Yealink

Yealink Technical White Paper

802.1X Authentification

Jul. 2016

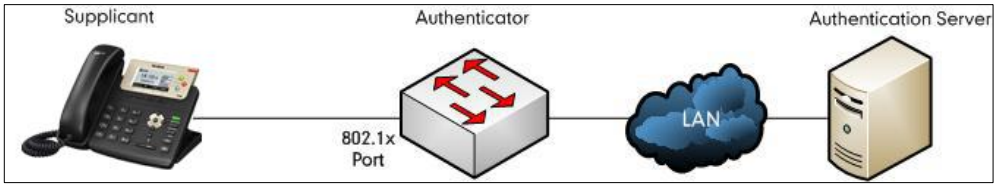
Table of Contents

About 802.1X

The IEEE 802.1X standard defines a Port-based Network Access Control (PNAC) and authentication protocol that restricts unauthorized clients from connecting to a LAN. The IEEE 802.1X defines the encapsulation of the Extensible Authentication Protocol (EAP) defined in RFC3748 which is known as "EAP over LAN" or EAPOL.

802.1X authentication involves three parties: a supplicant, an authenticator and an authentication server. The supplicant is a client device (such as an IP phone) that wishes to attach to the network. The authenticator is a network device, such as an Ethernet switch. And the authentication server is typically a host running software supporting the RADIUS and EAP protocols.

The authenticator acts like a security guard to a protected network. The supplicant is not allowed access through the authenticator to the protected side of the network until the supplicant's identity has been validated and authorized. An analogy to this is like providing a valid visa at the airport's arrival immigration before being allowed to enter the country. With 802.1X port-based authentication, the supplicant provides credentials, such as user name, password or digital certificate for the authenticator, and the authenticator forwards the credentials to the authentication server for verification. If the authentication server determines the credentials are valid, the supplicant is allowed to access resources located on the protected side of the network.



Yealink IP Phones Compatible with 802.1X

802.1X is the most widely accepted form of port-based network access control in use and is available on Yealink IP phones. Yealink IP phones support 802.1X authentication based on EAP-MD5, EAP-TLS, EAP-PEAP/MSCHAPv2, EAP-TTLS/EAP-MSCHAPv2, EAP-PEAP/GTC, EAP-TTLS/EAP-GTC and EAP-FAST protocols.

The table below lists the protocols supported by Yealink SIP IP phones with different versions.

Authentication Protocol	IP Phone Models	Firmware Version
EAP-MD5	All IP phones	All Versions
EAP-TLS	T28P, T26P, T22P, T20P	Firmware version 70 or later
	T21P, T19P, T46G, T42G, T41P, CP860	Firmware version 71 or later

Authentication Protocol	IP Phone Models	Firmware Version
	T48G	Firmware version 72 or later
	T49G, T40P, T29G, T27P/G, T23P/G, T21(P) E2, T19(P) E2, W56P	Firmware version 80 or later
EAP-PEAP/MSCHAPv2	T28P, T26P, T22P, T20P	Firmware version 70 or later
	T21P, T19P, T46G,T42G, T41P, CP860	Firmware version 71 or later
	T48G	Firmware version 72 or later
	T49G, T40P, T29G, T27P/G, T23P/G, T21(P) E2, T19(P) E2, W56P	Firmware version 80 or later
EAP-TTLS/EAP-MSCHAPv2	T28P, T26P, T22P, T20P, T21P, T19P, T46G, T42G, T41P, CP860	Firmware version 71 or later
	T48G	Firmware version 72 or later
	T49G, T40P, T29G, T27P/G, T23P/G, T21(P) E2, T19(P) E2, W56P	Firmware version 80 or later
EAP-PEAP/GTC	T28P, T26P, T22P, T20P, T48G, T46G, T42G, T41P	Firmware version 73 or later
	T49G, T40P, T29G, T27P/G, T23P/G, T21(P) E2, T19(P) E2, CP860, W56P	Firmware version 80 or later
EAP-TTLS/EAP-GTC	T28P, T26P, T22P, T20P, T48G, T46G, T42G, T41P	Firmware version 73 or later
	T49G, T40P, T29G, T27P/G, T23P/G, T21(P) E2, T19(P) E2, CP860, W56P	Firmware version 80 or later
EAP-FAST	T29G, T27P/G, T23P/G, T21(P) E2, T19(P) E2,	Firmware version 80 or later

Authentication Protocol	IP Phone Models	Firmware Version
	T49G, T48G, T46G, T42G, T41P, T40P, CP860, W56P	

Yealink IP phones support 802.1X as a supplicant, both Pass-thru Mode and Pass-thru Mode with Proxy Logoff. When the device connected to the phone disconnects from the PC port, the Yealink IP phone can provide additional security by sending an EAPOL Logoff message to the Ethernet switch. This functionality, also known as proxy logoff, prevents another device from using the port without first authenticating via 802.1X. The Pass-thru Mode is available on Yealink IP phones running specified firmware version. You can ask your system administrator or contact Yealink Field Application Engineer (FAE) for more information.

Configuring 802.1X Settings

The 802.1X authentication on Yealink IP phones is disabled by default. You can configure the 802.1X authentication in one of the following three ways:

- Configuring 802.1X using configuration files
- Configuring 802.1X via web user interface
- Configuring 802.1X via phone user interface

For detailed descriptions of the authentication parameters in configuration files, you can refer to Configuring 802.1X using configuration files on page 5. When setting up a large number of IP phones, Yealink recommends using configuration files. If you are provisioning a few phones, you can use the web user interface or phone user interface to configure 802.1X feature.

If the EAP-TLS, EAP-PEAP/MSCHAPv2, EAP-TTLS/EAP-MSCHAPv2, EAP-PEAP/GTC, EAP-TTLS/EAP-GTC or EAP-FAST protocol is preferred in your 802.1X environment, make sure that the firmware running on your new phone supports the protocol.

The followings provide system administrator with the procedures to successfully configure Yealink IP phones in a secure 802.1X environment, and take configurations of a SIP-T23G IP phone running firmware version 80 as examples.

Configuring 802.1X using configuration files

1. Add/Edit 802.1X authentication parameters in configuration files.

   The following table shows the information of parameters:

Parameters	Permitted Values	Default
network.802_1x.mode	0, 1, 2, 3, 4, 5, 6 or 7	0

Parameters	Permitted Values	Default
Description: Configures the 802.1x authentication method. **0**-Disabled **1**-EAP-MD5 **2**-EAP-TLS **3**-EAP-PEAP/MSCHAPv2 **4**-EAP-TTLS/EAP-MSCHAPv2 **5**-EAP-PEAP/GTC **6**-EAP-TTLS/EAP-GTC **7**-EAP-FAST **Web User Interface:** Network->Advanced->802.1x->802.1x Mode **Phone User Interface:** Menu->Settings->Advanced Settings->Network->802.1x Settings->802.1x Mode		
network.802_1x.identity	**String within 32 characters**	**Blank**
Description: Configures the user name for 802.1x authentication. **Note**: It works only if the value of the parameter "network.802_1x.mode" is set to 1, 2, 3, 4, 5, 6 or 7. **Web User Interface:** Network->Advanced->802.1x->MD5 Password **Phone User Interface:** Menu->Settings->Advanced Settings->Network->802.1x Settings->MD5 Password		
network.802_1x.md5_password	**String within 32 characters**	**Blank**
Description: Configures the password for 802.1x authentication. **Note**: It works only if the value of the parameter "network.802_1x.mode" is set to 1, 3, 4, 5, 6 or 7. **Web User Interface:** Network->Advanced->802.1x->MD5 Password **Phone User Interface:** Menu->Settings->Advanced Settings->Network->802.1x Settings->MD5		

Parameters	Permitted Values	Default
Password		
network.802_1x.root_cert_url	**URL within 511 characters**	**Blank**
Description: Configures the access URL of the CA certificate. **Note**: It works only if the value of the parameter "network.802_1x.mode" is set to 2, 3, 4, 5, 6 or 7. The format of the certificate must be *.pem, *.crt, *.cer or *.der. **Web User Interface:** Network->Advanced->802.1x->CA Certificates **Phone User Interface:** None		
network.802_1x.client_cert_url	**URL within 511 characters**	**Blank**
Description: Configures the access URL of the device certificate. **Note**: It works only if the value of the parameter "network.802_1x.mode" is set to 2 (EAP-TLS). The format of the certificate must be *.pem. **Web User Interface:** Network->Advanced->802.1x->Device Certificates **Phone User Interface:** None		

The following shows an example of the EAP-TLS protocol for 802.1X authentication in configuration files:

```
network.802_1x.mode = 2
network.802_1x.identity = yealink
network.802_1x.root_cert_url = http://192.168.1.8:8080/ca.crt
network.802_1x.client_cert_url = http:// 192.168.1.8:8080/client.pem
```

2. Upload configuration files, CA certificate and client certificate to the root directory of the configuration server.

Applying the Configuration Files to your Phone

Once you have edited a configuration file (e.g., y0000000000xx.cfg) using the parameters introduced above, you need to do the following to apply the files to your phone:

1. Connect your phone to a network that is not 802.1X-enabled.
2. Perform the auto provisioning process to apply the configuration files to the phone.

