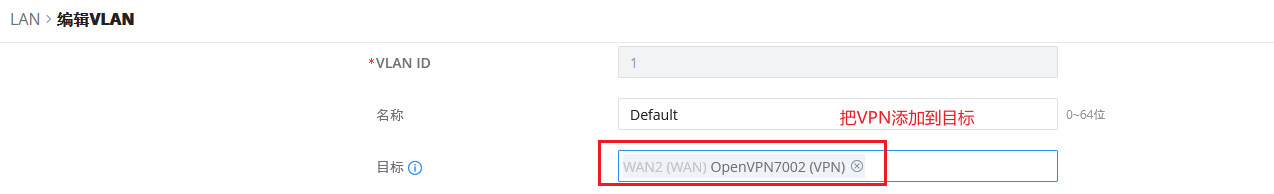






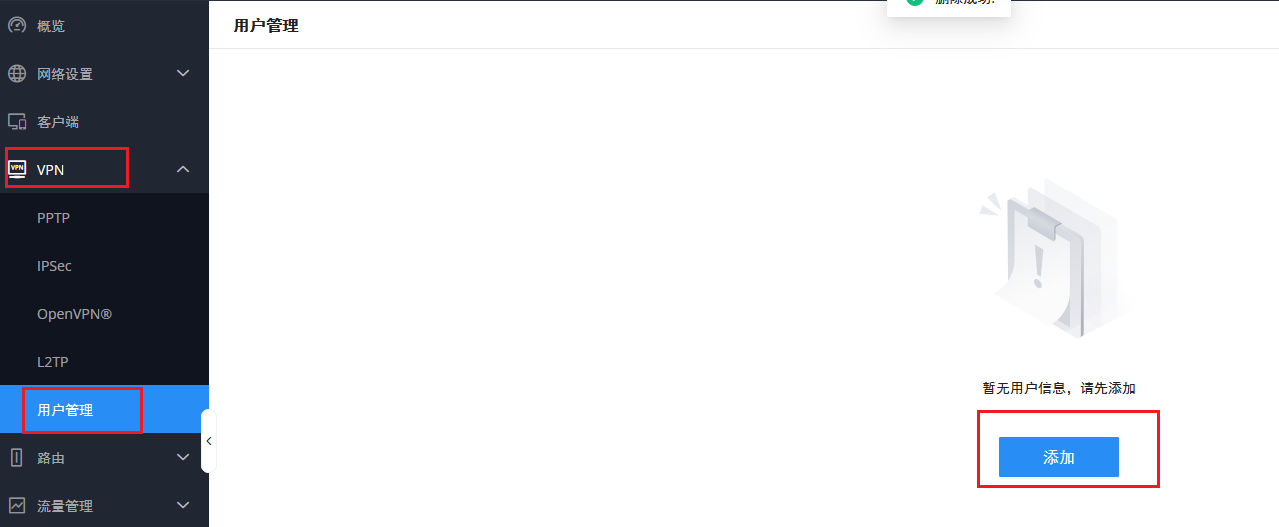
1. **配置LAN的目标**

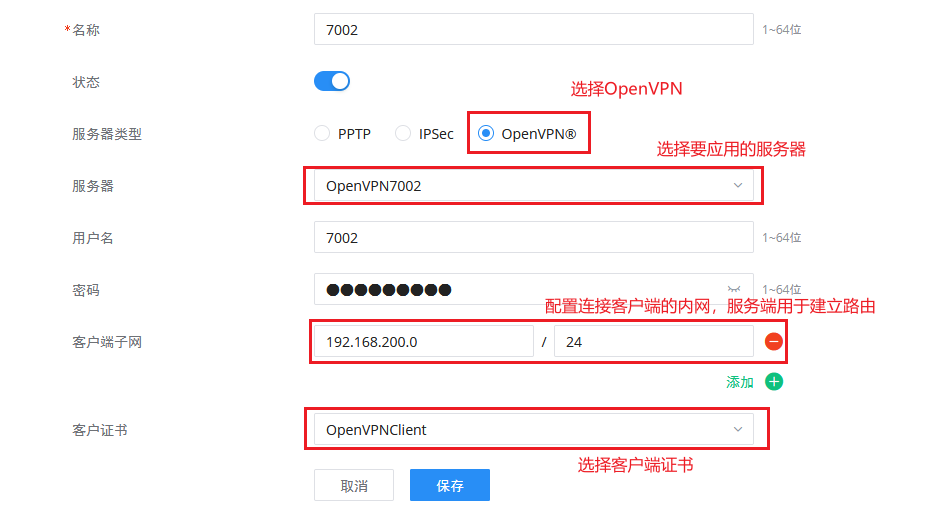




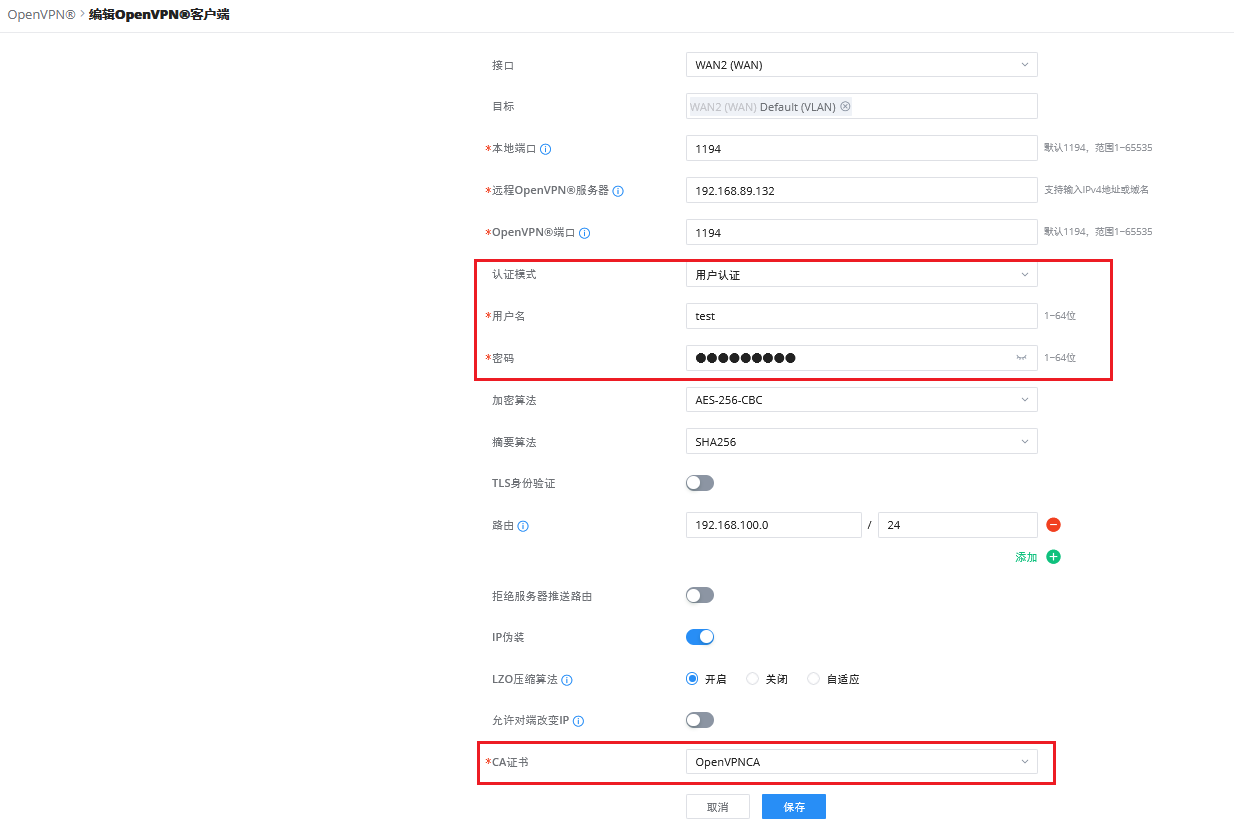
