CHAPTER 38

Secure Media and SRTP Passthrough

Cisco Unified Border Element (SP Edition) supports two methods of encrypted data streams—Secure Real-Time Protocol (SRTP) Passthrough and Secure Media. The preferred method is to use SRTP Passthrough because it allows the end points themselves to signal their encryption capabilities.

The Secure Media feature is enabled on the global level for all calls and is disabled by default. When Secure Media is turned on globally, the SBC assumes that all end points are going to use encrypted data streams regardless of the actual end point capabilities.

Starting with Cisco IOS XE Release 2.6, using the Unsignaled Secure Media feature you are able to configure secure media on a granular level for specific calls and adjacencies using Call Admission Control (CAC) table entry commands.

You can configure SRTP Passthrough on a granular basis using CAC policy.

Regardless of the method used to configure the Cisco Unified Border Element (SP Edition) to accept encrypted media packets, Cisco Unified Border Element (SP Edition) reserves additional bandwidth to ensure these packets pass through. Typically, the bandwidth of a media stream is determined by the codecs that the endpoints use. However, the use of the encryption in the media streams increases the packet size. As a rule of thumb, the bandwidth requirements are 10% more than the unencrypted codec. However, this increase is not reflected in the media flow statistics.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to the *Cisco Unified Border Element (SP Edition) Command Reference: Unified Model* at:

http://www.cisco.com/en/US/docs/ios/sbc/command/reference/sbcu_book.html

For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

Feature History for Secure Media and SRTP Passthrough

Release	Modification
Cisco IOS XE Release 2.4	These features were introduced on the Cisco ASR 1000 Series Aggregation Services Routers.
Cisco IOS XE Release 2.6	The Unsignaled Secure Media feature was introduced to allow configuration at a granular level using CAC table entry commands. With the introduction of this feature, the Configuring Secure Media-Global Level feature has been deprecated.

Cisco IOS XE Release 3.1S	The SRTP to RTP Interworking and SRTP Passthrough features were added.
Cisco IOS XE Release 3.4S	The SRTP Support for RTCP Multiplexed with RTP and for SSRC-Based Multiplexing feature was added.

Contents

This chapter contains the following sections:

Prerequisites for Secure Media and SRTP Passthrough

The following prerequisites are required to implement both features:

Before implementing the Secure Media and SRTP Passthrough features, Cisco Unified Border Element (SP Edition) must already be configured.

Restrictions for Secure Media

The following is a restriction for Global and Unsignaled Secure Media:

- With this feature enabled, RTCP related statistics displayed in the **show sbc dbe media-flow-stats** command are displayed as unknown.

The following is a restriction for Unsignaled (granular-level) Secure Media:

- Both caller and callee sides of the call need to be configured with the **caller secure-media** and **callee secure-media** commands. If only one leg of the call is configured, then the call will fail.

Note In some scenarios, the **branch** command can be used as an alternative to the **caller** and **callee** commands. The **branch** command has been introduced in Release 3.5.0. See the ?$paranum>Configuring Directed Nonlimiting CAC Policies? section on page 7-37 for information about this command.

Information About Secure Media

Typically, an endpoint will indicate that the media traffic is encrypted through the SIP signaling. The encryption keys are either exchanged through Session Description Protocol (SDP) or using the Datagram Transport Layer Security (DTLS) mechanism.

In Cisco IOS XE Release 2.4 and Release 2.5, Cisco Unified Border Element (SP Edition) interworked with end points or SIP device that use encrypted media (DTLS or Secure-RTP [SRTP]), but the endpoints did not indicate this in the SIP signaling. In those earlier releases, the SBC supported a globally enabled Secure Media configuration where all calls on the SBC were treated as consisting of SRTP media. Even though the endpoint may not have signaled for SRTP media, media pinholes were created as if the traffic was SRTP. A global configuration under the SBE submode indicates that the endpoints are using encrypted SRTP media, but they will not be using SIP signaling to communicate and negotiate as such. The consequence of this configuration being applied at a global level is that even for flows that are not encrypted, additional bandwidth is reserved and RTP and RTCP checking and validations are disabled.

When interworking with a SIP device that does not have full support for signaling SRTP media streams, the SBC cannot know in advance that the media will be SRTP because it is not signaled as SRTP. Starting with Cisco IOS XE Release 2.6, the Unsignaled Secure Media feature allows the SBC to successfully interoperate with SIP devices that generate SRTP media but signal this as a regular RTP media stream.

You are able to configure the SBC to know which SIP devices it communicates with require support for unsignaled SRTP. Such SIP devices are assumed to always send SRTP media. Minimally you must granularly configure all devices on a given adjacency to require support for SRTP. In configuring secure media on a granular level, you use Call Admission Control (CAC) table entry commands. We highly recommend you use the granular level configuration because, instead of turning on secure media globally, you can specify the calls and adjacencies where you want to use secure media. Using the granular option of Unsignaled Secure Media, additional bandwidth is allocated and RTCP no check is performed only for those calls that match the CAC match criteria. Unsignaled Secure Media, like the global option, is disabled by default.

In Cisco IOS XE Release 2.6, when you configure the SBC to allow unsignaled SRTP media on a granular level for adjacencies, observe these recommended guidelines:

- If the adjacencies are trusted to allow secure calls—use either the **security trusted-encrypted** or **security trusted-unencrypted** command to configure both adjacencies where caller and callee side are located for SRTP passthrough first. Both sides need to be configured because it is a passthrough. This is the default where SRTP calls are allowed between trusted adjacencies.
- If an adjacency is not trusted, you can still configure granular-level Unsignaled Secure Media on that adjacency by configuring SRTP Passthrough in a CAC configuration on the untrusted adjacency. Use the **srtp support** command to allow an SRTP call on the adjacency where the CAC policy is applied.
- Configure both legs of the call to enable the granular-level Unsignaled Secure Media—use the **caller secure-media** command on the caller side, and the **callee secure-media** command on the callee side.

Note In some scenarios, the **branch** command can be used as an alternative to the **caller** and **callee** commands. The **branch** command has been introduced in Release 3.5.0. See the ?$paranum>Configuring Directed Nonlimiting CAC Policies? section on page 7-37 for information about this command.

For information on the configuration steps, see the ?$paranum>Configuring Unsignaled Secure Media at a Granular Level? section on page 38-13.

Information About SRTP Passthrough

Cisco Unified Border Element (SP Edition) supports SIP calls between endpoints using Transport Layer Security (TLS) for SIP signaling encryption and Secure Real-Time Protocol (SRTP) to provide RTP media encryption. However, these two encryption mechanisms may not be deployed simultaneously, depending on the required call flow invoked on the associated configuration.

Before delving further into SRTP passthrough configuration, it would be useful to understand the two concepts—the *trusted* vs. *untrusted* and *encrypted* vs. *unencrypted*.

The “trusted” implies that an associated adjacency is trusted to allow secure calls. Calls to a standard SIP: URI will be accepted. Calls to a secure SIPS: URI will be accepted and routed over a trusted adjacency (encrypted or unencrypted). The “untrusted” indicates that an associated adjacency is not trusted to carry secure calls. The calls to standard SIP: URI will be accepted. Calls to a secure SIPS: URI will be rejected immediately.

The “encrypted” implies that an associated adjacency uses TLS for SIP signaling and the “unencrypted” implies that an associated adjacency does not use TLS for SIP signaling.

The trusted/untrusted are configured in conjunction with encrypted/unencrypted as outlined in the following four (4) combinations. This is invoked using the **security** command:

- **untrusted-unencrypted**: The adjacency is untrusted and unencrypted. The adjacency is not trusted to carry secure SIP calls (calls with SIPS URI) and it does not use TLS encryption for SIP signaling.
- **untrusted-encrypted**: The adjacency is untrusted and encrypted. The adjacency is not trusted to carry secure SIP calls (calls with SIPS URI) and it does use TLS encryption for SIP signaling.
- **trusted-unencrypted**: The adjacency is trusted and unencrypted. The adjacency is trusted to carry secure SIP calls (calls with SIPS URI) and it does not use TLS encryption for SIP signaling.
- **trusted-encrypted**: The adjacency is trusted and encrypted. The adjacency is trusted to carry secure SIP calls (calls with SIPS URI) and it does use TLS encryption for SIP signaling.

When Cisco Unified Border Element (SP Edition) comes up, the default is to allow SRTP calls to pass through on the trusted interfaces.

