Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint (For Cisco IOS Releases 12.4(4)T and 12.4(9)T Only)

First Published: October 30, 2005
Last Updated: June 25, 2008

> **Note** The information in this document applies only to the Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint feature for Cisco IOS Releases 12.4(4)T and 12.4(9)T. This feature has been upgraded and released as the Cisco V.150.1 Minimum Essential Requirements feature beginning in Cisco IOS Release 15.1(4)M (dated March 25, 2011) and later releases. For detailed information about the upgraded feature, see the *Cisco V.150.1 Minimum Essential Requirements* document.

The Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint feature provides the following Cisco IOS gateway capabilities:

- Support for establishing secure calls between gateway-attached secure terminal equipment (STE) devices and IP-STE devices
- Ability to configure modem transport methods
- Ability to configure V.150.1 modem relay parameters

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint" section on page 35.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Contents

Prerequisites for Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint

Make sure the following tasks have been completed before configuring the Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint feature:

- Cisco Unified CallManager 4.1.2 or a later release is running.
- Cisco IOS Release 12.4(9)T or a later release is running.
- Cisco IOS Advanced Enterprise Services image is installed.
- V.150.1 modem relay capability is enabled.

Note In addition to V.150.1 modem relay, Cisco offers a proprietary, named signaling event (NSE)-based modem-relay implementation. The Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint feature supports only V.150.1 modem relay.

Restrictions for Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint

- IP-STE fallback to Survivable Remote Site Telephony (SRST) mode is not supported.
- Secure calls between secure telephone unit (STU) and IP-STE are not supported.
- Secure communications are supported for the call scenarios listed in Table 1, and on the platforms and network modules listed in Table 2, Supported Gateways, Modules, and VICs for the Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint Feature.

- MGCP trunk-side support is available only on Cisco 2691, Cisco 2811, Cisco 2821, Cisco 2851, Cisco 3725, Cisco 3745, Cisco 3825, and Cisco 3845 platforms using 5510 digital signal processors (DSPs).
- Skinny Client Control Protocol (SCCP) line-side support is available only on Cisco 2801, Cisco 2811, Cisco 2821, Cisco 2851, Cisco 3725, Cisco 3745, Cisco 3825, and Cisco 3845 platforms with analog FXS and BRI ports using 5510 DSPs. SCCP line-side support on the VG224 platform is available only on analog FXS ports using 5510 DSPs.

Information About Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint

To configure the Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint feature, you should understand the following concepts:

- Benefits of Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint
- Feature Design of Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint

Benefits of Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint

- Supports the use of existing STE in voice networks, by allowing secure calls between gateway-connected legacy analog and BRI STE devices to IP-STE devices.
- Facilitates transition of STE and STU infrastructure to VoIP.

Feature Design of Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint

The Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint feature implements full IP-STE interoperability in a VoIP network by providing support for secure communications mode. This feature implements modem-relay support to gateway-connected endpoints controlled by Cisco CallManager. Prior to the Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint feature, Cisco IOS gateways supported secure voice and data calls between legacy on-net STE and STU devices using modem pass-through. This support did not extend to IP-STE, which require modem relay transmission to communicate. The new feature includes support for IP-STE by introducing a standards-based implementation of a subset of ITU-T Recommendation V.150.1, the standard for modem relay transmission. This modem relay capability allows Cisco Unified CallManager Skinny Client Control Protocol (SCCP)-controlled endpoints to communicate with the IP-STE or on-net legacy STE.

The Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint feature supports the following capabilities:

- Secure voice and secure data modes from STE devices connected to Cisco IOS gateway foreign exchange station (FXS) and BRI ports to an IP-STE.
- Support for the state signaling events (SSE) protocol, allowing for modem signaling end-to-end and VoIP to modem over IP (MoIP) transition and operation.

- Interoperation between line-side and trunk-side gateways and Cisco Unified CallManager to determine codec support and V.150.1 negotiation. You can configure gateway-attached devices to support either modem relay, modem pass-through, both modem transport methods, or neither method.
- Ability to tune V.150.1 modem-relay parameters to address specific network conditions.

Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint Architecture

Companion Cisco IOS features provide an overall architecture for call control and secure calls on the following on-network and off-network devices:

- Existing encryption and multi-level precedence and preemption (MLPP)- capable STE and STU
- IP-STE

The MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways feature provides support for Cisco Unified CallManager-controlled analog and BRI voice ports using SCCP on line-side access gateways. The Secure Communication Between IP-STE Endpoint and Trunk-Side STE Endpoint feature provides Cisco V.150.1 implementation on trunk-side gateways, enabling off-network STE to communicate with IP-STE endpoints. The Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint feature provides line-side V.150.1 modem relay capabilities to FXS and BRI ports, allowing Cisco Unified CallManager-controlled endpoints to communicate with IP-STE.

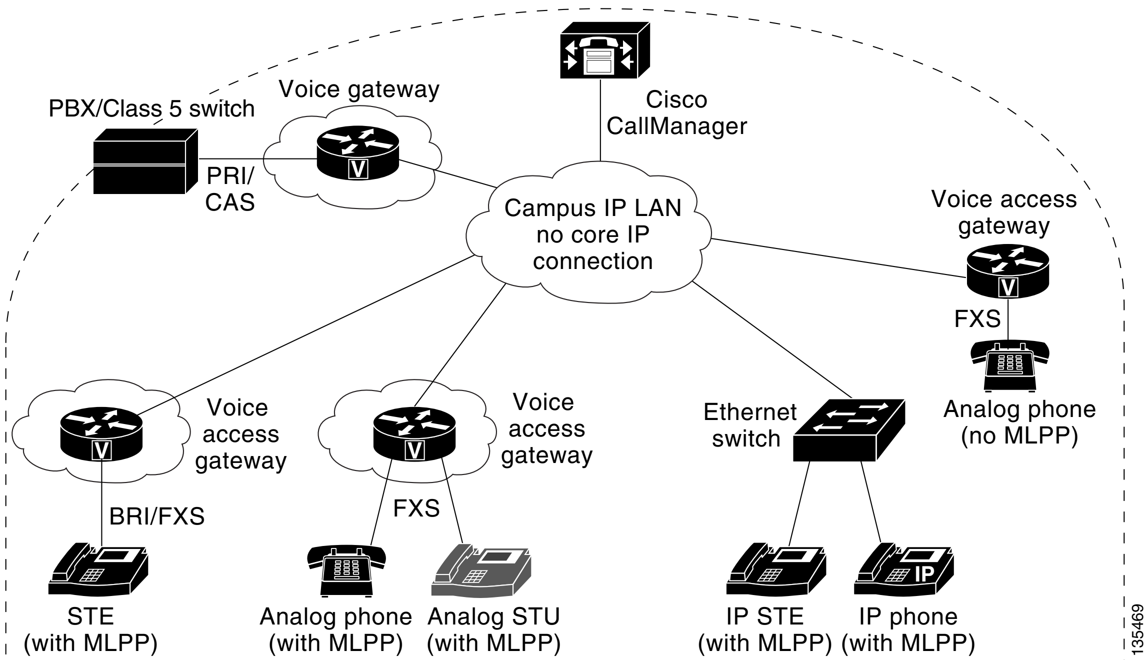
The SCCP Controlled Analog FXS Ports with Supplementary Features in Cisco IOS Gateways feature provides support for SCCP supplementary features for analog FXS ports on Cisco Integrated Services Routers under the control of Cisco Unified CallManager or a Cisco Unified CallManager Express (Cisco Unified CME) system.

These features use Cisco proprietary SCCP to communicate call control messages between the Cisco CallManager and gateway endpoints (phones). Endpoints can be FXS analog phones or BRI ISDN encryption-enabled phones. Cisco implements this support by the use of the SCCP Telephony Control Application (STCAPP), software that runs on existing line-side gateways. The line-side gateway translates call control messages between the Cisco Unified CallManager SCCP and the call control application programming interface (CCAPI), allowing the attached analog and BRI phones to be controlled by the Cisco Unified CallManager in the same way that Cisco IP phones are controlled.

See Table 1 for a matrix of supported secure call scenarios and feature interoperation.

Figure 1 shows a typical topology where the Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint feature is deployed and how the devices connect.

Figure 1 Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint Feature in a Telephony Network



The new feature enables secure calls over the network using modem connections. There are two ways to transport modem traffic over VoIP networks, modem pass-through and modem relay:

- With modem pass-through, also known as voice band data (VBD), modem traffic is carried between gateways in Real-time Transport Protocol (RTP) packets, using an uncompressed voice codec, G.711 mu-law, or G.711 a-law. Modem pass-through traffic is susceptible to packet loss, jitter, and latency in the IP network, with packet redundancy being used to mitigate the effects of packet loss.
- With modem relay, the modem signals are demodulated at one gateway, converted into digital form, and carried in Simple Packet Relay Transport (SPRT) protocol (which is a protocol layered upon User Datagram Protocol [UDP]) packets to the other gateway. There the modem signal is re-created and remodulated, then passed to the receiving modem. In this implementation, the call starts as a voice call, then switches briefly to modem pass-through mode, and then switches into modem relay mode for the duration of the secure call.

To signal media state transitions and to achieve fast media state synchronization of media gateways and endpoints, V.150.1 modem relay uses state signalling events (SSEs). These RTP-encoded event messages coordinate switching between different media states, that is, between initial audio state (a voice call), VBD or modem pass-through, and modem relay. This implementation of modem relay supports V.32 and V.34 modulation; it does not support data compression or error correction. The STE provides error correction for secure data transmissions.

The Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint feature provides a command line interface (CLI) that allows you to configure modem transport capabilities for gateway endpoints. You can specify modem transport method by registering the endpoint with the capability of modem relay, modem pass-through, both, or neither. When STCAPP devices register with Cisco CallManager, the application sends codec capabilities messages. Cisco Unified CallManager determines the codec to be used during call setup based on device capabilities sent during endpoint registration and Cisco Unified CallManager therefore controls the enabling of modem relay on the VoIP call leg for FXS and BRI ports. Because Cisco Unified CallManager prioritizes modem pass-through ahead of modem relay, any voice port, such as an on-net STE, that registers with both modem-relay and modem pass-through capability is, in effect, enabled as modem pass-through-capable. If, for example, modem relay is the desired transport method for an on-net STE, you need to register the STE as modem-relay-capable only. However, if you register an STE as modem-relay-capable only, it cannot communicate with the modem pass-through-capable STU. For more information on configuring device registration capabilities, see the "Configuring Modem Transport Methods for STCAPP Devices" section. For more information on endpoint registration, see "Configuring the SCCP Gateway and Gateway Endpoints on Cisco CallManager" and "Configuring Cisco Unified CallManager Download on Cisco IOS Gateways" sections.

The Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint feature allows you to configure V.150.1 modem-relay parameters to meet the needs of various network conditions. For more information, see the "Configuring V.150.1 Modem Relay Parameters" section. You may also configure V.150.1 parameters on a gateway that is not STCAPP enabled or one that provides PSTN gateway functionality. For more information, see the "Configuring the Secure Communications Between IP-STE Endpoint and Trunk-Side STE Endpoint Feature" section of this document.

Supported Endpoints for Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint

The Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint feature supports the following encryption-capable endpoints:

- STE, specialized encryption-capable analog or BRI phones, that can communicate over V.150.1 modem relay or over modem pass-through, also known as VBD.
- IP-STE, specialized encryption-capable IP phones that communicate only over V.150.1 modem relay.
- STU, specialized encryption-capable analog phones, that operate only over NSE-based modem pass-through connections.

Note Secure calls between IP-STE and STU are not supported. V.150.1 modem relay, using Future Narrow Band Digital Terminal (FNBDT) signaling over a V.32 or V.34 data pump, is the only transport method supported by IP-STE for secure communication; STU devices communicate using modem pass-through, do not support FNBDT signaling, and use a proprietary data pump.

Table 1 lists call scenarios between devices on and off the IP network, along with modem transport methods, that the Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint feature supports.

Table 1 Supported Secure Call Scenarios and Modem Transport Methods

Device Type	STU	On-net STE Gateway A[1]	On-Net STE Gateway B[2]	Off-Net STE (PSTN) [3]	IP-STE
STU	Pass-through	Pass through	Pass-through	Pass-through	None
On-net STE[1]	Pass-through	Pass-through	Pass-through	Pass-through	None
On-Net STE Gateway B[2]	Pass-through	Pass-through	Pass-through or Relay	Pass-through or Relay	Relay

Table 1 Supported Secure Call Scenarios and Modem Transport Methods (continued)

Device Type	STU	On-net STE Gateway A[1]	On-Net STE Gateway B[2]	Off-Net STE (PSTN)[3]	IP-STE
Off-Net STE (PSTN)[3]	Pass-through	Pass-through	Pass-through or Relay	Pass-through	Relay
IP-STE	None	None	Relay	Relay	Relay

1. Gateway A has the MLPP for Analog and BRI Endpoints on Cisco IOS Gateways feature enabled.
2. Gateway B has the Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint feature enabled.
3. The off-net STE calls through a gateway that has the Secure Communication Between IP-STE Endpoint and Trunk-Side Endpoint feature enabled.

Supported Gateways, Modules, and Voice Interface Cards

Table 2 lists supported gateways, modules, and voice interface cards (VICs).

Table 2 Supported Gateways, Modules, and VICs for the Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint Feature

Supported Gateways	Supported Extension Modules	Supported Network Modules and Expansion Modules	Supported VICs
• Cisco 2801 • Cisco 2811 • Cisco 2821 • Cisco 2851 • Cisco 3825 • Cisco 3845	—	• NM-HD-1V • NM-HD-2V • NM-HD-2VE	• VIC2-2FXS • VIC-4FXS/DID • VIC2-2BRI-NT/TE
• Cisco 2801 • Cisco 2821 • Cisco 2851 • Cisco 3825 • Cisco 3845	• EVM-HD	• EVM-HD-8FXS/DID • EM-3FXS/4FXO • EM-HDA-8FXS • EM-4BRI-NT/TE	—
• Cisco 2801 • Cisco 2811 • Cisco 2821 • Cisco 2851 • Cisco 3825 • Cisco 3845	—	• NM-HDV2 • NM-HDV2-1T1/E1 • NM-HDV2-2T1/E1	• VIC2-2FXS • VIC-4FXS/DID • VIC2-2BRI-NT/TE
• Cisco VG 224	—	—	—

How to Configure the Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint Feature

This section contains the following procedures:

- Configuring the SCCP Gateway and Gateway Endpoints on Cisco CallManager, page 8 (required)
- Configuring Cisco Unified CallManager Download on Cisco IOS Gateways, page 11 (required)
- Configuring SCCP on Cisco IOS Gateways, page 14 (required)
- Configuring Modem Pass-through Calls on Cisco IOS Gateways, page 15 (optional)
- Configuring Modem Transport Methods for STCAPP Devices, page 16 (required)
- Configuring V.150.1 Modem Relay Parameters, page 17 (optional)
- Configuring the Secure Communications Between IP-STE Endpoint and Trunk-Side STE Endpoint Feature, page 18 (optional)
- Verifying and Troubleshooting Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint, page 19 (optional)

Configuring the SCCP Gateway and Gateway Endpoints on Cisco CallManager

This task configures the SCCP gateway and gateway controlled endpoints on the Cisco CallManager, enabling them to support the Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint feature.

Cisco Unified CallManager Auto configuration

There are two methods of configuring the SCCP gateway, either by using Cisco Unified CallManager autoconfiguration or by manually configuring STCAPP on the gateway. The first method allows you to configure the SCCP gateway and SCCP gateway controlled endpoints on the Cisco Unified CallManager, then download eXtensible Markup Language (XML) configuration files for the endpoints to the Cisco IOS gateway. The second method requires you to manually enable the STCAPP applications and manually configure gateway endpoints. We recommend the Cisco Unified CallManager autoconfiguration method because it allows you to configure more devices in one place, eliminating endpoint configuration on the gateway.

Perform the following task to use Cisco UnifiedCallManager autoconfiguration. Figure 2 shows the Cisco CallManager gateway configuration window.

SUMMARY STEPS

1. Choose **Add a New Device** or **Add a New Gateway** in the Cisco Unified CallManager menu.
2. Choose the new SCCP gateway settings.
3. Configure the gateway MAC address, network modules, voice interface cards, and ports.
4. Save the configuration.
5. Verify the configuration.

DETAILED STEPS

Step 1 In the drop-down list in Cisco Unified CallManager, choose **Device** > **Add a New Device** > **Gateway** (from the **Device Type**) or choose **Device** > **Gateway** > **Add a New Gateway**.

Step 2 Choose the appropriate settings for the SCCP gateway:

- **a.** Choose the gateway type.
- **b.** Choose the SCCP option for device protocol.

Step 3 Enter the appropriate SCCP gateway MAC address and configure the network modules, voice interface cards, and ports:

- **a.** Enter the last ten characters of the MAC address of the interface used to register with the Cisco CallManager. Use the **show interface** command on the SCCP configured interface on the gateway to determine the gateway MAC address. (This MAC address must be the same address as that of the SCCP gateway local interface manually configured in Step 4 of the "DETAILED STEPS" section on page 14).
- **b.** Enter the gateway name.
- **c.** Enter the Cisco Unified CallManager group.
- **d.** Configure the appropriate network modules, voice interface cards, and ports.

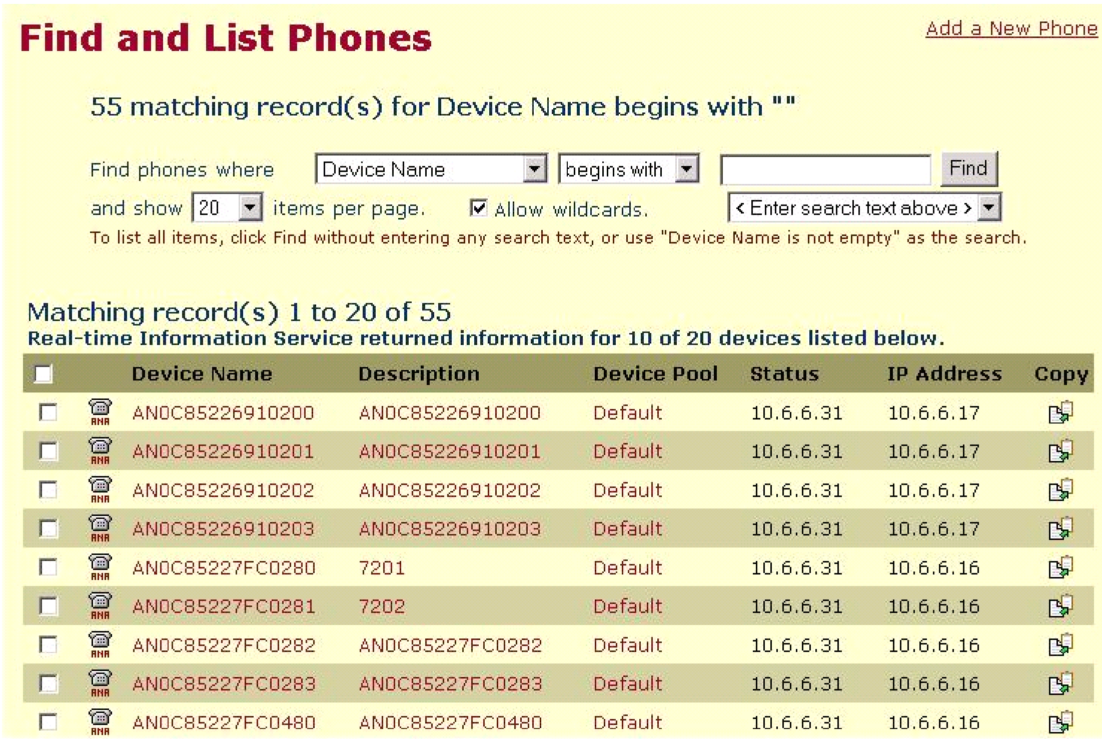
Note Gateway VIC port and slot numbers are referred to as Endpoint Identifiers on the Cisco Unified CallManager. For more information on Cisco Unified CallManager gateway configuration, refer to the section "Adding a Cisco IOS SCCP Gateway," in the *Cisco CallManager Administration Guide*, Release 4.1(2).

