

# Navigate Client EKU Sunset with Expressway x15.5

## Introduction

This document describes navigating the client EKU sunset with Cisco Expressway x15.5.

## Background Information

Digital certificates are electronic credentials issued by trusted Certificate Authorities (CAs) that secure communication between servers and clients by ensuring authentication, data integrity, and confidentiality. These certificates contain Extended Key Usage (EKU) fields that define their purpose:

- **Server Authentication EKU (`id-kp-serverAuth`)** is used when a server presents its certificate to prove identity.
- **Client Authentication EKU (`id-kp-clientAuth`)** is used in mutual TLS (mTLS) connections where both parties authenticate each other.

Traditionally, a single certificate could contain both Server and Client Authentication EKUs, allowing it to serve dual purposes. This is particularly important for products like Cisco Expressway that act as both server and client in different connection scenarios.

## Problem Definition

### Chrome Root Program Policy Change

Effective June 2026, the Chrome Root Program Policy restricts Root Certificate Authority (CA) certificates included in the Chrome Root Store, phasing out multi-purpose roots to align all public-key infrastructure (PKI) hierarchies to serve only TLS server authentication use cases.

### Key Policy Requirements

- Public Root CAs must assert Extended Key Usage (EKU) ONLY for Server Authentication (`id-kp-serverAuth`).
- Including Client Authentication EKU in these certificates is prohibited.
- No more mixed-use root CAs for public server TLS certificates.
- Enforcement Timeline: June 2026

## Public CA Response Timeline

- October 2025: Many public CAs (DigiCert, Sectigo, SSL) began issuing server-only certificates by default.
- May 2026: Public CA servers stop issuing Client Authentication EKU certifications
- June 2026: Chrome Root Program Policy becomes fully effective



**Note:** This policy applies only to certificates issued by public CAs. Private PKI and self-signed certificates are not affected by this policy.

---

If you are interested in reading about impact of sunseting of client EKU on Expressways, refer to [Prepare Expressway for Client Auth EKU Sunset in Public CA Certificates.](#)

## Expressway Release x15.5 with Solution

### Expressway x15.5

Expressway x15.5 comes with a proposed fix for a problem which arises due to sunseting of client EKU by all public certificate authorities. This is a global problem and affects all vendors/deployments who choose to use public PKI certificates.

x15.4, a release prior, had a CLI command switch which allowed admin to upload Server EKU only certificate (no client EKU present) on Expressway E.

xConfiguration XCP TLS Certificate CVS EnableServerEkuUpload: On



**Note:** This command is deprecated on x15.5.

---

### X15.5 Certificate Store Addition

x15.5 has two certificate stores:

1. Server certificate store
2. Client certificate store

Expressways (single Nic or dual Nic): Both Expressway interfaces are able to use 2 certificate stores on an


as-needed basis.


Example:


- When expressway acts as a client during TLS handshake, client certificate is presented.
- When expressway acts as server during TLS handshake, server certificate is presented.





**Note:** Both certificate stores (Client and Server) use same Trusted CA library. Ensure that the CA who signed server and client certificates are uploaded correctly on the Trust store. Diagnostic logs now include server certificate and client certificate in PEM file format.


 ca\_vcs8c\_2026-03-25\_03\_20\_11.pem


 client\_vcs8c\_2026-03-25\_03\_20\_11.pem

 eth0\_diagnostic\_logging\_tcpdump00\_vcs8c\_2026-03-25\_03\_20\_11.pcap

 loggingsnapshot\_vcs8c\_2026-03-25\_03\_20\_11.txt

 server\_vcs8c\_2026-03-25\_03\_20\_11.pem

 xconf\_dump\_vcs8c\_2026-03-25\_03\_20\_11.txt

 xconf\_dump\_vcs8c\_2026-03-25\_03\_20\_11.xml

 xstat\_dump\_vcs8c\_2026-03-25\_03\_20\_11.txt

 xstat\_dump\_vcs8c\_2026-03-25\_03\_20\_11.xml

## Upgrade from X15.4 or Prior Version to X15.5

When an upgrade is performed, the server certificate from x15.4 or prior version, the Expressway server certificate store is copied to client certificate store on x15.5. Client and server certificate stores on x15.5 have same certificate.

## Example with Screenshots

Expressway server on 15.4, current server certificate Serial Number 46:df:76:aa:00:00:00:00:29

Certificate:

Version: 3 (0x2)

Serial Number:

46:df:76:aa:00:00:00:00:29

Validity

Not Before: Mar 14 02:37:40 2026 GMT

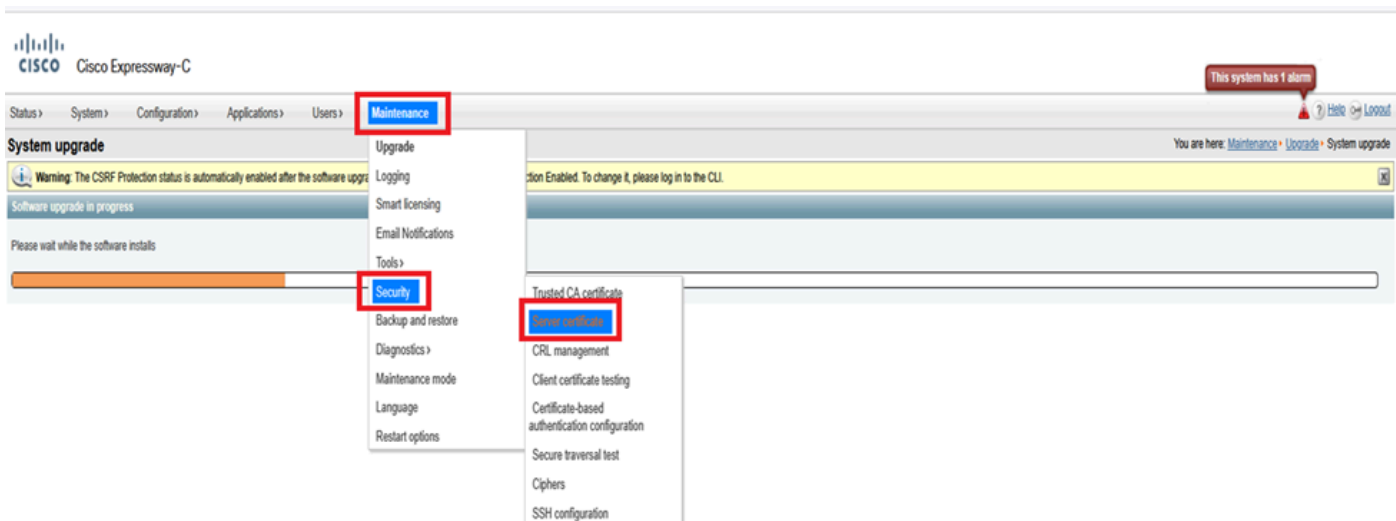
Not After : Mar 14 02:47:40 2028 GMT

Subject: C = IN, ST = KA, L = KA, O = Cisco, OU = TAc, CN = cluster.s.com

Expressway file system persistent/cert directory on x15.4:

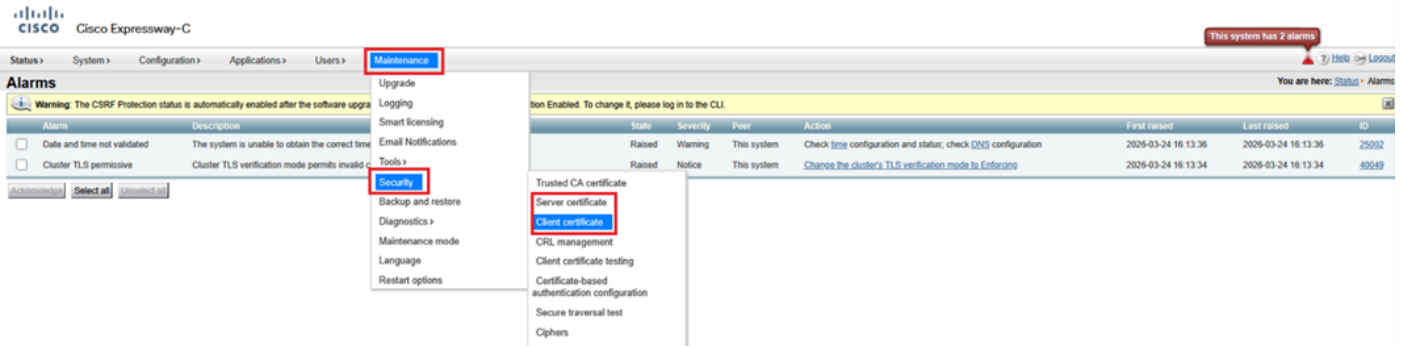
Name	Size	Changed	Rights	Owner
..		3/24/2026 3:16:20 PM	rw-r--r--	root
generated_csr		3/14/2026 8:20:12 AM	rw-r--r--	_nobody
multidomaincerts		3/17/2026 6:19:48 PM	rw-r--r--	root
saml		2/4/2026 3:56:54 PM	rw-r--r--	root
ca.pem	16 KB	3/14/2026 7:37:55 AM	rw-r--r--	_nobody
client-ca.crl	2 KB	2/4/2026 3:54:04 PM	rw-r--r--	_nobody
client-ca.crl.default	2 KB	2/4/2026 3:54:04 PM	rw-r--r--	root
crl-update.conf	1 KB	3/24/2026 3:16:09 PM	rw-r--r--	root
mtls_ca.pem	9 KB	1/19/2026 8:34:16 PM	rw-r--r--	_nobody
policy-services.crl	2 KB	1/19/2026 8:34:16 PM	rw-r--r--	_nobody
policy-services.crl.default	2 KB	1/19/2026 8:34:16 PM	rw-r--r--	root
privkey.pem	4 KB	3/14/2026 8:17:09 AM	r--r-----	root
server.pem	3 KB	3/14/2026 8:19:20 AM	rw-r--r--	_nobody
server-ssh.pem	6 KB	3/24/2026 3:16:12 PM	rw-r--r--	_pfwd

Expressway menu (**Maintenance > Security > Server** certificate) on x15.4 (only server certificate field present):



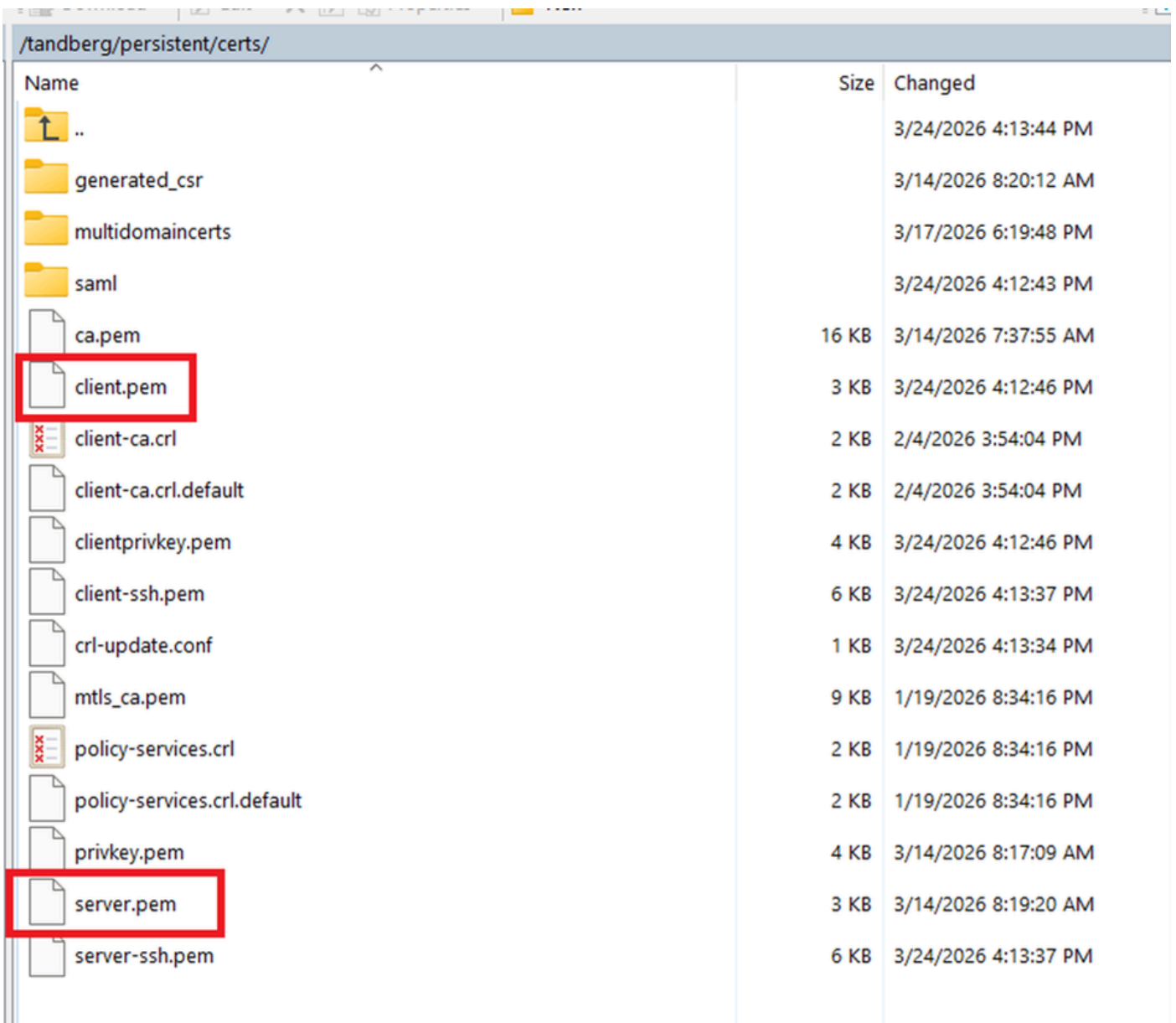
### After Successful Upgrade to x15.5

Here, you see 2 certificate options under **Maintenance > Security > client certificate** and server certificates. After upgrading to x15.5, both the Server and Client certificate portals on web admin shows the same certificate because the server certificate from x15.4 was copied to the client certificate store on x15.5.