The following are conditions of the SRTP Passthrough feature:

- SRTP Passthrough must be configured on both legs of the call. If the target adjacency does not support SRTP Passthrough, then the call is rejected by error message 415 (Unsupported Media Type).
- "m= .. RTP/SAVP .." and a="crypto:..." fields coming in on an Invite from one adjacency are passed on in an Invite to the target adjacency.
- “m= …RTP/SAVP…” is a required field in the Invite to trigger SRTP Passthrough behavior in the SBC.

The following shows a sample SRTP Invite and Response call flow from endpoints, as described in RFC-4568.

Offerer sends:

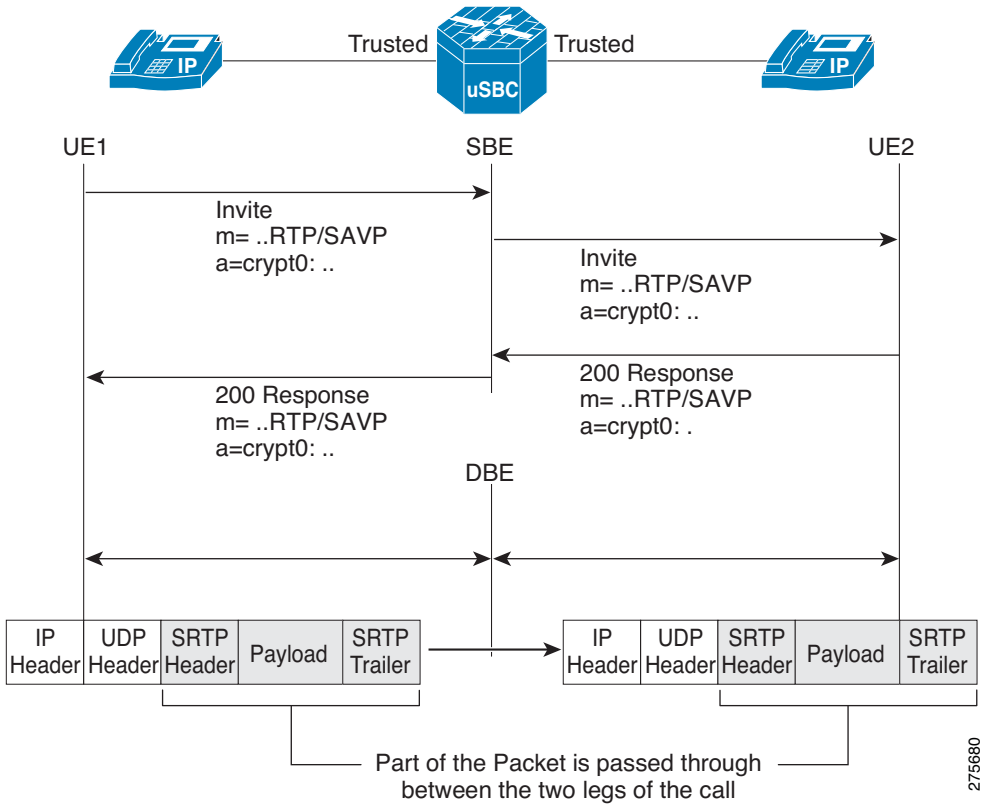
```
v=0
o=sam 2890844526 2890842807 IN IP4 10.47.16.5
s=SRTP Discussion
i=A discussion of Secure RTP
u=http://www.example.com/seminars/srtp.pdf
e=marge@example.com (Marge Simpson)
c=IN IP4 168.2.17.12
t=2873397496 2873404696
m=audio 49170 RTP/SAVP 0
a=crypto:1 AES_CM_128_HMAC_SHA1_80
 inline:WVNfX19zZW1jdGwgKCkgewkyMjA7fQp9CnVubGVz|2^20|1:4
 FEC_ORDER=FEC_SRTP
a=crypto:2 F8_128_HMAC_SHA1_80
 inline:MTIzNDU2Nzg5QUJDREUwMTIzNDU2Nzg5QUJjZGVm|2^20|1:4;
 inline:QUJjZGVmMTIzNDU2Nzg5QUJDREUwMTIzNDU2Nzg5|2^20|2:4
 FEC_ORDER=FEC_SRTP
```

Answerer replies:

```
v=0
o=jill 25690844 8070842634 IN IP4 10.47.16.5
s=SRTP Discussion
i=A discussion of Secure RTP
u=http://www.example.com/seminars/srtp.pdf
e=homer@example.com (Homer Simpson)
c=IN IP4 168.2.17.11
t=2873397526 2873405696
m=audio 32640 RTP/SAVP 0
a=crypto:1 AES_CM_128_HMAC_SHA1_80
 inline:PS1uQCVeeCFCanVmcjkpPywjNWhcYD0mXXtxaVBR|2^20|1:4
```

Figure 38-1 diagram illustrates an SRTP Passthrough Call Flow.

Figure 38-1 SRTP Passthrough Call Flow



The SRTP Passthrough feature defines a new Call Admission Control (CAC) entry variable, called "srtp transport," in the admission control table. If you configure the "srtp transport" variable, then CAC policy has the option to set the policy for the adjacency to either "allowed," "disallowed," or "trust only."

Calls using SRTP Passthrough are allowed on the adjacencies specified by the policy. Where there are conflicting policies, "disallowed" overrides "allowed" which overrides "trusted-only." If you configure the CAC policy, but you do not define the "srtp transport" variable, then the CAC policy takes the default value of "trusted-only" and restricts the SRTP calls between trusted endpoints.

See the **srtp support** command which sets the adjacency CAC policy for more information. The **no** form of the command sets the "srtp support" variable to "trusted-only." The **show sbc sbe cac-policy-set table entry** command is modified to display a "SRTP Transport" field and whether the policy for the adjacency is to allow, disallow, or trust only for SRTP Transport.

You can set the CAC policy to allow SRTP passthrough and allow configuration of certain security policing, such as the following:

- Preventing secure calls on a given adjacency
- Ensuring that all media sent over a given adjacency is secure
- Ensuring that secure streams are signaled over secure SIP adjacencies.

Information About SRTP to RTP Interworking and SRTP Passthrough

Secure Real-time Transport Protocol (SRTP) to Real-time Transport Protocol (RTP) interworking is supported on Session Border Controller (SBC) services on Cisco ASR 1000 Series Aggregation Services Routers.

System Administrators may configure SRTP to RTP interworking to enable their networks to communicate with other networks and add additional security to a network. SRTP to RTP interworking allows networks that use SRTP to accept calls from networks that use RTP.

The SRTP to RTP interworking feature provides SBC with the ability to encrypt and decrypt data streams to and from both types of networks, SRTP networks and RTP networks.

SRTP to RTP interworking can be deployed on both User to Network Interfaces (UNI) and Network to Network Interfaces (NNI).

Features Supported

The following SRTP to RTP interworking features are supported by SBC:

- SBC-generated SRTP encryption and decryption keys.
- Configurable policies, for SRTP pass-through, termination, and re-origination when both caller and callee CAC policies support SRTP.
- SRTP to RTP interworking in distributed DBE mode via H.248.
- PD logs with information for verifying SBC call handling for different SRTP preference and policy settings. (Encryption keys are not displayed in PD logs.)
- Stateful Switchover (SSO) for SRTP streams.

CAC policies can support the following types of SRTP to RTP interworking:

- RTP-only
- SRTP-only
- SRTP-optional
- SRTP-prefer

When a CAC policy uses SRTP-only:

- All media streams associated to that CAC policy use SRTP. The SRTP stream is end-to-end if the peer adjacency supports SRTP. If the peer adjacency does not support SRTP, or if the policy configuration is set to terminate and re-originate, SBC performs the necessary SRTP encryption and decryption.
- SBC rejects incoming RTP calls and sends the appropriate response code.

When a CAC policy uses RTP-only:

- All media streams associated to that CAC policy use RTP. The RTP stream is end-to-end if the peer adjacency does not require SRTP. If the peer adjacency requires SRTP, SBC perform RTP to SRTP interworking.
- SBC rejects incoming SRTP calls and sends the appropriate response code.

When a CAC policy uses SRTP-optional:

- SRTP-optional is by negotiation on inbound calls.
- SBC accepts both incoming RTP and incoming SRTP calls.
- No RTP to SRTP interworking is needed for incoming RTP calls unless the callee CAC policy uses SRTP-only.
- No SRTP encryption is needed for incoming SRTP calls unless the callee CAC policy uses RTP-only, or the policy configuration prohibits pass-through mode

When a CAC policy uses SRTP-prefer:

- SBC accepts either RTP or SRTP offers from endpoints.
- SBC offers SRTP to endpoints whether the inbound offer is RTP or SRTP.