Figure 2 Gateway Configuration Window

Step 4 Click **Insert** to save a new gateway configuration, or click **Update** to save an existing gateway configuration.

Step 5 Verify your configuration by displaying the list of configured analog and BRI phones. In the Cisco CallManager menu choose **Device** > **Phone** > **Find**. Analog phone device names begin with "AN" and BRI phone device names begin with "BR," as shown in Figure 3.

Figure 3 Find and List Phones Window



127496

Configuring Cisco Unified CallManager Download on Cisco IOS Gateways

This task configures automatic download capability of XML dial-peer configuration files from the Cisco Unified CallManager and enables Cisco Unified CallManager autoconfiguration. Perform this task to enable automatic download of dial-peer configuration files from Cisco CallManager to the SCCP gateway.

Note Although you may manually configure dial peers to work with the Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint feature, we recommend that you use Cisco Unified CallManager autoconfiguration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ccm-manager sccp local** *interface-type interface-number*
4. **ccm-manager config** [**dialpeer-prefix** *prefix* | **server** {*ip-address* | *name*}]
5. **ccm-manager sccp**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`ccm-manager sccp local`** *`interface-type interface-number`* **Example:** `Router(config)# ccm-manager sccp local fastethernet 0/0`	Selects the local interface that the SCCP application should use to register with Cisco Unified CallManager. This interface must be specified before you enable the autoconfiguration process. • *interface-type*—Interface type that the SCCP application uses to register with Cisco Unified CallManager. • *interface-number*—Interface number that the SCCP application uses to register with Cisco Unified CallManager.

	Command or Action	Purpose
Step 4	**ccm-manager config** [**dialpeer-prefix** *prefix* | **server** {*ip-address* | *name*}] **Example:** `Router(config)# ccm-manager config dialpeer-prefix 888`	Enables the download of Cisco Unified CallManager XML configuration files. **Note** We recommend that you configure the gateway to run some form of name resolution prior to running the XML download, so that the gateway is able to resolve the Cisco Unified CallManager name in the XML file to an IP address. Otherwise, phones may not register with the Cisco Unified CallManager. • **dialpeer prefix** *prefix*—Configures the prefix to use for autogenerated dial peers. Range is 0 to 999999. The default is 999. **Note** When manually adding a dial-peer prefix, select a prefix number other than the autoconfigured dial-peer prefix (999 by default), to keep manually added dial peers from being deleted from the running configuration when the Cisco CallManager download happens in the gateway. • **server** *ip-address*—Specifies the IP address of the TFTP server from which to download the XML configuration files to the Cisco IOS gateway. The TFTP server is typically your Cisco Unified CallManager. • **server** *name*—Specifies the TFTP server name from which the Cisco IOS gateway downloads Cisco CallManager XML configuration files. The TFTP server is typically your Cisco Unified CallManager.
Step 5	**ccm-manager sccp** **Example:** `Router(config)# ccm-manager sccp`	Enables the auto configuration process. This command immediately triggers the TFTP download of the XML configuration file.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits the current configuration mode.

Configuring SCCP on Cisco IOS Gateways

This task configures SCCP on the Cisco IOS gateway. SCCP messaging enables Cisco CallManager endpoint call control usingthe STCAPP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sccp ccm** {*ip-address* | *dns*} **identifier** *identifier-number* [**port** *port-number*][**version** *version-number*]
4. **sccp local** *interface-type interface-number*
5. **sccp ccm group** *group-number*
6. **associate ccm** *identifier-number* **priority** *priortiy-number*
7. **sccp**
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**sccp ccm** {*ip-address* | *dns*} **identifier** *identifier-number* [**port** *port-number*] [**version** *version-number*] **Example:** `Router(config)# sccp ccm 10.1.1.1 identifier 5`	Adds a Cisco Unified CallManager server to the list of available servers and sets various parameters. • *ip address*—Specifies the IP address of the Cisco CallManager server. • *identifier-number*—Identifies the Cisco Unified CallManager associated with the Cisco Unified CallManager *group-number* configured in Step 5 . Valid entries are from 1 to 65535. There is no default value.

	Command or Action	Purpose
Step 4	**sccp local** *interface-type interface-number* **Example:** `Router(config)# sccp local fastethernet 0/0`	Selects the local interface that the SCCP application uses to register with Cisco Unified CallManager. (This interface is the interface whose MAC address is specified for SCCP gateway registration using Cisco Unified CallManager autoconfiguration in Step 3 of the "DETAILED STEPS" section on page 9.) • *interface-type*—Specifies the interface type that the SCCP application uses to register with Cisco Unified CallManager . • *interface-number*—Specifies the interface number that the SCCP application uses to register with Cisco CallManager.
Step 5	**sccp ccm group** *group-number* **Example:** `Router(config)# sccp ccm group 1`	Creates a Cisco Unified CallManager group. • *group-number*—Associates the Cisco Unified CallManager group with the Cisco Unified CallManager group *identifier* configured in Step 3. Range is 1 to 65535. There is no default value.
Step 6	**associate ccm** *identifier-number* **priority** *priority-number* **Example:** `Router(config)# associate ccm 1 priority 1`	Associates a Cisco Unified CallManager with a Cisco Unified CallManager group. • *identifier-number*—Identifies the Cisco Unified CallManager associated with the Cisco Unified CallManager *group-number* configured in Step 5 . Valid entries are from 1 to 65535. There is no default value. • *priority-number*— Priority of the Cisco Unified CallManager within the Cisco Unified CallManager group. Range is 1 to 4. There is no default value. The highest priority is 1.
Step 7	**sccp** **Example:** `Router(config)# sccp`	Enables SCCP and its related applications.
Step 8	**exit** **Example:** `Router(config)# exit`	Exits the current configuration mode.

Configuring Modem Pass-through Calls on Cisco IOS Gateways

This task configures modem pass-through operation on the gateway. Perform this task to configure modem pass-through to allow for STU fallback, and also to enable interoperation with the SCCP gateway running versions of Cisco IOS software prior to Cisco IOS release 12.4(4)T that are not V.150.1-capable.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **voice service voip**
4. **modem passthrough nse** [**payload-type** *number*] **codec** {**g711ulaw** | **g711alaw**} [**redundancy** [**maximum-sessions** *sessions*]]
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice-service configuration mode and specifies VoIP encapsulation.
Step 4	**modem passthrough nse** [**payload-type** *number*] **codec** {**g711ulaw** | **g711alaw**} [**redundancy** [**maximum-sessions** *sessions*]] **Example:** `Router(config-voi-serv)# modem passthrough nse codec g711ulaw`	Configures modem pass-through over VoIP globally for all dial peers.
Step 5	**exit** **Example:** `Router(config-voi-serv)# exit`	Exits the current configuration mode.

Configuring Modem Transport Methods for STCAPP Devices

This task configures modem transport methods for STCAPP controlled devices. Perform this task to specify modem transport capability.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **stcapp register capability** *voice-port* [**both** | **modem-passthrough** | **modem-relay**]
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`stcapp register capability`** *`voice-port`* [**`both`** | **`modem-passthrough`** | **`modem-relay`**] **Example:** `Router(config)# stcapp register capability 1/0/0 modem-relay`	Specifies device modem transport capability. • *voice-port*—Specifies the voice interface slot number. • **both**—Specifies the device supports both modem relay and modem pass-through. • **modem-passthrough**—Specifies the device supports modem pass-through (voice band data). • **modem-relay**—Specifies the device supports V.150.1 modem relay.
Step 4	**`exit`** **Example:** `Router(config)# exit`	Exits the current configuration mode.

Configuring V.150.1 Modem Relay Parameters

This task configures optional V.150.1 modem-relay parameters. Configure these parameters to address specific network conditions for latency, redundancy and V.14 parameters.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **modem relay latency** *milliseconds*
5. **modem relay sse** [**redundancy** [**interval** *milliseconds* | **packet** *number*]] [**retries** *value*] [**t1** *milliseconds*]
6. **modem relay sprt v14** [**receive playback hold-time** *milliseconds* | **transmit hold-time** *milliseconds* | **transmit maximum hold-count** *characters*]
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice-service configuration mode and specifies VoIP encapsulation.
Step 4	**modem relay latency** *milliseconds* **Example:** `Router(config-voi-serv)# modem relay latency 250`	Specifies the estimated one-way delay across the IP network, in milliseconds
Step 5	**modem relay sse** [**redundancy** [**interval** *milliseconds* | **packet** number]] [**retries** ***value***] [**t1** *milliseconds*] **Example:** `Router(config-voi-serv)# modem relay sse redundancy packet 2`	Configures modem relay parameters.
Step 6	**modem relay sprt v14** [**receive playback hold-time** *milliseconds* | **transmit hold-time** *milliseconds* | **transmit maximum hold-count** *characters*] **Example:** `Router(config-voi-serv)# modem relay sprt v14 receive palyback holdtime 25`	Configures V.14 parameters.
Step 7	**exit** **Example:** `Router(config-voi-serv)# exit`	Exits the current configuration mode.

