Yealink

Yealink Technical White Paper

# 802.1X Authentication

Nov. 2013

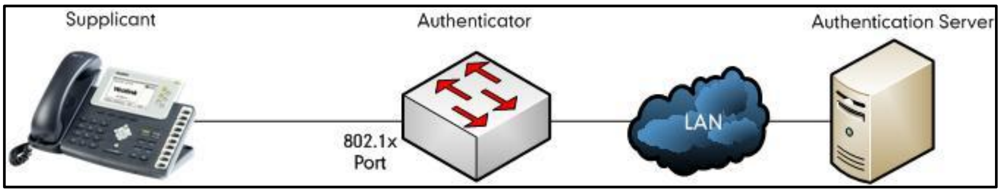
# Table of Contents

# 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, PEAP-MSCHAPv2 and EAP-TTLS/EAP-MSCHAPv2 protocols. The EAP-MD5 protocol is available on all Yealink IP phones. The EAP-TLS and PEAP-MSCHAPv2 protocols are available on Yealink SIP-T28P, SIP-T26P, SIP-T22P and SIP-T20P IP phones running firmware version 70 or later, and Yealink SIP-T21P, SIP-T19P, SIP-T46G, SIP-T42G, SIP-T41P IP phones running firmware version 71 or later. The EAP-TTLS/EAP-MSCHAPv2 protocol is available on Yealink SIP-T28P, SIP-T26P, SIP-T22P, SIP-T20P, SIP-T21P, SIP-T19P, SIP-T46G, SIP-T42G and SIP-T41P IP phones running firmware version 71 or later.

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 for 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 using configuration files, via web user interface or phone user interface. If the EAP-TLS, PEAP-MSCHAPv2 or EAP-TTLS/EAP-MSCHAPv2 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-T28P IP phone running firmware version 71 as examples.

**To configure the 802.1X authentication using configuration files:**

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

   The following table shows the information of parameters:

| Parameter | Description | Valid Value | Default Value |
|---|---|---|---|
| network.802_1x.mode | Specifies the protocol for 802.1X authentication on the IP phone.<br>**0**: Disable<br>**1**: EAP-MD5<br>**2**: EAP-TLS<br>**3**: PEAP-MSCHAPv2<br>**4**: EAP-TTLS/EAP-MSCHAPv2 | 0,1,2,3 or 4 | 0 |
| EAP-MD5 | | | |
| network.802_1x.identity | Specifies the user name for 802.1X authentication. | String | blank |
| network.802_1x.md5_password | Specifies the password for 802.1X authentication. | String | blank |
| EAP-TLS | | | |
| network.802_1x.identity | Specifies the user name for 802.1X authentication. | String | blank |

| | | | |
|---|---|---|---|
| network.802_1x.root_cert_url | Specifies the access URL of the CA certificate (*.pem, *.cer, *.crt or *.der). | String | blank |
| **Parameter** | **Description** | **Valid Value** | **Default Value** |
| network.802_1x.client_cert_url | Specifies the access URL of the client certificate (*.pem or *.cer). | String | blank |
| **PEAP-MSCHAPv2** | | | |
| network.802_1x.identity | Specifies the user name for 802.1X authentication. | String | blank |
| network.802_1x.md5_password | Specifies the password for 802.1X authentication. | String | blank |
| network.802_1x.root_cert_url | Specifies the access URL of the CA certificate (*.pem, *.cer, *.crt or *.der). | String | blank |
| **EAP-TTLS/EAP-MSCHAPv2** | | | |
| network.802_1x.identity | Specifies the user name for 802.1X authentication. | String | blank |
| network.802_1x.md5_password | Specifies the password for 802.1X authentication. | String | blank |
| network.802_1x.root_cert_url | Specifies the access URL of the CA certificate (*.pem, *.cer, *.crt or *.der). | String | blank |

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 and trigger IP phones to perform an auto provisioning for configuration update. The CA certificate and client certificate may be optional according to the specified 802.1X authentication protocol.

   For more information on auto provisioning, refer to Yealink IP Phones Auto Provisioning Guide.

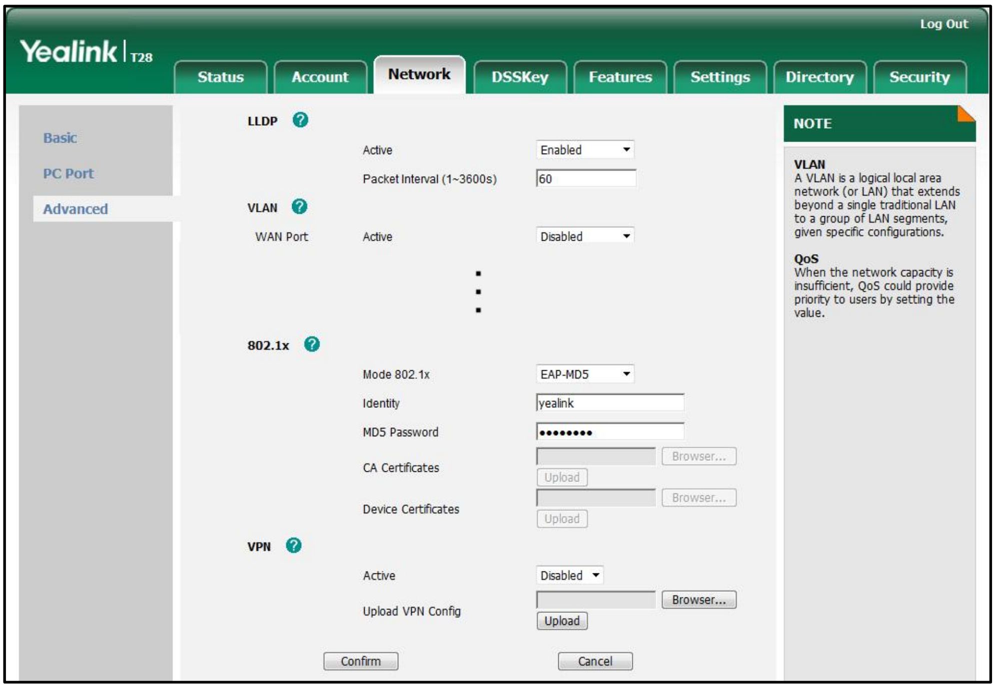
**To configure the 802.1X authentication via web user interface:**

1. Click on **Network**->**Advanced**.
2. Select the desired authentication protocol from the pull-down list of **Mode 802.1x**.

