Instructions on TLS/SSL Certificates on Yealink Phones

Instructions on TLS/SSL Certificates on Yealink Phones

Update log		
Version	Time	Update by
1.0	2010-09-09	

1. Summary

Up to the date when this is being written (or updated), Yealink phones are supporting TLS 1.0 on T20/T22/T26/T28. TLS 1.0, which is defined by RFC 2246 (The TLS Protocol Version 1.0), can be considered as an upgrade to SSL 3.0. There are just tiny differences between TLS 1.0 and SSL 3.0, so in this article we treat them as the same. TLS provides endpoint authentication and communications confidentiality over the Internet using cryptography. For the technical details of TLS, you can refer to wiki: http://en.wikipedia.org/wiki/SSL_certificate#TLS_version_1.1 .
This article will just describe the usage of it on Yealink phones in a somewhat "superficial" way.

2. Encryption, decryption and the keys

Before we talk about certificates, we must know something basic about keys which are used to encrypt or decrypt certain traffic. The sender uses an encryption key to encrypt the messages it sends, while the receiver uses a right decryption key to decrypt the messages it receives. The encryption key and the decryption key match each other, meaning that if one piece of message is encrypted by a key, it is supposed that there's only one decryption key that can decrypt it. Generally, there are two kinds of encryption and the keys:

Symmetric encryption: For symmetric encryption, the encryption key and the corresponding decryption key can be told by each other. In most cases, the encryption key and the decryption key are the same one.

Asymmetric encryption: For asymmetric encryption, you cannot tell the decryption key from the encryption key and vice versa. There are Public Key and Private Key in this case and they are a match. The information encrypted by the Public Key can only be decrypted by the corresponding Private Key and vice versa. Usually, the receiver keeps its private key. The public key is known by the sender, so the sender sends the information encrypted by the known public key, and then the receiver uses the private key to decrypt it.

3. SSL connection flow

After the establishment of a SSL connection, the data transmits between each other is encrypted by a session key which is a symmetric key. The process of the connection is like:

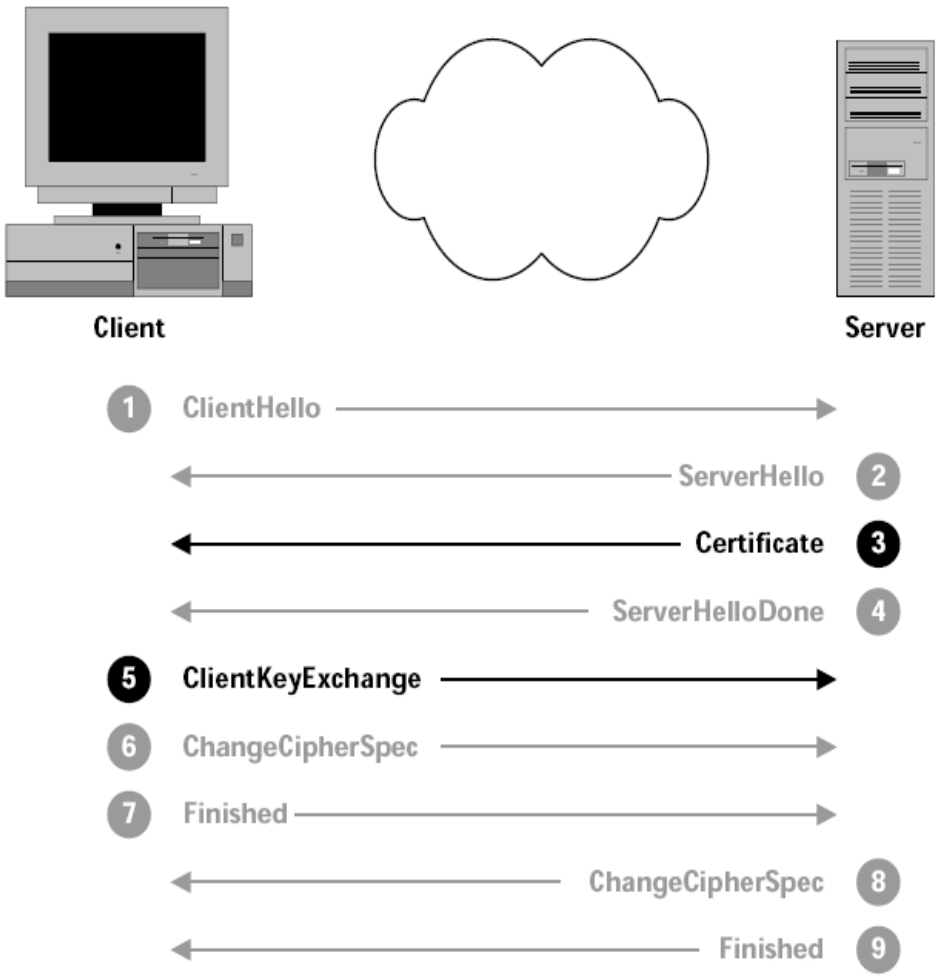

Figure 1

A. The client "say hello" to the HTTPS server to initiate a connection request. The request contains the encryption methods for negotiation with the server. (Step ① in the above flow)