Configuring the Secure Communications Between IP-STE Endpoint and Trunk-Side STE Endpoint Feature

This task configures the Secure Communications Between IP-STE Endpoint and Trunk-Side STE Endpoint feature. Perform this task to enable V.150.1 modem relay on trunk-side or non-STCAPP-enabled gateways.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mgcp modem relay mode voip sse [redundancy {interval** *number*| **packet** *number*}][**retries** *value*] [**t1** *time*]
4. **mgcp modem relay voip sprt v14** {**receive playback hold-time** *milliseconds* | **transmit hold-time** *milliseconds* | **transmit maximum hold-count** *characters*}
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mgcp modem relay mode voip sse** [**redundancy** {**interval** *number*}{**packet** *number*}][**retries** *value*] [**t1** *time*] **Example:** `Router(config)# mgcp modem relay mode voip sse redundancy packet 5`	Specifies SSE modem-relay parameters.
Step 4	**mgcp modem relay voip sprt v14** {**receive playback hold-time** *milliseconds* | **transmit hold-time** *milliseconds* | **transmit maximum hold-count** *characters*} **Example:** `Router(config)# modem relay voip sprt v14 transmit hold-time 250`	Specifies SPRT modem-relay parameters.
Step 5	**exit** **Example:** `Router(config)# exit`	Exits the current configuration mode.

Verifying and Troubleshooting Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint

This task verifies and troubleshoots the Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint configuration.

SUMMARY STEPS

1. **show call active voice**
2. **show stcapp device voice-port** *port-number*
3. **show voice dsp**
4. **debug voip application stcapp all**
5. **debug voip application stcapp port**

DETAILED STEPS

Step 1 **show call active voice**

Use the **show call active voice** command to display call information for voice calls in progress:

```
Router# show call active voice

Telephony call-legs: 1
SIP call-legs: 0
H323 call-legs: 0
Call agent controlled call-legs: 1
SCCP call-legs: 0
Multicast call-legs: 0
Total call-legs: 2

 GENERIC:
SetupTime=526091080 ms
Index=1
PeerAddress=
PeerSubAddress=
PeerId=999110
PeerIfIndex=15
LogicalIfIndex=10
ConnectTime=526098210 ms
CallDuration=00:04:34 sec
CallState=4
CallOrigin=2
ChargedUnits=0
InfoType=speech
TransmitPackets=707
TransmitBytes=37724
ReceivePackets=705
ReceiveBytes=32174
ConnectionId=[0x5112503A 0x249A11D7 0x802383F6 0x82EFEC1F]
IncomingConnectionId=[0x5112503A 0x249A11D7 0x802383F6 0x82EFEC1F]
CallID=328
TxDuration=1570 ms
VoiceTxDuration=1570 ms
FaxTxDuration=0 ms
CoderTypeRate=modem-relay
NoiseLevel=0
ACOMLevel=0
OutSignalLevel=0
InSignalLevel=0
InfoActivity=0
ERLLevel=0
SessionTarget=
ImgPages=0
CallerName=
CallerIDBlocked=False
OriginalCallingNumber=
```

```
OriginalCallingOctet=0x0
OriginalCalledNumber=
OriginalCalledOctet=0x80
OriginalRedirectCalledNumber=
OriginalRedirectCalledOctet=0x0
TranslatedCallingNumber=
TranslatedCallingOctet=0x0
TranslatedCalledNumber=
TranslatedCalledOctet=0x80
TranslatedRedirectCalledNumber=
TranslatedRedirectCalledOctet=0x0
Modem Relay Local Rx Speed=9600 bps
Modem Relay Local Tx Speed=9600 bps
Modem Relay Remote Rx Speed=9600 bps
Modem Relay Remote Tx Speed=9600 bps
Modem Relay Phy Layer Protocol=v32
Modem Relay Ec Layer Protocol=v14
SPRTInfoFramesReceived=0
SPRTInfoTFramesSent=0
SPRTInfoTFramesResent=0
SPRTXidFramesReceived=0
SPRTXidFramesSent=0
SPRTTotalInfoBytesReceived=0
SPRTTotalInfoBytesSent=0
SPRTPacketDrops=0

DSPIdentifier=1/1:1

 GENERIC:
SetupTime=526098210 ms
Index=1
PeerAddress=
PeerSubAddress=
PeerId=0
PeerIfIndex=0
LogicalIfIndex=0
Separate H245 Connection=FALSE

H245 Tunneling=FALSE

SessionProtocol=other
ProtocolCallId=
SessionTarget=
OnTimeRvPlayout=1500
GapFillWithSilence=0 ms
GapFillWithPrediction=0 ms
GapFillWithInterpolation=0 ms
GapFillWithRedundancy=0 ms
HiWaterPlayoutDelay=65 ms
LoWaterPlayoutDelay=64 ms
TxPakNumber=78
TxSignalPak=0
TxComfortNoisePak=0
TxDuration=1570
TxVoiceDuration=1570
RxPakNumber=78
RxSignalPak=0
RxDuration=0
TxVoiceDuration=1550
VoiceRxDuration=1500
RxOutOfSeq=0
RxLatePak=0
RxEarlyPak=0
PlayDelayCurrent=64
```

```
PlayDelayMin=64
PlayDelayMax=65
PlayDelayClockOffset=2125253855
PlayDelayJitter=0
PlayErrPredictive=0
PlayErrInterpolative=0
PlayErrSilence=0
PlayErrBufferOverFlow=0
PlayErrRetroactive=0
PlayErrTalkspurt=0
OutSignalLevel=0
InSignalLevel=0
LevelTxPowerMean=0
LevelRxPowerMean=0
LevelBgNoise=0
ERLLevel=0
ACOMLevel=0
ErrRxDrop=0
ErrTxDrop=0
ErrTxControl=0
ErrRxControl=0
PlayoutMode = undefined
PlayoutInitialDelay=0 ms
ReceiveDelay=64 ms
LostPackets=0
EarlyPackets=0
LatePackets=0
SRTP = off
VAD = disabled
CoderTypeRate=modem-relay
CodecBytes=160
Media Setting=flow-around
CallerName=
CallerIDBlocked=False
OriginalCallingNumber=
OriginalCallingOctet=0x0
OriginalCalledNumber=
OriginalCalledOctet=0x0
OriginalRedirectCalledNumber=
OriginalRedirectCalledOctet=0x0
TranslatedCallingNumber=
TranslatedCallingOctet=0x0
TranslatedCalledNumber=
TranslatedCalledOctet=0x0
TranslatedRedirectCalledNumber=
TranslatedRedirectCalledOctet=0x0
MediaInactiveDetected=no
MediaInactiveTimestamp=
MediaControlReceived=
Username=
Telephony call-legs: 1
SIP call-legs: 0
H323 call-legs: 0
Call agent controlled call-legs: 1
SCCP call-legs: 0
Multicast call-legs: 0
Total call-legs: 2
```

Step 2 **show stcapp device voice-port** *port number*

Use the **show stcapp device voice-port** *port number* commmand to display detailed device-level information:

```
Router# show stcapp device voice-port 1/1/0

Port Identifier:  1/1/0
Device Type:      ALG
Device Id:        7
Device Name:      AN0D65D8DD40280
```

The following line shows the modem transport method configured on the voice port:

```
Modem Capability: Both
Device State:     IS
Diagnostic:       None
Directory Number: 5902
Dial Peer(s):     999110
Last Event:       STCAPP_CC_EV_CALL_DISCONNECT_DONE
Line State:       IDLE
```

Step 3 **show voice dsp**

Use the **show voice dsp** command to display the current status of all digital signal processor (DSP) voice channels:

```
Router# show voice dsp

----------------------------FLEX VOICE CARD 1 ------------------------------
                           *DSP VOICE CHANNELS*
DSP   DSP             DSPWARE CURR  BOOT                             PAK   TX/RX
TYPE  NUM CH CODEC    VERSION STATE STATE   RST AI VOICEPORT TS ABRT PACK COUNT
===== === == ======== ======= ===== ======= === == ========= == ==== ============
C5510 001 01 modem-re 4.5.909 busy  idle      0  0 1/1/0     05    0      298/353
                           *DSP SIGNALING CHANNELS*
DSP   DSP             DSPWARE CURR  BOOT                             PAK   TX/RX
TYPE  NUM CH CODEC    VERSION STATE STATE   RST AI VOICEPORT TS ABRT PACK COUNT
===== === == ======== ======= ===== ======= === == ========= == ==== ============
C5510 001 05 {flex}   4.5.909 alloc idle      0  0 1/1/3     02    0         15/0
C5510 001 06 {flex}   4.5.909 alloc idle      0  0 1/1/2     02    0         17/0
C5510 001 07 {flex}   4.5.909 alloc idle      0  0 1/1/1     06    0         31/0
C5510 001 08 {flex}   4.5.909 alloc idle      0  0 1/1/0     06    0        321/0
------------------------END OF FLEX VOICE CARD 1 ----------------------------
```

Step 4 **debug voip application stcapp all**

Use the **debug voip application stcapp all** command to display debugging information for the components of the STCAPP:

```
Router# debug voip application stcapp all

*Jan 11 12:24:18.443: stcapp_start
*Jan 11 12:24:18.443:     stcapp process started
*Jan 11 12:24:18.443: stcapp_init_symphony
*Jan 11 12:24:18.443:     CCAPI successfully initialized
*Jan 11 12:24:18.443: stcapp_init_rtp
*Jan 11 12:24:18.443: stcapp_vp_shut
*Jan 11 12:24:18.443: stcapp_port_up_down
*Jan 11 12:24:18.443:     RTP successfully brought in service
*Jan 11 12:24:18.443: stcapp_create_dcbs_from_dialpeers
*Jan 11 12:24:18.447: 1/1/0: stcapp_create_device
*Jan 11 12:24:18.447: 1/1/0:     Endpoint base name generated->AN0D65D8DD40280
*Jan 11 12:24:18.447: 1/1/0:     New dialpeer id: 999110
*Jan 11 12:24:18.447: 1/1/0:     Analog device is ready to be registered
```