用户认证模式（这里只从客户端的角度说明不同点，其他的创建步骤与SSL认证模式相同）

**1、服务器端创建用户**



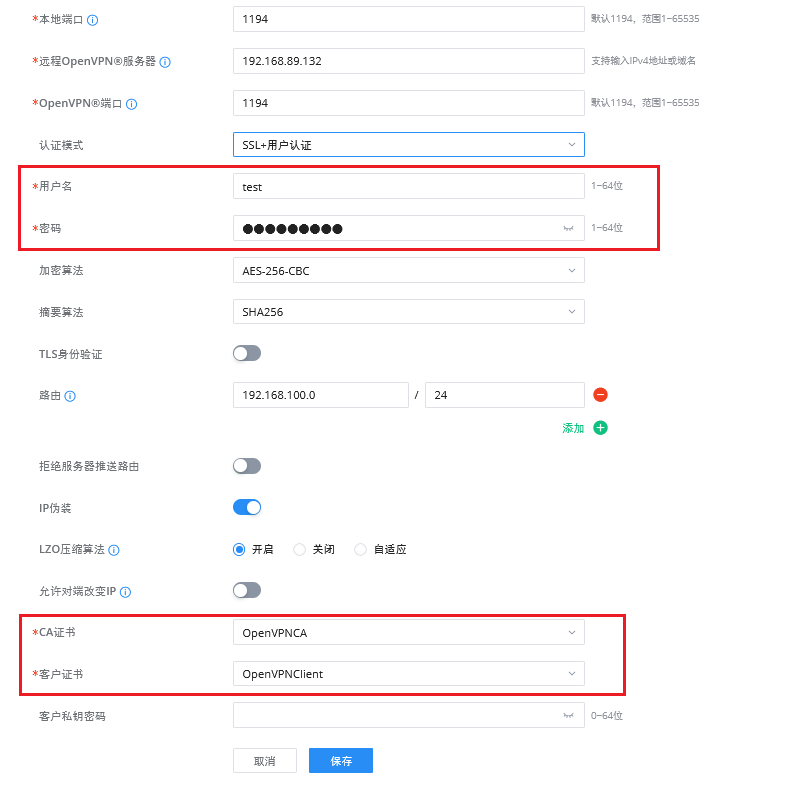


**2、客户端填写用户信息以及上传对应的CA证书**



SSL+用户认证模式

**1、用户信息、CA证书客户端证书都需要，客户端证书和服务器创建用户时使用的是同一个证书。**



PSK认证模式（预共享密钥在上述有提供）

