Cisco Secure SRST Configuration Example

Contents

Introduction

This document provides a sample configuration for Cisco Secure Survivable Remote Site Telephony (SRST).

Secure Cisco IP phones that are located at remote sites and that are attached to gateway routers can communicate securely over the WAN with Cisco CallManager. But if the WAN link or Cisco CallManager goes down, all communication through the remote phones becomes nonsecure. In order to overcome this situation, gateway routers can now function in secure SRST mode, which activates when

the WAN link or Cisco CallManager goes down. When the WAN link or Cisco CallManager is restored, Cisco CallManager resumes secure call-handling capabilities.

Secure SRST provides new SRST security features such as authentication, integrity, and media encryption. Authentication provides assurance to one party that another party is whom it claims to be. Integrity provides assurance that the given data is not altered between the entities. Encryption implies confidentiality, which means that no one can read the data except the intended recipient. These security features allow privacy for SRST voice calls and protect against voice security violations and identity theft.

SRST security is achieved when:

- End devices are authenticated with certificates.
- Signaling is authenticated and encrypted with Transport Layer Security (TLS) for TCP.
- A secure media path is encrypted with Secure Real-Time Transport Protocol (SRTP).
- Certificates are generated and distributed by a certificate authority (CA).

Prerequisites

Requirements

Ensure that you meet these requirements before you attempt this configuration:

Public Key Infrastructure Requirements

- Set the clock, either manually or with Network Time Protocol (NTP). This ensures synchronicity with Cisco CallManager.
- Enable the IP HTTP server (Cisco IOS® processor) with the **ip http server** command, if not already enabled. Refer to Cisco IOS Certificate Server for more information on public key infrastructure (PKI) deployment.
- If the certificate server is part of your startup configuration, you can potentially see these messages during the boot procedure:

```
% Failed to find Certificate Server's trustpoint at startup

% Failed to find Certificate Server's cert.
```

  These messages are informational messages and indicate a temporary inability to configure the certificate server, because the startup configuration is not fully parsed yet. The messages are useful in order to debug, in case the startup configuration is corrupted. You can verify the status of the certificate server after the boot procedure with the **show crypto pki server** command.

SRST Requirements

- Secure SRST services cannot be enrolled while SRST is active. Therefore disable SRST with the

no call-manager-fallback command.

- Refer to Media and Signaling Authentication and Encryption Feature for Cisco IOS MGCP Gateways for a list of supported Cisco IP phones, routers, network modules, and codecs for secure SRST.

- Refer to Cisco Unified SRST 4.0 Supported Firmware, Platforms, Memory, and Voice Products for the most up-to-date information about the maximum number of Cisco IP phones, the maximum number of directory numbers (DNs) or virtual voice ports, and the memory requirements for Cisco SRST.

Components Used

The information in this document is based on these software and hardware versions:

- Secure Cisco IP phones supported in secure SRST must have certificates installed and encryption enabled.

- The SRST router must have a certificate. Acertificate can be generated by a third party or by the Cisco IOS certificate authority (CA). The Cisco IOS CA can run on the same gateway as SRST.

- Certificate trust lists (CTLs) on Cisco CallManager must be enabled. For complete instructions, refer to the *Configuring Secure IP Telephony Calls* section of Media and Signaling Authentication and Encryption Feature for Cisco IOS MGCP Gateways.

- Cisco CallManager 4.1(2) or later must be installed and must support security mode (authenticate and encryption mode).

- Gateway routers that run secure SRST must support voice- and security-enabled Cisco IOS images (a “k9” cryptographic software image). Two images are supported: Advanced IP Services, which includes a number of advanced security features, and Advanced Enterprise Services, which includes full Cisco IOS software.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Conventions

Refer to Cisco Technical Tips Conventions for more information on document conventions.

Restrictions

General Restrictions

- Cryptographic software features (“k9”) are under export controls. This product contains cryptographic features and is subject to United States and local country laws governing import, export, transfer, and use. Delivery of Cisco cryptographic products does not imply third-party authority to import, export, distribute, or use encryption. Importers, exporters, distributors, and

users are responsible for compliance with U.S. and local country laws. By using this product you agree to comply with applicable laws and regulations. If you are unable to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products can be found at:

http://www.cisco.com/wwl/export/crypto/tool/

If you require further assistance, please contact us by sending e-mail to export@cisco.com.

- When a Secure Real-Time Transport Protocol (SRTP) encrypted call is made between Cisco IP phone endpoints or from a Cisco IP phone to a gateway endpoint, a lock icon is displayed on the IP phones. The lock indicates security only for the IP leg of the call. Security of the PSTN leg is not implied.
- Secure SRST is supported only within the scope of a single router.

Unsupported Features and Software in Secure SRST Mode

- Cisco CallManager versions prior to 4.1(2)
- Secure music on hold (MoH)
- Secure transcoding or conferencing
- Secure H.323 or SIP
- Hot Standby Router Protocol (HSRP)

Supported Calls in Secure SRST Mode

Only voice calls are supported in secure SRST mode. Specifically, these voice calls are supported:

- Basic call
- Call forward (busy, no-answer, all)
- Shared line (IP phones)
- Call transfer (consult and blind)
- Hold and resume

Background Information

Cisco IP Phones Clear-Text Fallback During SRST

Cisco SRST versions earlier than Cisco IOS Software Release 12.3(14)T are not able to support secure connections or have security enabled. If an SRST router is not capable of secure SRST as a fallback

mode—that is, it is not able to complete a TLS handshake with Cisco CallManager—its certificate is not added to the configuration file of the Cisco IP phone. The absence of an SRST router certificate causes the Cisco IP phone to use nonsecure (clear-text) communication when in SRST fallback mode. The capability to detect and fallback in clear-text mode is built into Cisco IP phone firmware. Refer to Media and Signaling Authentication and Encryption Feature for Cisco IOS MGCP Gateways for more information on clear-text mode.

SRST Routers and the TLS Protocol

Transport Layer Security (TLS) Version 1.0 provides secure TCP channels between Cisco IP phones, secure SRST routers, and Cisco CallManager. The TLS process begins when the Cisco IP phone establishes a TLS connection when it registers with Cisco CallManager. If Cisco CallManager is configured to fallback to SRST, the TLS connection between the Cisco IP phones and the secure SRST router is also established. If the WAN link or Cisco CallManager fails, call control reverts to the SRST router.

SRST Routers and PKI

The transfer of certificates between an SRST router and Cisco CallManager is mandatory for secure SRST functionality. Public key infrastructure (PKI) commands are used to generate, import, and export the certificates for secure SRST. The certificates for each supported Cisco IP phone are shown in this table.

Table 1 - Supported Cisco IP Phones and Certificates

Cisco IP Phone 7940	Cisco IP Phone 7960	Cisco IP Phone 7970
The phone receives the locally significant certificate (LSC) from the Certificate Authority Proxy Function (CAPF) in Distinguished Encoding Rules (DER) format. Certificate filename: 59fe77ccd.0 The filename can change based on the CAPF certificate subject name and the	The phone receives the locally significant certificate (LSC) from the Certificate Authority Proxy Function (CAPF) in Distinguished Encoding Rules (DER) format. Certificate filename: 59fe77ccd.0 The filename can change based on the CAPF certificate subject name and the	The phone contains a manufacturing installed certificate (MIC) used for device authentication. If the Cisco 7970 implements MIC, two public certificate files are needed: • CiscoCA.pem (Cisco Root CA, used to authenticate the certificate) • a69d2e04.0, in Privacy

CAPF certificate issuer. If Cisco CallManager uses a third-party certificate provider, there can be multiple .0 files (from two to ten). Each .0 certificate file must be imported individually during the configuration. Manual enrollment is supported only.	CAPF certificate issuer. If Cisco CallManager uses a third-party certificate provider, there can be multiple .0 files (from two to ten). Each .0 certificate file must be imported individually during the configuration. Manual enrollment is supported only.	Enhanced Mail (PEM) format If Cisco CallManager uses a third-party certificate provider, there can be multiple .0 files (from two to ten). Each .0 certificate file must be imported individually during the configuration. Manual enrollment is supported only.