The following lines show the codec subtype, which indicates the modem transport method: 0=None, 1=V.150.1, and 2=VBD:

```
*Jan 11 12:24:18.447: 1/1/0:     reg caps including codec=5 (g711ulaw) subtype=2
```

```
*Jan 11 12:24:18.447: 1/1/0:     reg caps including codec=1 (g729ar8) subtype=2
*Jan 11 12:24:18.447: 1/1/0:     reg caps including codec=5 (g711ulaw) subtype=1
*Jan 11 12:24:18.447: 1/1/0:     reg caps including codec=1 (g729ar8) subtype=1
*Jan 11 12:24:18.447: 1/1/0:     reg caps including codec=5 (g711ulaw) subtype=0
*Jan 11 12:24:18.447: 1/1/0:     reg caps including codec=6 (g711alaw) subtype=0
*Jan 11 12:24:18.447: 1/1/0:     reg caps including codec=1 (g729ar8) subtype=0
*Jan 11 12:24:18.447: 1/1/0:     reg caps including codec=2 (g726r16) subtype=0
*Jan 11 12:24:18.447: 1/1/0:     reg caps including codec=3 (g726r24) subtype=0
*Jan 11 12:24:18.447: 1/1/0:     reg caps including codec=4 (g726r32) subtype=0
*Jan 11 12:24:18.447: 1/1/0:     reg caps including codec=7 (g728) subtype=0
*Jan 11 12:24:18.447: 1/1/0:     reg caps including codec=8 (g723r63) subtype=0
*Jan 11 12:24:18.447: 1/1/0:     reg caps including codec=9 (g723r53) subtype=0
*Jan 11 12:24:18.447: 1/1/0:     reg caps including codec=12 (g729br8) subtype=0
*Jan 11 12:24:18.447: 1/1/0:     reg caps including codec=14 (g723ar63) subtype=0
*Jan 11 12:24:18.447: 1/1/0:     reg caps including codec=15 (g723ar53) subtype=0
*Jan 11 12:24:18.447: 1/1/0:     Device: AN0D65D8DD40280 Id: 7 successfully registered 
with CM
*Jan 11 12:24:18.447: 1/1/0:     New DCB hash count:1
*Jan 11 12:24:18.455: ==> Received event:STCAPP_DC_EV_DEVICE_REGISTER_DONE
*Jan 11 12:24:18.455: 1/1/0:     Device State:OOS
*Jan 11 12:24:18.455: 1/1/0: stcapp_dev_default_eh
*Jan 11 12:24:18.455: 1/1/0:     New State = INIT
*Jan 11 12:24:18.455: ==> Received event:STCAPP_DC_EV_DEVICE_CAP_REQ
*Jan 11 12:24:18.455: 1/1/0:     Device State:INIT
*Jan 11 12:24:18.455: 1/1/0: stcapp_cap_req_eh
*Jan 11 12:24:18.455: 1/1/0:     Sending dcDeviceHeadsetStatus for devID:7
*Jan 11 12:24:18.455: 1/1/0:     Sending dcDeviceButtonTemplateReq for devID:7
*Jan 11 12:24:18.455: 1/1/0:     No state change
*Jan 11 12:24:18.647: ==> Received event:STCAPP_DC_EV_DEVICE_BUTTON_TEMP_RES
*Jan 11 12:24:18.647: 1/1/0:     Device State:INIT
*Jan 11 12:24:18.647: 1/1/0: stcapp_button_templ_res_eh
*Jan 11 12:24:18.647: 1/1/0:     Sending dcDeviceLineStatReq for devID:7
*Jan 11 12:24:18.647: 1/1/0:     No state change
*Jan 11 12:24:18.647: ==> Received event:STCAPP_DC_EV_DEVICE_FORWARD_STAT_RES
*Jan 11 12:24:18.647: 1/1/0:     Device State:INIT
*Jan 11 12:24:18.647: 1/1/0: stcapp_forward_stat_res_eh
*Jan 11 12:24:18.647: 1/1/0:     lineNumber: 1
*Jan 11 12:24:18.647: 1/1/0:     forwardAllActive: 0
*Jan 11 12:24:18.647: 1/1/0:     forwardBusyActive: 0
*Jan 11 12:24:18.647: 1/1/0:     forwardNoAnswerActive: 0
*Jan 11 12:24:18.651: 1/1/0:     ForwardAllDirNumber:
*Jan 11 12:24:18.651: 1/1/0:     No state change
*Jan 11 12:24:18.651: ==> Received event:STCAPP_DC_EV_DEVICE_LINE_STAT_RES
*Jan 11 12:24:18.651: 1/1/0:     Device State:INIT
*Jan 11 12:24:18.651: 1/1/0: stcapp_line_stat_eh
*Jan 11 12:24:18.651: 1/1/0:     lineNumber: 1
*Jan 11 12:24:18.651: 1/1/0:     lineDirNumber: 5902
*Jan 11 12:24:18.651: 1/1/0:     display name: 5902
*Jan 11 12:24:18.651: 1/1/0:     Sending dcDeviceRegAvailableLines for devID:7
*Jan 11 12:24:18.651: 1/1/0:     Sending dcDeviceDateTimeReq for devID:7
*Jan 11 12:24:18.651: 1/1/0:     No state change
*Jan 11 12:24:18.823: ==> Received event:STCAPP_DC_EV_DEVICE_DEFINE_DATE_TIME_RES
*Jan 11 12:24:18.827: 1/1/0:     Device State:INIT
*Jan 11 12:24:18.827: 1/1/0: stcapp_define_date_time_eh
*Jan 11 12:24:18.827: 1/1/0:     New State = IS
*Jan 11 12:24:18.827: ==> Received event:STCAPP_DC_EV_DEVICE_DISPLAY_PROMPT_STATUS
*Jan 11 12:24:18.827: 1/1/0:     Device State:IS
*Jan 11 12:24:18.827: 1/1/0: stcapp_display_prompt_status_eh
*Jan 11 12:24:18.827: 1/1/0:     lineNumber: 0
*Jan 11 12:24:18.827: 1/1/0:     call reference: 0
*Jan 11 12:24:18.827: 1/1/0:     promptStatus: Your current options
*Jan 11 12:24:18.827: 1/1/0:     device control type: 3
*Jan 11 12:24:18.827: 1/1/0:     No state change
```