Then the IP phone will reboot to make the settings effective.

For more information on auto provisioning, refer to *Yealink_SIP-T2_Series_T19(P)_E2_T4_Series_CP860_W56P_IP_Phones_Auto_Provisioning_Guide*.

3. Connect the phone to the 802.1X-enabled network and reboot the phone.

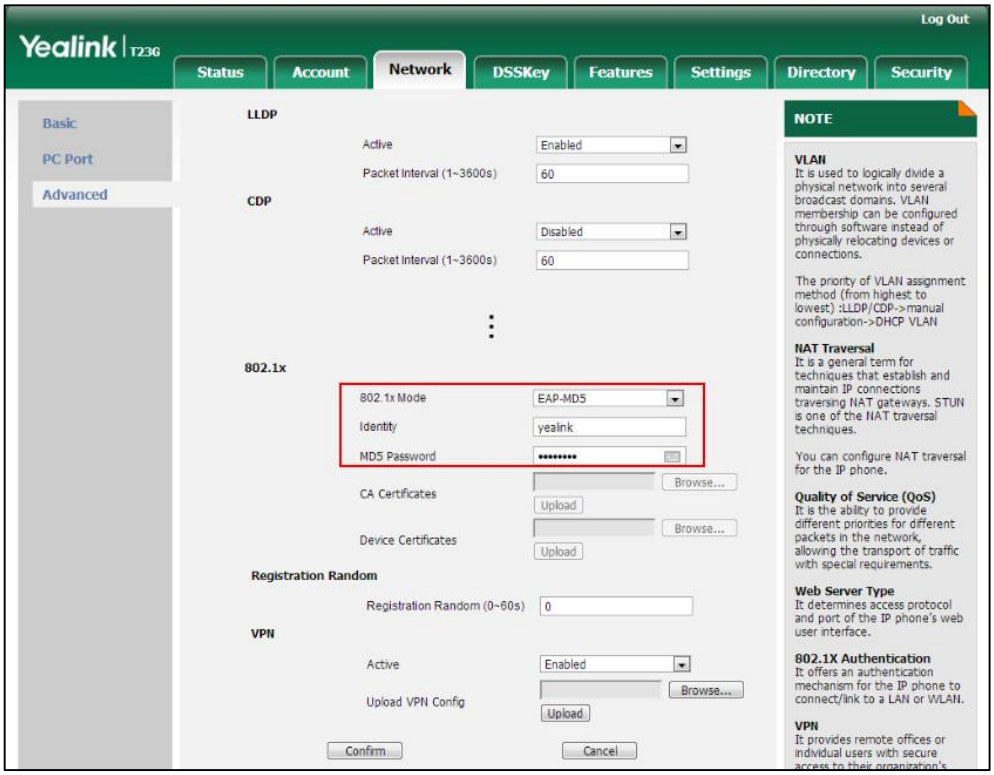
   You can make a phone call to verify whether the phone is authenticated.

Configuring 802.1X via web user interface

1. Connect your phone to a network that is not 802.1X-enabled.
2. Login to the web user interface of the phone.
3. Click on **Network->Advanced**.
4. In the **802.1x** block, select the desired protocol from the pull-down list of **802.1x Mode**.

   a) If you select **EAP-MD5**:

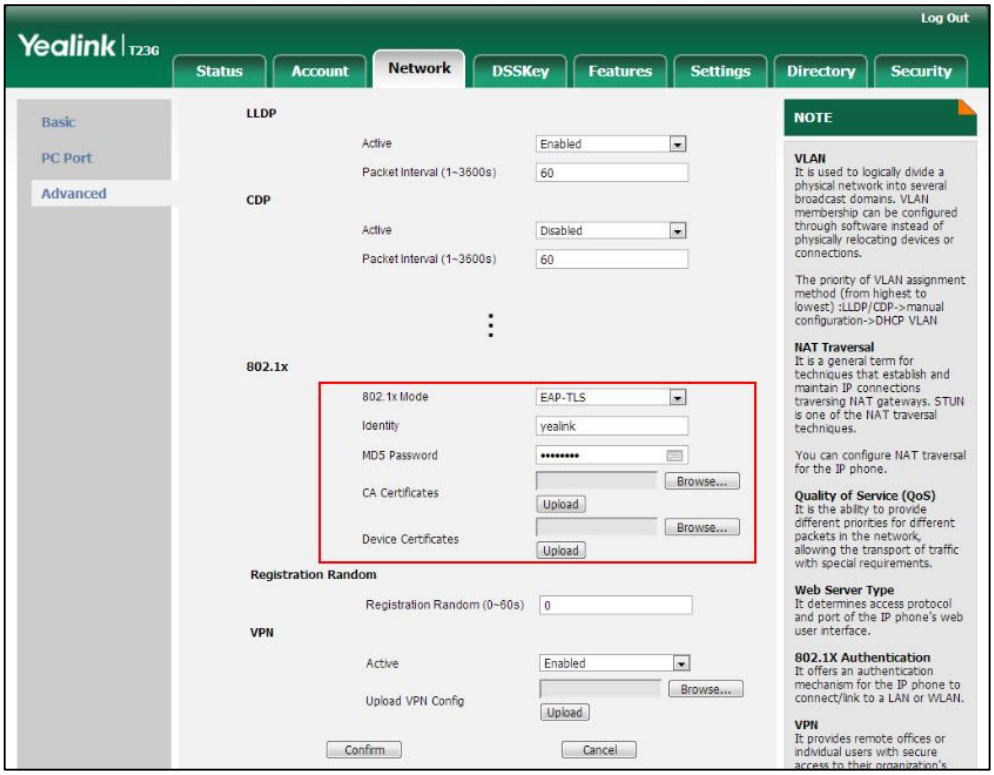
      1) Enter the user name for authentication in the **Identity** field.
      2) Enter the password for authentication in the **MD5 Password** field.



   b) If you select **EAP-TLS**:

      1) Enter the user name for authentication in the **Identity** field.
      2) Leave the **MD5 Password** field blank.
      3) In the **CA Certificates** field, click **Browse** to select the desired CA certificate (*.pem, *.crt, *.cer or *.der) from your local system.

4) In the **Device Certificates** field, click **Browse** to select the desired client (*.pem or *.cer) certificate from your local system.