   - 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.



LLDP
Active: Enabled
Packet Interval (1~3600s): 60

VLAN
WAN Port Active: Disabled

⋮

802.1x
Mode 802.1x: EAP-MD5
Identity: yealink
MD5 Password: ••••••••
CA Certificates
Device Certificates

VPN
Active: Disabled
Upload VPN Config

NOTE

VLAN
A VLAN is a logical local area network (or LAN) that extends beyond a single traditional LAN to a group of LAN segments, given specific configurations.

QoS
When the network capacity is insufficient, QoS could provide priority to users by setting the value.

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

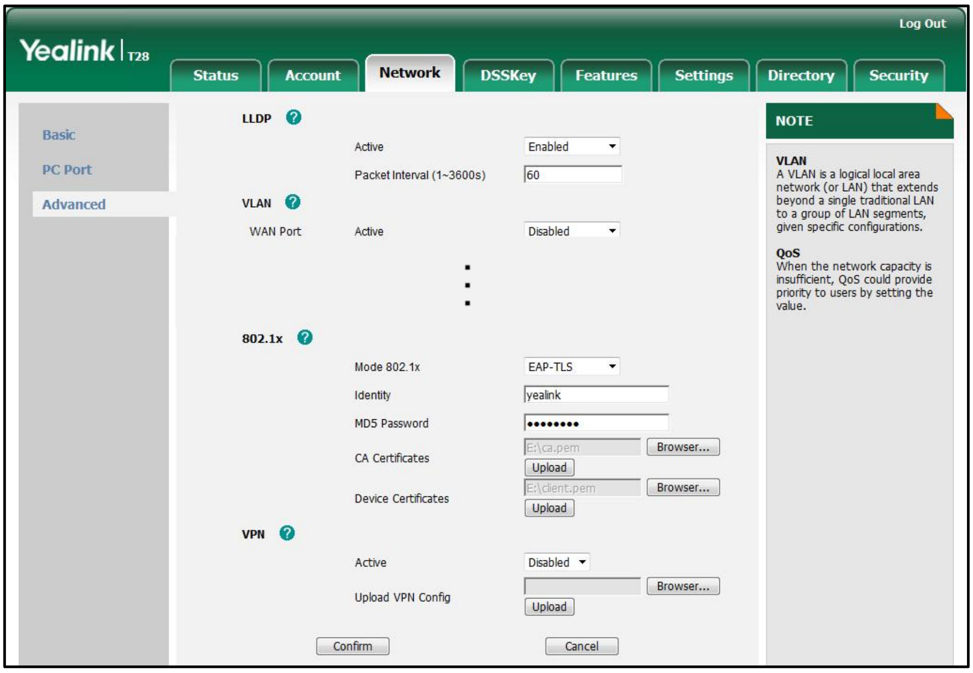
     1) Enter the user name for authentication in the **Identity** field.

     2) Leave the **MD5 Password** field blank.

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

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

5) Click the **Upload** button to upload the CA and client certificates.



LLDP
Active: Enabled
Packet Interval (1~3600s): 60

VLAN
WAN Port Active: Disabled

802.1x
Mode 802.1x: EAP-TLS
Identity: yealink
MD5 Password: ••••••••
CA Certificates: E:\ca.pem Browser... Upload
Device Certificates: E:\client.pem Browser... Upload

VPN
Active: Disabled
Upload VPN Config: Browser... Upload

NOTE

VLAN
A VLAN is a logical local area network (or LAN) that extends beyond a single traditional LAN to a group of LAN segments, given specific configurations.

QoS
When the network capacity is insufficient, QoS could provide priority to users by setting the value.

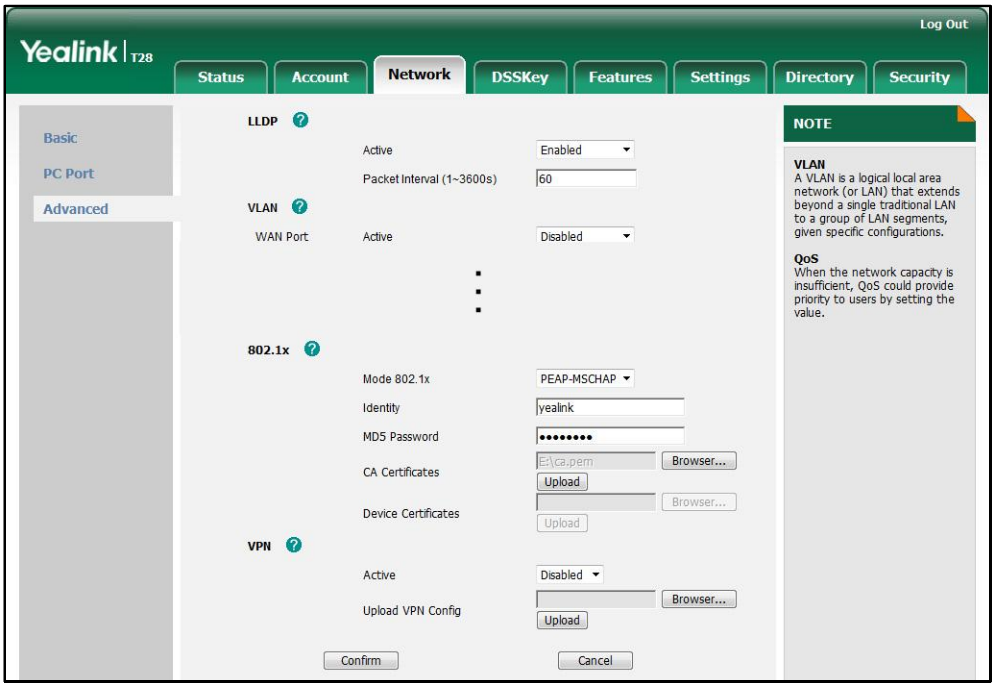
- If you select **PEAP-MSCHAPv2**:

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 locate the certificate (*.pem, *.cer, *.crt or *.der) from your local system.