Step 5 **debug voip application stcapp port** *port-number*

Use the **debug voip application stcapp port** *port-number* command to enable STCAPP debugging for a specific port:

```
Router# debug voip application stcapp port 1/1/0


*Jan 11 12:37:48.631: ==> Received event:STCAPP_CC_EV_CALL_SETUP_IND
(evId:CC_EV_CALL_SETUP_IND) for CallId: 326
*Jan 11 12:37:48.631: 1/1/0:     Call State:IDLE
*Jan 11 12:37:48.631: 1/1/0: stcapp_setup_ind_eh
*Jan 11 12:37:48.631: 1/1/0: stcapp_get_ccb
*Jan 11 12:37:48.631: 1/1/0:     dcb->lcb[line_inst - 1].num_ccbs=0
*Jan 11 12:37:48.631: 1/1/0:     Acquired CCB 0x65D932B8 for device id:7
*Jan 11 12:37:48.631: 1/1/0:     num_ccbs++, num_ccbs=1
*Jan 11 12:37:48.631: 1/1/0:     Voice Setup: callID:326, vdb_ptr:666581AC
*Jan 11 12:37:48.631: 1/1/0:     Sending StationOffHook to CallManager
*Jan 11 12:37:48.631: 1/1/0:     Sending ccCallSetupAck to Symphony for voice call id:326
*Jan 11 12:37:48.631: 1/1/0:     New State = OFFHOOK
*Jan 11 12:37:48.643: 1/1/0:     No line (line=0) found... most likely old Call Ref: event
STCAPP_DC_EV_DEVICE_SET_RINGER
*Jan 11 12:37:48.643: ==> Received event:STCAPP_DC_EV_DEVICE_CALL_STATE_OFFHOOK
(evID:DC_EV_DEVICE_CALL_STATE_OFFHOOK)
*Jan 11 12:37:48.643: 1/1/0:     Call State:OFFHOOK
*Jan 11 12:37:48.643: 1/1/0: stcapp_cs_offhook_eh
*Jan 11 12:37:48.643: 1/1/0:     call_ref=16777250
*Jan 11 12:37:48.643: 1/1/0: stcapp_get_ccb_ptr
*Jan 11 12:37:48.643: 1/1/0: stcapp_get_ccb_ptr
*Jan 11 12:37:48.643: 1/1/0:     Using call_ref 0 to get ccb=0x65D932B8
*Jan 11 12:37:48.643: 1/1/0:     No state change
*Jan 11 12:37:48.643: ==> Received event:STCAPP_DC_EV_DEVICE_DISPLAY_PROMPT_STATUS
*Jan 11 12:37:48.643: 1/1/0:     Device State:IS
*Jan 11 12:37:48.643: 1/1/0: stcapp_display_prompt_status_eh
*Jan 11 12:37:48.643: 1/1/0:     lineNumber: 1
*Jan 11 12:37:48.643: 1/1/0:     call reference: 16777250
*Jan 11 12:37:48.643: 1/1/0:     promptStatus: Enter Number
*Jan 11 12:37:48.643: 1/1/0:     No state change
*Jan 11 12:37:48.643: ==> Received event:STCAPP_DC_EV_DEVICE_START_TONE
(evID:DC_EV_DEVICE_START_TONE)
*Jan 11 12:37:48.643: 1/1/0:     Call State:OFFHOOK
*Jan 11 12:37:48.643: 1/1/0: stcapp_start_tone_eh
*Jan 11 12:37:48.643: 1/1/0: stcapp_get_ccb_ptr
*Jan 11 12:37:48.643: 1/1/0:     call_ref=16777250, ccb=0x65D932B8, tone=8(0x8)
*Jan 11 12:37:48.643: 1/1/0:     Sending ccGenerateTone(8(0x8))
*Jan 11 12:37:48.643: 1/1/0:     Sending ccCallReportDigits
*Jan 11 12:37:48.643: 1/1/0:     No state change
*Jan 11 12:37:48.643: ==> Received event:STCAPP_CC_EV_CALL_REPORT_DIGITS_DONE
(evId:CC_EV_CALL_REPORT_DIGITS_DONE) for CallId: 326
*Jan 11 12:37:48.647: 1/1/0:     Call State:OFFHOOK
*Jan 11 12:37:48.647: 1/1/0: stcapp_report_digits_done_eh
*Jan 11 12:37:48.647: 1/1/0:     No state change
*Jan 11 12:37:52.643: ==> Received event:STCAPP_CC_EV_CALL_DIGIT_BEGIN
(evId:CC_EV_CALL_DIGIT_BEGIN) for CallId: 326
*Jan 11 12:37:52.643: 1/1/0:     Call State:OFFHOOK
*Jan 11 12:37:52.643: 1/1/0:     Uninteresting event
*Jan 11 12:37:52.683: ==> Received event:STCAPP_CC_EV_CALL_DIGIT_END
(evId:CC_EV_CALL_DIGIT_END) for CallId: 326
*Jan 11 12:37:52.683: 1/1/0:     Call State:OFFHOOK
*Jan 11 12:37:52.683: 1/1/0: stcapp_digit_end_eh
*Jan 11 12:37:52.683: 1/1/0:     Digit received is (5)
*Jan 11 12:37:52.683: 1/1/0:     Sending StationKeypadButton(5) to CallManager
*Jan 11 12:37:52.683: 1/1/0:     No state change
```

```
*Jan 11 12:37:52.687: ==> Received event:STCAPP_DC_EV_DEVICE_STOP_TONE
(evID:DC_EV_DEVICE_STOP_TONE)
*Jan 11 12:37:52.687: 1/1/0:     Call State:OFFHOOK
*Jan 11 12:37:52.687: 1/1/0: stcapp_stop_tone_eh
*Jan 11 12:37:52.687: 1/1/0:     call_ref=16777250
*Jan 11 12:37:52.687: 1/1/0: stcapp_get_ccb_ptr
*Jan 11 12:37:52.687: 1/1/0:     Sending ccGenerateTone(NULL)
*Jan 11 12:37:52.687: 1/1/0:     No state change
*Jan 11 12:37:52.775: ==> Received event:STCAPP_CC_EV_CALL_DIGIT_BEGIN
(evId:CC_EV_CALL_DIGIT_BEGIN) for CallId: 326
*Jan 11 12:37:52.775: 1/1/0:     Call State:OFFHOOK
*Jan 11 12:37:52.775: 1/1/0:     Uninteresting event
*Jan 11 12:37:52.823: ==> Received event:STCAPP_CC_EV_CALL_DIGIT_END
(evId:CC_EV_CALL_DIGIT_END) for CallId: 326
*Jan 11 12:37:52.823: 1/1/0:     Call State:OFFHOOK
*Jan 11 12:37:52.823: 1/1/0: stcapp_digit_end_eh
*Jan 11 12:37:52.823: 1/1/0:     Digit received is (8)
*Jan 11 12:37:52.823: 1/1/0:     Sending StationKeypadButton(8) to CallManager
*Jan 11 12:37:52.823: 1/1/0:     No state change
*Jan 11 12:37:52.923: ==> Received event:STCAPP_CC_EV_CALL_DIGIT_BEGIN
(evId:CC_EV_CALL_DIGIT_BEGIN) for CallId: 326
*Jan 11 12:37:52.923: 1/1/0:     Call State:OFFHOOK
*Jan 11 12:37:52.923: 1/1/0:     Uninteresting event
*Jan 11 12:37:52.963: ==> Received event:STCAPP_CC_EV_CALL_DIGIT_END
(evId:CC_EV_CALL_DIGIT_END) for CallId: 326
*Jan 11 12:37:52.963: 1/1/0:     Call State:OFFHOOK
*Jan 11 12:37:52.963: 1/1/0: stcapp_digit_end_eh
*Jan 11 12:37:52.963: 1/1/0:     Digit received is (0)
*Jan 11 12:37:52.963: 1/1/0:     Sending StationKeypadButton(0) to CallManager
*Jan 11 12:37:52.963: 1/1/0:     No state change
*Jan 11 12:37:53.063: ==> Received event:STCAPP_CC_EV_CALL_DIGIT_BEGIN
(evId:CC_EV_CALL_DIGIT_BEGIN) for CallId: 326
*Jan 11 12:37:53.063: 1/1/0:     Call State:OFFHOOK
*Jan 11 12:37:53.063: 1/1/0:     Uninteresting event
*Jan 11 12:37:53.103: ==> Received event:STCAPP_CC_EV_CALL_DIGIT_END
(evId:CC_EV_CALL_DIGIT_END) for CallId: 326
*Jan 11 12:37:53.103: 1/1/0:     Call State:OFFHOOK
*Jan 11 12:37:53.103: 1/1/0: stcapp_digit_end_eh
*Jan 11 12:37:53.103: 1/1/0:     Digit received is (2)
*Jan 11 12:37:53.103: 1/1/0:     Sending StationKeypadButton(2) to CallManager
*Jan 11 12:37:53.103: 1/1/0:     No state change
*Jan 11 12:37:53.235: ==> Received event:STCAPP_DC_EV_DEVICE_CALL_STATE_PROCEED
(evID:DC_EV_DEVICE_CALL_STATE_PROCEED)
*Jan 11 12:37:53.235: 1/1/0:     Call State:OFFHOOK
*Jan 11 12:37:53.235: 1/1/0: stcapp_cs_proceed_eh
*Jan 11 12:37:53.235: 1/1/0: stcapp_get_ccb_ptr
*Jan 11 12:37:53.235: 1/1/0:     Sending ccCallProceeding for voice call id:326
*Jan 11 12:37:53.235: 1/1/0:     Stopping the initial and inter digit timer!
*Jan 11 12:37:53.235: 1/1/0:     New State = PROCEEDING
*Jan 11 12:37:53.235: ==> Received event:STCAPP_DC_EV_DEVICE_CALL_INFO
(evID:DC_EV_DEVICE_CALL_INFO)
*Jan 11 12:37:53.235: 1/1/0:     Call State:PROCEEDING
*Jan 11 12:37:53.235: 1/1/0: stcapp_proceed_call_info_eh
*Jan 11 12:37:53.235: 1/1/0: stcapp_get_ccb_ptr
*Jan 11 12:37:53.239: 1/1/0:     No state change
*Jan 11 12:37:53.239: ==> Received event:STCAPP_DC_EV_DEVICE_START_TONE
(evID:DC_EV_DEVICE_START_TONE)
*Jan 11 12:37:53.239: 1/1/0:     Call State:PROCEEDING
*Jan 11 12:37:53.239: 1/1/0: stcapp_start_tone_eh
*Jan 11 12:37:53.239: 1/1/0: stcapp_get_ccb_ptr
*Jan 11 12:37:53.239: 1/1/0:     call_ref=16777250, ccb=0x65D932B8, tone=1(0x1)
*Jan 11 12:37:53.239: 1/1/0:     Sending ccCallAlert(signal:1) for voice call id:326
*Jan 11 12:37:53.239: 1/1/0:     No state change
```

```
*Jan 11 12:37:53.239: ==> Received event:STCAPP_DC_EV_DEVICE_CALL_STATE_RINGOUT
(evID:DC_EV_DEVICE_CALL_STATE_RINGOUT)
*Jan 11 12:37:53.239: 1/1/0:     Call State:PROCEEDING
*Jan 11 12:37:53.239: 1/1/0: stcapp_set_call_state_eh
*Jan 11 12:37:53.239: 1/1/0:     call_ref=16777250, call_state=2
*Jan 11 12:37:53.239: 1/1/0: stcapp_get_ccb_ptr
*Jan 11 12:37:53.239: 1/1/0:     No state change
*Jan 11 12:37:53.239: ==> Received event:STCAPP_DC_EV_DEVICE_DISPLAY_PROMPT_STATUS
*Jan 11 12:37:53.239: 1/1/0:     Device State:IS
*Jan 11 12:37:53.239: 1/1/0: stcapp_display_prompt_status_eh
*Jan 11 12:37:53.239: 1/1/0:     lineNumber: 1
*Jan 11 12:37:53.239: 1/1/0:     call reference: 16777250
*Jan 11 12:37:53.239: 1/1/0:     promptStatus: Ring Out
*Jan 11 12:37:53.239: 1/1/0:     No state change
*Jan 11 12:37:53.239: ==> Received event:STCAPP_DC_EV_DEVICE_CALL_INFO
(evID:DC_EV_DEVICE_CALL_INFO)
*Jan 11 12:37:53.239: 1/1/0:     Call State:PROCEEDING
*Jan 11 12:37:53.239: 1/1/0: stcapp_proceed_call_info_eh
*Jan 11 12:37:53.239: 1/1/0: stcapp_get_ccb_ptr
*Jan 11 12:37:53.239: 1/1/0:     No state change
*Jan 11 12:37:56.635: ==> Received event:STCAPP_DC_EV_DEVICE_STOP_TONE
(evID:DC_EV_DEVICE_STOP_TONE)
*Jan 11 12:37:56.635: 1/1/0:     Call State:PROCEEDING
*Jan 11 12:37:56.635: 1/1/0: stcapp_stop_tone_eh
*Jan 11 12:37:56.635: 1/1/0:     call_ref=16777250
*Jan 11 12:37:56.635: 1/1/0: stcapp_get_ccb_ptr
*Jan 11 12:37:56.639: 1/1/0:     Sending ccGenerateTone(NULL)
*Jan 11 12:37:56.639: 1/1/0:     No state change
*Jan 11 12:37:56.639: ==> Received event:STCAPP_DC_EV_MEDIA_OPEN_RCV_CHNL
(evID:DC_EV_MEDIA_OPEN_RCV_CHNL)
*Jan 11 12:37:56.639: 1/1/0:     Call State:PROCEEDING
*Jan 11 12:37:56.639: 1/1/0: stcapp_open_rcv_chnl_eh
*Jan 11 12:37:56.639: 1/1/0:     call_ref=16777250
*Jan 11 12:37:56.639: 1/1/0: stcapp_get_ccb_ptr
*Jan 11 12:37:56.639: 1/1/0: stcapp_set_up_voip_leg
*Jan 11 12:37:56.639: 1/1/0: stcapp_get_ccb_ptr
*Jan 11 12:37:56.639: 1/1/0:     Codec: 5 ptime :20, codecbytes: 160
```