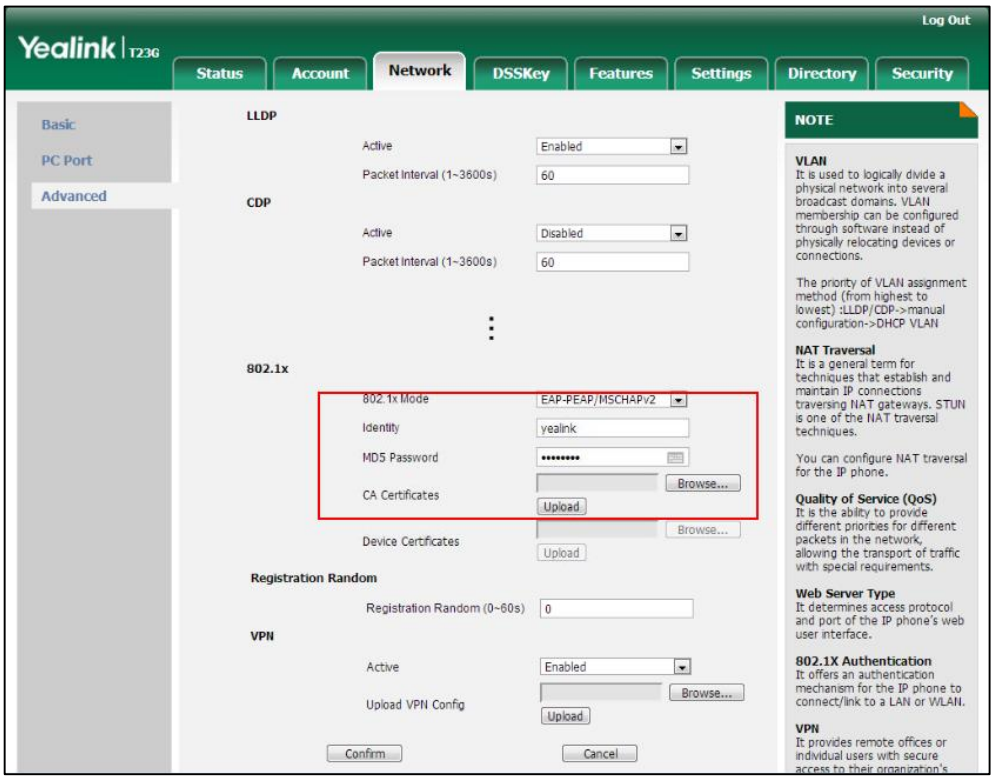
5) Click **Upload** to upload the certificates.

c) If you select **EAP-PEAP/MSCHAPv2**:

1) Enter the user name for authentication in the **Identity** field.

2) Enter the password for authentication in the **MD5 Password** field.

3) In the **CA Certificates** field, click **Browse** to select the desired CA certificate (*.pem, *.crt, *.cer or *.der) from your local system.



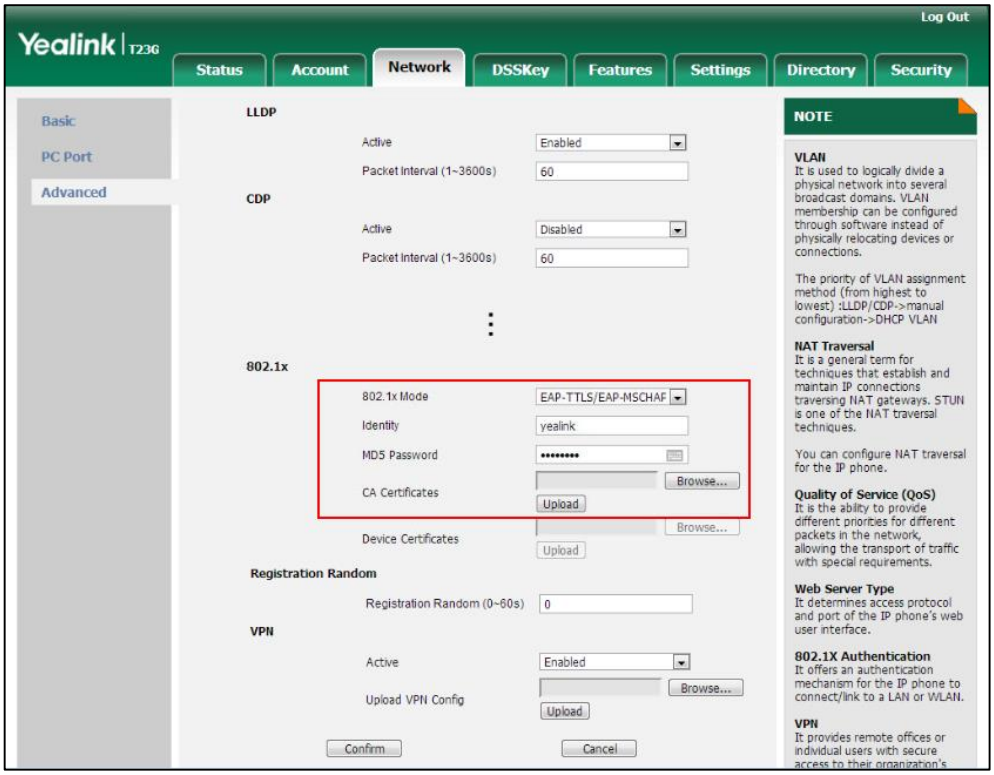
4) Click **Upload** to upload the certificate.

d) If you select **EAP-TTLS/EAP-MSCHAPv2**:

1) Enter the user name for authentication in the **Identity** field.

2) Enter the password for authentication in the **MD5 Password** field.

3) In the **CA Certificates** field, click **Browse** to select the desired CA certificate (*.pem, *.crt, *.cer or *.der) from your local system.



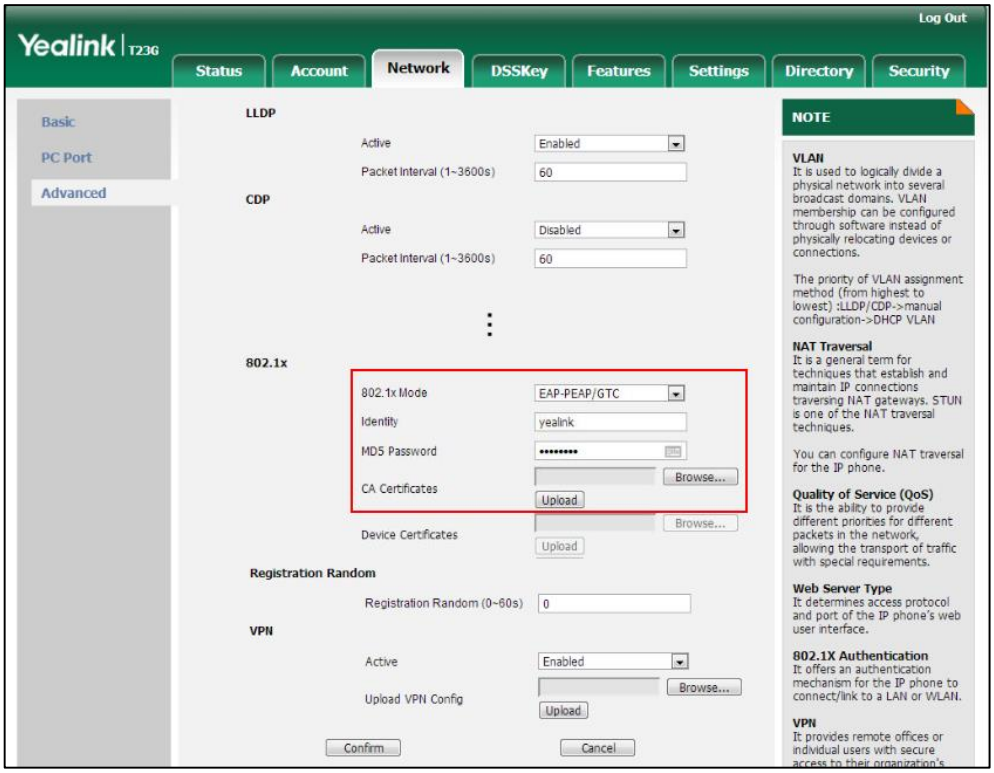
4) Click **Upload** to upload the certificate.

e) If you select **EAP-PEAP/GTC**:

1) Enter the user name for authentication in the **Identity** field.

2) Enter the password for authentication in the **MD5 Password** field.

3) In the **CA Certificates** field, click **Browse** to select the desired CA certificate (*.pem, *.crt, *.cer or *.der) from your local system.