Post-upgrade to x15.5 existing certificate and private key has been copied to the client certificate store.

Expressway file system persistent/cert directory on x15.5:



**X15.5 EKU Check During TLS Handshake**

On x15.5, a new CLI command has been introduced to check Extended key usage (EKU) during TLS handshake. Default value is “ON.” The command set is valid on Expressway Core and Edge.

The command set triggers a check for all INBOUND SIP TLS connections into Expressway. (inbound client hellos/certificate presented). When turned “ON,” this checks whether or not the presented certificate by the TLS initiator contains client ECU in certificate. IF turned OFF, the check is bypassed; however, the server ECU is checked if it is present on certificate.

xconfiguration SIP TLS Certificate ExtendedKeyUsage Checking Mode: ON/OFF:



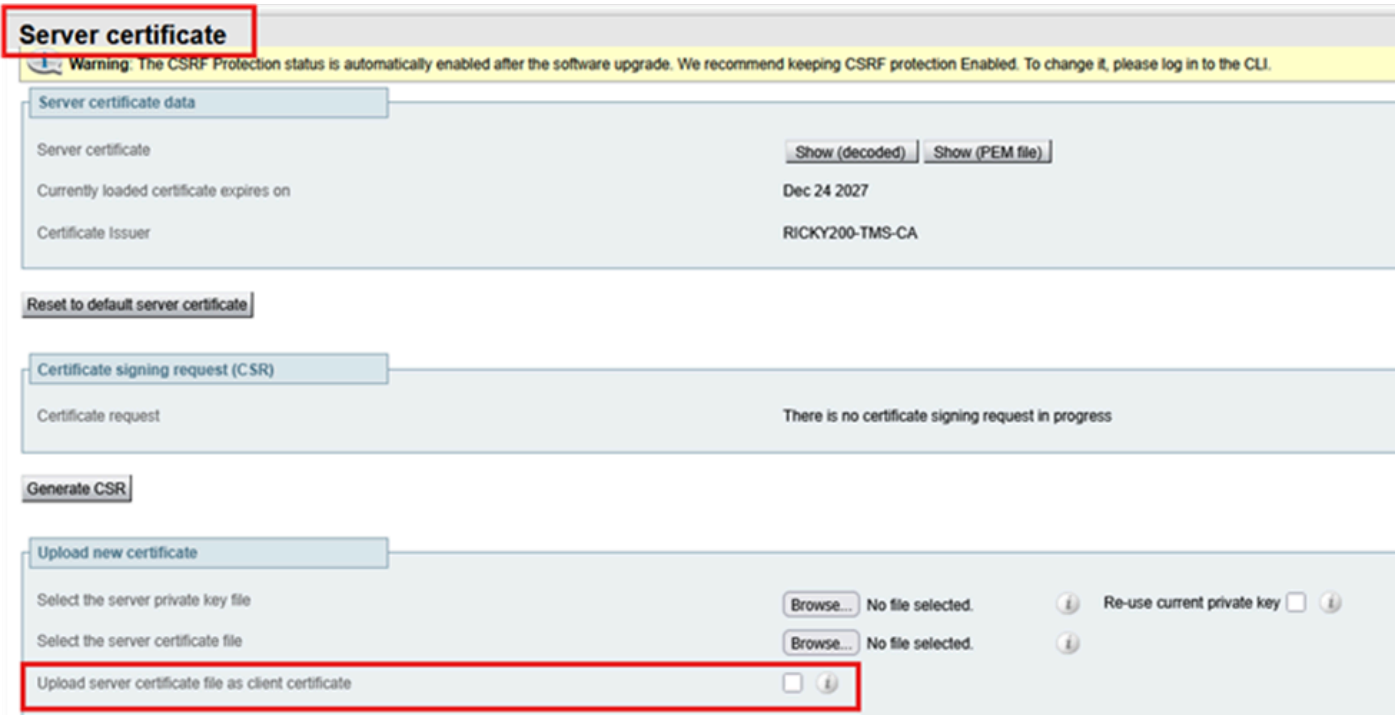
**Note:** If you generate a client certificate, signing a CSR which does not contain client ECU (an example of public CA signed certificate), you are *not* able to upload this certificate manually on the client certificate store. So, you need to ensure that certificates generated by signing a CSR always contain the client ECU (a private CA can be used to insert the client ECU).



**Tip:** This error becomes evident when you attempt to upload a CSR signed certificate, which is missing the client ECU, from client certificate store.

The screenshot shows the Cisco Expressway-E web interface. At the top left is the Cisco logo and the text "Cisco Expressway-E". Below this is a navigation bar with links for "Status >", "System >", "Configuration >", "Applications >", "Users >", and "Maintenance >". The main heading is "Client certificate". Below the heading, there is a yellow error message box with a red border that reads: "Invalid certificate: The file provided does not have a client usage attribute. Services requiring mutual TLS may not work." Below the error message is a yellow warning box that reads: "Warning: The CSRF Protection status is automatically enabled after the software upgrade. We recommend keeping CSRF protection Enabled. To change it, please log in to the CLI." At the bottom of the screenshot, there is a blue box labeled "Client certificate data".

However, if you choose to upload a certificate which has a server ECU only (no client ECU) via server certificate store and select **Upload server certificate file as client certificate**, the certificate is copied to the client certificate store. Admins who do not want to use a private CA signed certificate on Expressway-Edge can choose to copy the server ECU only from the server certificate store to the client certificate store.

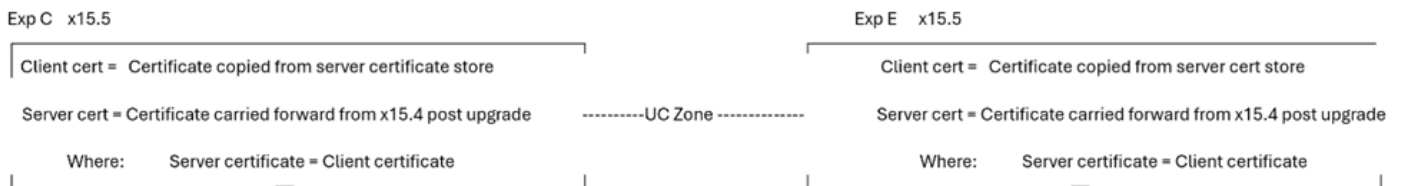


## Multiple Certificate Stores, Multiple Deployment Scenarios

Since now there are two certificate stores on Expressway, there are multiple scenarios of certificate stores.