The following lines show the modem transport method and modem parameters that will be used:

```
*Jan 11 12:37:56.639: 1/1/0:     CCM directive -> enabling modem relay
*Jan 11 12:37:56.639: 1/1/0:     MR parms: sprt_retries=10, sprt_latency=250,
sprt_rx_v14_pb_hold_time=32, sprt_tx_v14_hold_time=12, sprt_tx_v14_hold_count=22,
gw_xid=1, dictsize=1024, stringlen=16, compressdir=3, sse_red_interval=16,
sse_red_pkt_count=2, sse_t1=2100, sse_retries=5
*Jan 11 12:37:56.639: 1/1/0:     Info provided to RTPSPI - sess_mode 2, desired_qos 0,
codec 5, pkt_period 20, lr_port 17180
*Jan 11 12:37:56.639: 1/1/0:     Sending ccIFCallSetupRequest for voip leg
*Jan 11 12:37:56.639: 1/1/0:     ccIFCallSetRequest returned voip call id:327
*Jan 11 12:37:56.639: 1/1/0:     Sending dcDeviceOpenReceiveChannelAck
*Jan 11 12:37:56.639: 1/1/0:     ORChnlAck Info: codec:5, loc_port:17180, chnl_id:16777521
*Jan 11 12:37:56.639: 1/1/0:     New State = CONNECTING
*Jan 11 12:37:56.643: ==> Received event:STCAPP_DC_EV_DEVICE_CALL_STATE_CONNECTED
(evID:DC_EV_DEVICE_CALL_STATE_CONNECTED)
*Jan 11 12:37:56.643: 1/1/0:     Call State:CONNECTING
*Jan 11 12:37:56.643: 1/1/0: stcapp_set_call_state_eh
*Jan 11 12:37:56.643: 1/1/0:     call_ref=16777250, call_state=6
*Jan 11 12:37:56.643: 1/1/0: stcapp_get_ccb_ptr
*Jan 11 12:37:56.643: 1/1/0:     No state change
*Jan 11 12:37:56.643: ==> Received event:STCAPP_DC_EV_DEVICE_DISPLAY_PROMPT_STATUS
*Jan 11 12:37:56.643: 1/1/0:     Device State:IS
*Jan 11 12:37:56.643: 1/1/0: stcapp_display_prompt_status_eh
*Jan 11 12:37:56.643: 1/1/0:     lineNumber: 1
*Jan 11 12:37:56.643: 1/1/0:     call reference: 16777250
```

```
*Jan 11 12:37:56.643: 1/1/0:     promptStatus: Connected
*Jan 11 12:37:56.643: 1/1/0:     No state change
*Jan 11 12:37:56.643: ==> Received event:STCAPP_DC_EV_DEVICE_CALL_INFO
(evID:DC_EV_DEVICE_CALL_INFO)
*Jan 11 12:37:56.643: 1/1/0:     Call State:CONNECTING
*Jan 11 12:37:56.643: 1/1/0: stcapp_conn_call_info_eh
*Jan 11 12:37:56.647: 1/1/0: stcapp_get_ccb_ptr
*Jan 11 12:37:56.647: 1/1/0:     stcapp_call_info_eh::caller_name=
*Jan 11 12:37:56.647: 1/1/0:     Irrelevant CALL_INFO message is ignore!
*Jan 11 12:37:56.647: 1/1/0:     No state change
*Jan 11 12:37:56.647: ==> Received event:STCAPP_DC_EV_DEVICE_STOP_TONE
(evID:DC_EV_DEVICE_STOP_TONE)
*Jan 11 12:37:56.647: 1/1/0:     Call State:CONNECTING
*Jan 11 12:37:56.647: 1/1/0: stcapp_stop_tone_eh
*Jan 11 12:37:56.647: 1/1/0:     call_ref=16777250
*Jan 11 12:37:56.647: 1/1/0: stcapp_get_ccb_ptr
*Jan 11 12:37:56.647: 1/1/0:     Sending ccGenerateTone(NULL)
*Jan 11 12:37:56.647: 1/1/0:     No state change
*Jan 11 12:37:56.647: ==> Received event:STCAPP_DC_EV_MEDIA_OPEN_XMT_CHNL
(evID:DC_EV_MEDIA_OPEN_XMT_CHNL)
*Jan 11 12:37:56.647: 1/1/0:     Call State:CONNECTING
*Jan 11 12:37:56.647: 1/1/0: stcapp_start_media_eh
*Jan 11 12:37:56.647: 1/1/0:     call_ref=16777250
*Jan 11 12:37:56.647: 1/1/0: stcapp_get_ccb_ptr
*Jan 11 12:37:56.647: 1/1/0:     New State = ACTIVE_PENDING
*Jan 11 12:37:56.647: ==> Received event:STCAPP_CC_EV_CALL_CONNECTED
(evId:CC_EV_CALL_CONNECTED) for CallId: 327
*Jan 11 12:37:56.647: 1/1/0:     Call State:ACTIVE_PENDING
*Jan 11 12:37:56.647: 1/1/0: stcapp_call_connected_eh
*Jan 11 12:37:56.647: 1/1/0: stcapp_create_conference
*Jan 11 12:37:56.647: 1/1/0:     Sending ccConferenceCreate to Symphony
*Jan 11 12:37:56.651: 1/1/0:     Conference created. voice call id:326, voip call id:327
*Jan 11 12:37:56.651: 1/1/0:     No state change
*Jan 11 12:37:56.651: ==> Received event:STCAPP_CC_EV_CONF_CREATE_DONE
(evId:CC_EV_CONF_CREATE_DONE) for CallId: 326
*Jan 11 12:37:56.651: 1/1/0:     Call State:ACTIVE_PENDING
*Jan 11 12:37:56.651: 1/1/0: stcapp_active_pending_eh
*Jan 11 12:37:56.651: 1/1/0:     Sending ccCallModify for voice call id:326
*Jan 11 12:37:56.651: 1/1/0:     codec=5, vad=0
*Jan 11 12:37:56.651: 1/1/0:     Stopping the initial and inter digit timer!
*Jan 11 12:37:56.651: 1/1/0:     Sending ccCallModify for voip call id:327
*Jan 11 12:37:56.651: 1/1/0:     Updated SMT info to RTPSPI - sess_mode:3,desired_qos:0,
codec:5, pkt_period:20,rem_port:18968 vad:0 ip_tos:4
*Jan 11 12:37:56.655: 1/1/0:     No state change
*Jan 11 12:37:56.655: ==> Received event:STCAPP_CC_EV_VOICE_MODE_DONE
(evId:CC_EV_VOICE_MODE_DONE) for CallId: 326
*Jan 11 12:37:56.655: 1/1/0:     Call State:ACTIVE_PENDING
*Jan 11 12:37:56.655: 1/1/0:     Uninteresting event
*Jan 11 12:37:56.655: ==> Received event:STCAPP_CC_EV_CALL_REPORT_DIGITS_DONE
(evId:CC_EV_CALL_REPORT_DIGITS_DONE) for CallId: 326
*Jan 11 12:37:56.655: 1/1/0:     Call State:ACTIVE_PENDING
*Jan 11 12:37:56.655: 1/1/0:     Uninteresting event
*Jan 11 12:37:56.655: ==> Received event:STCAPP_CC_EV_CALL_MODIFY_DONE
(evId:CC_EV_CALL_MODIFY_DONE) for CallId: 326
*Jan 11 12:37:56.655: 1/1/0:     Call State:ACTIVE_PENDING
*Jan 11 12:37:56.655: 1/1/0: stcapp_default_eh
*Jan 11 12:37:56.655: 1/1/0:     call_ref=0, call_state=0
*Jan 11 12:37:56.655: 1/1/0:     New State = ACTIVE
*Jan 11 12:37:56.655: ==> Received event:STCAPP_CC_EV_CALL_MODIFY_DONE
(evId:CC_EV_CALL_MODIFY_DONE) for CallId: 327
*Jan 11 12:37:56.655: 1/1/0:     Call State:ACTIVE
*Jan 11 12:37:56.655: 1/1/0:     Uninteresting event
*Jan 11 12:37:59.963: ==> Received event:STCAPP_CC_EV_CALL_FEATURE_OFFHOOK
(evId:CC_EV_CALL_FEATURE) for CallId: 326
```

```
*Jan 11 12:37:59.963: 1/1/0:     Call State:ACTIVE
*Jan 11 12:37:59.963: 1/1/0: stcapp_call_feature_eh
*Jan 11 12:37:59.963: 1/1/0:     lcb->num_ccbs = 1
*Jan 11 12:37:59.963: 1/1/0:     No CC_FEATURE match!
*Jan 11 12:37:59.967: 1/1/0:     No state change... call remaining
```

Configuration Examples for Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint

This section provides the following configuration example:

- Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint: Example, page 29

Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint: Example

The following example shows how to configure the Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint feature:

```
Router# show running-config

*Jan  9 08:51:42.763: %SYS-5-CONFIG_I: Configured from console by console
Building configuration...

Current configuration : 3502 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
logging buffered 25000000 debugging
!
username lab password 0 lab
no aaa new-model
!
resource manager
clock timezone EST -5
no network-clock-participate slot 1
no network-clock-participate slot 2
ip subnet-zero
no ip cef
!
!
no ip dhcp use vrf connected
!
!
no ip domain lookup
```

```
ip domain name cisco.com
ip name-server 172.18.138.14
no ip ips deny-action ips-interface
```

The following lines show STCAPP device registration:

```
stcapp register capability 1/1/0 modem-relay
stcapp register capability 1/1/1 modem-passthrough
stcapp register capability 1/1/2 both
```