The following SRTP and RTP statistics are collected and available in show commands at the global level and the adjacency level:

- Number of calls rejected due to RTP requested
- Number of calls rejected due to SRTP requested
- Number of calls using SRTP pass-through
- Number of calls performing RTP to SRTP interworking
- Number of calls using RTP
- Number of calls using SRTP

SIP SRTP Offer Retry Feature

When the SIP SRTP Offer Retry feature is configured, using the **srtp {branch | callee | caller} retry rtp** command, and a 415 or 488 reject error code is generated in response to a prior SRTP (RTP/SAVP) offer, SBC reissues the offer, using RTP (RTP/AVP). This allows SBC to attempt to configure SRTP on a call leg and downgrade it to RTP if SRTP is not supported.

Note 415 and 488 error codes are general purpose errors. After the SRTP Offer Retry feature is configured, the SBC interprets that the 415 and 488 error codes are caused by an initial RTP/SAVP offer.

Downgraded Response to an SRTP Offer

The **srtp {branch | callee | caller} response downgrade** command allows SBC to send an RTP/AVP answer in response to an RTP/SAVP offer and downgrade media security. For instance, if SRTP interworking is not configured in the CAC policy, and the caller offers RTP/SAVP, but the callee answers with RTP/AVP, this command allows the SBC to downgrade the answer to RTP/AVP instead of rejecting the call.

If downgrade is not set, SBC provides strict adherence to the offer/answer protocol and rejects RTP/SAVP offers that are not supported.

This is a non-standard procedure, and is not widely supported. SBC always supports receiving an SRTP downgrade answer, but only sends a downgrade answer when this downgrade flag is set.

Both of the following cases, for SRTP fallback to RTP, are subject to the overall per-side SRTP policy and RTP-SRTP interworking policy:

- If the policy does not allow RTP at all, SBC does not attempt fallback.
- If the policy does not allow RTP-SRTP interworking, SBC allows a fallback on the answer side, but only if SBC can downgrade the offer side as well.

How SBC Processes SRTP

STRP policies behave differently depending on how the following commands are set:

- **srtp branch forbid | mandate | allow | prefer**
- **srtp caller forbid | mandate | allow | prefer**
- **srtp callee forbid | mandate | allow | prefer**
- **srtp media interworking forbid | allow**
- **srtp interworking forbid | allow**

The settings for these commands are defined as follows:

- **forbid**—SRTP is not supported on the caller side or the callee side of the call.
- **mandate**—SRTP is mandatory on the caller side or the callee side of the call.
- **allow**—SRTP is optional on the caller side or the callee side of the call.
- **prefer**—SRTP is preferred on this adjacency. Both RTP and SRTP are accepted inbound, but only SRTP is offered outbound. When the prefer option is set on the offer side of a call, it functions the same as allow. When the prefer option is set on the answer side of the call, and there is a choice between offering RTP or SRTP, SRTP is offered.

SRTP Policy Passthrough Tables

The following tables show the behavior of SBC based on the configuration of SRTP policies for each side of a call.

Table 38-1 shows how SBC selects the SRTP passthrough type for a stream offered as RTP when an SRTP policy is present.

Table 38-1 SBC Processing of RTP Offers with Presence of SRTP Policies

SRTP Policy		SRTP Passthrough Type	
Offer Side	**Answer Side**	**Interworking Not Possible**	**Interworking Possible**
Mandate	*	Reject	Reject
Forbid	Mandate	Reject	RTP-SRTP
Forbid	Forbid	RTP-RTP	RTP-RTP
Forbid	Allow	RTP-RTP	RTP-RTP
Forbid	Prefer	RTP-RTP	RTP-SRTP
Allow/Prefer	Mandate	Reject	RTP-SRTP
Allow/Prefer	Forbid	RTP-RTP	RTP-RTP
Allow/Prefer	Allow	RTP-RTP	RTP-RTP
Allow/Prefer	Prefer	RTP-RTP	RTP-SRTP

Table 38-2 shows how SBC selects the SRTP passthrough type for a stream offered as SRTP.

Table 38-2 SBC Processing of SRTP Policies for SRTP Offers

SRTP Policy		SRTP Passthrough Type		
Offer Side	Answer Side	No Interworking No Downgrade	Interworking Possible	Downgrade Possible
Forbid	Mandate	Reject	Reject	RTP-SRTP
Forbid	Forbid	Reject	Reject	RTP-RTP
Forbid	Allow	Reject	Reject	RTP-RTP
Forbid	Prefer	Reject	Reject	RTP-SRTP
Mandate	Mandate	SRTP-SRTP	SRTP-SRTP	SRTP-SRTP
Mandate	Forbid	Reject	SRTP-RTP	Reject
Mandate	Allow/Prefer	SRTP-SRTP	SRTP-SRTP	SRTP-SRTP
Allow/Prefer	Mandate	SRTP-SRTP	SRTP-SRTP	SRTP-SRTP
Allow/Prefer	Forbid	Reject	SRTP-RTP	RTP-RTP (3)
Allow/Prefer	Allow/Prefer	SRTP-SRTP	SRTP-SRTP	SRTP-SRTP

Table 38-3 shows how SBC selects the SRTP passthrough type when it receives a SIP 415 or SIP 488 rejection code in response to its SRTP offer, and Retry SRTP as RTP is set.

Table 38-3 SBC Processing of SRTP Policies for SRTP Rejection with Retry SRTP as RTP

SRTP Policy		SRTP Passthrough Type	
Offer Side	Answer Side	Interworking Not Possible	Interworking Possible
*	Mandate	Reject	Reject
Mandate	Allow/Prefer	SRTP-RTP	Reject
Allow/Prefer	Allow/Prefer	SRTP-RTP	RTP-RTP

Table 38-4 shows how SBC selects the SRTP passthrough type when it receives an RTP downgrade answer to an SRTP offer.

Table 38-4 SBC Processing of SRTP Policies for SRTP to RTP Downgrade Answer

SRTP Policy		SRTP Passthrough Type	
Offer Side	Answer Side	Interworking Not Possible	Interworking Possible
*	Mandate	Fail Call	Fail Call
Mandate	Allow/Prefer	SRTP-RTP	Fail Call
Allow/Prefer	Allow/Prefer	SRTP-RTP	RTP-RTP

Restrictions

The following restrictions apply to SRTP to RTP interworking and SRTP passthrough:

- Packet cable event messages continue to bill SRTP/RTP interworking calls and SRTP passthrough calls, and the billing does not indicate whether SRTP was used on one or both call legs.
- In late to early interworking and SRTP to RTP interworking, SBC does not support SRTP in a generated SDP offer. The call is forced to be an RTP-RTP call. If this violates the configured call policy, the event is logged and the call fails at setup.
- If a call has multiple streams (multiple m= lines in the SDP), each stream may have a different passthrough type. If any specific stream cannot be satisfied, the call is rejected. Calls with multiple streams and different passthrough types can occur when:
  - An offer is received containing a mix of RTP and SRTP streams.
  - An answer is downgraded from SRTP streams to a subset of RTP streams.
  - Some streams require interworking, others do not.
- SRTP capability is not signaled in H.248 and hence cannot be discovered automatically by SBC. This capability must be manually configured on SBC.
- SBC MG selection does not select an MG on the basis of which crypto-suites it supports.
- SBC does not allow the user to configure distinct SRTP session parameters on a per-call basis.
- SBC SRTP features do not work in conjunction with unsignaled SRTP.
- SBC will fail an SRTP call if it receives a SIP forking answer.
- SIP late-early interworking does not support SRTP.
- H.323-SIP interworking does not support SRTP.
- SBC cannot terminate RFC5027 security preconditions signaling in RTP-SRTP calls.
- SBC does not support local call transfer of SRTP calls.
- SBC currently only supports the AES_CM_128_HMAC_SHA1_32 crypto suite.
- SBC does not refresh any master keys that it generates.
- SBC does not renegotiate master key rotation when the packet usage count is reached (as specified in RFC3711).
- If the transcoder does not support SRTP (such as MGX), SBC does not allow an SRTP-SRTP call. SBC cannot perform SRTP-RTP interworking on the two media gates on either side of the transcoder.
- RTP-SRTP and SRTP-RTP calls can be transcoded by a third-party transcoder. In such cases, the media through the transcoder RTP, and the interworking is performed by SBC on the side closest to the SRTP endpoint.