Cisco IOS Credentials Server on Secure SRST Routers

Secure SRST introduces a credentials server that runs on a secure SRST router. When the client, Cisco CallManager, requests a certificate through the TLS channel, the credentials server provides the SRST router certificate to Cisco CallManager. Cisco CallManager inserts the SRST router certificate in the Cisco IP phone configuration file and downloads the configuration files to the phones. The secure Cisco IP phone uses the certificate to authenticate the SRST router during fallback operations. The credentials service runs on default TCP port 2445.

Five new Cisco IOS commands configure the credentials server in call-manager-fallback mode and provide server debugging and verification capabilities:

- credentials
- debug credentials
- ip source-address (credentials)
- show credentials
- trustpoint (credentials)

Establishment of Secure SRST to the Cisco IP Phone

This diagram shows the interworking of the credentials server on the SRST router, Cisco CallManager, and the Cisco IP phone, which establishes secure SRST to the Cisco IP phone.

1. The Cisco IP phone configures DHCP and gets the TFTP server address.

2. The Cisco IP phone retrieves a CTL file from the TFTP server. The CTL file contains the certificates that the phone is meant to trust.

3. The Cisco IP phone opens a Transport Layer Security (TLS) protocol channel and registers to Cisco CallManager.

Cisco CallManager exports secure SRST router information and the SRST router certificate to the Cisco IP phone. The phone places the certificate into its configuration. Once the phone has the SRST certificate, the SRST router is considered secure.

If the Cisco IP phone is configured as “authenticated” or “encrypted” and Cisco CallManager is configured in mixed mode, the phone looks for an SRST certificate in its configuration file. If it finds an SRST certificate, it opens a standby TLS connection to the default port. The default port is the Cisco IP phone TCP port plus 443, which is port 2443 on an SRST router. The connection to the SRST router happens automatically, as long as there is not a secondary Cisco CallManager and the SRST is configured as the backup device.

Cisco CallManager must be configured in mixed mode, which is its secure mode.

In case of WAN failure, the Cisco IP phone starts SRST registration. The Cisco IP phone registers with the SRST router at the default port for secure communications.

Configure

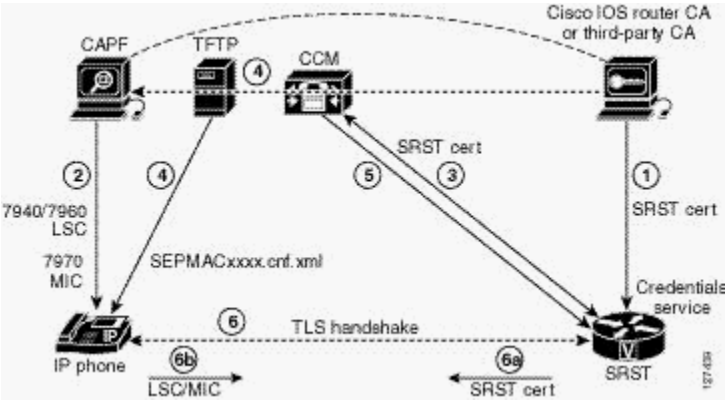
In this section, you are presented with the information to configure the features described in this document.

Note: Use the Command Lookup Tool (registered customers only) in order to obtain more information on the commands used in this section.

The secure SRST router and the Cisco IP phones must request mutual authentication during the TLS handshake. The TLS handshake occurs when the phone registers with the SRST router, either before or after the WAN link fails. The configuration example does not include the use of a third-party CA. It assumes the use of the Cisco IOS certificate server to generate your certificates.

Network Diagram

This document uses the network setup shown in this diagram. The diagram illustrates the process of secure SRST authentication and encryption.



1. The CA server, whether it is a Cisco IOS router CA or a third-party CA, issues a device certificate to the SRST gateway, which enables credentials service. Optionally, the certificate can be self-generated by the SRST router with a Cisco IOS CA server.

   The CA router is the ultimate trustpoint for the Certificate Authority Proxy Function (CAPF). Refer to Cisco CallManager Security Guide for more information on CAPF.

2. The CAPF is a process where supported devices can request a locally significant certificate (LSC). The CAPF utility generates a key pair and certificate that is specific for CAPF, copies this certificate to all Cisco CallManager servers in the cluster, and provides the LSC to the Cisco IP phone. An LSC is required for Cisco IP phones that do not have a manufacturing installed certificate (MIC). The Cisco 7970 is equipped with a MIC and therefore does not need to go through the CAPF process.

3. Cisco CallManager requests the SRST certificate from the credentials server, and the credentials server responds with the certificate.

4. For each device, Cisco CallManager uses the TFTP process and inserts the certificate into the SEPMACxxxx.cnf.xml configuration file of the Cisco IP phone.

5. Cisco CallManager provides the PEM format files that contain phone certificate information to the SRST router. The PEM files are provided to the SRST router manually. When the SRST router has the PEM files, it can authenticate the IP phone and validate the issuer of the IP phone certificate during the TLS handshake.

6. The TLS handshake occurs, certificates are exchanged, and mutual authentication and registration occurs between the Cisco IP phone and the SRST router.

   a. The SRST router sends its certificate, and the phone validates the certificate to the certificate that it received from Cisco CallManager in Step 4.

   b. The Cisco IP phone provides the SRST router with the LSC or MIC, and the router validates the LSC or MIC with the PEM format files that it received in Step 5.

   Note: The media is encrypted automatically once the phone and router certificates are exchanged

and the TLS connection is established with the SRST router.

Before You Configure

Cisco CallManager

Complete these steps:

1. Once the credentials service runs on the SRST router, an SRST reference to Cisco CallManager needs to be added, because Cisco CallManager connects to the SRST router for its device certificate. Refer to the *Survivable Remote Site Telephony Configuration* section of Cisco CallManager Administration Guide, Release 4.1(2) for complete information on how to add SRST to Cisco CallManager.

2. SRST fallback must be configured on Cisco CallManager. In order to do this assign the device pool to SRST. Refer to the *Device Pool Configuration* section of Cisco CallManager Administration Guide, Release 4.1(2) for complete information on how to add a device pool to Cisco CallManager.

3. Certificate Authority Proxy Function (CAPF) must be configured on Cisco CallManager. The CAPF process allows supported devices, such as Cisco CallManager, to request LSC certificates from Cisco IP phones. The CAPF utility generates a key pair and certificate that is specific for CAPF, and the utility copies this certificate to all Cisco CallManager servers in the cluster. Refer to Cisco IP Phone Authentication and Encryption for Cisco CallManager 4.0(1) for complete instructions on how to configure CAPF in Cisco CallManager.

Security Cautions

- The **grant auto** command allows certificates to be issued and must be activated when you define your root CA. However, for security reasons, the **grant auto** command must not remain active and must be disabled after certificates are issued.