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



LLDP
Active: Enabled
Packet Interval (1~3600s): 60

VLAN
WAN Port Active: Disabled

802.1x
Mode 802.1x: PEAP-MSCHAP
Identity: yealink
MD5 Password: ••••••••
CA Certificates: E:\ca.pem Browser... Upload
Device Certificates: Browser... Upload

VPN
Active: Disabled
Upload VPN Config: Browser... Upload

NOTE

VLAN
A VLAN is a logical local area network (or LAN) that extends beyond a single traditional LAN to a group of LAN segments, given specific configurations.

QoS
When the network capacity is insufficient, QoS could provide priority to users by setting the value.

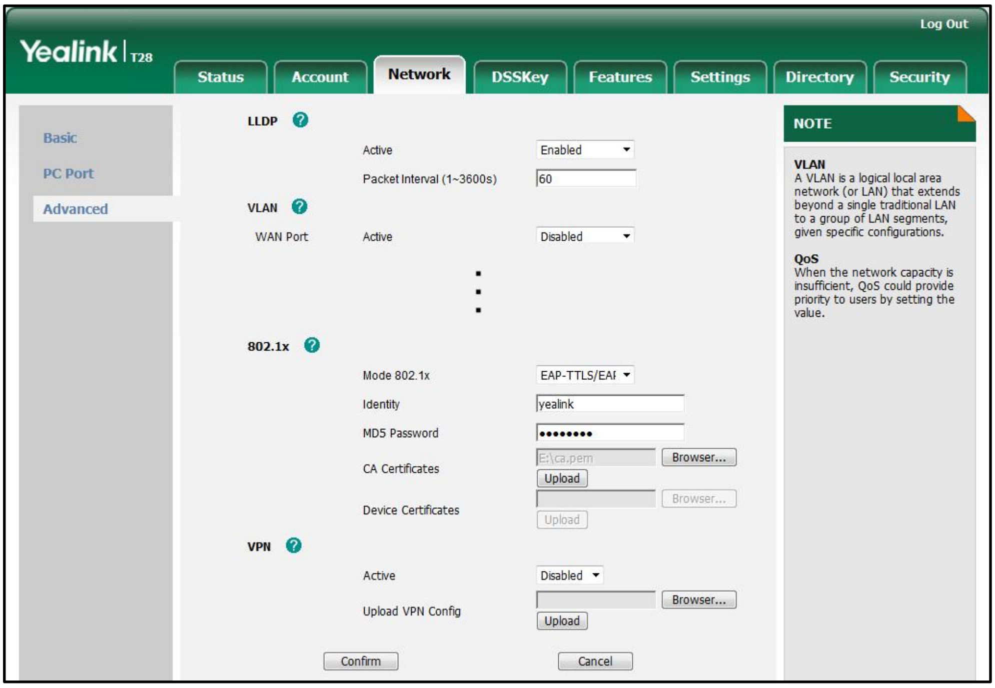
- If you select **EAP-TTLS/EAP-MSCHAPv2**:

  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 locate the certificate (*.pem, *.cer, *.crt or *.der) from your local system.

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



LLDP

Active: Enabled

Packet Interval (1~3600s): 60

VLAN

WAN Port Active: Disabled

802.1x

Mode 802.1x: EAP-TTLS/EAI

Identity: yealink

MD5 Password: ••••••••

CA Certificates: E:\ca.pem

Device Certificates

VPN

Active: Disabled

Upload VPN Config

NOTE

VLAN
A VLAN is a logical local area network (or LAN) that extends beyond a single traditional LAN to a group of LAN segments, given specific configurations.

QoS
When the network capacity is insufficient, QoS could provide priority to users by setting the value.

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

   The web user interface pops up a dialog box to prompt that the 802.1X settings will take effect after a 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.

**To configure the 802.1X authentication via phone user interface:**

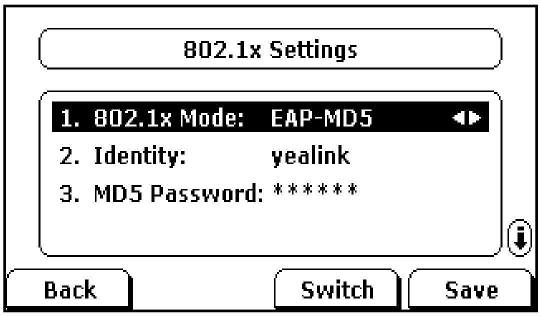
1. Press **Menu**->**Settings**->**Advanced Settings** (password: admin) ->**Network**->**802.1x Settings**.

2. Press ◀ or ▶ , or the **Switch** soft key to select the desired authentication protocol from the **802.1x Mode** field.

   - 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.



802.1x Settings

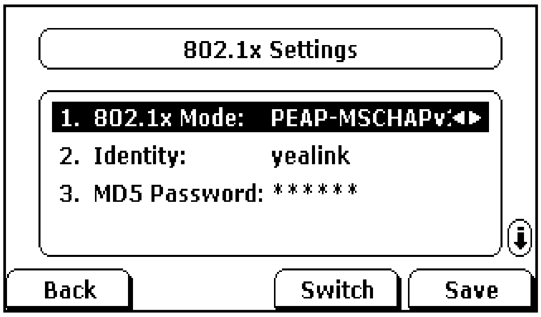
1. 802.1x Mode: EAP-MD5
2. Identity: yealink
3. MD5 Password: ******

- If you select **PEAP-MSCHAPv2**:

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

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

  You should upload the CA certificate using configuration files or via web user interface.