### Condition 1: Upgrade

When Expressway is upgraded from x15.4 or prior to x15.5, this condition is true. Existing certificates from x15.4 version are copied into two (2) certificate stores. On the x15.5 client and server, the certificates are the same.

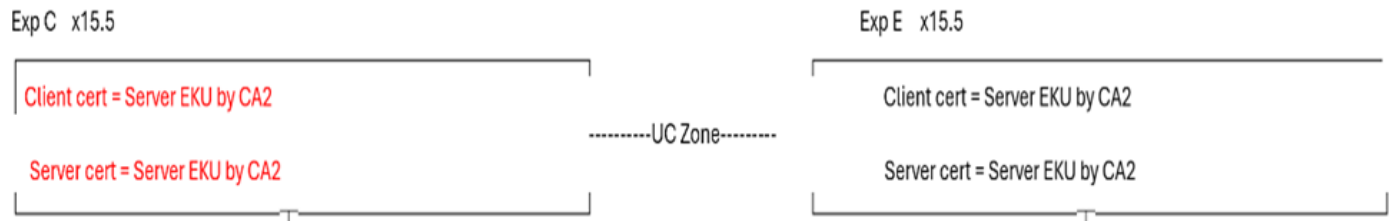


### Condition 2: When the Admin Installs New Certificate on x15.5 (Existing Certs Expired)

CA 1 = Internal CA

CA 2 = Public CA

In the Figure shown next, Expressway Core has a client certificate with server ECU only signed by CA 2 (Public CA) and a server certificate with server ECU only signed by CA 2 (Public CA). Similarly, Expressway E has a client certificate with the server ECU signed by CA2 (public CA) and a server certificate with server ECU only signed by CA 2 (Public CA).



If the Expressway core server certificate does not have a client ECU, Unified communications traversal zone, MRA, the WebRTC proxy does not work. Ensure that the Expressway Core server certificate has a client ECU. This is a common use case where users choose to sign all certificates from public CA. Since public CA does not include Client ECU in certificates, the Unified communications traversal zone does become active.

To make UC zone active, a quick fix is to turn off the ECU check on Expressway E. This brings up the UC zone. However, SSH tunnels stay inactive. As of today, the SSH tunnel communication on 2222 requires validation of the client ECU.

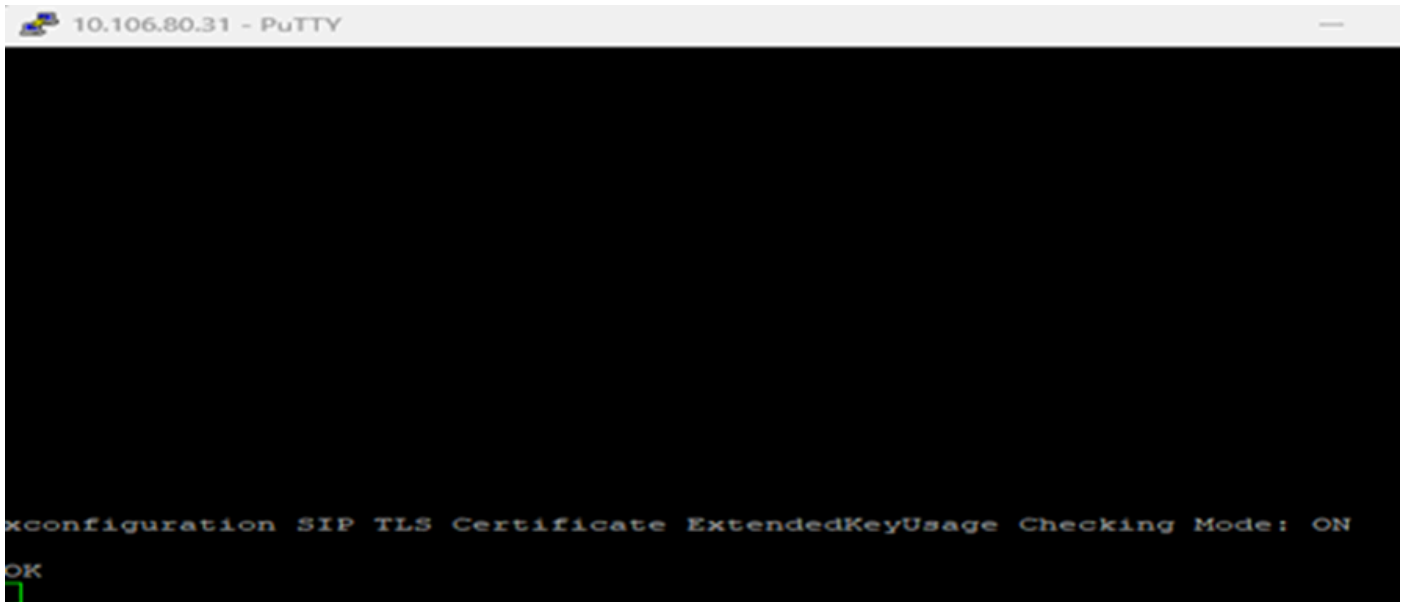
MRA client login and WebRTC proxy functions do not function. You could have to resort to private CA.

### Test Case 1

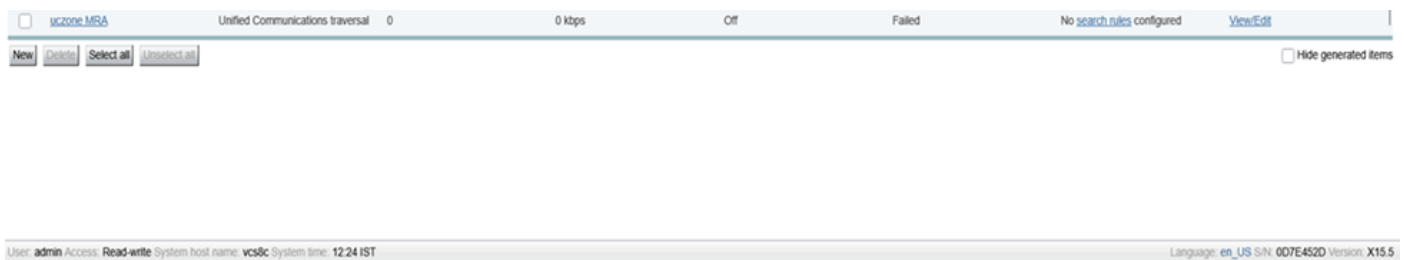
- When ECU check is “ON” on Expressway E
- When Client and Server certificate on Expressway core has Server ECU only
- UC zone status is FAILED

On Expressway-Edge ExtendedKeyUsage Check ON.

xconfiguration SIP TLS Certificate ExtendedKeyUsage Checking Mode: On:



Unified communication zone failure:



Expressway E logs show where 10.106.80.16 = Expressway Core, 10.106.80.31 = Expressway Edge:

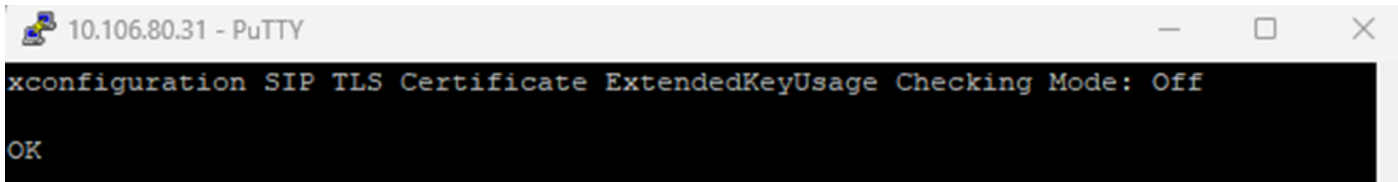


## Test Case 2

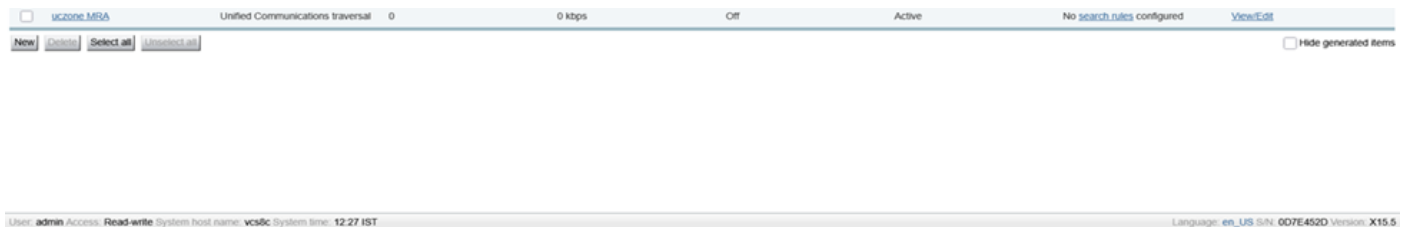
- When EKU check is OFF on Expressway E
- When Client and Server certificate on Expressway Core has server only EKU
- UC zone status is ACTIVE

Turn OFF EKU check on Expressway E.

xconfiguration SIP TLS Certificate ExtendedKeyUsage Checking Mode: Off



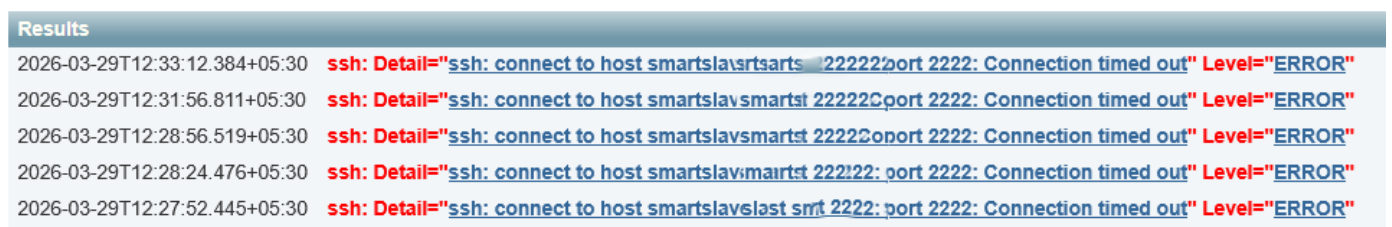
Unified communication zone Active:



However, ssh tunnels still failed:



Expressway event logs:



## Condition 2.1: Success Case

CA 1 = Internal CA

CA 2 = Public CA

- Where Expressway core client certificate is signed by CA 1 (internal CA) and includes, Client/Server ECU both.
- Expressway core server certificate is signed by CA 2 public CA and includes Server ECU only.
- Expressway Edge Server certificate is signed by CA 2 public CA and includes Server ECU only.
- Expressway Edge client certificate is signed by CA 2 public CA and includes Server ECU only.

Exp C x15.5

```

Client cert = Client/Server EKU by CA1
Server cert = Server EKU by CA2

```

-----UC Zone-----

Exp E x15.5

```

Client cert = Server EKU by CA2
Server cert = Server EKU by CA2

```

This condition is a success case. Irrespective of whether the EKU check mode is ON/OFF, the Unified Communication zone and SSH tunnel both become active. MRA clients work.

It does not matter whether the Expressway Edge EKU check is OFF or ON. The Expressway core client certificate contains client EKU:

```

10.106.80.31 - PuTTY
xconfiguration SIP TLS Certificate ExtendedKeyUsage Checking Mode: Off
OK

```

```

10.106.80.31 - PuTTY
xConfiguration SIP TLS Certificate ExtendedKeyUsage Checking Mode: "On"
OK

```

SSH Tunnels on Expressway core Active:

**CISCO Cisco Expressway-C**

Status > System > Configuration > Applications > Users > Maintenance >

**Unified Communications SSH tunnels status**

**Warning:** The CSRF Protection status is automatically enabled after the software upgrade. We recommend keeping CSRF protection Enabled. To change it, please log in to the CLI.

Target	Domain	Status	Tunnel Created
smartslave.vikduttia.com	tomcat.com	Active	29/03/2026 07:21:27
smartslave.vikduttia.com	555.federation.com	Active	29/03/2026 07:19:26

SSH Tunnels on Expressway Edge Active:

**CISCO Cisco Expressway-E**

Status > System > Configuration > Applications > Users > Maintenance >

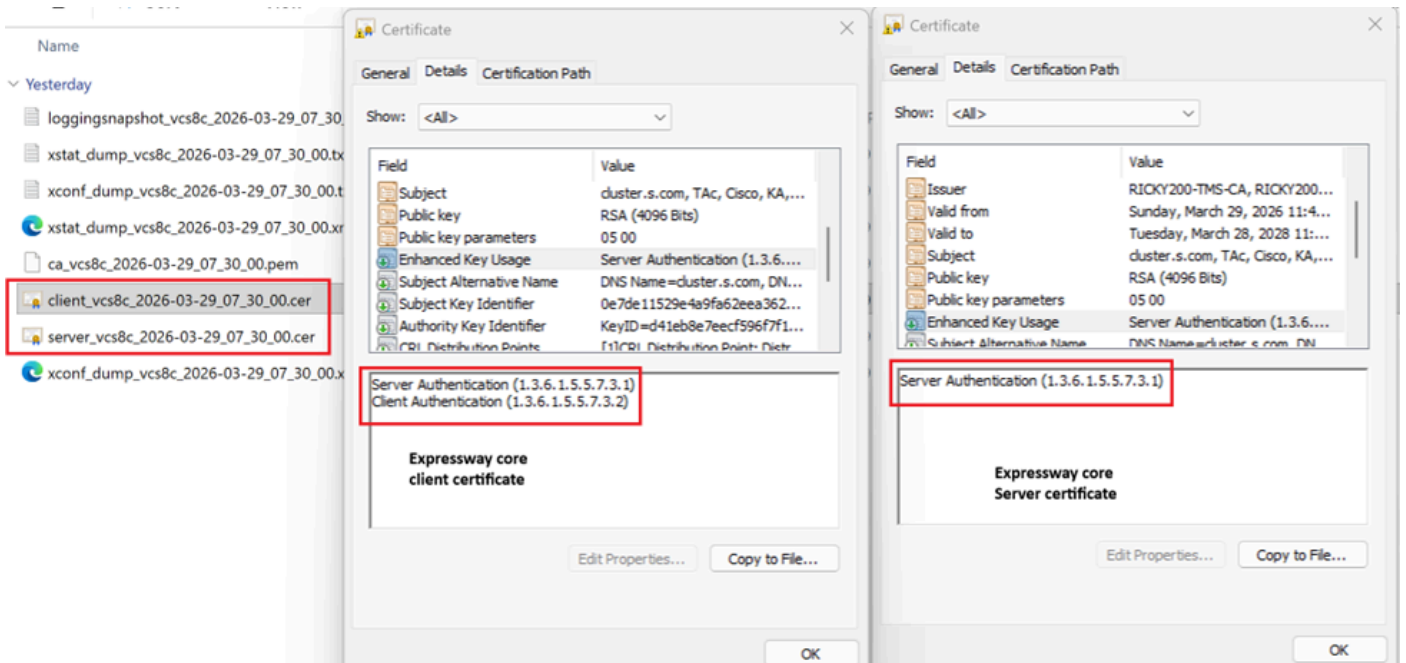
**Unified Communications SSH tunnels status**

**Warning:** The CSRF Protection status is automatically enabled after the software upgrade. We recommend keeping CSRF protection Enabled. To change it, please log in to the CLI.

Target	Domain	Status	Tunnel Created
vcs8c	tomcat.com	Active	29/03/2026 07:21:27
vcs8c	555.federation.com	Active	29/03/2026 07:19:26

Unified Communication MRA Zone status Active:

- Expressway-Core client certificate has Server ECU and Client ECU.
- Expressway Core Server certificate has Server ECU only.



MRA Client logs in and registered:

The screenshot shows the Cisco Jabber interface with a 'Connection Status' window open. The window title is 'Cisco Jabber' and the version is 'Version 12.6.1 (284405)'. The status is as follows:

Component	Status	Protocol	Address	Device	Line
Softphone	Connected	SIP	10.106.79.162 (CCMCIP - Expressway) (IPv4)	CSFHanu	7777
Deskphone	Not connected	CTI	(CTI) (Unknown)		
Outlook address book	Last connection successful	MAPI	Outlook (Unknown)		
Directory	Last connection successful				

The IP address '10.106.79.162 (CCMCIP - Expressway) (IPv4)' and the device name 'CSFHanu' are highlighted with a red box in the original image.

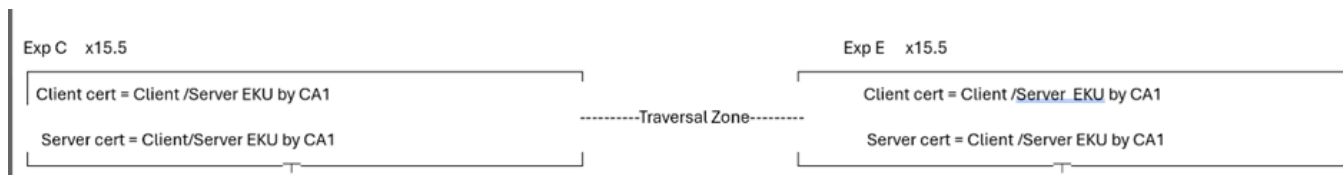


**Note:** Compare and take note of EKUs present in certificates for MRA and WebRTC proxy to work. It is a comparison of working and non-working deployment.

### Condition 3: Signs all Certificates with Private CA

CA 1 = Internal CA

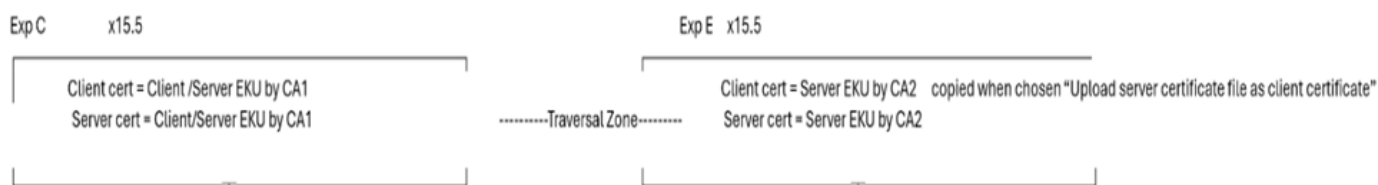
## CA 2 = Public CA



In condition 3, all certificates are signed by internal CA (CA1) .

- When Expressway-E sends out a TLS connection, CA 1 root/intermediate need to be exchanged with far-end entity. If far-end does not have capability or does not allow private CA certificate to be uploaded, TLS connection is unsuccessful.
- MRA clients get certificates to accept pop ups if the private certificate is not in the OS trust store.

## Condition 4: Expressway Edge has Public Certificates with Server ECU Only



In Condition 4, the Expressway core client and server certificates are (CA1) internal CA signed and have both client and server ECU present. The Expressway E server certificate is public CA signed and has server ECU only. Server certificate is copied to client certificate store choosing **Upload server certificate file as client certificate**.

In Condition 4, when TLS connection is made to far-end, if Expressway -E sends a TLS client hello, far-end has to disable client ECU check (as client certificate does not have client auth ECU) else TLS connection is unsuccessful.

There can be many more conditions or scenarios in the field based on user deployment and use cases and all cannot be covered due to my limited stream of thought. However, points to remember are:

- # IF Expressway becomes a client during TLS handshake, the client certificate is presented to peers.
- #IF Expressway becomes Server during TLS handshake; the server certificate is presented to peer.

This reasoning has been established with these tests cases.

## Scenario 1

For this scenario, Expressway presents client certificate during the MTLS handshake with Webex.

A video call to Webex meeting:

Sample call flow Jabber -à CUCM -à Exp Core --à Exp Edge --à Webex

10.106.80.31= Expressway Edge

163.129.37.33 = Webex

```
2026-03-24T11:54:26.106+00:00 smartslave tvcs: UTCTime="2026-03-24 11:54:26,106"  
Module="network.sip" Level="DEBUG": Action="Sent" Local-ip="10.106.80.31" Local-port="25002"  
Dst-ip="163.129.37.33" Dst-port="5061"
```

Expressway Edge has Client certificate with this serial number  
(2f0000004c869c77c8981becde00000000004c).