To configure SRTP to RTP interworking, see the ?$paranum>Configuring CAC Policies for SRTP to RTP Interworking? section on page 38-23 and the ?$paranum>CAC Policies for SRTP to RTP Interworking Configuration: Example? section on page 38-33

You can display policy failure statistics for a specified source adjacency, using this existing command that has been updated for SRTP:

show sbc sbe call-stats src-adjacency

You can display all the calls on the SBEs, using this existing command that has been updated for SRTP:

show sbc sbe calls srtp-iw

Configuring Secure Media—Global Level

Note The Unsignaled Secure Media feature was introduced in Cisco IOS XE Release 2.6 to allow configuration of secure media at a granular level using CAC table entry commands. With the introduction of this feature, the Configuring Secure Media-Global Level feature has been deprecated. If you are upgrading from a release earlier than Release 2.6, see the procedure described in the ?$paranum>Configuring Unsignaled Secure Media at a Granular Level? section on page 38-13.

Perform the following steps to configure secure media globally.

SUMMARY STEPS

1. **configure terminal**
2. **sbc** *sbc-name*
3. **sbe**
4. **secure-media**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enables global configuration mode.
Step 2	**sbc** *sbc-name* **Example:** `Router(config)# sbc mysbc`	Creates the SBC service on Cisco Unified Border Element (SP Edition) and enters into SBC configuration mode.
Step 3	**sbe** **Example:** `Router(config-sbc)# sbe`	Enters the mode of the signaling border element (SBE) function of the SBC.

Step 4	`secure-media`
	
	Example:
	`Router(config-sbc-sbe)# secure-media`
		Configures the SBC to treat every media flow as an encrypted media flow. This allows media packets, such as DTLS and SRTP packets, to pass through the SBC.

Step 5	**end** **Example:** `Router(config-sbc-sbe)# end`	Exits SBE mode and returns to Privileged EXEC mode.

Configuring Unsignaled Secure Media at a Granular Level

Use the following steps to configure both adjacencies and both call legs using CAC policy set to enable Unsignaled Secure Media at a granular level.

Note The **caller** and **callee** commands have been used in this procedure. In some scenarios, the **branch** command can be used as an alternative to the **caller** and **callee** command pair. The **branch** command has been introduced in Release 3.5.0. See the ?$paranum>Configuring Directed Nonlimiting CAC Policies? section on page 7-37 for information about this command.

SUMMARY STEPS

1. **configure terminal**
2. **sbc** *sbc-name*
3. **sbe**
4. **adjacency {sip | h323}** *adjacency-name*
5. **security** [**untrusted | trusted-encrypted | untrusted-encrypted | trusted-unencrypted**]
6. **exit**
7. **adjacency {sip | h323}** *adjacency-name*
8. **security** [**untrusted | trusted-encrypted | untrusted-encrypted | trusted-unencrypted**]
9. **exit**
10. **cac-policy-set** *policy-set-id*
11. **first-cac-table** *table-name*
12. **cac-table** *table-name*
13. **table-type limit** *list of limit tables*
14. **entry** *entry-id*
15. **match-value** *key*
16. **srtp support [allow | disallow | trusted-only**]
17. **caller secure-media**
18. **callee secure-media**
19. **action** {**cac-complete | next-table** *goto-table-name*}
20. **exit**
21. **complete**
22. **exit**
23. **active-cac-policy-set** *policy-set-id*
24. **end**
25. **show sbc** *sbc-name* **sbe cac-policy-set** [id [table name [entry id]] | active [table name [entry id]]] [detail]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enables global configuration mode.
Step 2	**sbc** *sbc-name* **Example:** `Router(config)# sbc mysbc`	Creates the SBC service on Cisco Unified Border Element (SP Edition) and enters into SBC configuration mode.
Step 3	**sbe** **Example:** `Router(config-sbc)# sbe`	Enters the mode of the signaling border element (SBE) function of the SBC.
Step 4	**adjacency** {**sip** | **h323**} *adjacency-name* **Example:** `Router(config-sbc-sbe)# adjacency sip client`	Configures the caller side SIP adjacency, that is named 'client' in the example. And enters the mode of an SBE SIP adjacency, often called adjacency sip mode.
Step 5	**security [untrusted | trusted-encrypted | untrusted-encrypted | trusted-unencrypted]** **Example:** `Router(config-sbc-sbe-adj-sip)# security trusted-encrypted`	Configures transport-level security (TLS) on a SIP adjacency. For granular-level Secure Media, configure the trusted adjacency as trusted-encrypted or trusted-unencrypted. Trusted means the adjacency is trusted to carry secure SIP calls (calls with SIPS URI). Encrypted means the adjacency uses TLS encryption for SIP signaling. Unencrypted means it does not use TLS encryption for SIP signaling. **Note** If this adjacency is *untrusted*, skip steps Step 4 through Step 6. You need to configure for an untrusted adjacency in a CAC policy table.

Step 6	`exit`
	
	Example:
	`Router(config-sbc-sbe-adj-sip)# exit`
		Exits the SBE SIP adjacency mode to the SBE mode.

Step 7	**adjacency** {**sip** | **h323**} *adjacency-name* **Example:** `Router(config-sbc-sbe)# adjacency sip server`	Configures the callee side SIP adjacency, that is named 'server' in the example. And enters the mode of an SBE SIP adjacency, often called adjacency sip mode.

	Command or Action	Purpose
Step 8	**security** [**untrusted** | **trusted-encrypted** | **untrusted-encrypted** | **trusted-unencrypted**] **Example:** `Router(config-sbc-sbe-adj-sip)# security trusted-unencrypted`	Configures transport-level security (TLS) on a SIP adjacency. For granular-level Secure Media, configure the trusted adjacency as trusted-encrypted or trusted-unencrypted. Trusted means the adjacency is trusted to carry secure SIP calls (calls with SIPS URI). Encrypted means the adjacency uses TLS encryption for SIP signaling. Unencrypted means it does not use TLS encryption for SIP signaling. **Note** If this adjacency is *untrusted*, skip steps Step 7 through Step 9. You need to configure for an untrusted adjacency in a CAC policy table.
Step 9	**exit** **Example:** `Router(config-sbc-sbe-adj-sip)# exit`	Exits the SBE SIP adjacency mode to the SBE mode.
Step 10	**cac-policy-set** *policy-set-id* **Example:** `Router(config-sbc-sbe)# cac-policy-set 1`	Enters the mode of CAC policy set configuration within an SBE entity, creating a new policy set if necessary. *policy-set-id*—Integer chosen by the user to identify the policy set. The range is 1 - 2147483647.
Step 11	**first-cac-table** *table-name* **Example:** `Router(config-sbc-sbe-cacpolicy)# first-cac-table testSecure`	Configures the name of the first policy table to process. A CAC policy may have many tables configured. To start the application of the CAC policy, the first table that is used needs to be defined. *table-name*—The admission control table that should be processed first.
Step 12	**cac-table** *table-name* **Example:** `Router(config-sbc-sbe-cacpolicy)# cac-table testSecure`	Enters the mode for configuration of an admission control table (creating one if necessary) within the context of an SBE policy set. *table-name*—Name of the admission control table.

	Command or Action	Purpose
Step 13	**table-type limit** ***list of limit tables*** **Example:** `Router(config-sbc-sbe-cacpolicy-cactable)# table-type limit all`	Configures a new CAC Limit table type where you enter the criteria that is used to match the entries. *list of limit tables* can be one of the following values: • **account**—Compare the name of the account. • **adj-group**—Compare the name of the adjacency group. • **adjacency**—Compare the name of the adjacency. • **all**—No comparison type. All events match this type. • **call-priority**—Compare with call priority. • **category**—Compare the number analysis assigned category. • **dst-account**—Compare the name of the destination account. • **dst-adj-group**—Compare the name of the destination adjacency group. • **dst-adjacency**—Compare the name of the destination adjacency. • **dst-prefix**—Compare the beginning of the dialed digit string. • **event-type**—Compare with CAC policy event types. • **src-account**—Compare the name of the source account. • **src-adj-group**—Compare the name of the source adjacency group. • **src-adjacency**—Compare the name of the source adjacency. • **src-prefix**—Compare the beginning of the calling number string.
Step 14	**entry** *entry-id* **Example:** `Router(config-sbc-sbe-cacpolicy-cactable)# entry 1`	Enters the mode to modify an entry in an admission control table. *entry-id*—Specifies the table entry.
Step 15	**match-value** *key* **Example:** `Router(config-sbc-sbe-cacpolicy-cactable-entry) # match-value call-update`	Configures the match-value of an entry in a CAC Limit table type.

	Command or Action	Purpose

Step 16	**srtp support [allow | disallow | trusted-only]**
	
	Example:
	`Router(config-sbc-sbe-cacpolicy-cactable-entry)`
	`# srtp support allow`
		If an adjacency is *untrusted* and you want granular-level Secure Media, you need to configure this step—configuring with **srtp support allow** will allow an SRTP call on the untrusted adjacency where the CAC policy is applied. Continue on to Step 17.
		