802.1x Settings

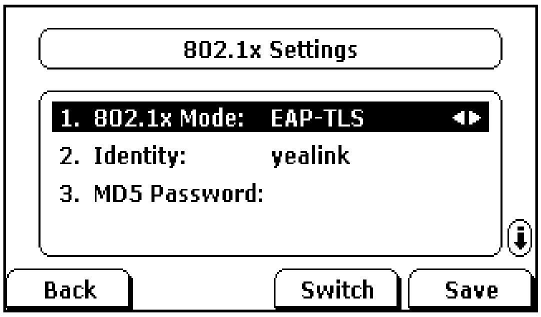
1. 802.1x Mode: PEAP-MSCHAPv
2. Identity: yealink
3. MD5 Password: ******

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

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

  2) Leave the **MD5 Password** field blank.

  You should upload the CA certificate and client certificate using configuration files or via web user interface.



802.1x Settings

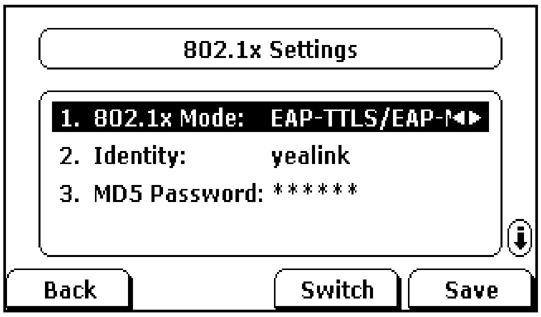
1. 802.1x Mode: EAP-TLS
2. Identity: yealink
3. MD5 Password:

- 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.

You should upload the CA certificate using configuration files or via web user interface.



802.1x Settings

1. 802.1x Mode: EAP-TTLS/EAP-M
2. Identity: yealink
3. MD5 Password: ******

3. Press the **Save** soft key to accept the change or the **Back** soft key to cancel.

   The IP phone reboots automatically to make the 802.1X 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:

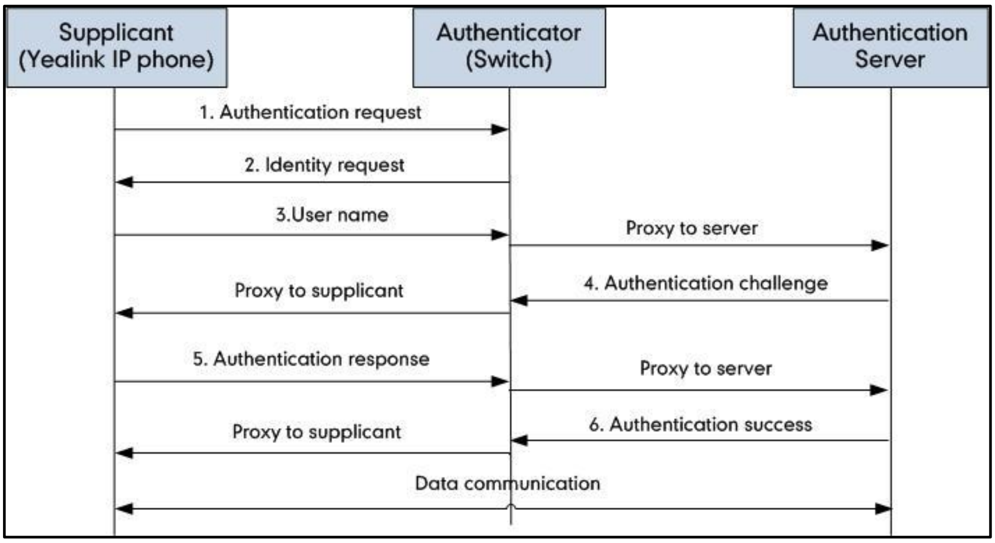
### Pre-authentication

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 the network on 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, 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.

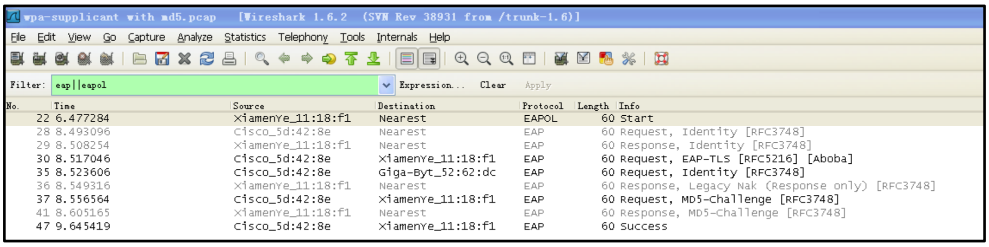
The following figure summarizes an implementation of the 802.1X authentication process using a RADIUS server as the authentication server:



For more details about the 802.1X authentication process using EAP-MD5, EAP-TLS, PEAP-MSCHAPv2 and EAP-TTLS/EAP-MSCHAPv2 protocols, refer to Appendix B: 802.1X Authentication Process on page 14.

