Troubleshooting

This chapter explains the following:

SIP TLS Negotiation Failures on Neighbor and Traversal Zones

If **TLS verify mode** is enabled, the neighbor system's FQDN or IP address, as specified in the **Peer address** field of the zone's configuration, is used to verify against the certificate holder's name in the X.509 certificate presented by that system. (The name must be in the SAN attribute of the certificate.) The certificate itself must also be valid and signed by a trusted certificate authority.

So when certificates have been generated with peer or cluster FQDNs, ensure that the zone's **Peer address** fields are configured with FQDNs rather than IP addresses.

Certificates with Key Length of 8192 Bits

SIP TLS zones may fail to become active if certificates use a key length of 8192 bits. We recommend using certificates with a key length of 4096 bits.

Service Failures when Using Mobile and Remote Access

Unified Communications mobile and remote access services can fail due to certificate errors if you upload a private key file that does not contain a trailing newline character.

Ensure that the private key file contains a trailing newline character.

Issues with SSH Failures and Unsupported OIDs

If you experience unknown ssh failures such as ssh tunnels failing to establish, please verify there are no unknown OIDs in the certificate. This can be done by checking that there are no undecoded numerical entries in the CN of the Issuer & Subject fields (from the GUI: **Maintenance** > **Security** > **Server Certificate** > **Show(decoded)** or from the console: `'openssl x509 -text -noout -in /tandberg/persistent/certs/server.pem')`

```
Invalid

subject=CN=blahdeblah,OU=IT

Security,O=BigBang,L=Washington,ST=District of

Columbia,C=US,1.3.6.1.4.1.6449.1.2.1.5.1 = #060C2B06010401B2310102010501

Valid

subject=CN=blahdeblah,OU=IT

Security,O=BigBang,L=Washington,ST=District of

Columbia,C=US,jurisdictionOfIncorporationLocalityName=Dover
```

CUCM Cipher Interop with Expressway

Servers during Transport Layer Security (TLS) handshake send Rivest Shamir Adleman (RSA)/Elliptic Curve Digital Signature Algorithm (ECDSA) ciphers. Expressway, as a client, can accept these ciphers.

Note Fresh install of Expressway comes default with ECDSA ciphers.

Expressway can negotiate an ECDSA cipher request.

Remember

- Certificate cipher with RSA, UCM sends either a *CallManager* or a *Tomcat* certificate.
- Certificate cipher with ECDSA, UCM sends either a *CallMananager-ECDSA* or a *Tomcat-ECDSA* certificate.
- Users must sequentially upload, to Expressway-C, signed Unified Call Manager (UCM) certificates as trusted Certificate Authority (CA) to verify the received certificate from UCM.

Reference Information

- **For Cipher Configuration:** Configuring ECDSA followed by RSA ciphers.

  ```
  ECDHE-ECDSA-AES128-GCM-SHAdefault:ECDHE-ECDSA-AES128-SHAdefault:ECDHE-ECDSA-
  AES128-SHA:ECDHE-ECDSA-AESdefault-GCM-SHA384:ECDHE-ECDSA-AESdefault-
  AES128-SHA:ECDHE-ECDSA-AESdefault-GCM-SHA384:ECDHE-ECDSA-AESdefault-
  GCM-SHA384
  ```

- **For Configuration in Expressway**

Add the below Ciphers under **Maintenance** > **Security** > **Ciphers**.

Note The following cipher change is required to send ECDSA as a high preference.

```
EECDH:EDH:HIGH:-
AES256+SHA:!MEDIUM:!LOW:!3DES:!MD5:!PSK:!eNULL:!aNULL:!aDH
```

Troubleshooting