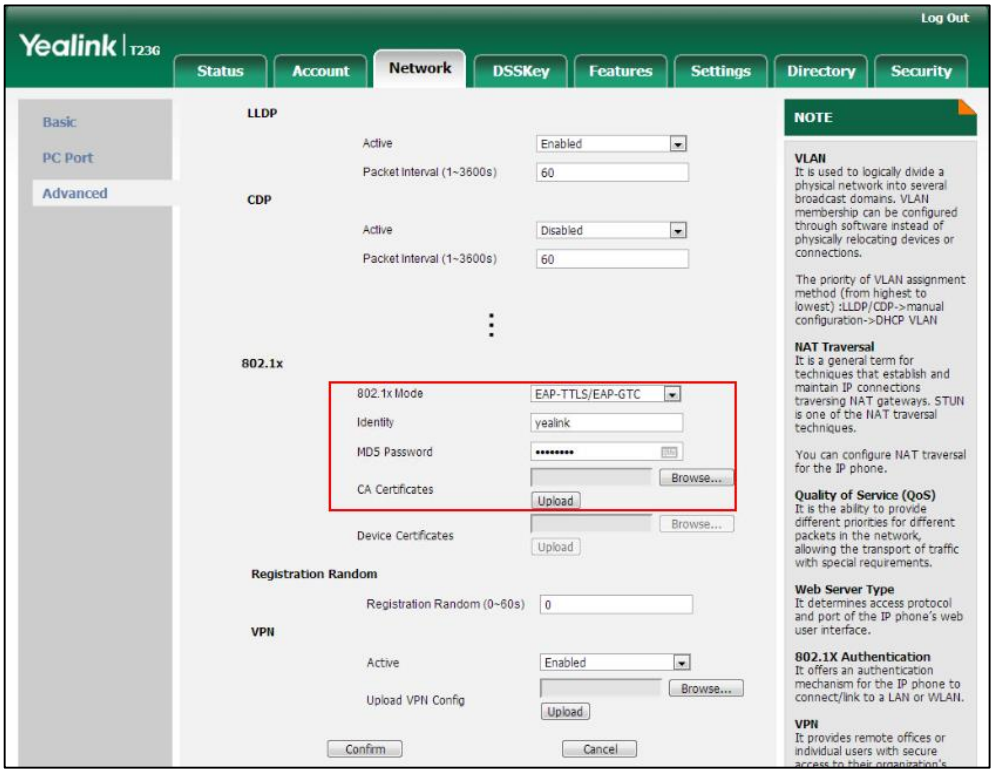
4) Click **Upload** to upload the certificate.

f) If you select **EAP-TTLS/EAP-GTC**:

1) Enter the user name for authentication in the **Identity** field.

2) Enter the password for authentication in the **MD5 Password** field.

3) In the **CA Certificates** field, click **Browse** to select the desired CA certificate (*.pem, *.crt, *.cer or *.der) from your local system.



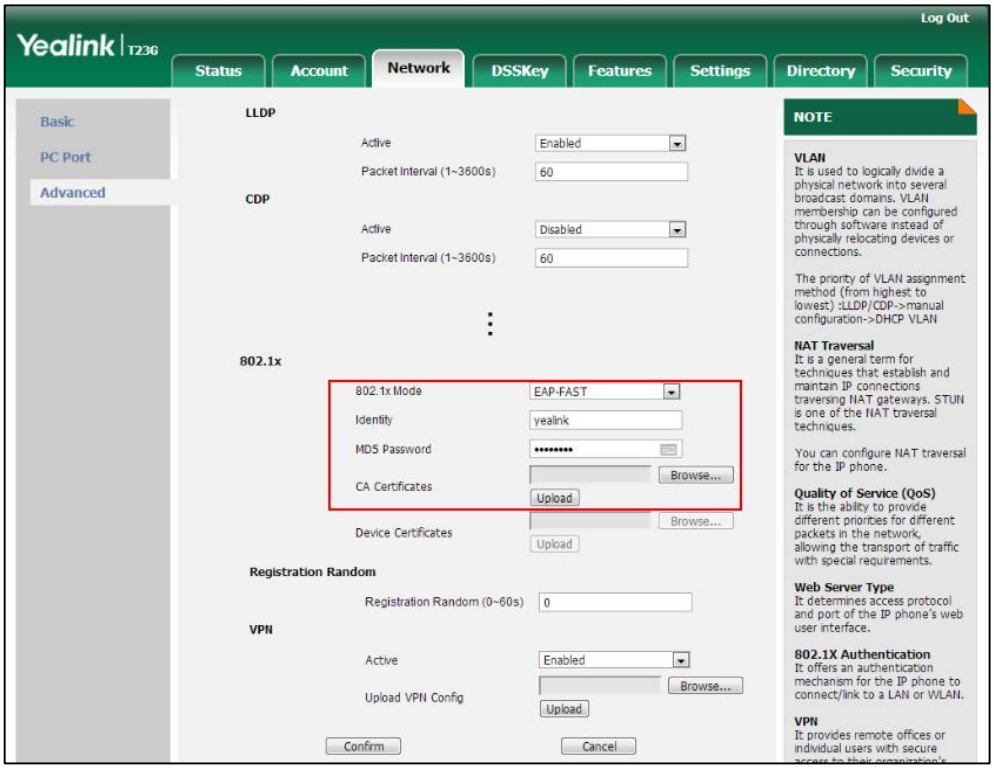
4) Click **Upload** to upload the certificate.

g) If you select **EAP-FAST**:

1) Enter the user name for authentication in the **Identity** field.

2) Enter the password for authentication in the **MD5 Password** field.

3) In the **CA Certificates** field, click **Browse** to select the desired CA certificate (*.pem, *.crt, *.cer or *.der) from your local system.



4) Click **Upload** to upload the certificate.

5. Click **Confirm** to accept the change.

   A dialog box pops up to prompt that settings will take effect after a reboot.

6. Click **OK** to reboot the phone.

7. Connect the phone to the 802.1X-enabled network after reboot.

Note If the Pass-thru mode is available on your new phone, you can select the Pass-thru mode from the pull-down list of **DOT1XSTAT Options** via web user interface.

Configuring 802.1X via phone user interface

If you select EAP-PEAP/MSCHAPv2, EAP-TTLS/EAP-MSCHAPv2, EAP-PEAP/GTC, EAP-TTLS/EAP-GTC or EAP-FAST mode, you should upload CA certificate in advance using configuration files or via web user interface.

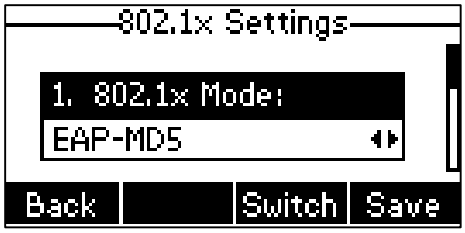
If you select EAP-TLS mode, you should upload CA certificate and device certificate in advance using configuration files or via web user interface.

To configure 802.1x via phone user interface:

1. Press **Menu->Settings->Advanced Settings** (default password: admin) **->Network->802.1x Settings**.

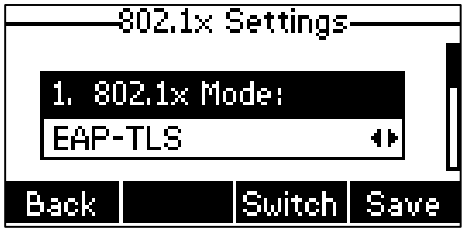
2. Press ◄ or ►, or the **Switch** soft key to select the desired value from the **802.1x Mode** field.

a) If you select **EAP-MD5**:



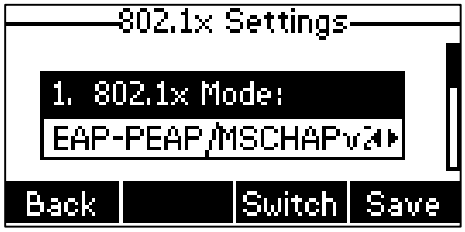
1) Enter the user name for authentication in the **Identity** field.
2) Enter the password for authentication in the **MD5 Password** field.

b) If you select **EAP-TLS**:



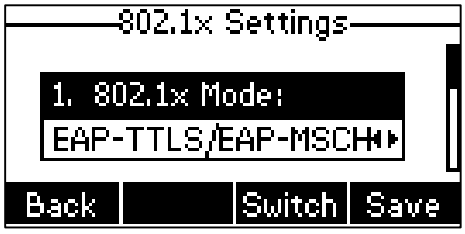
1) Enter the user name for authentication in the **Identity** field.
2) Leave the **MD5 Password** field blank.

c) If you select **EAP-PEAP/MSCHAPv2**:



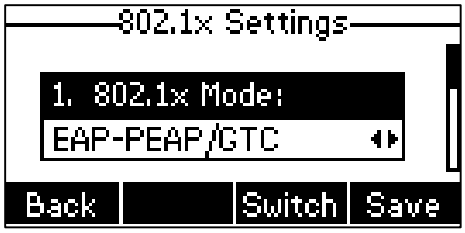
1) Enter the user name for authentication in the **Identity** field.
2) Enter the password for authentication in the **MD5 Password** field.

d) If you select **EAP-TTLS/EAP-MSCHAPv2**:



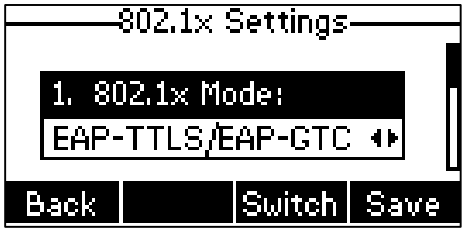
1) Enter the user name for authentication in the **Identity** field.
2) Enter the password for authentication in the **MD5 Password** field.

e) If you select **EAP-PEAP/GTC**:



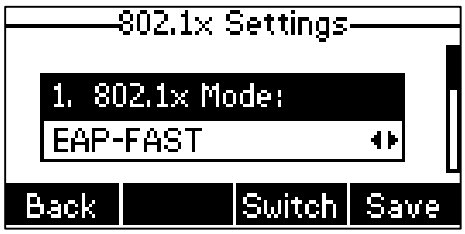
1) Enter the user name for authentication in the **Identity** field.
2) Enter the password for authentication in the **MD5 Password** field.

f) If you select **EAP-TTLS/EAP-GTC**:



1) Enter the user name for authentication in the **Identity** field.
2) Enter the password for authentication in the **MD5 Password** field.

g) If you select **EAP-FAST**:



1) Enter the user name for authentication in the **Identity** field.
2) Enter the password for authentication in the **MD5 Password** field.

3. Click **Save** to accept the change.

   The IP phone reboots automatically to make the settings effective after a period of time.

802.1X Authentication Process

Reboot the phone to activate the 802.1X authentication on the phone. The 802.1X authentication process is divided into two basic stages:

Pre-authentication

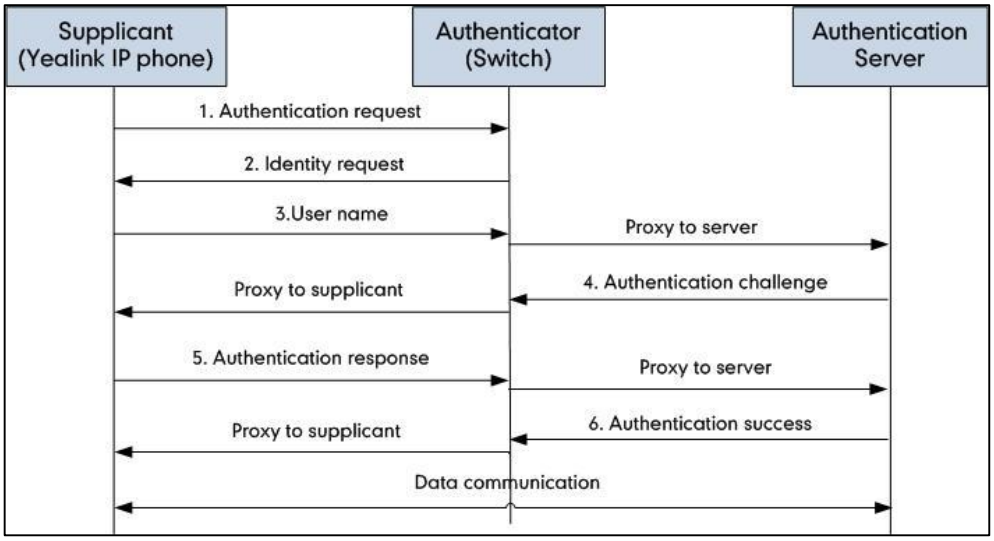
The 802.1X pre-authentication process begins with the IP phone that contains a supplicant service used for negotiation and authentication. When the IP phone connects to an unauthorized port, the authenticator blocks the IP phone from connecting to the

network. Using one of the authentication protocols, the authenticator establishes a security negotiation with the IP phone and creates an 802.1X session. The IP phone provides its authentication information for the authenticator, and then the authenticator forwards the information to the authentication server.

Authentication

After the authentication server authenticates the IP phone, the authentication server initiates the authentication stage of the process. During this phase, the authenticator facilitates an exchange of keys between the IP phone and the authentication server. After these keys are established, the authenticator grants the IP phone access to the protected network on an authorized port.

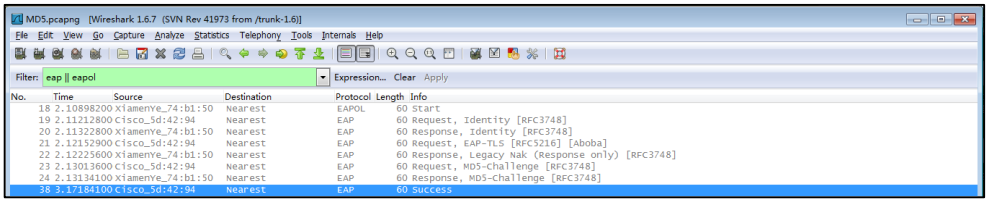
The following figure summarizes an implementation of the 802.1X authentication process using a RADIUS server as the authentication server:



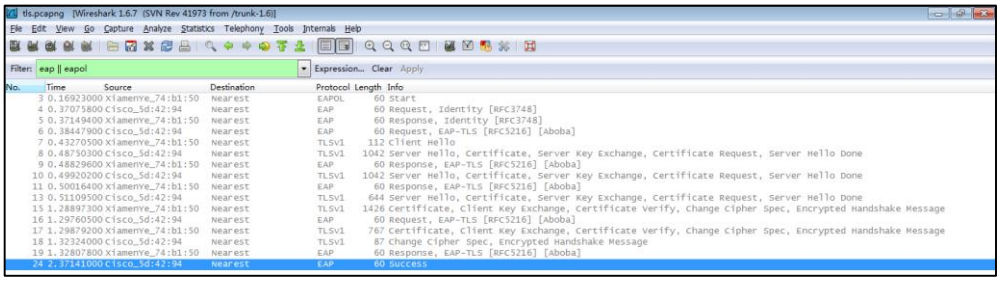
For more details about the 802.1X authentication process using EAP-MD5, EAP-TLS, EAP-PEAP/MSCHAPv2, EAP-TTLS/EAP-MSCHAPv2, EAP-PEAP/GTC, EAP-TTLS/EAP-GTC and EAP-FAST protocols, refer to Appendix B: 802.1X Authentication Process on page 21.

If you are interested in the packets exchanged during the authentication process, we recommend you to use the Wireshark tool. Refer to http://wiki.wireshark.org for more information about the Wireshark tool.