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 EAP-MD5 protocol:

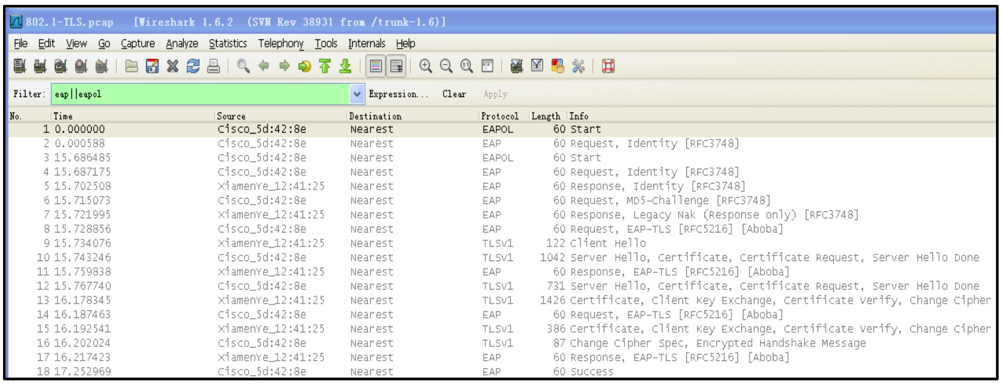


wpa-supplicant with md5.pcap [Wireshark 1.6.2 (SVN Rev 38931 from /trunk-1.6)]

Filter: eap||eapol

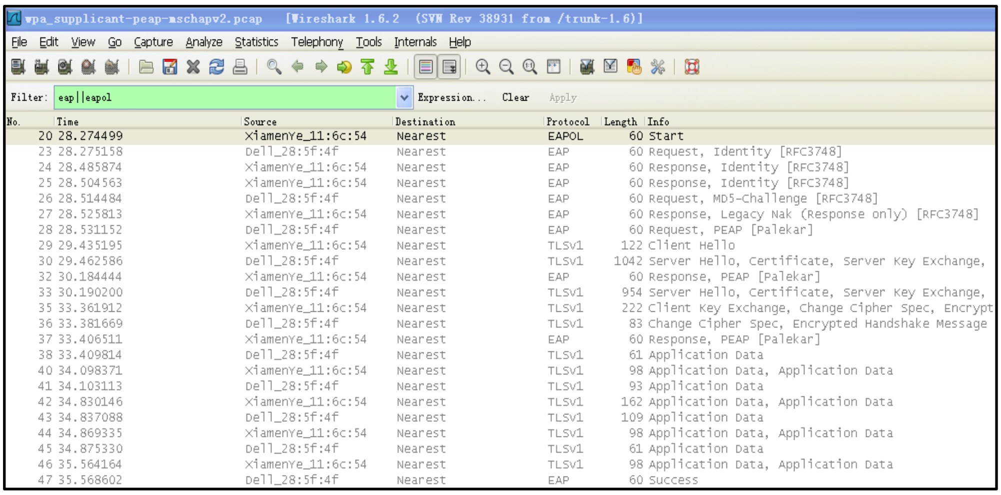
```
No.  Time        Source              Destination         Protocol  Length  Info
  22 6.477284    XiamenYe_11:18:f1   Nearest             EAPOL         60  Start
  28 8.493096    Cisco_5d:42:8e      Nearest             EAP           60  Request, Identity [RFC3748]
  29 8.508254    XiamenYe_11:18:f1   Nearest             EAP           60  Response, Identity [RFC3748]
  30 8.517046    Cisco_5d:42:8e      XiamenYe_11:18:f1   EAP           60  Request, EAP-TLS [RFC5216] [Aboba]
  35 8.523606    Cisco_5d:42:8e      Giga-Byt_52:62:dc   EAP           60  Request, Identity [RFC3748]
  36 8.549316    XiamenYe_11:18:f1   Nearest             EAP           60  Response, Legacy Nak (Response only) [RFC3748]
  37 8.556564    Cisco_5d:42:8e      XiamenYe_11:18:f1   EAP           60  Request, MD5-Challenge [RFC3748]
  41 8.605165    XiamenYe_11:18:f1   Nearest             EAP           60  Response, MD5-Challenge [RFC3748]
  47 9.645419    Cisco_5d:42:8e      XiamenYe_11:18:f1   EAP           60  Success
```

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


802.1-TLS.pcap [Wireshark 1.6.2 (SVN Rev 38931 from /trunk-1.6)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply

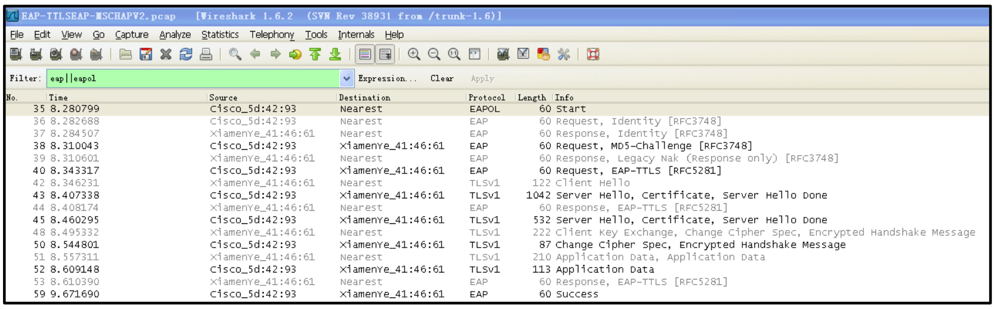
```
Filter: eap||eapol
No.  Time       Source              Destination  Protocol Length Info
   1 0.000000   Cisco_5d:42:8e      Nearest      EAPOL       60 Start
   2 0.000588   Cisco_5d:42:8e      Nearest      EAP         60 Request, Identity [RFC3748]
   3 15.686485  Cisco_5d:42:8e      Nearest      EAPOL       60 Start
   4 15.687175  Cisco_5d:42:8e      Nearest      EAP         60 Request, Identity [RFC3748]
   5 15.702508  XiamenYe_12:41:25   Nearest      EAP         60 Response, Identity [RFC3748]
   6 15.715073  Cisco_5d:42:8e      Nearest      EAP         60 Request, MD5-Challenge [RFC3748]
   7 15.721995  XiamenYe_12:41:25   Nearest      EAP         60 Response, Legacy Nak (Response only) [RFC3748]
   8 15.728856  Cisco_5d:42:8e      Nearest      EAP         60 Request, EAP-TLS [RFC5216] [Aboba]
   9 15.734076  XiamenYe_12:41:25   Nearest      TLSv1      122 Client Hello
  10 15.743246  Cisco_5d:42:8e      Nearest      TLSv1     1042 Server Hello, Certificate, Certificate Request, Server Hello Done
  11 15.759838  XiamenYe_12:41:25   Nearest      EAP         60 Response, EAP-TLS [RFC5216] [Aboba]
  12 15.767740  Cisco_5d:42:8e      Nearest      TLSv1      731 Server Hello, Certificate, Certificate Request, Server Hello Done
  13 16.178345  XiamenYe_12:41:25   Nearest      TLSv1     1426 Certificate, Client Key Exchange, Certificate Verify, Change Cipher
  14 16.187463  Cisco_5d:42:8e      Nearest      EAP         60 Request, EAP-TLS [RFC5216] [Aboba]
  15 16.192541  XiamenYe_12:41:25   Nearest      TLSv1      386 Certificate, Client Key Exchange, Certificate Verify, Change Cipher
  16 16.202024  Cisco_5d:42:8e      Nearest      TLSv1       87 Change Cipher Spec, Encrypted Handshake Message
  17 16.217423  XiamenYe_12:41:25   Nearest      EAP         60 Response, EAP-TLS [RFC5216] [Aboba]
  18 17.252969  Cisco_5d:42:8e      Nearest      EAP         60 Success
```