		Configures the srtp support variable in the CAC table to allow or disallow SRTP Passthrough of secure media on the adjacency where the policy is applied.
		
		• **allow**—allows SRTP Transport when an event matches this CAC policy.
		• **disallow**—do not allow SRTP Transport when an event matches this CAC policy.
		• **trusted-only**—allows SRTP Transport on a trusted adjacency (default) when an event matches this CAC policy.
		
		Calls using SRTP Passthrough are allowed on the adjacencies specified by the policy.

Step 17	**caller secure-media**
	
	Example:
	`Router(config-sbc-sbe-cacpolicy-cactable-entry)`
	`# caller secure-media`
		Configures a Secure Media call on the caller side.

Step 18	**callee secure-media**
	
	Example:
	`Router(config-sbc-sbe-cacpolicy-cactable-entry)`
	`# callee secure-media`
		Configures a Secure Media call on the callee side.

Step 19	**action {cac-complete | next-table *goto-table-name*}**
	
	Example:
	`Router(config-sbc-sbe-cacpolicy-cactable-entry)`
	`# action cac-complete`
		Configures the action to perform after this entry in an admission control table. Each entry requires a match criteria and an action. The action is to accept the transport.
		
		action is one of the following:
		
		• **cac-complete**—When an event matches, this CAC policy is complete.
		• **next-table**—Specifies the name of the next cac table.
		• *goto-table-name*—Specifies the table name identifying the next CAC table to process (or cac-complete, if processing should stop).

Step 20	**exit**
	
	Example:
	`Router(config-sbc-sbe-cacpolicy-cactable-entry)`
	`# exit`
		Exits CAC Table Entry mode and enters CAC Policy-set configuration mode.

	Command or Action	Purpose
Step 21	**complete** **Example:** `Router(config-sbc-sbe-cacpolicy)# complete`	Completes the CAC-policy set after committing the full set.
Step 22	**exit** **Example:** `Router(config-sbc-sbe-cacpolicy)# exit`	Exits CAC Policy-set configuration mode and enters SBE mode.
Step 23	**active-cac-policy-set** *policy-set-id* **Example:** `Router(config-sbc-sbe)# active-cac-policy-set 1`	Sets the newly created CAC policy to be active. When the policy is active, it can no longer be modified. *policy-set-id*—Identifies the policy set that is made active. Range is 1 to 2147483647.
Step 24	**end** **Example:** `Router(config-sbc-sbe)# end`	Exits the SBE mode and returns to Privileged EXEC mode.
Step 25	**show sbc** *name* **sbe cac-policy-set** [*id* [**table** *name* [**entry** *id*]]|**active** [**table** *name* [**entry** *id*]]] [**detail**] **Example:** `Router# show sbc mysbc sbe cac-policy-set 1 detail`	Displays detailed information for a given entry in a CAC policy table. In this example, that includes the caller/callee unsignaled secure media: Allowed fields and the security trusted-unencrypted for both adjacencies of the Secure Media call.

Configuring SRTP Passthrough

These steps show how to configure the CAC policy set to allow SRTP Passthrough.

SUMMARY STEPS

1. **configure terminal**
2. **sbc** *sbc-name*
3. **sbe**
4. **cac-policy-set** ***policy-set-id***
5. **first-cac-scope** ***scope-name***
6. **first-cac-table** ***table-name***
7. **cac-table** ***table-name***
8. **table-type limit** ***list of limit tables***
9. **entry** ***entry-id***
10. **match-value** ***key***
11. **srtp support [allow | disallow | trusted-only]**
12. **action [cac-complete | next-table |** ***goto-table-name*** **]**

13. **exit**
14. **exit**
15. **complete**
16. **exit**
17. **active-cac-policy-set** *policy-set-id*
18. **end**
19. **show sbc** *sbc-name* **sbe cac-policy-set** *id* **table** *name* **entry** *entry*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enables global configuration mode.
Step 2	**sbc** *sbc-name* **Example:** `Router(config)# sbc mysbc`	Creates the SBC service on Cisco Unified Border Element (SP Edition) and enters into SBC configuration mode.
Step 3	**sbe** **Example:** `Router(config-sbc)# sbe`	Enters the mode of the signaling border element (SBE) function of the SBC.
Step 4	**cac-policy-set** *policy-set-id* **Example:** `Router(config-sbc-sbe)# cac-policy-set 1`	Enters the mode of CAC policy set configuration within an SBE entity, creating a new policy set if necessary. *policy-set-id*—Integer chosen by the user to identify the policy set. The range is 1 - 2147483647.

	Command or Action	Purpose
Step 5	**first-cac-scope** *scope-name* **Example:** `Router(config-sbc-sbe-cacpolicy)# first-cac-scope call`	Configures scope at which limits should be initially defined when performing the admission control stage of the policy. Each CAC policy has a scope that is applied to it. This CAC policy applies on a per call basis. *scope-name* has one of the following values: • **adj-group**—Limits for events from members of the same adjacency group. • **call**—Limits are per single call. • category—Limits per category. • **dst-account**—Limits for events sent to the same account. • **dst-adj-group**—Limits for events sent to the same adjacency group. • **dst-adjacency**—Limits for events sent to the same adjacency. • **dst-number**—Limits for events that have the same adjacency number. • **global**—Limits are global (May not be combined with any other option). • **src-account**—Limits for events from the same account. • **src-adj-group**—Limits for events from the same adjacency group. • **arc-adjacency**—Limits for events from the same adjacency. • **src-number**—Limits for events that have the same source number.
Step 6	**first-cac-table** *table-name* **Example:** `Router(config-sbc-sbe-cacpolicy)# first-cac-table testSecure`	Configures the name of the first policy table to process. A CAC policy may have many tables configured. To start the application of the CAC policy, the first table that is used needs to be defined. *table-name*—The admission control table that should be processed first.
Step 7	**cac-table** *table-name* **Example:** `Router(config-sbc-sbe-cacpolicy)# cac-table testSecure`	Enters the mode for configuration of an admission control table (creating one if necessary) within the context of an SBE policy set. *table-name*—Name of the admission control table.

	Command or Action	Purpose
Step 8	**table-type limit** *list of limit tables* **Example:** `Router(config-sbc-sbe-cacpolicy-cactable)# table-type limit all`	Configures a new CAC Limit table type where you enter the criteria that is used to match the entries. *list of limit tables* can be one of the following values: • **account**—Compare the name of the account. • **adj-group**—Compare the name of the adjacency group. • **adjacency**—Compare the name of the adjacency. • **all**—No comparison type. All events match this type. • **call-priority**—Compare with call priority. • **category**—Compare the number analysis assigned category. • **dst-account**—Compare the name of the destination account. • **dst-adj-group**—Compare the name of the destination adjacency group. • **dst-adjacency**—Compare the name of the destination adjacency. • **dst-prefix**—Compare the beginning of the dialed digit string. • **event-type**—Compare with CAC policy event types. • **src-account**—Compare the name of the source account. • **src-adj-group**—Compare the name of the source adjacency group. • **src-adjacency**—Compare the name of the source adjacency. • **src-prefix**—Compare the beginning of the calling number string.
Step 9	**entry** *entry-id* **Example:** `Router(config-sbc-sbe-cacpolicy-cactable)# entry 1`	Enters the mode to modify an entry in an admission control table. *entry-id*—Specifies the table entry.
Step 10	**match-value** *key* **Example:** `Router(config-sbc-sbe-cacpolicy-cactable-entry) # match-value call-update`	Configures the match-value of an entry in a CAC Limit table type.

	Command or Action	Purpose
Step 11	**srtp support [allow | disallow | trusted-only]** **Example:** `Router(config-sbc-sbe-cacpolicy-cactable-entry)# srtp support allow`	Configures the srtp support variable in the CAC table to allow or disallow SRTP Passthrough of secure media on the adjacency where the policy is applied. • **allow**—allows SRTP Transport when an event matches this CAC policy. • **disallow**—do not allow SRTP Transport when an event matches this CAC policy. • **trusted-only**—allows SRTP Transport on a trusted adjacency (default) when an event matches this CAC policy. Calls using SRTP Passthrough are allowed on the adjacencies specified by the policy. Where there are conflicting policies, "disallowed" overrides "allowed" which overrides "trusted-only."
Step 12	**action** [**cac-complete** | **next-table** *goto-table-name*] **Example:** `Router(config-sbc-sbe-cacpolicy-cactable-entry)# action cac-complete`	Configures the action to perform after this entry in an admission control table. Each entry requires a match criteria and an action. The action is to accept the transport. action is one of the following: • **cac-complete**—When an event matches, this CAC policy is complete. • **next-table**—Specifies the name of the next cac table. • *goto-table-name*—Specifies the table name identifying the next CAC table to process (or cac-complete, if processing should stop).
Step 13	**exit** **Example:** `Router(config-sbc-sbe-cacpolicy-cactable-entry)# exit`	Exits CAC table entry submode and enters into cacpolicy cactable mode
Step 14	**exit** **Example:** `Router(config-sbc-sbe-cacpolicy-cactable)# exit`	Exits cacpolicy cactable submode and enters into cacpolicy mode.
Step 15	**complete** **Example:** `Router(config-sbc-sbe-cacpolicy)# complete`	Completes the CAC policy after all the entries within the CAC tables have been configured.
Step 16	**exit** **Example:** `Router(config-sbc-sbe-cacpolicy)# exit`	Exits the cacpolicy submode and enters into SBE mode.