B. The server responds "say hello" to confirm the encryption method and sends its certificate. The certificate contains a public key of the server. (Step ②③④ in the above flow)

C. If the client trusts the server, it will send back a session key which is encrypted by the public key from the server and tell the server that the coming up information will be encrypted by this session key. (Step ⑤⑥⑦ in the above flow)

D. The server confirms and the information it sends will also be encrypted by the session key. (Step ⑧⑨ in the above flow)

E. Now the data transmission between each other will be encrypted and both sides know the right key to decrypt since the session key is a symmetric key.

The trace T28-https.pcap can be referred if you want to know the detail.

4. The instructions on a certificate

It is not difficult to encrypt and decrypt the data by the sender and receiver. The tricky part is the negotiation of the encryption key (session key). The key cannot be sent freely because if someone else gets the information, your conversation will be at risk. It is done by using the public key and private key mechanism to encrypt the session key. You get the public

key from the sender, so then the problem comes to how you can trust the sender. The certificate is a solution to these problems.

The normal certificate, or accurately the so-called digital certificate (also known as a public key certificate), is actually an electronic document that mainly contains a public key and identity information of the certificate owner. And there will be other information such as the unique serial number, the issuer, the validity date of the certificate. By verifying the information in the certificate, it can be told that whether the sender of the certificate is trustable. If no, there won't be further transmission. If yes, the receiver will use the public key in the certificate to go further like from step ⑤⑥⑦ in Figure 1. The phone uses X.509 certificates whose common filename extensions can be .pem, .cer.

The phone can act as either a client or a server based on who initiates the connection request.

4.1 Phone acts as a client

When the phone initiates the SSL connection, we say it acts as a client. The case is like when the phone connects to the setting HTTPS URL for provisioning. Usually, the server should be verified by the client but not vice versa. That is, the client should verify that whether the server can be trusted. So when the phone requests the connection, the HTTPS server will send its certificate to the phone. If the phone doesn't enable the option "**Only Accept Trusted Certificates**", the connection will be done anyway. If the option is enabled, the phone will check the certificate based on the existing Trusted Certificates list. The related webpage of the phone is as below:

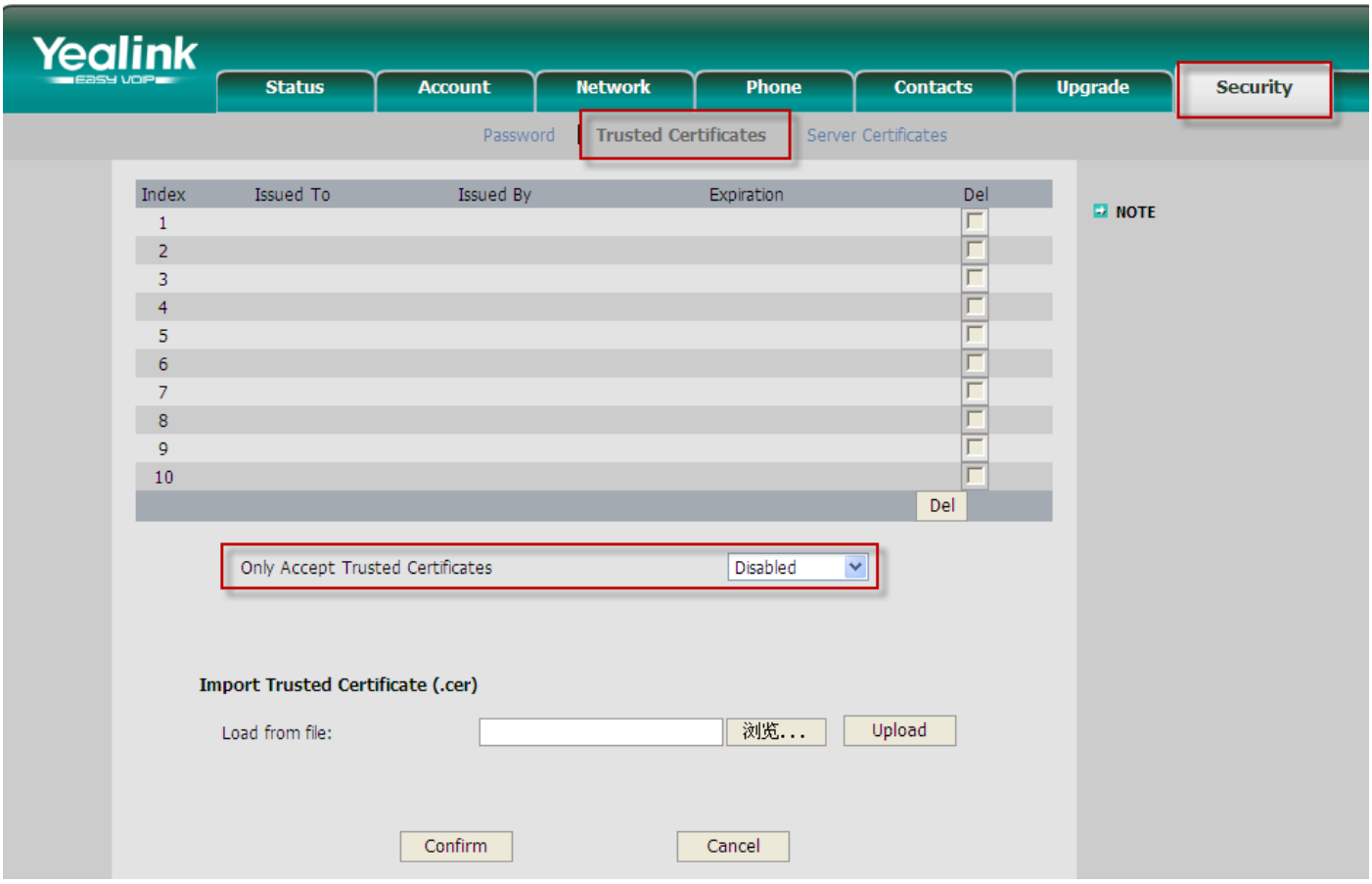

Figure 2

Up to the date when this document is being written, Yealink phones don't have any inbuilt Trusted Certificates. You can upload them by your own.

4.2 Phone acts as a server

When certain PC browser connects to the phone's webpage, we say the phone acts as a server. In this case, it is the phone's turn to send its certificate to the browser. The certificate belonging to the phone can be uploaded by your own too. Yealink phones don't have inbuilt server certificate either. The related webpage of the phone is as below:

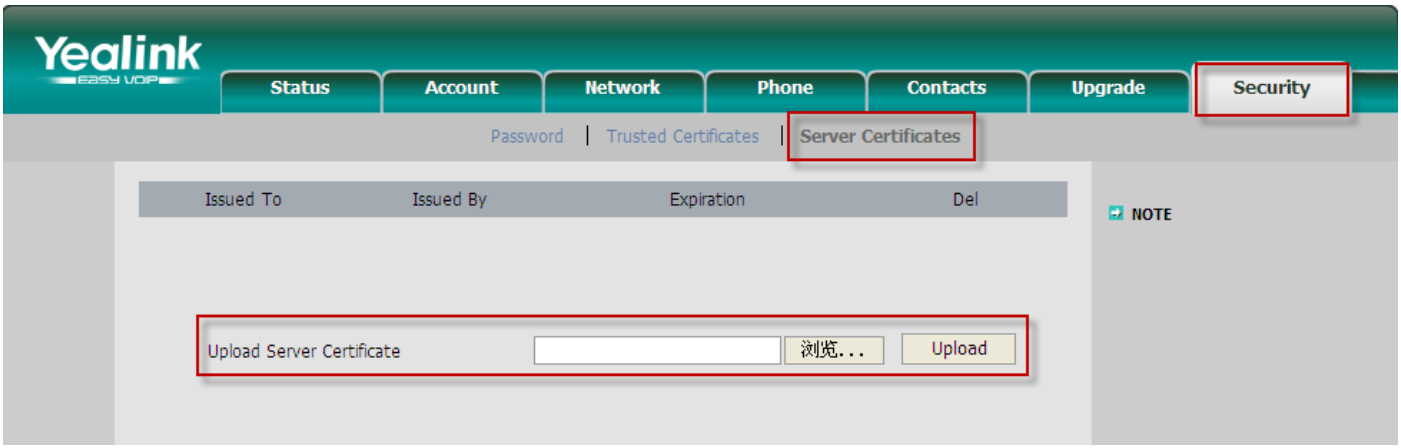

Figure 3

4.3 Phone's client certificate

We just said usually it is the server that is verified by the client. Actually there is possibility that the server also verifies the client, depending on the server's configuration. In this case, during the connection with a HTTPS server, the phone should send its client certificate to the server as well. For Yealink phones, the client certificate is the same one as the server certificate which you upload on webpage shown in Figure 3.

5. Frequently Asked Questions

1. Q: When I choose the TLS, which protocol (TCP or UDP) is used?
   A: TCP.

2. Q: Is it possible for the phone to upload certificates via auto provision?
   A: Not for now. But it is in our schedule of V70.

3. Q: There's an existing server certificate I uploaded before, what if I upload another one?
   A: The old one will be replaced by the newly uploaded one.