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


wpa_supplicant-peap-mschapv2.pcap [Wireshark 1.6.2 (SVN Rev 38931 from /trunk-1.6)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply

```
Filter: eap||eapol
No.  Time       Source              Destination  Protocol Length Info
  20 28.274499  XiamenYe_11:6c:54   Nearest      EAPOL       60 Start
  23 28.275158  Dell_28:5f:4f       Nearest      EAP         60 Request, Identity [RFC3748]
  24 28.485874  XiamenYe_11:6c:54   Nearest      EAP         60 Response, Identity [RFC3748]
  25 28.504563  XiamenYe_11:6c:54   Nearest      EAP         60 Response, Identity [RFC3748]
  26 28.514484  Dell_28:5f:4f       Nearest      EAP         60 Request, MD5-Challenge [RFC3748]
  27 28.525813  XiamenYe_11:6c:54   Nearest      EAP         60 Response, Legacy Nak (Response only) [RFC3748]
  28 28.531152  Dell_28:5f:4f       Nearest      EAP         60 Request, PEAP [Palekar]
  29 29.435195  XiamenYe_11:6c:54   Nearest      TLSv1      122 Client Hello
  30 29.462586  Dell_28:5f:4f       Nearest      TLSv1     1042 Server Hello, Certificate, Server Key Exchange,
  32 30.184444  XiamenYe_11:6c:54   Nearest      EAP         60 Response, PEAP [Palekar]
  33 30.190200  Dell_28:5f:4f       Nearest      TLSv1      954 Server Hello, Certificate, Server Key Exchange,
  35 33.361912  XiamenYe_11:6c:54   Nearest      TLSv1      222 Client Key Exchange, Change Cipher Spec, Encrypt
  36 33.381669  Dell_28:5f:4f       Nearest      TLSv1       83 Change Cipher Spec, Encrypted Handshake Message
  37 33.406511  XiamenYe_11:6c:54   Nearest      EAP         60 Response, PEAP [Palekar]
  38 33.409814  Dell_28:5f:4f       Nearest      TLSv1       61 Application Data
  40 34.098371  XiamenYe_11:6c:54   Nearest      TLSv1       98 Application Data, Application Data
  41 34.103113  Dell_28:5f:4f       Nearest      TLSv1       93 Application Data
  42 34.830146  XiamenYe_11:6c:54   Nearest      TLSv1      162 Application Data, Application Data
  43 34.837088  Dell_28:5f:4f       Nearest      TLSv1      109 Application Data
  44 34.869335  XiamenYe_11:6c:54   Nearest      TLSv1       98 Application Data, Application Data
  45 34.875330  Dell_28:5f:4f       Nearest      TLSv1       61 Application Data
  46 35.564164  XiamenYe_11:6c:54   Nearest      TLSv1       98 Application Data, Application Data
  47 35.568602  Dell_28:5f:4f       Nearest      EAP         60 Success
```

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


EAP-TTLSEAP-MSCHAPV2.pcap [Wireshark 1.6.2 (SVN Rev 38931 from /trunk-1.6)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply

```
Filter: eap||eapol
No.  Time      Source              Destination         Protocol Length Info
  35 8.280799  Cisco_5d:42:93      Nearest             EAPOL       60 Start
  36 8.282688  Cisco_5d:42:93      Nearest             EAP         60 Request, Identity [RFC3748]
  37 8.284507  XiamenYe_41:46:61   Nearest             EAP         60 Response, Identity [RFC3748]
  38 8.310043  Cisco_5d:42:93      XiamenYe_41:46:61   EAP         60 Request, MD5-Challenge [RFC3748]
  39 8.310601  XiamenYe_41:46:61   Nearest             EAP         60 Response, Legacy Nak (Response only) [RFC3748]
  40 8.343317  Cisco_5d:42:93      XiamenYe_41:46:61   EAP         60 Request, EAP-TTLS [RFC5281]
  42 8.346231  XiamenYe_41:46:61   Nearest             TLSv1      122 Client Hello
  43 8.407338  Cisco_5d:42:93      XiamenYe_41:46:61   TLSv1     1042 Server Hello, Certificate, Server Hello Done
  44 8.408174  XiamenYe_41:46:61   Nearest             EAP         60 Response, EAP-TTLS [RFC5281]
  45 8.460295  Cisco_5d:42:93      XiamenYe_41:46:61   TLSv1      532 Server Hello, Certificate, Server Hello Done
  48 8.495332  XiamenYe_41:46:61   Nearest             TLSv1      222 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
  50 8.544801  Cisco_5d:42:93      XiamenYe_41:46:61   TLSv1       87 Change Cipher Spec, Encrypted Handshake Message
  51 8.557311  XiamenYe_41:46:61   Nearest             TLSv1      210 Application Data, Application Data
  52 8.609148  Cisco_5d:42:93      XiamenYe_41:46:61   TLSv1      113 Application Data
  53 8.610390  XiamenYe_41:46:61   Nearest             EAP         60 Response, EAP-TTLS [RFC5281]
  59 9.671690  Cisco_5d:42:93      XiamenYe_41:46:61   EAP         60 Success
```