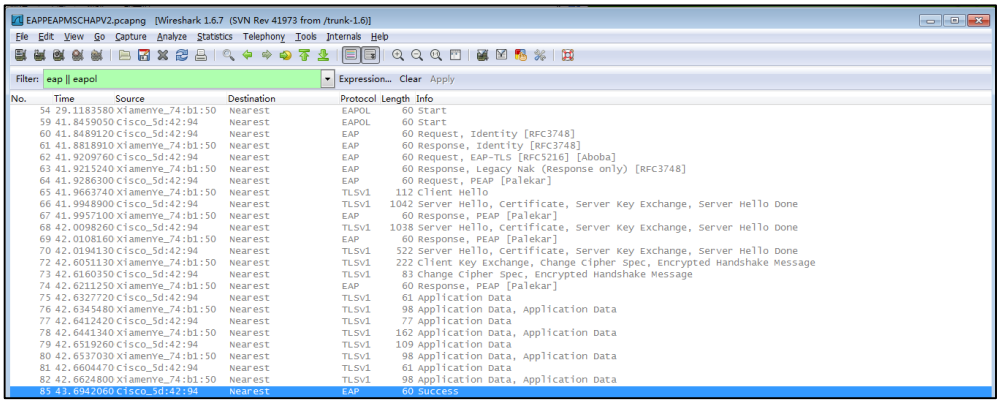
The following screenshot of the Wireshark shows a sample of a successful authentication process using the EAP-MD5 protocol:



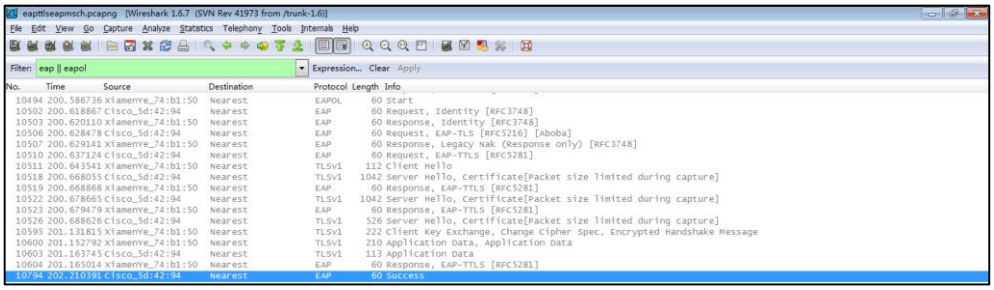
The following screenshot of the Wireshark shows a sample of a successful authentication process using the EAP-TLS protocol:



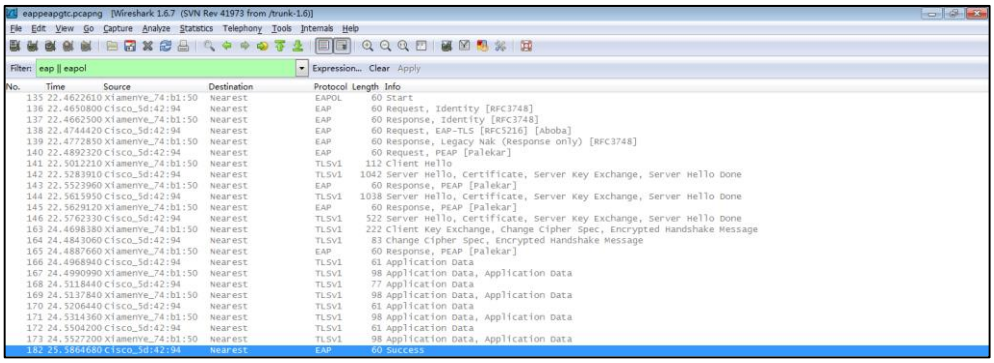
The following screenshot of the Wireshark shows a sample of a successful authentication process using the EAP-PEAP/MSCHAPv2 protocol:



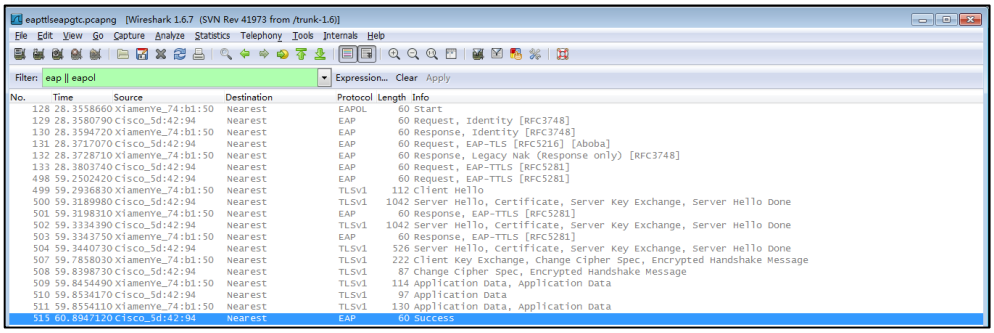
The following screenshot of the Wireshark shows a sample of a successful authentication process using the EAP-TTLS/EAP-MSCHAPv2 protocol:



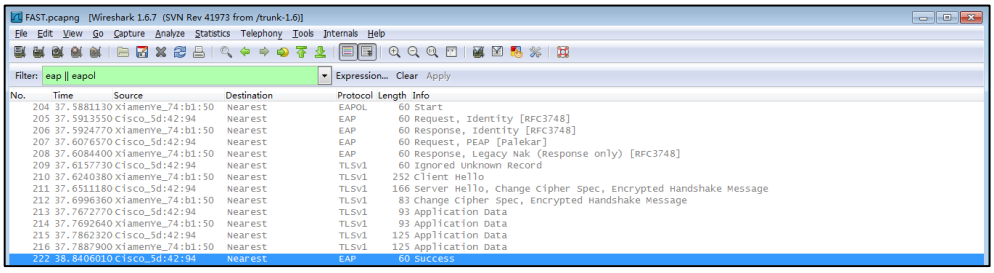
The following screenshot of the Wireshark shows a sample of a successful authentication process using the EAP-PEAP/GTC protocol:



The following screenshot of the Wireshark shows a sample of a successful authentication process using the EAP-TTLS/EAP-GTC protocol:



The following screenshot of the Wireshark shows a sample of a successful authentication process using the EAP-FAST protocol:



Troubleshooting

Why doesn't the IP phone pass 802.1X authentication?

Do the following in sequence:

- Ensure that the 802.1X authentication environment is operational.
  - a) Connect another device (e.g., a computer) to the switch port.
  - b) Check if the device is authenticated successfully, and an IP address is assigned to it. If the device fails the authentication, check the configurations on the switch and authentication server.
- Ensure that the user name and password configured on the phone are correct. If EAP-TLS, EAP-PEAP/MSCHAPv2, EAP-TTLS/EAP-MSCHAPv2, EAP-PEAP/GTC, EAP-TTLS/EAP-GTC and EAP-FAST protocols are used, ensure that the certificate uploaded to the phone is valid.
  - a) Double click the certificate to check the validity time.
  - b) Check if the time and date on the phone is within the validity time of the uploaded certificate. If not, re-generate a certificate and upload it the phone.
- Ensure that the failure is not caused by network settings.
  - a) Disable LLDP feature and manually configure a VLAN ID for the Internet port of

the phone to check if the authentication is successful. If the phone is authenticated successfully, contact your network administrator to troubleshoot the LLDP-related problem.

b) Disable VLAN feature on the phone to check if the authentication passes successfully. If the phone is authenticated successfully, capture the packet and feed back to your network administrator.

- Contact Yealink FAE for support when the above steps cannot solve your problem.
  a) Capture the packet and export configurations of the phone, switch and authentication server.
  b) Provide the related information to Yealink FAE.

Appendix A: Glossary

IEEE (Institute of Electrical and Electronics Engineers) –A professional association headquartered in New York City that is dedicated to advancing technological innovation and excellence.

802.1X –A port-based network access control, meaning it only provides an authentication mechanism for devices wishing to attach to a LAN.

EAP (Extensible Authentication Protocol) –An authentication framework which supports multiple authentication methods.

TLS (Transport Layer Security) –Provides for mutual authentication, integrity-protected cipher suite negotiation between two endpoints.

MD5 (Message-Digest Algorithm) –Only provides authentication of the EAP peer for the EAP server but not mutual authentication.

PEAP (Protected Extensible Authentication Protocol) –A protocol that encapsulates the EAP within an encrypted and authenticated TLS tunnel.

MSCHAPv2 (Microsoft Challenge Handshake Authentication Protocol version 2) – Provides for mutual authentication, but does not require a supplicant-side certificate.