- A security best practice is to protect the credentials service port with control plane policing. Control plane policing protects the gateway and maintains packet forwarding and protocol states despite a heavy traffic load. Refer to Control Plane Policing for more information on control planes. A configuration example also appears in the Configuration 2 section of this document.

Configurations

This document uses these configurations:

- Configuration 1—Configure your router according to this **show running-config** example.

- Configuration 2—A security best practice is to protect the credentials service port with control plane policing. If you use control plane policing, configure your router according to this partial **show running-config** example.

Configuration 1
Router#**show running-config**

```
.
.
.

!--- Define Cisco CallManager.

ccm-manager fallback-mgcp
ccm-manager mgcp
ccm-manager music-on-hold
ccm-manager config server 10.1.1.13
ccm-manager config
!

!--- Define root CA.
!--- For SRST routers to provide secure communications, there must be a
!--- CA server that issues the device certificate in the network.
!--- The CA server can be a third-party CA or one generated from a
!--- Cisco IOS certificate server. The Cisco IOS certificate server
!--- provides a certificate generation option to users who do not
!--- have a third-party CA in their network. The Cisco IOS certificate
!--- can run on the SRST router or on a different Cisco IOS router.

crypto pki server srstcaserver
 database level complete
 database url nvram
 issuer-name CN=srstcaserver

!
!--- The secure SRST router needs to define a trustpoint. That is,
!--- it must obtain a device certificate from the CA server. The procedure
!--- is called certificate enrollment. Once enrolled, the secure SRST router
!--- can be recognized by Cisco CallManager as a secure SRST router. There
!--- are three options to enroll the secure SRST router to a CA server:
!--- autoenrollment, cut and paste, and TFTP. When the CA server is a
!--- Cisco IOS certificate server, autoenrollment can be used. Otherwise, manual
!--- enrollment is required. Manual enrollment refers to cut and paste or TFTP.
!--- Issue the enrollment URL command for autoenrollment and the
!--- crypto pki authenticate command in order to authenticate the SRST router.
!--- Issue the crypto ca enroll command in order to obtain the SRST router
!--- certificate from the CA.

crypto pki trustpoint srstca
 enrollment url http://10.1.1.22:80
 revocation-check none
!
crypto pki trustpoint srstcaserver
 revocation-check none
 rsakeypair srstcaserver
!

!--- Define the CTL/7970/7960 trustpoint to authenticate secure SRST.
!--- Repeat the enrollment procedure for each phone or PEM file.

crypto pki trustpoint 7970
 enrollment terminal
 revocation-check none
!
crypto pki trustpoint PEM
 enrollment terminal
```

```
 revocation-check none
!
crypto pki trustpoint 7960
 enrollment terminal
 revocation-check none
!

!--- This is the SRST router device certificate.

crypto pki certificate chain srstca
 certificate 02
  308201AD 30820116 A0030201 02020102 300D0609 2A864886 F70D0101 04050030
  17311530 13060355 0403130C 73727374 63617365 72766572 301E170D 30343034
  31323139 35323233 5A170D30 35303431 32313935 3232335A 30343132 300F0603
  55040513 08443042 39453739 43301F06 092A8648 86F70D01 09021612 6A61736F
  32363931 2E636973 636F2E63 6F6D305C 300D0609 2A864886 F70D0101 01050003
  4B003048 024100D7 0CC354FB 5F7C1AE7 7A25C3F2 056E0485 22896D36 6CA70C19
  C98F9BAE AE9D1F9B D4BB7A67 F3251174 193BB1A3 12946123 E5C1CCD7 A23E6155
  FA2ED743 3FB8B902 03010001 A330302E 300B0603 551D0F04 04030205 A0301F06
  03551D23 04183016 8014F829 CE97AD60 18D05467 FC293963 C2470691 F9BD300D
  06092A86 4886F70D 01010405 00038181 007EB48E CAE9E1B3 D1E7A185 D7F0D565
  CB84B17B 1151BD78 B3E39763 59EC650E 49371F6D 99CBD267 EB8ADF9D 9E43A5F2
  FB2B18A0 34AF6564 11239473 41478AFC A86E6DA1 AC518E0B 8657CEBB ED2BDE8E
  B586FE67 00C358D4 EFDD8D44 3F423141 C2D331D3 1EE43B6E 6CB29EE7 0B8C2752
  C3AF4A66 BD007348 D013000A EA3C206D CF
  quit
 certificate ca 01
  30820207 30820170 A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  17311530 13060355 0403130C 73727374 63617365 72766572 301E170D 30343034
  31323139 34353136 5A170D30 37303431 32313934 3531365A 30173115 30130603
  55040313 0C737273 74636173 65727665 7230819F 300D0609 2A864886 F70D0101
  01050003 818D0030 81890281 8100C3AF EE1E4BB1 9922A8DA 2BB9DC8E 5B1BD332
  1051C9FE 32A971B3 3C336635 74691954 98E765B1 059E24B6 32154E99 105CA989
  9619993F CC72C525 7357EBAC E6335A32 2AAF9391 99325BFD 9B8355EB C10F8963
  9D8FC222 EE8AC831 71ACD3A7 4E918A8F D5775159 76FBF499 5AD0849D CAA41417
  DD866902 21E5DD03 C37D4B28 0FAB0203 010001A3 63306130 0F060355 1D130101
  FF040530 030101FF 300E0603 551D0F01 01FF0404 03020186 301D0603 551D0E04
  160414F8 29CE97AD 6018D054 67FC2939 63C24706 91F9BD30 1F060355 1D230418
  30168014 F829CE97 AD6018D0 5467FC29 3963C247 0691F9BD 300D0609 2A864886
  F70D0101 04050003 8181007A F71B25F9 73D74552 25DFD03A D8D1338F 6792C805