# 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.

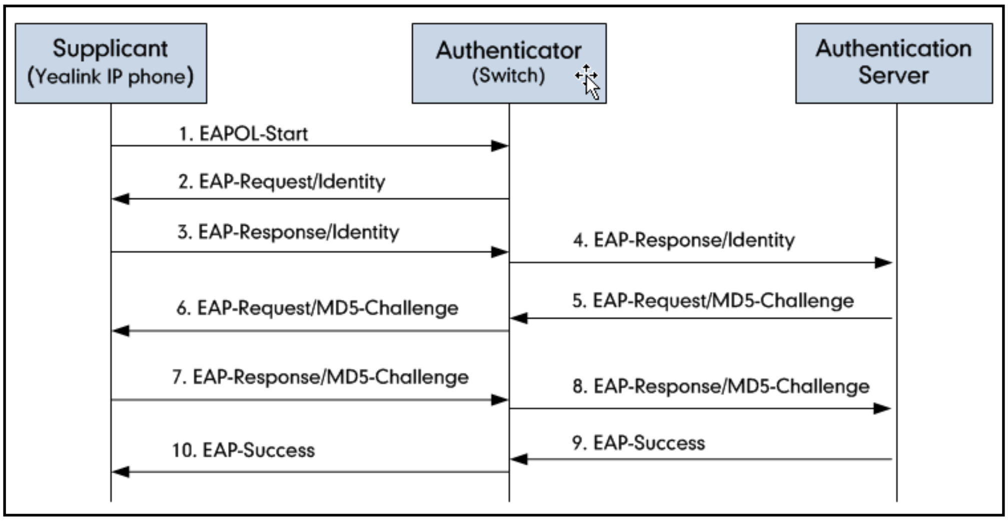
**TTLS** (Tunneled Transport Layer Security) –Extends TLS to improve some weak points, but it does not require a supplicant-side certificate.

**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 EAP-MD5 protocol.

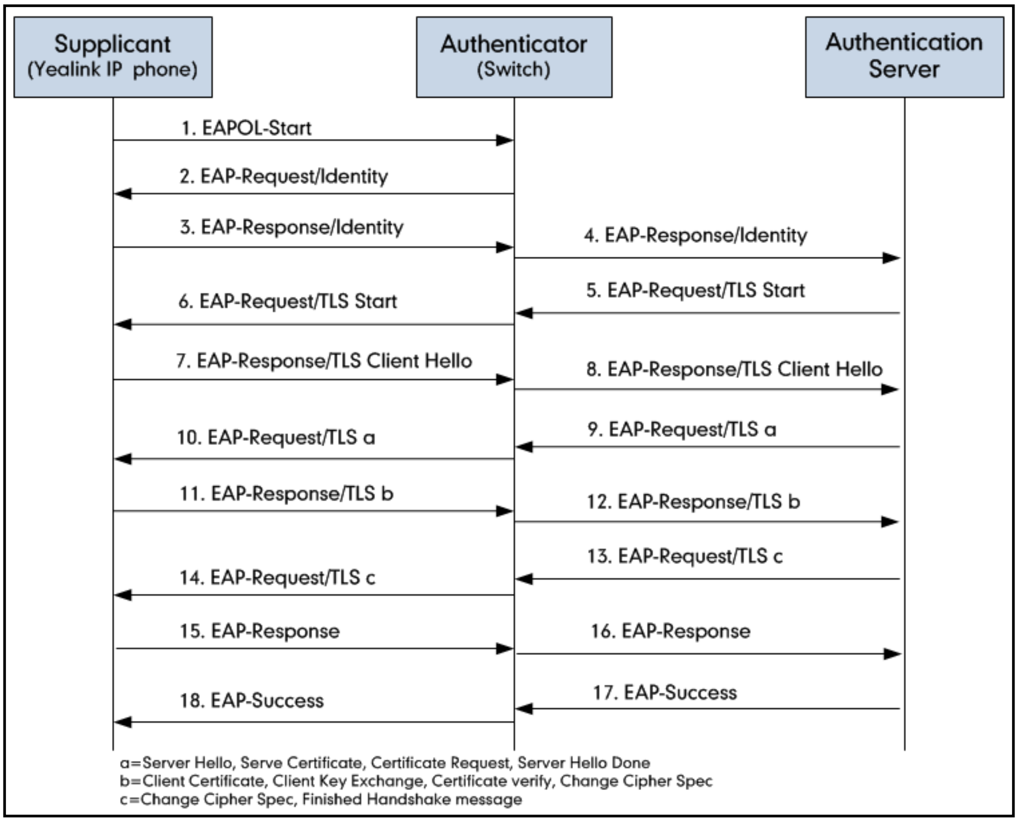


1. The supplicant sends an "EAPOL-Start" packet to the authenticator.
2. The authenticator responds with an "EAP-Request/Identity" packet to the supplicant.
3. The supplicant responds with an "EAP-Response/Identity" packet to the authenticator.
4. The authenticator strips the Ethernet header and encapsulates 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. When the supplicant is disabled or reset, the supplicant sends an EAPOL-Logoff message, which prompts the authenticator to block access to the LAN Success message.