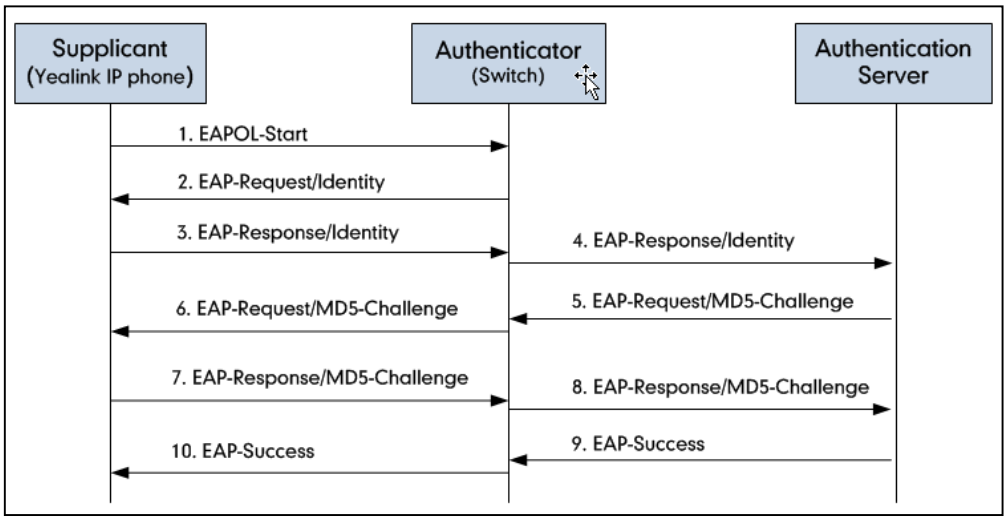
TTLS (Tunneled Transport Layer Security) –Extends TLS to improve some weak points, but it does not require a supplicant-side certificate.

EAPOL (Extensible Authentication Protocol over Local Area Network) –A delivery mechanism and doesn't provide the actual authentication mechanisms.

Appendix B: 802.1X Authentication Process

A Successful Authentication Using EAP-MD5 Protocol

The following figure illustrates the scenario of a successful 802.1X authentication process using the EAP-MD5 protocol.



1. The supplicant sends an "EAPOL-Start" packet to the authenticator.
2. The authenticator responds with an "EAP-Request/Identity" packet to the supplicant.
3. The supplicant responds with an "EAP-Response/Identity" packet to the authenticator.
4. The authenticator strips the Ethernet header and encapsulates the remaining EAP frame in the RADIUS format, and then sends it to the authentication server.
5. The authentication server recognizes the packet as an EAP-MD5 type and sends back a Challenge message to the authenticator.
6. The authenticator strips the authentication server's frame header, encapsulates the remaining EAP frame into the EAPOL format, and sends it to the supplicant.
7. The supplicant responds to the Challenge message.
8. The authenticator passes the response to the authentication server.
9. The authentication server validates the authentication information and sends an authentication success message.
10. The authenticator passes the successful message to the supplicant.

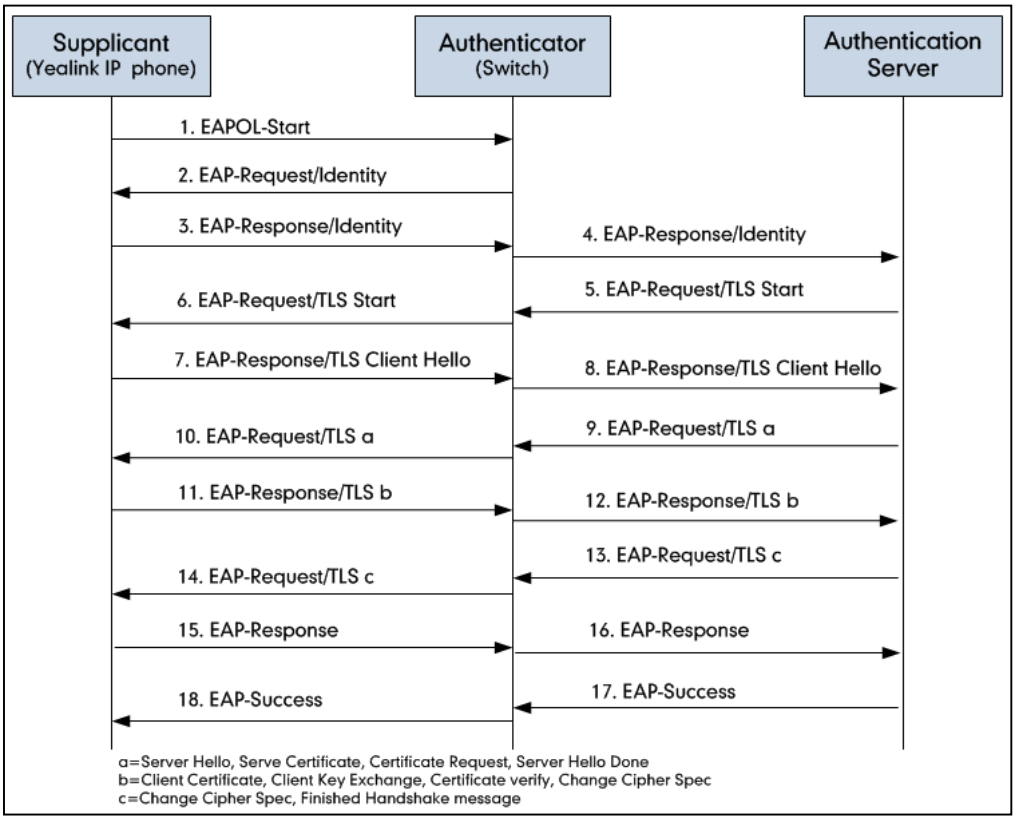
    After the supplicant is authenticated successfully, the authenticator provides network access permissions. If the supplicant does not provide proper identification, the authentication server responds with a rejection message. The authenticator passes the message onto the supplicant and blocks access to the LAN.

    If the supplicant is disabled or reset after successful authentication, the supplicant sends an EAPOL-Logoff message, which prompts the authenticator to block access

to the LAN.

A Successful Authentication Using EAP-TLS Protocol

The following figure illustrates the scenario of a successful 802.1X authentication process using the EAP-TLS protocol.



1. The supplicant sends an “EAPOL-Start” packet to the authenticator.
2. The authenticator responds with an “EAP-Request/Identity” packet to the supplicant.
3. The supplicant responds with an "EAP-Response/Identity" packet to the authenticator.
4. The authenticator strips the Ethernet header and encapsulates the remaining EAP frame in the RADIUS format, and then sends it to the authentication server.
5. The authentication server recognizes the packet as an EAP-TLS type and sends an “EAP-Request” packet with a TLS start message to the authenticator.
6. The authenticator strips the authentication server’s frame header, encapsulates the remaining EAP frame in the EAPOL format, and then sends it to the supplicant.
7. The supplicant responds with an “EAP-Response” packet containing a TLS client hello handshake message to the authenticator. The client hello message includes the TLS version supported by the supplicant, a session ID, a random number and a set of cipher suites.
8. The authenticator passes the response to the authentication server.