  47A81019 795B5AAE 035400BB F859DABF 21892B5B E71A8283 08950414 8633A8B2
  C98565A6 C09CA641 88661402 ACC424FD 36F23360 ABFF4C55 BB23C66A C80A3A57
  5EE85FF8 C1B1A540 E818CE6D 58131726 BB060974 4E1A2F4B E6195522 122457F3
  DEDBAAD7 3780136E B112A6
  quit
crypto pki certificate chain srstcaserver
 certificate ca 01
  30820207 30820170 A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  17311530 13060355 0403130C 73727374 63617365 72766572 301E170D 30343034
  31323139 34353136 5A170D30 37303431 32313934 3531365A 30173115 30130603
  55040313 0C737273 74636173 65727665 7230819F 300D0609 2A864886 F70D0101
  01050003 818D0030 81890281 8100C3AF EE1E4BB1 9922A8DA 2BB9DC8E 5B1BD332
  1051C9FE 32A971B3 3C336635 74691954 98E765B1 059E24B6 32154E99 105CA989
  9619993F CC72C525 7357EBAC E6335A32 2AAF9391 99325BFD 9B8355EB C10F8963
  9D8FC222 EE8AC831 71ACD3A7 4E918A8F D5775159 76FBF499 5AD0849D CAA41417
  DD866902 21E5DD03 C37D4B28 0FAB0203 010001A3 63306130 0F060355 1D130101
  FF040530 030101FF 300E0603 551D0F01 01FF0404 03020186 301D0603 551D0E04
  160414F8 29CE97AD 6018D054 67FC2939 63C24706 91F9BD30 1F060355 1D230418
  30168014 F829CE97 AD6018D0 5467FC29 3963C247 0691F9BD 300D0609 2A864886
```

```
  F70D0101 04050003 8181007A F71B25F9 73D74552 25DFD03A D8D1338F 6792C805
  47A81019 795B5AAE 035400BB F859DABF 21892B5B E71A8283 08950414 8633A8B2
  C98565A6 C09CA641 88661402 ACC424FD 36F23360 ABFF4C55 BB23C66A C80A3A57
  5EE85FF8 C1B1A540 E818CE6D 58131726 BB060974 4E1A2F4B E6195522 122457F3
  DEDBAAD7 3780136E B112A6
  quit
crypto pki certificate chain 7970
 certificate ca 353FB24BD70F14A346C1F3A9AC725675
  308203A8 30820290 A0030201 02021035 3FB24BD7 0F14A346 C1F3A9AC 72567530
  0D06092A 864886F7 0D010105 0500302E 31163014 06035504 0A130D43 6973636F
  20537973 74656D73 31143012 06035504 03130B43 41502D52 54502D30 3032301E
  170D3033 31303130 32303138 34395A17 0D323331 30313032 30323733 375A302E
  31163014 06035504 0A130D43 6973636F 20537973 74656D73 31143012 06035504
  03130B43 41502D52 54502D30 30323082 0120300D 06092A86 4886F70D 01010105
  00038201 0D003082 01080282 010100C4 266504AD 7DC3FD8D 65556FA6 308FAE95
  B570263B 575ABD96 1CC8F394 5965D9D0 D8CE02B9 F808CCD6 B7CD8C46 24801878
  57DC4440 A7301DDF E40FB1EF 136212EC C4F3B50F BCAFBB4B CD2E5826 34521B65
  01555FE4 D4206776 03368357 83932638 D6FC953F 3A179E44 67255A73 45C69DEE
  FB4D221B 21D7A3AD 38184171 8FD8C271 42183E65 09461434 736C77CC F380EEBF
  632C7B3F A5F92AA6 A8EF3490 8724A84F 4DAF7FD7 0928F585 764D3558 3C0FE9AF
  1ED8763F A299A802 970004AD 1912D265 7DE335B4 BCB6F789 DC68B9FA C8FDF85E
  8A28AD8F 0F4883C0 77112A47 141DBEE0 948FBE53 FE67B308 D40C8029 87BD790E
  CDAB9FD7 A190C1A2 A462C5F2 4A6E0B02 0103A381 C33081C0 300B0603 551D0F04
  04030201 86300F06 03551D13 0101FF04 05300301 01FF301D 0603551D 0E041604
  1452922B E288EE2E 098A4E7E 702C56A5 9AB4D49B 96306F06 03551D1F 04683066
  3064A062 A060862D 68747470 3A2F2F63 61702D72 74702D30 30322F43 65727445
  6E726F6C 6C2F4341 502D5254 502D3030 322E6372 6C862F66 696C653A 2F2F5C5C
  6361702D 7274702D 3030325C 43657274 456E726F 6C6C5C43 41502D52 54502D30
  30322E63 726C3010 06092B06 01040182 37150104 03020100 300D0609 2A864886
  F70D0101 05050003 82010100 56838CEF C4DA3AD1 EA8FBB15 2FFE6EE5 50A1972B
  D4D7AF1F D298892C D5A2A76B C3462866 13E0E55D DC0C4B92 5AA94B6E 69277F9B
  FC73C697 11266E19 451C0FAB A55E6A28 901A48C5 B9911EE6 348A8920 0AEDE1E0
  B6EA781C FFD97CA4 B03C0E34 0E5B0649 8B0A34C9 B73A654E 09050C1F 4DA53E44
  BF78443D B08C3A41 2EEEB873 78CB8089 34F9D16E 91512F0D 3A8674AD 0991ED1A
  92841E76 36D7740E CB787F11 685B9E9D 0C67E85D AF6D05BA 3488E86D 7E2F7F65
  6918DE0F BD3C7F67 D8A33F70 9C4A596E D9F62B3B 1EDEE854 D5882AD4 3D71F72B
  8FAB7F3C 0B5F0759 D9828F83 954D7BB1 57A638EC 7D72BFF1 8933C16F 760BCA94
  4C5B1931 67947A4F 89A1BDB5
  quit
crypto pki certificate chain PEM
 certificate ca 7612F960153D6F9F4E42202032B72356
  308203A8 30820290 A0030201 02021076 12F96015 3D6F9F4E 42202032 B7235630
  0D06092A 864886F7 0D010105 0500302E 31163014 06035504 0A130D43 6973636F
  20537973 74656D73 31143012 06035504 03130B43 41502D52 54502D30 3031301E
  170D3033 30323036 32333237 31335A17 0D323330 32303632 33333633 345A302E
  31163014 06035504 0A130D43 6973636F 20537973 74656D73 31143012 06035504
  03130B43 41502D52 54502D30 30313082 0120300D 06092A86 4886F70D 01010105
  00038201 0D003082 01080282 010100AC 55BBED18 DE9B8709 FFBC8F2D 509AB83A
  21C1967F DEA7F4B0 969694B7 80CC196A 463DA516 54A28F47 5D903B5F 104A3D54
  A981389B 2FC7AC49 956262B8 1C143038 5345BB2E 273FA7A6 46860573 CE5C998D
  55DE78AA 5A5CFE14 037D695B AC816409 C6211F0B 3BBF09CF B0BBB2D4 AC362F67
  0FD145F1 620852B3 1F07E2F1 AA74F150 367632ED A289E374 AF0C5B78 CE7DFB9F
  C8EBBE54 6ECF4C77 99D6DC04 47476C0F 36E58A3B 6BCB24D7 6B6C84C2 7F61D326
  BE7CB4A6 60CD6579 9E1E3A84 8153B750 5527E865 423BE2B5 CB575453 5AA96093
  58B6A2E4 AA3EF081 C7068EC1 DD1EBDDA 53E6F0D6 E2E0486B 109F1316 78C696A3
  CFBA84CC 7094034F C1EB9F81 931ACB02 0103A381 C33081C0 300B0603 551D0F04
  04030201 86300F06 03551D13 0101FF04 05300301 01FF301D 0603551D 0E041604