Expressway Edge sends out client hello to ‘Webex during TLS negotiation, then sends out client certificate.

Serial number 2f0000004c869c77c8981becde00000000004c:

1. Expressway Edge sends out client hello (pkt= 13699) to ‘Webex during mTLS negotiation.
2. Webex sends a server hello to Expressway Edge (pkt=13701).
3. Webex sends its certificate to Expressway Edge (pkt=13711).
4. Webex requests Expressway edge certificate “CertificateRequest ” (pkt=13715).
5. Expressway Edge sends its certificate to Webex (pkt=13718).

(screenshot)

```

13698 2026-03-24 17:25:20.911700 10.106.00.31 163.129.37.32 TCP 66 25003 + 5061 [ACK] Seq=1 Ack=1 Win=64512 Len=0 TSval=840949379 TSecr=3608271268
13699 2026-03-24 17:25:20.912773 10.106.00.31 163.129.37.32 TLSv1.2 583 Client Hello
13700 2026-03-24 17:25:20.956852 163.129.37.32 10.106.00.31 TCP 66 5061 + 25003 [ACK] Seq=1 Ack=518 Win=28544 Len=0 TSval=3608271312 TSecr=840949380
13701 2026-03-24 17:25:20.956925 163.129.37.32 10.106.00.31 TLSv1.2 156 Server Hello
13702 2026-03-24 17:25:20.956963 10.106.00.31 163.129.37.32 TCP 66 25003 + 5061 [ACK] Seq=518 Ack=91 Win=64512 Len=0 TSval=840949424 TSecr=3608271313
13703 2026-03-24 17:25:20.957044 163.129.37.32 10.106.00.31 TCP 1300 5061 + 25003 [ACK] Seq=91 Ack=518 Win=28544 Len=1242 TSval=3608271313 TSecr=840949380 [TCP PDU reassembled in 13711]
13704 2026-03-24 17:25:20.957049 10.106.00.31 163.129.37.32 TCP 66 25003 + 5061 [ACK] Seq=518 Ack=1333 Win=67584 Len=0 TSval=840949425 TSecr=3608271313
13705 2026-03-24 17:25:20.957163 163.129.37.32 10.106.00.31 TCP 1300 5061 + 25003 [ACK] Seq=1333 Ack=518 Win=28544 Len=1242 TSval=3608271313 TSecr=840949380 [TCP PDU reassembled in 13711]
13706 2026-03-24 17:25:20.957170 10.106.00.31 163.129.37.32 TCP 66 25003 + 5061 [ACK] Seq=518 Ack=2575 Win=70656 Len=0 TSval=840949425 TSecr=3608271313
13707 2026-03-24 17:25:20.957175 163.129.37.32 10.106.00.31 TCP 1300 5061 + 25003 [ACK] Seq=2575 Ack=518 Win=28544 Len=1242 TSval=3608271313 TSecr=840949380 [TCP PDU reassembled in 13711]
13708 2026-03-24 17:25:20.957179 10.106.00.31 163.129.37.32 TCP 66 25003 + 5061 [ACK] Seq=518 Ack=3817 Win=72704 Len=0 TSval=840949425 TSecr=3608271313
13709 2026-03-24 17:25:20.957184 163.129.37.32 10.106.00.31 TCP 1300 5061 + 25003 [ACK] Seq=3817 Ack=518 Win=28544 Len=1242 TSval=3608271313 TSecr=840949380 [TCP PDU reassembled in 13711]
13710 2026-03-24 17:25:20.957188 10.106.00.31 163.129.37.32 TCP 66 25003 + 5061 [ACK] Seq=518 Ack=5059 Win=71680 Len=0 TSval=840949425 TSecr=3608271313
13711 2026-03-24 17:25:20.957193 163.129.37.32 10.106.00.31 TLSv1.2 378 Certificate
13712 2026-03-24 17:25:20.957215 10.106.00.31 163.129.37.32 TCP 66 25003 + 5061 [ACK] Seq=518 Ack=5371 Win=72704 Len=0 TSval=840949425 TSecr=3608271313
13713 2026-03-24 17:25:20.958101 163.129.37.32 10.106.00.31 TLSv1.2 404 Server Key Exchange
13714 2026-03-24 17:25:20.958110 10.106.00.31 163.129.37.32 TCP 66 25003 + 5061 [ACK] Seq=518 Ack=5709 Win=73728 Len=0 TSval=840949426 TSecr=3608271314
13715 2026-03-24 17:25:20.958341 163.129.37.32 10.106.00.31 TLSv1.2 124 Certificate Request, Server Hello Done
13716 2026-03-24 17:25:20.958350 10.106.00.31 163.129.37.32 TCP 66 25003 + 5061 [ACK] Seq=518 Ack=5767 Win=73728 Len=0 TSval=840949426 TSecr=3608271315
13717 2026-03-24 17:25:20.967687 10.106.00.31 163.129.37.32 TCP 2550 25003 + 5061 [PSH, ACK] Seq=518 Ack=5767 Win=73728 Len=2484 TSval=840949435 TSecr=3608271315 [TCP PDU reassembled in 13718]
13718 2026-03-24 17:25:20.967707 10.106.00.31 163.129.37.32 TLSv1.2 1170 Certificate, Client Key Exchange, Certificate Verify, Change Cipher Spec, Encrypted Handshake Message
13719 2026-03-24 17:25:20.971327 10.106.00.31 163.129.37.32 TCP 66 5061 + 25003 [ACK] Seq=5767 Ack=3802 Win=26112 Len=0 TSval=3608271365 TSecr=840949435
13720 2026-03-24 17:25:21.008884 163.129.37.32 10.106.00.31 TCP 66 5061 + 25003 [ACK] Seq=5767 Ack=3802 Win=26112 Len=0 TSval=3608271365 TSecr=840949435
13721 2026-03-24 17:25:21.010881 163.129.37.32 10.106.00.31 TLSv1.2 72 Change Cipher Spec

```

Length: 2936

Certificates Length: 2933

Certificates (2933 bytes)

Certificate Length: 2834

```

Certificate [-]: 308207ee308206d6e00302010201132f0000004c869c77c8981becde00000000004c300006092a86486f700101060500304f31133011060a0992268993f22c6401191603636fd3118301004
  signedCertificate
    version: v3 (2)
    serialNumber: 0x2f0000004c869c77c8981becde0000000004c
    signature (sha256withRSAEncryption)
    issuer: rdnsSequence (0)
    rdnsSequence: 3 items (id-at-commonName=bgluc1ab-WIN-DC-01-CA,dc=bgluc1ab,dc=com)
      rdnsSequence item: 1 item (dc=com)
      rdnsSequence item: 1 item (dc=bgluc1ab)
      rdnsSequence item: 1 item (id-at-commonName=bgluc1ab-WIN-DC-01-CA)
    validity
      notBefore: utcTime (0)
      notAfter: utcTime (0)
    subject: rdnsSequence (0)