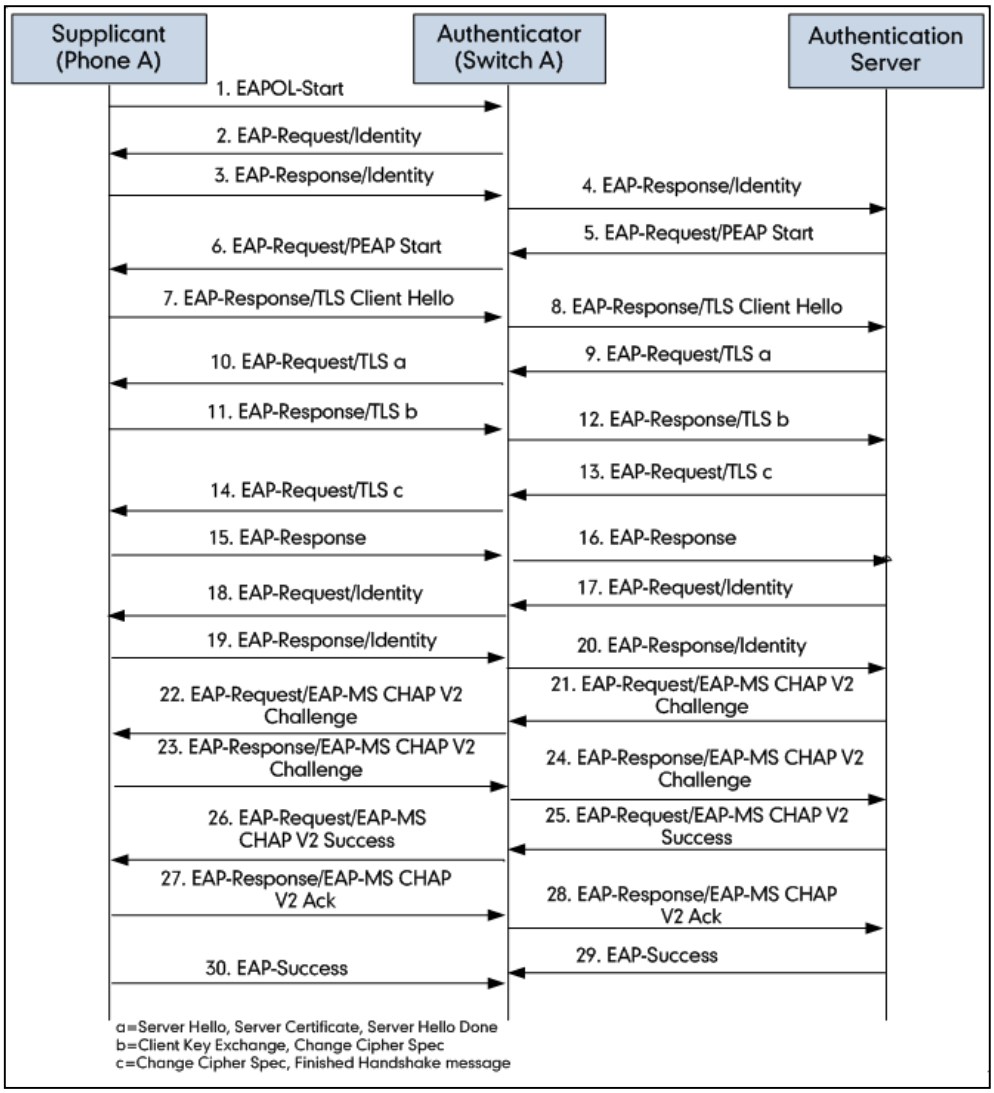
9. The authentication server sends an "EAP-Request" packet to the authenticator. The packet includes a TLS server hello handshake message, a server certificate message, a certificate request message and a server hello done message.
10. The authenticator passes the request to the supplicant.
11. The supplicant responds with an "EAP-Response" packet to the authenticator. The packet includes a TLS change cipher spec message, a client certificate message, a client key exchange message and a certificate verify message.
12. The authenticator passes the response to the authentication server.
13. The authentication server sends an "EAP-Request" packet to the authenticator. The packet includes a TLS change cipher spec message and a finished handshake message. The change cipher spec message is sent to notify the authenticator that subsequent records will be protected under the newly negotiated cipher spec.
14. The authenticator passes the request to the supplicant.
15. The supplicant responds with an "EAP-Response" packet to the authenticator.
16. The authenticator passes the response to the authentication server.
17. The authentication server responds with a success message indicating the supplicant and the authentication server have successfully authenticated each other.
18. The authenticator passes the message to the supplicant.

    After the supplicant is authenticated successfully, the authenticator provides network access permissions. If the supplicant does not provide proper identification, the authentication server responds with a rejection message. The authenticator passes the message to the supplicant and blocks access to the LAN.

    If the supplicant is disabled or reset after successful authentication, the supplicant sends an EAPOL-Logoff message, which prompts the authenticator to block access to the LAN.

A Successful Authentication Using EAP-PEAP/MSCHAPv2 Protocol

The following figure illustrates the scenario of a successful 802.1X authentication process using the EAP-PEAP/MSCHAPv2 protocol.



1. The supplicant sends an "EAPOL-Start" packet to the authenticator.
2. The authenticator responds with an "EAP-Request/Identity" packet to the supplicant.
3. The supplicant responds with an "EAP-Response/Identity" packet to the authenticator.
4. The authenticator strips the Ethernet header and encapsulates the remaining EAP frame in the RADIUS format, and then sends it to the authentication server.
5. The authentication server recognizes the packet as a PEAP type and sends an "EAP-Request" packet with a PEAP start message to the authenticator.
6. The authenticator strips the authentication server's frame header, encapsulates the

remaining EAP frame in the EAPOL format, and then sends it to the supplicant.

7. The supplicant responds with an "EAP-Respond" packet containing a TLS client hello handshake message to the authenticator. The TLS client hello message includes TLS version supported by the supplicant, a session ID, a random number and a set of cipher suites.
8. The authenticator passes the respond to the authentication server.
9. The authentication server sends an "EAP-Request" packet to the authenticator. The packet includes a TLS server hello handshake message, a server certificate message and a server hello done message.
10. The authenticator passes the request to the supplicant.
11. The supplicant responds with an "EAP-Response" packet to the authenticator. The packet includes a TLS change cipher spec message and a certificate verify message.
12. The authenticator passes the response to the authentication server.
13. The authentication server sends an "EAP-Request" packet to the authenticator. The packet includes a TLS change cipher spec message and a finished handshake message. The change cipher spec message is sent to notify the authenticator that subsequent records will be protected under the newly negotiated cipher spec.
14. The authenticator passes the request to the supplicant.
15. The supplicant responds with an "EAP-Response" packet to the authenticator.
16. The authenticator passes the response to the authentication server. The TLS tunnel is established.
17. The authentication server sends an "EAP-Request/Identity" packet to the authenticator.
18. The authenticator passes the request to the supplicant.
19. The supplicant responds with an "EAP-Response/Identity" packet to the authenticator.
20. The authenticator passes the response to the authentication server.
21. The authentication server sends an "EAP-Request" packet to the authenticator. The packet includes an MSCHAPv2 challenge message.
22. The authenticator passes the request to the supplicant.
23. The supplicant responds a challenge message to the authenticator.
24. The authenticator passes the message to the authentication server.
25. The authentication server sends a success message indicating that the supplicant provides proper identity.
26. The authenticator passes the message to the supplicant.
27. The supplicant responds with an ACK message to the authenticator.
28. The authenticator passes the respond message to the authentication server.
29. The authentication server sends a successful message to the authenticator.
30. The authenticator passes the message to the supplicant.

After the supplicant is authenticated successfully, the authenticator provides network access permissions. If the supplicant does not provide proper identification, the authentication server responds with a rejection message. The authenticator passes the message to the supplicant and blocks access to the LAN.

If the supplicant is disabled or reset after successful authentication, the supplicant sends an EAPOL-Logoff message, which prompts the authenticator to block access to the LAN.

A Successful Authentication Using EAP-TTLS/EAP-MSCHAPv2 Protocol

The 802.1X authentication process using the EAP-TTLS/EAP-MSCHAPv2 protocol is quite similar to that using the EAP-PEAP/MSCHAPv2 protocol. For more information, refer to the network resource.

A Successful Authentication Using EAP-PEAP/GTC Protocol

The 802.1X authentication process using the EAP-PEAP/GTC protocol is quite similar to that using the EAP-PEAP/MSCHAPv2 protocol. For more information, refer to the network resource.

A Successful Authentication Using EAP-TTLS/EAP-GTC Protocol

The 802.1X authentication process using the EAP-TTLS/EAP-GTC protocol is quite similar to that using the EAP-PEAP/MSCHAPv2 protocol. For more information, refer to the network resource.

A Successful Authentication Using EAP-FAST Protocol

The 802.1X authentication process using the EAP-FAST protocol is quite similar to that using the EAP-PEAP/MSCHAPv2 protocol. For more information, refer to the network resource.

Customer Feedback

We are striving to improve our documentation quality and we appreciate your feedback. Email your opinions and comments to DocsFeedback@yealink.com.