  14E917B1 82C71FCF ACA91B6E F4A9269C 70AE05A0 9A306F06 03551D1F 04683066
  3064A062 A060862D 68747470 3A2F2F63 61702D72 74702D30 30312F43 65727445
```

```
  6E726F6C 6C2F4341 502D5254 502D3030 312E6372 6C862F66 696C653A 2F2F5C5C
  6361702D 7274702D 3030315C 43657274 456E726F 6C6C5C43 41502D52 54502D30
  30312E63 726C3010 06092B06 01040182 37150104 03020100 300D0609 2A864886
  F70D0101 05050003 82010100 AB64FDEB F60C32DC 360F0E10 5FE175FA 0D574AB5
  02ACDCA3 C7BBED15 A4431F20 7E9286F0 770929A2 17E4CDF4 F2629244 2F3575AF
  E90C468C AE67BA08 AAA71C12 BA0C0E79 E6780A5C F814466C 326A4B56 73938380
  73A11AED F9B9DE74 1195C48F 99454B8C 30732980 CD6E7123 8B3A6D68 80B97E00
  7F4BD4BA 0B5AB462 94D9167E 6D8D48F2 597CDE61 25CFADCC 5BD141FB 210275A2
  0A4E3400 1428BA0F 69953BB5 50D21F78 43E3E563 98BCB2B1 A2D4864B 0616BACD
  A61CD9AE C5558A52 B5EEAA6A 08F96528 B1804B87 D26E4AEE AB7AFFE9 2FD2A574
  BAFE0028 96304A8B 13FB656D 8FC60094 D5A53D71 444B3CEF 79343385 3778C193
  74A2A6CE DC56275C A20A303D
  quit
crypto pki certificate chain 7960
 certificate ca F301
  308201F7 30820160 A0030201 020202F3 01300D06 092A8648 86F70D01 01050500
  3041310B 30090603 55040613 02555331 1A301806 0355040A 13114369 73636F20
  53797374 656D7320 496E6331 16301406 03550403 130D4341 50462D33 35453038
  33333230 1E170D30 34303430 39323035 3530325A 170D3139 30343036 32303535
  30315A30 41310B30 09060355 04061302 5553311A 30180603 55040A13 11436973
  636F2053 79737465 6D732049 6E633116 30140603 55040313 0D434150 462D3335
  45303833 33323081 9F300D06 092A8648 86F70D01 01010500 03818D00 30818902
  818100C8 BD9B6035 366B44E8 0F693A47 250FF865 D76C35F7 89B1C4FD 1D122CE0
  F5E5CDFF A4A87EFF 41AD936F E5C93163 3E55D11A AF82A5F6 D563E21C EB89EBFA
  F5271423 C3E875DC E0E07967 6E1AAB4F D3823E12 53547480 23BA1A09 295179B6
  85A0E83A 77DD0633 B9710A88 0890CD4D DB55ADD0 964369BA 489043BB B667E60F
  93954B02 03010001 300D0609 2A864886 F70D0101 05050003 81810056 60FD3AB3
  6F98D2AD 40C309E2 C05B841C 5189271F 01D864E8 98BCE665 2AFBCC8C 54007A84
  8F772C67 E3047A6C C62F6508 B36A6174 B68C1D78 C2228FEA A89ECEFB CC8BA9FC
  0F30E151 431670F9 918514D9 868D1235 18137F1E 50DFD32E 1DC29CB7 95EF4096
  421AF22F 5C1D5804 B83F8E8E 95B04F45 86563BFE DF976C5B FB490A
  quit
!
!
no crypto isakmp enable
!

!--- Enable IPsec.

crypto isakmp policy 1
 authentication pre-share
 lifetime 28800
crypto isakmp key cisco123 address 10.1.1.13

!--- The crypto key must match the key configured on Cisco CallManager.
!
!--- The crypto IPSec configuration must match your Cisco CallManager
!--- configuration.

crypto ipsec transform-set rtpset esp-des esp-md5-hmac
!
!
crypto map rtp 1 ipsec-isakmp
 set peer 10.1.1.13
 set transform-set rtpset
 match address 116
!
!
interface FastEthernet0/0
```

```
 ip address 10.1.1.22 255.255.255.0
 duplex auto
 speed auto
 crypto map rtp
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
ip classless
!
ip http server
no ip http secure-server
!
!

!--- Define the traffic to be encrypted by IPsec.

access-list 116 permit ip host 10.1.1.22 host 10.1.1.13
!
!
control-plane
!
!
call application alternate DEFAULT
!
!
voice-port 1/0/0
!
voice-port 1/0/1
!
voice-port 1/0/2
!
voice-port 1/0/3
!
voice-port 1/1/0
 timing hookflash-out 50
!
voice-port 1/1/1
!
voice-port 1/1/2
!
voice-port 1/1/3
!

!--- Enable the MGCP voice protocol.

mgcp
mgcp call-agent 10.1.1.13 2427 service-type mgcp version 0.1
mgcp dtmf-relay voip codec all mode out-of-band
mgcp rtp unreachable timeout 1000 action notify
mgcp package-capability rtp-package
mgcp package-capability sst-package
no mgcp package-capability fxr-package
no mgcp timer receive-rtcp
mgcp sdp simple
mgcp fax t38 inhibit
```

```
mgcp rtp payload-type g726r16 static
!
mgcp profile default
!
!
dial-peer voice 81235 pots
 application mgcpapp
 destination-pattern 81235
 port 1/1/0
 forward-digits all
!
dial-peer voice 81234 pots
 application mgcpapp
 destination-pattern 81234
 port 1/0/0
!
dial-peer voice 999100 pots
 application mgcpapp
 port 1/0/0
!
dial-peer voice 999110 pots
 application mgcpapp
 port 1/1/0
!
!

!--- Enable the credentials service on the gateway.
!--- Cisco CallManager takes the certificate retrieved from the secure SRST
!--- device certificate and places it in the configuration file of the
!--- Cisco IP phone. Activate credentials service on all SRST routers.
!--- Enable the SRST router to receive messages from Cisco CallManager. The
!--- IP address is the preexisting router IP address, typically one of the
!--- addresses of the Ethernet port of the router. The default port number is 2445.

credentials
 ip source-address 10.1.1.22 port 2445


!--- Specify the name of the trustpoint that is to be associated with the SRST
!--- router certificate. The trustpoint name must be the same as the one already
!--- declared.

 trustpoint srstca
!
!

!--- Enable SRST mode on the SRST router to support Cisco IP phone functions.

call-manager-fallback
 secondary-dialtone 9
 transfer-system full-consult
 ip source-address 10.1.1.22 port 2000
 max-ephones 15
 max-dn 30
 transfer-pattern .....
.
.
.
```