	Command or Action	Purpose
Step 17	**active-cac-policy-set** *policy-set-id* **Example:** `Router(config-sbc-sbe)# active-cac-policy-set 1`	Sets the newly created CAC policy to be active. When the policy is active, it can no longer be modified. *policy-set-id*—Identifies the policy set that is made active. Range is 1 to 2147483647.
Step 18	**end** **Example:** `Router(config-sbc-sbe)# end`	Exits the SBE mode and returns to Privileged EXEC mode.
Step 19	`show sbc` *sbc-name* `sbe cac-policy-set` *id* `table` *name* `entry` *entry* **Example:** `Router# show sbc mysbc sbe cac-policy-set 1 table testSecure entry 1`	Displays detailed output, including a "SRTP Transport" field and whether the policy for the adjacency is to allow, disallow, or trust only for SRTP Transport.

Configuring CAC Policies for SRTP to RTP Interworking

Use the following procedure to configure the CAC policies for the caller side and the callee side of a call for SRTP to RTP interworking.

SUMMARY STEPS

1. **configure terminal**
2. **sbc** *sbc-name*
3. **sbe**
4. **cac-policy-set** *policy-set-id*
5. **first-cac-table** *table-name*

CAC Table for Caller Side of the Call

6. **cac-table** *table-name*
7. **table-type limit** *list of limit tables*

(repeat steps 8 through 14 as many times as needed)

8. **entry** *entry-id*
9. **match-value** *key*
10. **srtp support allow**
11. **action next-table** *goto-table-name*
12. **srtp caller forbid | mandate | allow | prefer**
13. **srtp interworking forbid | allow**
14. **srtp media interworking forbid | allow**

CAC Table for Callee Side of the Call

15. **cac-table** *table-name*

16. **table-type limit** *list of limit tables*

(repeat steps 17 through 23 as many times as needed)

17. **entry** *entry-id*
18. **match-value** *key*
19. **srtp support allow**
20. **action cac-complete**
21. **srtp callee forbid | mandate | allow**
22. **srtp interworking forbid | allow**
23. **srtp media interworking forbid | allow**

(issue **complete** *command after all entries are configured)*

24. **complete**
25. **end**
26. **show sbc** *name* **sbe cac-policy-set** *id* **detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**sbc** *sbc-name* **Example:** `Router(config)# sbc SBC1`	Creates the SBC service on Cisco Unified Border Element (SP Edition) and enters into SBC configuration mode.
Step 3	**sbe** **Example:** `Router(config-sbc)# sbe`	Enters the mode of the signaling border element (SBE) function of the SBC.
Step 4	**cac-policy-set** *policy-set-id* **Example:** `Router(config-sbc-sbe)# cac-policy-set 44`	Enters the mode of CAC policy set configuration within an SBE entity, creating a new policy set. • *policy-set-id*—Integer chosen by the user to identify the policy set. The range is 1 to 2147483647.
Step 5	**first-cac-table** *table-name* **Example:** `Router(config-sbc-sbe-cacpolicy)# first-cac-table 44`	Specifies which CAC table is processed first. • *table-name*—The name table to be processed first.

CAC Table for Caller Side of the Call

	Command or Action	Purpose
Step 6	**cac-table** *table-name* **Example:** `Router(config-sbc-sbe-cacpolicy)# cac-table 44`	Enters the mode for configuration of an admission control table (creating one if necessary) within the context of an SBE policy set. • *table-name*—Name of the admission control table.
Step 7	**table-type limit** *list of limit tables* **Example:** `Router(config-sbc-sbe-cacpolicy-cactable)# table-type limit src-adjacency`	Configures the limit of the table types to be matched by the **match-value** command. For this example, use the following table type: • *src-adjacency*—Compare the name of the source adjacency.

Repeat steps 8 through 14 as many times as necessary to configure as many entries as needed.

Step 8	**entry** *entry-id* **Example:** `Router(config-sbc-sbe-cacpolicy-cactable)# entry 1`	Enters the mode to modify an entry in an admission control table. • *entry-id*—Specifies the table entry.
Step 9	**match-value** *key* **Example:** `Router(config-sbc-sbe-cacpolicy-cactable-entry) # match-value A`	Configures the match-value of an entry in a Call Admission Control (CAC) Limit Table. • *key*—Specifies the keyword used to match events. The format of the key is determined by the table-type limit.
Step 10	**srtp support allow** **Example:** `Router(config-sbc-sbe-cacpolicy-cactable-entry) # srtp support allow`	Configures SRTP support.
Step 11	**action next-table** *goto-table-name* **Example:** `Router(config-sbc-sbe-cacpolicy-cactable-entry) # action next-table 45`	Configures the action to take when this routing entry is chosen. • *goto-table-name*—Specifies the next routing table to process when an event matches the entry.
Step 12	**srtp caller forbid | mandate | allow | prefer** **Example:** `Router(config-sbc-sbe-cacpolicy-cactable-entry) # srtp caller forbid`	Configures SRTP for the caller side of the call with one of the following SRTP settings. • **forbid**—SRTP is not supported on the caller side of the call. • **mandate**—SRTP is mandatory on the caller side of the call. • **allow**—SRTP is optional on the caller side of the call. • **prefer**—SRTP is preferred on this adjacency. Both RTP and SRTP are accepted inbound, but only SRTP is offered outbound.

	Command or Action	Purpose
Step 13	**srtp interworking forbid | allow** **Example:** `Router(config-sbc-sbe-cacpolicy-cactable-entry) # srtp interworking allow`	Configures SRTP to RTP interworking. • **forbid**—Prohibits SRTP to RTP interworking on the call. • **allow**—Allows SRTP to RTP interworking on the call.
Step 14	**srtp media interworking forbid | allow** **Example:** `Router(config-sbc-sbe-cacpolicy-cactable-entry) # srtp media interworking allow`	Configures SRTP to RTP media interworking. • **forbid**—Prohibits SRTP to RTP media interworking on the call. • **allow**—Allows SRTP to RTP media interworking on the call.

CAC Table for Callee Side of the Call

Step 15	**cac-table** *table-name* **Example:** `Router(config-sbc-sbe-cacpolicy)# cac-table 45`	Enters the mode for configuration of an admission control table (creating one if necessary) within the context of an SBE policy set. • *table-name*—Name of the admission control table.
Step 16	**table-type limit** *list of limit tables* **Example:** `Router(config-sbc-sbe-cacpolicy-cactable)# table-type limit src-adjacency`	Configures the limit of the table types to be matched by the **match-value** command. For this example, use the following table type: • *src-adjacency*—Compare the name of the source adjacency.

Repeat steps 17 through 23 as many times as necessary to configure as many entries as needed.

Step 17	**entry** *entry-id* **Example:** `Router(config-sbc-sbe-cacpolicy-cactable)# entry 1`	Enters the mode to modify an entry in an admission control table. • *entry-id*—Specifies the table entry.
Step 18	**match-value** *key* **Example:** `Router(config-sbc-sbe-cacpolicy-cactable-entry) # match-value A`	Configures the match-value of an entry in a Call Admission Control (CAC) Limit Table. • *key*—Specifies the keyword used to match events. The format of the key is determined by the table-type limit.
Step 19	**srtp support allow** **Example:** `Router(config-sbc-sbe-cacpolicy-cactable-entry) # srtp support allow`	Configures SRTP support.
Step 20	**action next-table** *goto-table-name* **Example:** `Router(config-sbc-sbe-cacpolicy-cactable-entry) # action next-table 45`	Configures the action to take when this routing entry is selected. • *goto-table-name*—Specifies the next routing table to process if the event matches the entry.