```

```

0040 30 4f 31 13 30 11 06 0a 09 92 2f
0050 01 19 16 03 43 6f 6d 31 18 30 31
0060 09 93 f2 2c 64 01 19 16 08 42 61
0070 62 31 1e 30 1c 06 03 55 04 03 11
0080 03 6c 61 62 2d 57 49 4e 26 44 41
0090 41 30 1e 17 0d 32 36 30 33 32 34
0100 37 5a 17 0d 32 38 30 33 32 33 33
0110 5a 30 5e 31 0d 30 09 06 03 55 04
0120 31 0d 30 09 06 03 55 04 08 13 0c
0130 0a 06 03 55 04 07 13 03 42 6c 71
0140 03 55 04 0a 13 05 63 69 73 63 6f
0150 03 55 04 0b 13 03 74 61 63 31 1f
0160 04 03 13 0d 63 6c 75 73 74 65 77
0170 6d 30 82 02 21 30 0d 06 09 2a 86
0180 02 01 05 00 03 82 02 0f 00 30 82
0190 01 00 c8 a6 f3 0b 06 f2 44 21 5f
01a0 5b 7d 9f 25 b2 ad 7f 51 25 47 28
01b0 24 87 88 a9 3d 64 38 73 ff eb 4e
01c0 ab de 92 20 62 ec ad 92 32 98 de
01d0 78 cb 2e 53 49 5f 37 a8 88 fe 41
01e0 1d f0 02 87 52 47 a0 63 2e 0f af
01f0

```

Client certificate from Expressway Edge:

The screenshot shows a file explorer window with a list of files. The file 'client\_smartslave\_2026-03-24\_11\_55\_47.pem' is selected. A 'Certificate' dialog box is open, displaying the following details:

Field	Value
Version	V3
Serial number	2f0000004c869c77c8981becde0000000004c
Signature algorithm	sha256RSA
Signature hash algorithm	sha256
Issuer	bgluc1ab-WIN-DC-01-CA, bglu...
Valid from	Tuesday, March 24, 2026 4:5...
Valid to	Thursday, March 23, 2028 4:5...
Subject	cluster.s.com, tac.riscn.BR

The serial number field is highlighted with a red box. Below the dialog box, the serial number '2f0000004c869c77c8981becde0000000004c' is also highlighted with a red box.

## Scenario 2

Expressway becomes a server entity during the mTLS handshake and presents its server certificate:

Where Expressway presents Server certificate, Expressway has a secure Neighbor zone over 5061 with verify name ON.

Secure neighbor zone between Expressway node x15.5 and Expressway node x8.11.4:

10.106.80.15 (x8.11.4) sends a client hello to 10.106.80.16 (x15.5) (pkt=736)

10.106.80.16 sends a server hello to 10.106.80.15 (pkt=738)

10.106.80.16 (x15.5) presents its server cert during TLS handshake (pkt=742) and requests client's cert

10.106.80.15 (x8.11.4) sends client certificate (pkt=744)

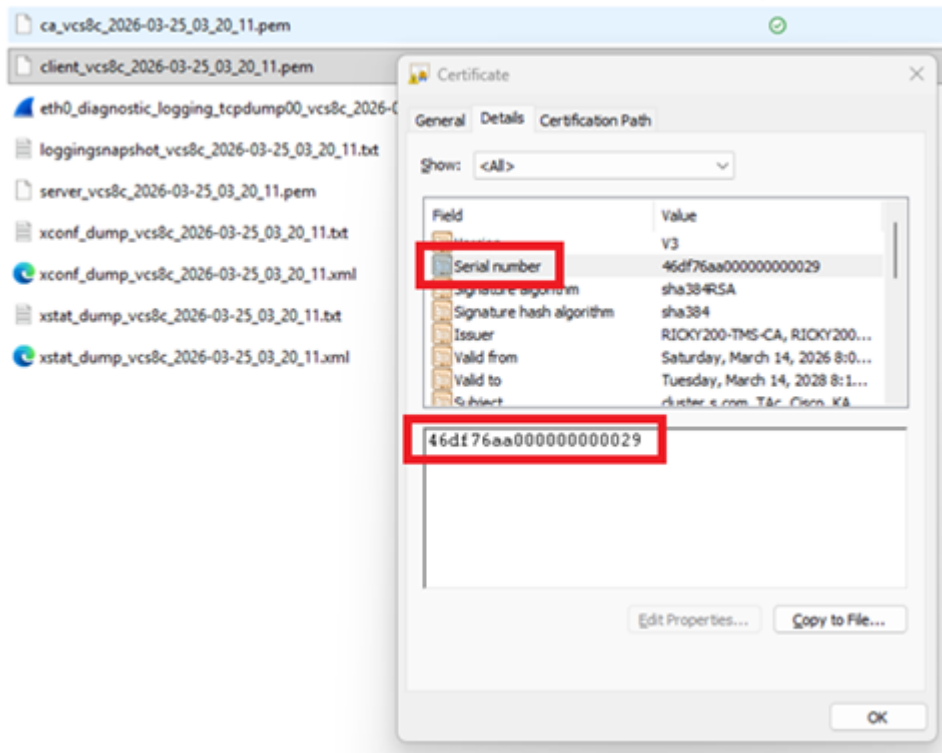
The screenshot displays a Wireshark network traffic capture of a TLS handshake between two hosts: 10.106.80.15 (client) and 10.106.80.16 (server). The capture shows several packets:

- 736: Client Hello from 10.106.80.15 to 10.106.80.16 (pkt=736).
- 738: Server Hello from 10.106.80.16 to 10.106.80.15 (pkt=738).
- 742: Certificate, Server Key Exchange, Certificate Request, Server Hello Done from 10.106.80.16 to 10.106.80.15 (pkt=742).
- 744: Certificate, Client Key Exchange, Certificate Verify, Change Cipher Spec, Encrypted Handshake Message from 10.106.80.15 to 10.106.80.16 (pkt=744).

The detailed view of the Certificate field (packet 742) shows the following structure:

- Handshake Protocol: Certificate
- Handshake Type: Certificate (11)
- Length: 2919
- Certificates Length: 2916
- signedCertificate
  - version: v3 (2)
  - serialNumber: 0x46df76aa0000000029
  - signature (sha384withRSAEncryption)
    - algorithm: sha384withRSAEncryption
  - Issuer: rdnSequence (0)
  - rdnSequence: 3 items (id-at-commonName=RICKY200-THIS-CA,dc=RICKY200,dc=com)
  - validity

This screenshot shows server certificate as serial number matches:



Test case 3: MRA Client is provisioned for login and the workflow includes traffic server certificate verification between Expressway Core and CUCM.

10.106.80.16 = Expressway Core x15.5

10.106.80.38 = CUCM

- Exp C 16 sends a client hello on 6972 TFTP.
- Exp C 16 sends a client certificate during the TLS handshake.