Configuration 2

```
!--- Allow trusted host traffic.

access-list 140 deny tcp host 10.1.1.11 any eq 2445

!--- Rate-limit all other traffic.

access-list 140 permit tcp any any eq 2445
access-list 140 deny ip any any

!--- Define class-map sccp-class.

class-map match-all sccp-class
 match access-group 140
policy-map control-plane-policy
 class sccp-class
  police 8000 1500 1500 conform-action drop exceed-action drop

!--- Define aggregate control plane service for the active Route Processor.

control-plane
 service-policy input control-plane-policy
```

Verify

Use this section in order to confirm that your configuration works properly.

The Output Interpreter Tool (registered customers only) (OIT) supports certain **show** commands. Use the OIT to view an analysis of **show** command output.

Verify Credential Settings

In order to verify credential settings on the SRST router that are supplied to Cisco CallManager for use during secure SRST fallback, issue the **show credentials** command.

```
Router#show credentials

Credentials IP: 10.1.1.22
Credentials PORT: 2445
Trustpoint: srstca
```

Verify Certificate Enrollment

If you used the Cisco IOS certificate server as your CA, issue the **show running-config** command in order to verify certificate enrollment or the **show crypto pki server** command in order to verify the status of the CA server.

1. Issue the **show running-config** command in order to verify the creation of the CA server (01) and device (02) certificates. This example shows the enrolled certificates.

```
! SRST router device certificate.
```

```
crypto pki certificate chain srstca
 certificate 02
  308201AD 30820116 A0030201 02020102 300D0609 2A864886 F70D0101 04050030
  17311530 13060355 0403130C 73727374 63617365 72766572 301E170D 30343034
  31323139 35323233 5A170D30 35303431 32313935 3232335A 30343132 300F0603
  55040513 08443042 39453739 43301F06 092A8648 86F70D01 09021612 6A61736F
  32363931 2E636973 636F2E63 6F6D305C 300D0609 2A864886 F70D0101 01050003
  4B003048 024100D7 0CC354FB 5F7C1AE7 7A25C3F2 056E0485 22896D36 6CA70C19
  C98F9BAE AE9D1F9B D4BB7A67 F3251174 193BB1A3 12946123 E5C1CCD7 A23E6155
  FA2ED743 3FB8B902 03010001 A330302E 300B0603 551D0F04 04030205 A0301F06
  03551D23 04183016 8014F829 CE97AD60 18D05467 FC293963 C2470691 F9BD300D
  06092A86 4886F70D 01010405 00038181 007EB48E CAE9E1B3 D1E7A185 D7F0D565
  CB84B17B 1151BD78 B3E39763 59EC650E 49371F6D 99CBD267 EB8ADF9D 9E43A5F2
  FB2B18A0 34AF6564 11239473 41478AFC A86E6DA1 AC518E0B 8657CEBB ED2BDE8E
  B586FE67 00C358D4 EFDD8D44 3F423141 C2D331D3 1EE43B6E 6CB29EE7 0B8C2752
  C3AF4A66 BD007348 D013000A EA3C206D CF
  quit
 certificate ca 01
  30820207 30820170 A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  17311530 13060355 0403130C 73727374 63617365 72766572 301E170D 30343034
  31323139 34353136 5A170D30 37303431 32313934 3531365A 30173115 30130603
  55040313 0C737273 74636173 65727665 7230819F 300D0609 2A864886 F70D0101
  01050003 818D0030 81890281 8100C3AF EE1E4BB1 9922A8DA 2BB9DC8E 5B1BD332
  1051C9FE 32A971B3 3C336635 74691954 98E765B1 059E24B6 32154E99 105CA989
  9619993F CC72C525 7357EBAC E6335A32 2AAF9391 99325BFD 9B8355EB C10F8963
  9D8FC222 EE8AC831 71ACD3A7 4E918A8F D5775159 76FBF499 5AD0849D CAA41417
  DD866902 21E5DD03 C37D4B28 0FAB0203 010001A3 63306130 0F060355 1D130101
  FF040530 030101FF 300E0603 551D0F01 01FF0404 03020186 301D0603 551D0E04
  160414F8 29CE97AD 6018D054 67FC2939 63C24706 91F9BD30 1F060355 1D230418
  30168014 F829CE97 AD6018D0 5467FC29 3963C247 0691F9BD 300D0609 2A864886
  F70D0101 04050003 8181007A F71B25F9 73D74552 25DFD03A D8D1338F 6792C805
  47A81019 795B5AAE 035400BB F859DABF 21892B5B E71A8283 08950414 8633A8B2
  C98565A6 C09CA641 88661402 ACC424FD 36F23360 ABFF4C55 BB23C66A C80A3A57
  5EE85FF8 C1B1A540 E818CE6D 58131726 BB060974 4E1A2F4B E6195522 122457F3
  DEDBAAD7 3780136E B112A6
  quit
```

2. Issue the **show crypto pki server** command in order to verify the status of the CA server after a boot procedure.

```
Router#show crypto pki server

Certificate Server srstcaserver:
Status: enabled
Server's configuration is locked (enter "shut" to unlock it)
Issuer name: CN=srstcaserver
CA cert fingerprint: AC9919F5 CAFE0560 92B3478A CFF5EC00
Granting mode is: auto
Last certificate issued serial number: 0x2
CA certificate expiration timer: 13:46:57 PST Dec 1 2007
CRL NextUpdate timer: 14:54:57 PST Jan 19 2005
Current storage dir: nvram
Database Level: Complete - all issued certs written as <serialnum>.cer
```

Verify Phone Status and Registrations

In order to verify or troubleshoot IP phone status and registration, complete these steps in privileged

EXEC mode.

1. Issue the **show ephone** command in order to display registered Cisco IP phones and their capabilities. This command also displays authentication and encryption status when used for secure SRST. In this example, authentication and encryption status is active with a TLS connection.

```
Router#show ephone

ephone-1 Mac:1000.1111.0002 TCP socket:[5] activeLine:0 REGISTERED in
SCCP ver 5 + Authentication + Encryption with TLS connection
mediaActive:0 offhook:0 ringing:0 reset:0 reset_sent:0 paging 0 debug:0
IP:10.1.1.40 32626 7970 keepalive 390 max_line 8
button 1: dn 14 number 2002 CM Fallback CH1 IDLE
ephone-2 Mac:1000.1111.000B TCP socket:[12] activeLine:0 REGISTERED in
SCCP ver 5 + Authentication + Encryption with TLS connection
mediaActive:0 offhook:0 ringing:0 reset:0 reset_sent:0 paging 0 debug:0
IP:10.1.1.40 32718 7970 keepalive 390 max_line 8
button 1: dn 21 number 2011 CM Fallback CH1 IDLE
ephone-3 Mac:1000.1111.000A TCP socket:[16] activeLine:0 REGISTERED in
SCCP ver 5 + Authentication + Encryption with TLS connection
mediaActive:0 offhook:0 ringing:0 reset:0 reset_sent:0 paging 0 debug:0
IP:10.1.1.40 32862 7970 keepalive 390 max_line 8
button 1: dn 2 number 2010 CM Fallback CH1 IDLE
```

2. Issue the **show ephone offhook** command in order to display Cisco IP phone status and quality for all phones that are off hook. In this example, authentication and encryption status is active with a TLS connection, and there is an active secure call.

```
Router#show ephone offhook

ephone-1 Mac:1000.1111.0002 TCP socket:[5] activeLine:1 REGISTERED in
SCCP ver 5 + Authentication + Encryption with TLS connection
mediaActive:1 offhook:1 ringing:0 reset:0 reset_sent:0 paging 0
:0
IP:10.1.1.40 32626 7970 keepalive 391 max_line 8
button 1: dn 14 number 2002 CM Fallback CH1 CONNECTED
Active Secure Call on DN 14 chan 1 :2002 10.1.1.40 29632 to 10.1.1.40 2561
via 10.1.1.40 G711Ulaw64k 160 bytes no vad
Tx Pkts 295 bytes 49468 Rx Pkts 277 bytes 46531 Lost 0
Jitter 0 Latency 0 callingDn 22 calledDn -1
ephone-2 Mac:1000.1111.000B TCP socket:[12] activeLine:1 REGISTERED in
SCCP ver 5 + Authentication + Encryption with TLS connection
mediaActive:1 offhook:1 ringing:0 reset:0 reset_sent:0 paging 0 debug:0
IP:10.1.1.40 32718 7970 keepalive 391 max_line 8
button 1: dn 21 number 2011 CM Fallback CH1 CONNECTED
Active Secure Call on DN 21 chan 1 :2011 10.1.1.40 16382 to 10.1.1.40
16382 via 10.1.1.40 G711Ulaw64k 160 bytes no vad
Tx Pkts 295 bytes 49468 Rx Pkts 277 bytes 46531 Lost 0
Jitter 0 Latency 0 callingDn -1 calledDn 11
```