	Command or Action	Purpose

Step 21	**srtp callee forbid | mandate | allow | prefer**
	
	Example:
	`Router(config-sbc-sbe-cacpolicy-cactable-entry)`
	`# srtp callee forbid`
		Configures SRTP for the callee side of the call.
		• **forbid**—SRTP is not supported on the callee side of the call.
		• **mandate**—SRTP is mandatory on the callee side of the call.
		• **allow**—SRTP is optional on the callee side of the call.
		• **prefer**—SRTP is preferred on this adjacency. Both RTP and SRTP are accepted inbound, but only SRTP is offered outbound.

Step 22	**srtp interworking forbid | allow**
	
	Example:
	`Router(config-sbc-sbe-cacpolicy-cactable-entry)`
	`# srtp interworking allow`
		Configures SRTP to RTP interworking.
		• **forbid**—Prohibits SRTP to RTP interworking on the call.
		• **allow**—Allows SRTP to RTP interworking on the call.

Step 23	**srtp media interworking forbid | allow**
	
	Example:
	`Router(config-sbc-sbe-cacpolicy-cactable-entry)`
	`# srtp media interworking allow`
		Configures SRTP to RTP media interworking.
		• **forbid**—Prohibits SRTP to RTP media interworking on the call.
		• **allow**—Allows SRTP to RTP media interworking on the call.

*Issue the **complete** command only after all entries are configured.*

Step 24	`complete`
	
	Example:
	`Router(config-sbc-sbe-cacpolicy-cactable-entry)`
	`# complete`
		Completes the CAC-policy after all entries are entered.

Step 25	**end**
	
	Example:
	`Router(config-sbc-sbe-cacpolicy-cactable-entry)`
	`# end`
		Exits configuration mode and returns to privileged EXEC mode.

Step 26	**show sbc** *name* **sbe cac-policy-set** *id* **detail**
	
	Example:
	`Router# show sbc SBC1 sbe cac-policy-set 1 detail`
		Displays detailed information for the given entry ID in a CAC policy table. In this case, it shows the default values for SRTP-RTP interworking. For example:
		`Caller SRTP support: Inherit (default)`
		`Callee SRTP support: Inherit (default)`
		`SRTP Interworking: Inherit (default)`
		`SRTP media Interworking: Inherit (default)`

SRTP Support for RTCP Multiplexed with RTP

In earlier releases, the SBC could process incoming RTP and RTCP streams that were sent over separate UDP channels. From Release 3.4S, the SBC can also process RTCP streams multiplexed with RTP streams and sent over a single UDP channel. The SBC distinguishes between RTCP and RTP streams by examining the payload format of each stream. This also applies to SRTCP streams multiplexed with SRTP streams.

Note RFC 5761 describes the multiplexing of RTCP streams with RTP streams. The same principle applies to SRTCP and SRTP.

This feature is an enhancement to the support for interworking of RTP-based and SRTP-based endpoints that are linked through the SBC. The Cisco TelePresence System is an example of an RTP-based endpoint, and Cisco Umi TelePresence is an example of an SRTP-based endpoint. With the introduction of this feature, the SBC processes RTCP streams multiplexed with RTP streams coming from the Cisco TelePresence System. In a similar manner, the SBC identifies and correctly processes SRTCP streams multiplexed with SRTP streams coming from Cisco Umi TelePresence.

By default, the detection of RTCP streams multiplexed with RTP streams is disabled in the SBC. You can enable this feature by performing the procedure described in the following section.

Configuring the Detection of RTCP Multiplexed with RTP

This task explains how to configure the detection of RTCP streams multiplexed with RTP streams.

Note The same procedure can be used to configure the detection of SRTCP streams multiplexed with SRTP streams.

SUMMARY STEPS

1. **configure terminal**
2. **sbc** *sbc-name*
3. **sbe**
4. **rtcp-mux**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters the global configuration mode.
Step 2	**sbc** *sbc-name* **Example:** `Router(config)# sbc MySbc`	Enters the SBC service mode. • *sbc-name*—Name of the SBC.
Step 3	**sbe** **Example:** `Router(config-sbc)# sbe`	Enters the SBE configuration mode.
Step 4	**rtcp-mux** **Example:** `Router(config-sbc-sbe)# rtcp-mux`	Enables the detection of RTCP streams multiplexed with RTP streams. By default, this feature is disabled.

SRTP Support for SSRC-Based Multiplexing

An SBC endpoint such as the Cisco TelePresence System multiplexes RTP streams of the same type (audio or video) on a single UDP channel. It uses the 32-bit synchronization source (SSRC) field of RTP streams to differentiate between discrete RTP streams originating from a single source.

When an SRTP-based or RTP-based endpoint sends multiplexed streams over a single UDP channel, the channel contains multiple streams and each stream has its own SSRC field. In earlier releases, the SBC could support only a single SSRC field in a UDP channel. Therefore, the SBC could not support interworking of endpoints that sent multiplexed SRTP and RTP. From Release 3.4S, the SBC can process multiple SSRC fields in multiplexed SRTP or RTP streams. In combination with SRTP support for RTCP multiplexed with RTP, this feature enhances interworking of RTP-based and SRTP-based endpoints.

Configuring Global Secure Media Example

This section provides a sample configuration for the Secure Media Passthrough feature.

```
Router# configure
Router(config)# sbc mysbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# secure-media
Router(config-sbc-sbe)# end
```

Configuring Unsignaled, Granular-Level Secure Media: Examples

The following configuration example shows how the client and server SIP adjacencies are configured as "security trusted-unencrypted" and how the CAC table entry 1 is configured for secure media on both the caller and callee sides.

Note The **caller** and **callee** commands have been used in this procedure. In some scenarios, the **branch** command can be used as an alternative to the **caller** and **callee** command pair. The **branch** command has been introduced in Release 3.5.0. See the ?$paranum>Configuring Directed Nonlimiting CAC Policies? section on page 7-37 for information about this command.

```
...
cac-policy-set 2
 first-cac-table 1
 cac-table 1
  table-type limit all
  entry 1
   match-value call-update
   caller secure-media
   callee secure-media
   action cac-complete
   exit
 complete
 exit
active-cac-policy-set 2

adjacency sip client
 nat force-off
 security trusted-unencrypted
 signaling-address ipv4 10.10.100.110
 signaling-port 9060
 remote-address ipv4 10.10.100.10 255.255.255.255
 signaling-peer 10.10.100.10
 signaling-peer-port 9060
 attach
adjacency sip server
 nat force-off
 security trusted-unencrypted
 signaling-address ipv4 10.10.100.110
 signaling-port 9070
 remote-address ipv4 10.10.100.10 255.255.255.255
 signaling-peer 10.10.100.10
 signaling-peer-port 9070
 attach
```

The following example shows how to configure granular-level unsignaled secure media where an adjacency is *untrusted* by using the **srtp support allow** command on the untrusted adjacency in a CAC policy table:

```
...
cac-policy-set 2
 first-cac-table 1
 cac-table 1
  table-type limit all
  entry 1
   match-value call-update
```

```
   srtp support allow
   caller secure-media
   callee secure-media
   action cac-complete
   exit
 complete
 exit
active-cac-policy-set 2
```

The following example lists detailed information pertaining to CAC policy set 2, and shows how secure media is configured on the caller and callee sides:

```
Router# show sbc asr sbe cac-policy-set 2 detail

SBC Service "asr"

CAC Policy Set 2
  Active policy set: Yes
  Description:
  Averaging period: 60 sec
  First CAC table: 1
  First CAC scope: global
  First CAC prefix length: 4294967256

  Table name: 1
    Description:
    Table type: policy-set                      Total call failures: 0

    Entry 1
    CAC scope:
    CAC scope prefix length: 0
    Action: CAC complete                    Number of calls rejected: 0
    Max calls per scope:    Unlimited       Max call rate per scope: Unlimited
    Max in-call rate:       Unlimited       Max out-call rate:       Unlimited
    Max reg. per scope:     Unlimited       Max reg. rate per scope: Unlimited
    Max channels per scope: Unlimited       Max updates per scope:   Unlimited
    Early media:            Allowed         Early media direction:   Both
    Early media timeout:    None            Transcoder per scope:    Allowed
    Callee Bandwidth-Field: None            Caller Bandwidth-Field:  None
    Media bypass:                   Allowed
    Renegotiate Strategy:           Delta
    Max bandwidth per scope:        Unlimited
    SRTP Transport:                 Trusted-Only (by default)
    Caller hold setting:            Standard
    Callee hold setting:            Standard
    Caller privacy setting:         Never hide
    Callee privacy setting:         Never hide
    Caller voice QoS profile:       Default
    Callee voice QoS profile:       Default
    Caller video QoS profile:       Default
    Callee video QoS profile:       Default
    Caller sig QoS profile:         Default
    Callee sig QoS profile:         Default
    Caller inbound SDP policy:      None
    Callee inbound SDP policy:      None
    Caller outbound SDP policy:     None
    Callee outbound SDP policy:     None
    Caller media disabled:
      Strip All Answer
    Callee media disabled:
      Strip All Offer
    Caller unsignaled secure media:  Allowed
    Callee unsignaled secure media:  Allowed
    Caller tel-event payload type:   Default
```

```
    Callee tel-event payload type:   Default
    Media flag:
      Ignore bandwidth-fields (b=), Telephone Event Interworking
    Restrict codecs to list:         Default
    Restrict caller codecs to list:  Default
    Restrict callee codecs to list:  Default
    Maximum Call Duration:           Unlimited
```