## A Successful Authentication Using EAP-TLS Protocol

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



a=Server Hello, Serve Certificate, Certificate Request, Server Hello Done
b=Client Certificate, Client Key Exchange, Certificate verify, Change Cipher Spec
c=Change Cipher Spec, Finished Handshake message

1. The supplicant sends an “EAPO-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-Respond” 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

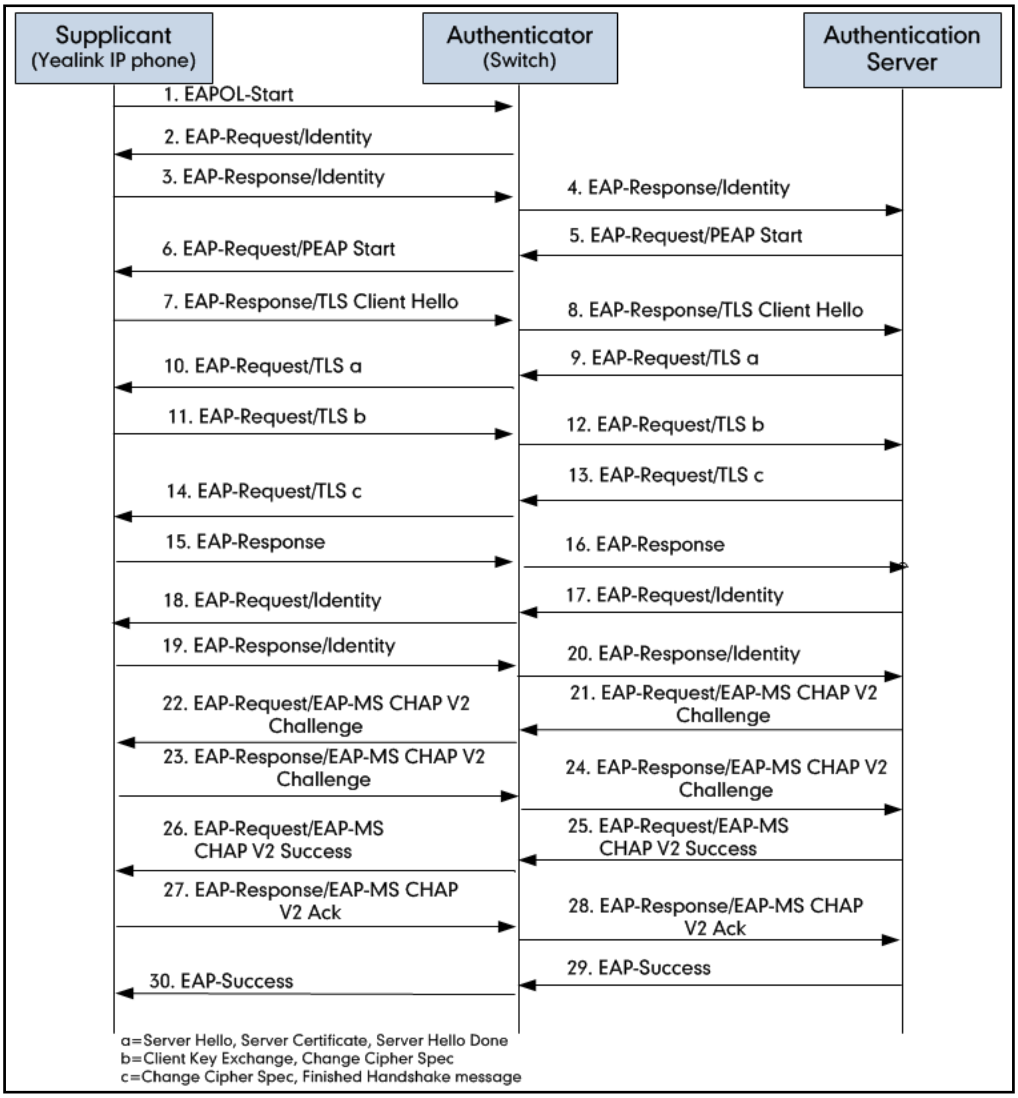
message, a certificate request message and a server hello done message.

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-Response” 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. When the supplicant is disabled or reset, the supplicant sends an EAPOL-Logoff message, which prompts the authenticator to block access to the LAN.

## A Successful Authentication Using PEAP-MSCHAPv2 Protocol

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


Supplicant (Yealink IP phone)
Authenticator (Switch)
Authentication Server
1. EAPOL-Start
2. EAP-Request/Identity
3. EAP-Response/Identity
4. EAP-Response/Identity
5. EAP-Request/PEAP Start
6. EAP-Request/PEAP Start
7. EAP-Response/TLS Client Hello
8. EAP-Response/TLS Client Hello
9. EAP-Request/TLS a
10. EAP-Request/TLS a
11. EAP-Request/TLS b
12. EAP-Request/TLS b
13. EAP-Request/TLS c
14. EAP-Request/TLS c
15. EAP-Response
16. EAP-Response
17. EAP-Request/Identity
18. EAP-Request/Identity
19. EAP-Response/Identity
20. EAP-Response/Identity
21. EAP-Request/EAP-MS CHAP V2 Challenge
22. EAP-Request/EAP-MS CHAP V2 Challenge
23. EAP-Response/EAP-MS CHAP V2 Challenge
24. EAP-Response/EAP-MS CHAP V2 Challenge
25. EAP-Request/EAP-MS CHAP V2 Success
26. EAP-Request/EAP-MS CHAP V2 Success
27. EAP-Response/EAP-MS CHAP V2 Ack
28. EAP-Response/EAP-MS CHAP V2 Ack
29. EAP-Success
30. EAP-Success
a=Server Hello, Server Certificate, Server Hello Done
b=Client Key Exchange, Change Cipher Spec
c=Change Cipher Spec, Finished Handshake message

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-Response" 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 a 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 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. When the supplicant is disabled or reset, 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 EAP-TTLS/EAP-MSCHAPv2 protocol is quite similar to that using 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.