3. Issue the **show voice call status** command in order to display the call status for all voice ports on the Cisco SRST router. This command is not applicable for calls between two POTS dial peers.

```
Router#show voice call status
```

```
CallID CID ccVdb Port DSP/Ch Called # Codec Dial-peers
0x1164 2BFE 0x8619A460 50/0/35.0 2014 g711ulaw 20035/20027
0x1165 2BFE 0x86144B78 50/0/27.0 *2014 g711ulaw 20027/20035
0x1166 2C01 0x861043D8 50/0/21.0 2012 g711ulaw 20021/20011
0x1168 2C01 0x860984C4 50/0/11.0 *2012 g711ulaw 20011/20021
0x1167 2C04 0x8610EC7C 50/0/22.0 2002 g711ulaw 20022/20014
0x1169 2C04 0x860B8894 50/0/14.0 *2002 g711ulaw 20014/20022
0x116A 2C07 0x860A374C 50/0/12.0 2010 g711ulaw 20012/20002
0x116B 2C07 0x86039700 50/0/2.0 *2010 g711ulaw 20002/20012
0x116C 2C0A 0x86119520 50/0/23.0 2034 g711ulaw 20023/20020
0x116D 2C0A 0x860F9150 50/0/20.0 *2034 g711ulaw 20020/20023
0x116E 2C0D 0x8608DC20 50/0/10.0 2022 g711ulaw 20010/20008
0x116F 2C0D 0x86078AD8 50/0/8.0 *2022 g711ulaw 20008/20010
0x1170 2C10 0x861398F0 50/0/26.0 2016 g711ulaw 20026/20028
0x1171 2C10 0x8614F41C 50/0/28.0 *2016 g711ulaw 20028/20026
0x1172 2C13 0x86159CC0 50/0/29.0 2018 g711ulaw 20029/20004
0x1173 2C13 0x8604E848 50/0/4.0 *2018 g711ulaw 20004/20029
0x1174 2C16 0x8612F04C 50/0/25.0 2026 g711ulaw 20025/20030
0x1175 2C16 0x86164F48 50/0/30.0 *2026 g711ulaw 20030/20025
0x1176 2C19 0x860D8C64 50/0/17.0 2032 g711ulaw 20017/20018
0x1177 2C19 0x860E4008 50/0/18.0 *2032 g711ulaw 20018/20017
0x1178 2C1C 0x860CE3C0 50/0/16.0 2004 g711ulaw 20016/20019
0x1179 2C1C 0x860EE8AC 50/0/19.0 *2004 g711ulaw 20019/20016
0x117A 2C1F 0x86043FA4 50/0/3.0 2008 g711ulaw 20003/20024
0x117B 2C1F 0x861247A8 50/0/24.0 *2008 g711ulaw 20024/20003
0x117C 2C22 0x8608337C 50/0/9.0 2020 g711ulaw 20009/20031
0x117D 2C22 0x8616F7EC 50/0/31.0 *2020 g711ulaw 20031/20009
0x117E 2C25 0x86063990 50/0/6.0 2006 g711ulaw 20006/20001
0x117F 2C25 0x85C6BE6C 50/0/1.0 *2006 g711ulaw 20001/20006
0x1180 2C28 0x860ADFF0 50/0/13.0 2029 g711ulaw 20013/20034
0x1181 2C28 0x8618FBBC 50/0/34.0 *2029 g711ulaw 20034/20013
0x1182 2C2B 0x860C3B1C 50/0/15.0 2036 g711ulaw 20015/20005
0x1183 2C2B 0x860590EC 50/0/5.0 *2036 g711ulaw 20005/20015
0x1184 2C2E 0x8617A090 50/0/32.0 2024 g711ulaw 20032/20007
0x1185 2C2E 0x8606E234 50/0/7.0 *2024 g711ulaw 20007/20032
0x1186 2C31 0x861A56E8 50/0/36.0 2030 g711ulaw 20036/20033
0x1187 2C31 0x86185318 50/0/33.0 *2030 g711ulaw 20033/20036
18 active calls found
```

Troubleshoot

This section provides information you can use to troubleshoot your configuration.

For additional information on how to troubleshoot, see the Related Information section.

Debug Credential Settings

This task debugs credential settings on the SRST router.

In order to set debugging on the credential settings that are supplied to Cisco CallManager for use during secure SRST fallback, issue the **debug credentials** command.

```
Router#debug credentials

Credentials server debugging is enabled
```

```
Router#
Sep 29 01:01:50.903: Credentials service: Start TLS Handshake 1 10.1.1.13 2187
Sep 29 01:01:50.903: Credentials service: TLS Handshake returns
OPSSLReadWouldBlockErr
Sep 29 01:01:51.903: Credentials service: TLS Handshake returns
OPSSLReadWouldBlockErr
Sep 29 01:01:52.907: Credentials service: TLS Handshake returns
OPSSLReadWouldBlockErr
Sep 29 01:01:53.927: Credentials service: TLS Handshake completes.
```

Debug IP Phone Registrations

These tasks debug Cisco IP phone registration and call setup.

1. In order to debug the process of Cisco IP phone registration, issue the **debug ephone register** command.

```
Router#debug ephone register

EPHONE registration debugging is enabled
*Jun 29 09:16:02.180: New Skinny socket accepted [2] (0 active)
*Jun 29 09:16:02.180: sin_family 2, sin_port 51617, in_addr 10.5.43.177
*Jun 29 09:16:02.180: skinny_socket_process: secure skinny sessions = 1
*Jun 29 09:16:02.180: add_skinny_secure_socket: pid =155, new_sock=0,
   ip address = 1.5.43.177
*Jun 29 09:16:02.180: skinny_secure_handshake: pid =155, sock=0,
   args->pid=155, ip address = 10.5.43.177
*Jun 29 09:16:02.184: Start TLS Handshake 0 10.5.43.177 51617
*Jun 29 09:16:02.184: TLS Handshake retcode OPSSLReadWouldBlockErr
*Jun 29 09:16:03.188: TLS Handshake retcode OPSSLReadWouldBlockErr
*Jun 29 09:16:04.188: TLS Handshake retcode OPSSLReadWouldBlockErr
*Jun 29 09:16:05.188: TLS Handshake retcode OPSSLReadWouldBlockErr
*Jun 29 09:16:06.188: TLS Handshake retcode OPSSLReadWouldBlockErr
*Jun 29 09:16:07.188: TLS Handshake retcode OPSSLReadWouldBlockErr
*Jun 29 09:16:08.188: CRYPTO_PKI_OPSSL - Verifying 1 Certs
*Jun 29 09:16:08.212: TLS Handshake completes
```