The following lines show that the STCAPP is enabled. The Cisco Unified CallManager group number must match the Cisco Unified CallManager identifier configured for the SPI using the **sccp ccm-group** command.

```
stcapp ccm-group 1
stcapp
!
no ftp-server write-enable
```

The next line shows the ISDN switch type configuration for BRI voice ports:

```
isdn switch-type basic-ni
voice-card 1
no dspfarm
!
voice-card 2
dspfarm
!
!
voice service voip
```

The next line shows modem pass-through configuration:

```
modem passthrough nse codec g711ulaw
```

The next lines show modem-relay parameter configuration:

```
modem relay nse codec g711ulaw
modem relay latency 250
modem relay sse redundancy interval 16
modem relay sse redundancy packet 2
modem relay sse t1 2100
modem relay sse retries 5
modem relay sprt v14 receive playback hold-time 32
modem relay sprt v14 transmit hold-time 12
modem relay sprt v14 transmit maximum hold-count 22
!
interface FastEthernet0/0
ip address 10.2.6.10 255.255.255.0
duplex auto
speed auto
no clns route-cache
!
interface FastEthernet0/1
no ip address
duplex auto
speed auto
no clns route-cache
!
```

The following lines show the required configuration for the ISDN BRI interface:

```
interface BRI1/0
no ip address
isdn switch-type basic-ni
```

The following **isdn timer** command is not user-configured. The command is automatically generated based on the presence of the **isdn switch-type basic ni** and **isdn protocol-emulate network** commands.

```
isdn timer t309 30000
isdn overlap-receiving T302 16000
isdn protocol-emulate network
isdn point-to-point-setup
isdn layer1-emulate network
isdn spid1 7705
isdn incoming-voice voice
```

The next line shows automatically enabled support for ISDN calling name display information.

```
isdn supp-service name calling
isdn sending-complete
```

The following **isdn skipsend-idverify** command is not user-configured. The command is automatically generated based on the presence of the **isdn switch-type basic ni** and **isdn protocol-emulate network** commands.

```
isdn skipsend-idverify
line-power
no clns route-cache
!
interface BRI1/1
no ip address
isdn switch-type basic-net3
isdn overlap-receiving T302 16000
isdn protocol-emulate network
isdn layer1-emulate network
isdn incoming-voice voice
isdn supp-service name calling
isdn skipsend-idverify
no clns route-cache
!
ip classless
ip route 0.0.0.0 0.0.0.0 FastEthernet0/0
!
!
ip http server
no ip http secure-server
!
ip access-list extended jeff
!
access-list 1 permit 10.6.6.31
access-list 107 deny   ip host 10.6.6.20 any log
access-list 107 permit ip any any
!
!
control-plane
!
!
voice-port 1/0/0
timeouts initial 60
timeouts interdigit 60
!
voice-port 1/0/1
timeouts initial 60
!
voice-port 1/0/2
timeouts initial 60
timeouts interdigit 60
!
voice-port 1/0/3
```

```
timeouts initial 60
!
voice-port 1/1/0
timeouts initial 60
!
voice-port 1/1/1
compand-type a-law
timeouts initial 60
bearer-cap Speech
!
ccm-manager music-on-hold
```

The following lines show the Cisco Unified CallManager configuration for automatic download capability of XML dial-peer configuration files. The IP address should match the address of the TFTP server from which to download XML files. The SCCP interface should match the interface specified for Cisco Unified CallManager registration.

```
ccm-manager config server 10.6.6.31
ccm-manager sccp local FastEthernet0/0
ccm-manager sccp
!
!
```

The following lines show SCCP configuration. The IP address should match the address of the Cisco Unified CallManager server.

```
sccp local FastEthernet0/0
sccp ccm 10.2.6.101 identifier 2
sccp ccm 10.2.6.100 identifier 1
sccp ip precedence 1
sccp
```

The following SCCP Cisco CallManager group number must match the STCAPP Cisco Unified CallManager group number specified using the **stcapp ccm-group** command:

```
sccp ccm group 1
associate ccm 2 priority 2
associate ccm 1 priority 1
!
!
```

The following lines show STCAPP-controlled dial peers. By default, Cisco Unified CallManager autoconfigured STCAPP dial peers begin with the prefix 999.

```
dial-peer voice 999100 pots
service stcapp
port 1/0/0
!
dial-peer voice 999101 pots
service stcapp
port 1/0/1
!
dial-peer voice 999103 pots
service stcapp
port 1/0/3
!
dial-peer voice 999110 pots
service stcapp
port 1/1/0
!
dial-peer voice 999111 pots
service stcapp
port 1/1/1
!
dial-peer voice 999102 pots
service stcapp
```

```
port 1/0/2
!
!
line con 0
exec-timeout 0 0
length 0
line aux 0
line vty 0 4
password cisco
login
!
!
end
```

Additional References

The following sections provide references related to the Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint feature.

Related Documents

Related Topic	Document Title
V.150.1 specification	ITU-T Recommendation V.150.1
Simple Packet Relay Transport protocol	ITU-T Recommendation V.150.1, Annex B
State Signaling Events protocol	ITU-T Recommendation V.150.1, Annex C
MLPP and other Cisco Unified CallManager controlled features	*MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways*
SCCP gateway controlled supplementary features	*SCCP Controlled Analog (FXS) Ports with Supplementary Features in Cisco IOS Gateways*
CiscoUnified CallManager administration	*Cisco CallManager Administration Guide*, Release 4.1(2)
Cisco Unified CallManager interoperability	*Cisco CallManager and Cisco IOS Interoperability Guide*
Cisco Unified CallManager security configuration	*Cisco CallManager Security Guide*, Release 4.1.2
ISDN configuration	*Cisco IOS ISDN Voice Configuration Guide,* Release 12.3, "Basic ISDN Voice-Interface Configuration" chapter
Cisco IOS debugging	*Cisco IOS Debug Command Reference*
Cisco IOS voice configuration	*Cisco IOS Voice Configuration Library*
Cisco IOS voice command reference	*Cisco IOS Voice Command Reference*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Voice Command Reference* at http://www.cisco.com/en/US/docs/ios/voice/command/reference/vr_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html.

- **debug voip application stcapp all**
- **debug voip application stcapp port**
- **modem relay sprt v14**
- **modem relay sse**
- **show stcapp device**

- **show voice dsp**
- **stcapp register capability**

Feature Information for Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint

Table 3 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 3 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 3 Fe ature Information for the Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint feature

Feature Name	Releases	Feature Information
Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint	12.4(4)T 12.4(9)T	This feature was introduced. In Cisco IOS Release 12.4(9)T, this feature was implemented on the following platforms: Cisco 2801, Cisco 2811, Cisco 2821, Cisco 2851, Cisco 3825, Cisco 3845, Cisco VG 224.

Glossary

API—application program interface. The means by which an application program “talks” to communications software.

BRI—Basic Rate Interface. ISDN interface composed of two B channels and one D channel for circuit-switched communication of voice, video, and data.

CCAPI—call control API.

DSP—digital signal processor.

FNBD—Future Narrow Band Digital Signaling.

FXS— Foreign Exchange Station. An FXS interface connects directly to a standard telephone and supplies ring, voltage, and dial tone. Cisco’s FXS interface is an RJ-11 connector that allows connections to basic telephone service equipment, keysets, and PBXs.

H.323—H.323 protocol allows dissimilar communication devices to communicate with each other by using a standardized communication protocol. H.323 defines a common set of codecs, call setup and negotiating procedures, and basic data transport methods.

IP—Internet Protocol.

IP-STE—Internet Protocol secure terminal equipment.

ISDN—Integrated Services Digital Network. Communication protocol offered by telephone companies that permits telephone networks to carry data, voice, and other source traffic.

MAC address—Standardized data link layer address that is required for every port or device that connects to a LAN. Other devices in the network use these addresses to locate specific ports in the network and to create and update routing tables and data structures. MAC addresses are 6 bytes long and are controlled by the IEEE. Also known as a hardware address, MAC layer address, and physical address.

MGCP—Media Gateway Control Protocol. A merging of the IPDC and SGCP protocols.

MLPP—Multilevel Precedence and Preemption.

MoIP—Modem over IP.

NM—network module.

NSE—named signaling events.

RTP—Real-Time Transport Protocol. Commonly used with IP networks. RTP is designed to provide end-to-end network transport functions for applications transmitting real-time data such as audio, video, or simulation data over multicast or unicast network services. RTP provides such services as payload type identification, sequence numbering, time-stamping, and delivery monitoring to real-time applications.

SCCP—Skinny Client Control Protocol.

SPRT—Simple Packet Relay Transport.

SRST—Survivable Remote Site Telephony.

SSE—state signaling events.

STCAPP—SCCP Telephony Control Application.

STE—Secure Terminal Equipment.

STU—Secure Telephone Unit.

TFTP— Trivial File Transfer Protocol. Simplified version of FTP that allows files to be transferred from one computer to another over a network, usually without the use of client authentication (for example, username and password).

UDP—User Datagram Protocol

V.150.1—Modem relay specification.

VBD—voice band data (modem pass-through).

VIC—Voice interface card.

VoIP— Voice over IP. The capability to carry normal telephony-style voice over an IP-based internet with POTS-like functionality, reliability, and voice quality. VoIP enables a router to carry voice traffic (for example, telephone calls and faxes) over an IP network. In VoIP, the DSP segments the voice signal into frames, which then are coupled in groups of two and stored in voice packets. These voice packets are transported using IP in compliance with ITU-T specification H.323.

XML— eXtensible Markup Language. A standard maintained by the World Wide Web Consortium (W3C). It defines a syntax that lets you create markup languages to specify information structures. Information structures define the type of information, for example, subscriber name or address, not how the information looks (bold, italic, and so on). External processes can manipulate these information structures and publish them in a variety of formats. Text markup language designed to enable the use of SGML on the World Wide Web. XML allows you to define your own customized markup language.