The following example shows an excerpt of detailed information for the callee side SIP adjacency 'server' showing that security trusted-unencrypted is configured:

```
Router# show sbc asr sbe adjacencies server detail

SBC Service "asr"
  Adjacency server (SIP)
    Status:                Attached
[snip]
    Security:              Trusted-Unencrypted
[snip]
```

Configuring SRTP Passthrough Example

The following shows a configuration where the "srtp transport" variable is set in the CAC policy set 1 table for an adjacency to allow SRTP Passthrough:

```
sbc SBE-NODE2-SBE1
   sbe
      cac-policy-set 1
         first-cac-scope global
         first-cac-table STANDARD-LIST-BY-ACCOUNT
         cac-table STANDARD-LIST-BY-ACCOUNT
            table-type limit dst-account
            entry 1
               media-bypass-forbid
                 match-value SIP-CUSTOMER-1
               max-num-calls 100
               max-call-rate 20
               max-bandwidth 1000000 bps
               callee-privacy never
               srtp support allow
               action cac-complete
               exit
            entry 2
               match-value SIP-CUSTOMER-2
               max-num-calls 100
               max-call-rate 20
               max-bandwidth 1000000 bps
               transcode-deny
               max-regs 500
               action cac-complete
               exit
               exit
               complete
      active-call-policy-set 1
```

The following example displays entries in table CAC1 for CAC policy set 100 and shows that the SRTP Transport variable has been set to allow SRTP Passthrough on whichever adjacency the policy is applied:

```
Router# show sbc SBC1 sbe cac-policy-set 100 table CAC1 entry 1000

SBC Service "SBC1"
```

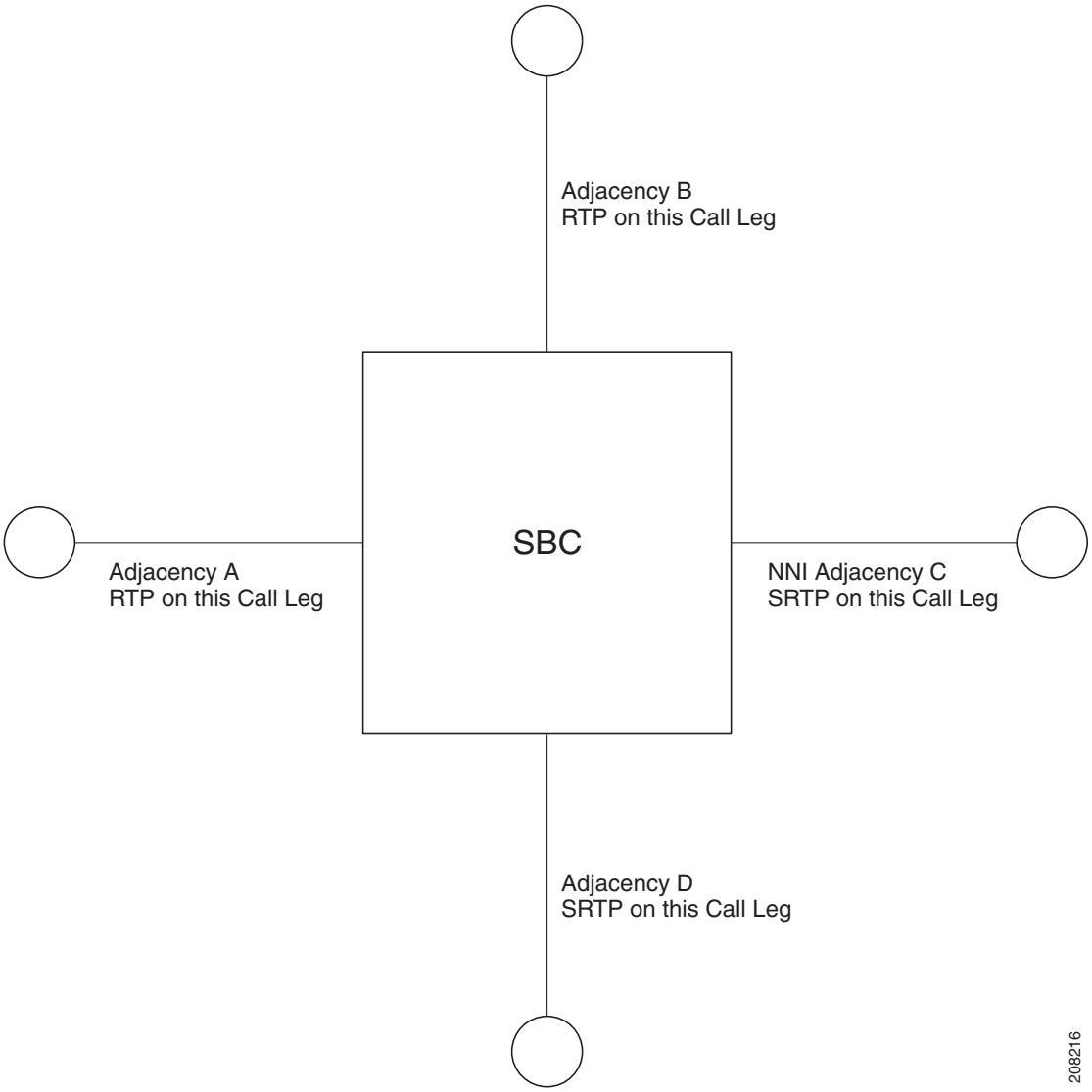
```
Policy set 100 table CAC1 entry 1000
  Match value               src-adjacency
  Action                    CAC policy complete
  Max calls                 Unlimited
  Max call rate             100
  Max registrations         Unlimited
  Max reg. rate             Unlimited
  Max bandwidth             Unlimited
  Max channels              Unlimited
  Transcoder                Allowed
  Caller privacy setting    Never hide
  Callee privacy setting    Never hide
  Early media               Allowed
  Early media direction     Both
  Early media timeout       0
  Restrict codecs to list   default
  Media bypass              Allowed
  Number of calls rejected by this entry    0
  SRTP Transport            Allowed
```

CAC Policies for SRTP to RTP Interworking Configuration: Example

The following example shows specific details of how to configure the CAC policies for the caller side and the callee side of a call for SRTP to RTP interworking. Multiple entries with specific settings are given.

Figure 38-2 shows the adjacencies that are used by the match-value command in this example.

Figure 38-2 Adjacencies A, B, C, and D for Example




```
configure terminal
sbc SBC1
sbe

cac-policy-set 44
    first-cac-table 44

    cac-table 44
     table-type limit src-adjacency

     entry 1
      match-value A
      srtp support allow
      action next-table 45
      srtp caller forbid
      srtp interworking allow
      srtp media interworking allow
```

```
   entry 2
     match-value B
     srtp support allow
     action next-table 45
     srtp caller forbid
     srtp interworking allow
     srtp media interworking allow

   entry 3
     match-value C
     srtp support allow
     action next-table 45
     srtp caller mandate
     srtp interworking allow
     srtp media interworking allow

entry 4
     match-value D
     srtp support allow
     action next-table 45
     srtp caller mandate
     srtp interworking allow
     srtp media interworking allow

   cac-table 45
    table-type limit dst-adjacency

    entry 1
     match-value A
     srtp support allow
     action cac-complete
     srtp callee forbid
     srtp interworking allow
     srtp media interworking allow

entry 2
     match-value B
     srtp support allow
     action cac-complete
     srtp callee forbid
     srtp interworking allow
     srtp media interworking allow

entry 3
     match-value C
     srtp support allow
     action cac-complete
     srtp callee mandate
     srtp interworking allow
     srtp media interworking allow

entry 4
     match-value D
     srtp support allow
     action cac-complete
     srtp callee mandate
     srtp interworking allow
     srtp media interworking allow

     complete
     end

show sbc sbc1 sbe cac-policy-set 44 detail
```