2. In order to review call setup between two secure Cisco IP phones, issue the **debug ephone state** command. The **debug ephone state** trace shows the generation and distribution of encryption and decryption keys between the two phones.

```
Router#debug ephone state

*Jan 11 18:33:09.231:%SYS-5-CONFIG_I:Configured from console by console
*Jan 11 18:33:11.747:ephone-2[2]:OFFHOOK
*Jan 11 18:33:11.747:ephone-2[2]:---SkinnySyncPhoneDnOverlays is onhook
*Jan 11 18:33:11.747:ephone-2[2]:SIEZE on activeLine 0 activeChan 1
*Jan 11 18:33:11.747:ephone-2[2]:SetCallState line 1 DN 2(-1) chan 1 ref 6
   TsOffHook
*Jan 11 18:33:11.747:ephone-2[2]:Check Plar Number
*Jan 11 18:33:11.751:DN 2 chan 1 Voice_Mode
*Jan 11 18:33:11.751:dn_tone_control DN=2 chan 1 tonetype=33:DtInsideDialT
   onoff=1 pid=232
*Jan 11 18:33:15.031:dn_tone_control DN=2 chan 1 tonetype=0:DtSilence
   onoff=0 pid=232
```

```
*Jan 11 18:33:16.039:ephone-2[2]:Skinny-to-Skinny call DN 2 chan 1 to DN 4
   chan 1 instance 1
*Jan 11 18:33:16.039:ephone-2[2]:SetCallState line 1 DN 2(-1) chan 1 ref 6
   TsProceed
*Jan 11 18:33:16.039:ephone-2[2]:SetCallState line 1 DN 2(-1) chan 1 ref 6
   TsRingOut
*Jan 11 18:33:16.039:ephone-2[2]::callingNumber 6000
*Jan 11 18:33:16.039:ephone-2[2]::callingParty 6000
*Jan 11 18:33:16.039:ephone-2[2]:Call Info DN 2 line 1 ref 6 call state 1
   called 6001 calling 6000 origcalled
*Jan 11 18:33:16.039:ephone-2[2]:Call Info DN 2 line 1 ref 6 called 6001
   calling 6000 origcalled 6001 calltype 2
*Jan 11 18:33:16.039:ephone-2[2]:Call Info for chan 1
*Jan 11 18:33:16.039:ephone-2[2]:Original Called Name 6001
*Jan 11 18:33:16.039:ephone-2[2]:6000 calling
*Jan 11 18:33:16.039:ephone-2[2]:6001
*Jan 11 18:33:16.047:ephone-3[3]:SetCallState line 1 DN 4(4) chan 1 ref 7
   TsRingIn
*Jan 11 18:33:16.047:ephone-3[3]::callingNumber 6000
*Jan 11 18:33:16.047:ephone-3[3]::callingParty 6000
*Jan 11 18:33:16.047:ephone-3[3]:Call Info DN 4 line 1 ref 7 call state 7
   called 6001 calling 6000 origcalled
*Jan 11 18:33:16.047:ephone-3[3]:Call Info DN 4 line 1 ref 7 called 6001
   calling 6000 origcalled 6001 calltype 1
*Jan 11 18:33:16.047:ephone-3[3]:Call Info for chan 1
*Jan 11 18:33:16.047:ephone-3[3]:Original Called Name 6001
*Jan 11 18:33:16.047:ephone-3[3]:6000 calling
*Jan 11 18:33:16.047:ephone-3[3]:6001
*Jan 11 18:33:16.047:ephone-3[3]:Ringer Inside Ring On
*Jan 11 18:33:16.051:dn_tone_control DN=2 chan 1
   tonetype=36:DtAlertingTone onoff=1 pid=232
*Jan 11 18:33:20.831:ephone-3[3]:OFFHOOK
*Jan 11 18:33:20.831:ephone-3[3]:---SkinnySyncPhoneDnOverlays is onhook
*Jan 11 18:33:20.831:ephone-3[3]:Ringer Off
*Jan 11 18:33:20.831:ephone-3[3]:ANSWER call
*Jan 11 18:33:20.831:ephone-3[3]:SetCallState line 1 DN 4(-1) chan 1 ref 7
   TsOffHook
*Jan 11 18:33:20.831:ephone-3[3][SEP000DEDAB3EBF]:Answer Incoming call
   from ephone-(2) DN 2 chan 1
*Jan 11 18:33:20.831:ephone-3[3]:SetCallState line 1 DN 4(-1) chan 1 ref 7
   TsConnected
*Jan 11 18:33:20.831:defer_start for DN 2 chan 1 at CONNECTED
*Jan 11 18:33:20.831:ephone-2[2]:SetCallState line 1 DN 2(-1) chan 1 ref 6
   TsConnected
*Jan 11 18:33:20.835:ephone-3[3]::callingNumber 6000
*Jan 11 18:33:20.835:ephone-3[3]::callingParty 6000
*Jan 11 18:33:20.835:ephone-3[3]:Call Info DN 4 line 1 ref 7 call state 4
   called 6001 calling 6000 origcalled
*Jan 11 18:33:20.835:ephone-3[3]:Call Info DN 4 line 1 ref 7 called 6001
   calling 6000 origcalled 6001 calltype 1
*Jan 11 18:33:20.835:ephone-3[3]:Call Info for chan 1
*Jan 11 18:33:20.835:ephone-3[3]:Original Called Name 6001
*Jan 11 18:33:20.835:ephone-3[3]:6000 calling
*Jan 11 18:33:20.835:ephone-3[3]:6001
*Jan 11 18:33:20.835:ephone-2[2]:Security Key Generation
! Ephone 2 generates a security key.
*Jan 11 18:33:20.835:ephone-2[2]:OpenReceive DN 2 chan 1 codec
   4:G711Ulaw64k duration 20 ms bytes 160
*Jan 11 18:33:20.835:ephone-2[2]:Send Decryption Key
! Ephone 2 sends the decryption key.
```

```
*Jan 11 18:33:20.835:ephone-3[3]:Security Key Generation
!Ephone 3 generates its security key.
*Jan 11 18:33:20.835:ephone-3[3]:OpenReceive DN 4 chan 1 codec
   4:G711Ulaw64k duration 20 ms bytes 160
*Jan 11 18:33:20.835:ephone-3[3]:Send Decryption Key
! Ephone 3 sends its decryption key.
*Jan 11 18:33:21.087:dn_tone_control DN=2 chan 1 tonetype=0:DtSilence
   onoff=0 pid=232
*Jan 11 18:33:21.087:DN 4 chan 1 Voice_Mode
*Jan 11 18:33:21.091:DN 2 chan 1 End Voice_Mode
*Jan 11 18:33:21.091:DN 2 chan 1 Voice_Mode
*Jan 11 18:33:21.095:ephone-2[2]:OpenReceiveChannelAck:IP 1.1.1.8,
   port=25552, dn_index=2, dn=2, chan=1
*Jan 11 18:33:21.095:ephone-3[3]:StartMedia 1.1.1.8 port=25552
*Jan 11 18:33:21.095:DN 2 chan 1 codec 4:G711Ulaw64k duration 20 ms bytes
   160
*Jan 11 18:33:21.095:ephone-3[3]:Send Encryption Key
! Ephone 3 sends its encryption key.
*Jan 11 18:33:21.347:ephone-3[3]:OpenReceiveChannelAck:IP 1.1.1.9,
   port=17520, dn_index=4, dn=4, chan=1
*Jan 11 18:33:21.347:ephone-2[2]:StartMedia 1.1.1.9 port=17520
*Jan 11 18:33:21.347:DN 2 chan 1 codec 4:G711Ulaw64k duration 20 ms bytes
   160
*Jan 11 18:33:21.347:ephone-2[2]:Send Encryption Key
!Ephone 2 sends its encryption key.
*Jan 11 18:33:21.851:ephone-2[2]::callingNumber 6000
*Jan 11 18:33:21.851:ephone-2[2]::callingParty 6000
*Jan 11 18:33:21.851:ephone-2[2]:Call Info DN 2 line 1 ref 6 call state 4
   called 6001 calling 6000 origc
*Jan 11 18:33:21.851:ephone-2[2]:Call Info DN 2 line 1 ref 6 called 6001
   calling 6000 origcalled 6001 calltype 2
*Jan 11 18:33:21.851:ephone-2[2]:Call Info for chan 1
*Jan 11 18:33:21.851:ephone-2[2]:Original Called Name 6001
*Jan 11 18:33:21.851:ephone-2[2]:6000 calling
*Jan 11 18:33:21.851:ephone-2[2]:6001
```

Cisco Support Community - Featured Conversations

Cisco Support Community is a forum for you to ask and answer questions, share suggestions, and collaborate with your peers. Below are just some of the most recent and relevant conversations happening right now.

Want to see more? Join us by clicking **here**

Start A New Discussion **Subscribe**

Related Information

- **Certificate Enrollment Enhancements**
- **Certification Authority Interoperability Commands**
- **Cisco SRST and SIP SRST Command Reference (All Versions)**
- **Control Plane Policing**
- **Manual Certificate Enrollment (TFTP and Cut-and-Paste)**
- **Voice Technology Support**
- **Voice and Unified Communications Product Support**
- **Recommended Reading: Troubleshooting Cisco IP Telephony** 
- **Technical Support & Documentation - Cisco Systems**
