



Network Data Encryption Commands

This chapter describes network data encryption commands. Cisco provides network data encryption as a means to safeguard network data that travels from one Cisco router to another, across unsecured networks.

Refer to the *Command Reference Master Index* or search online to find complete descriptions of other commands used when configuring network data encryption.

For configuration information, refer to the chapter “Configuring Network Data Encryption” in the *Security Configuration Guide*.

access-list (encryption)

To define an encryption access list by number, use the extended IP **access-list** global configuration command. To remove a numbered encryption access list, use the **no** form of this command.

```
access-list access-list-number [dynamic dynamic-name [timeout minutes]] {deny | permit}
    protocol source-wildcard destination destination-wildcard [precedence precedence]
    [tos tos] [log]
no access-list access-list-number
```

For Internet Control Message Protocol (ICMP), you can also use the following syntax:

```
access-list access-list-number [dynamic dynamic-name [timeout minutes]] {deny | permit}
    icmp source source-wildcard destination destination-wildcard [icmp-type [icmp-code] |
    icmp-message] [precedence precedence] [tos tos] [log]
```

For Internet Group Management Protocol (IGMP), you can also use the following syntax:

```
access-list access-list-number [dynamic dynamic-name [timeout minutes]] {deny | permit}
    igmp source source-wildcard destination destination-wildcard [igmp-type]
    [precedence precedence] [tos tos] [log]
```

For TCP, you can also use the following syntax:

```
access-list access-list-number [dynamic dynamic-name [timeout minutes]] {deny | permit}
    tcp source source-wildcard [operator port [port]] destination destination-wildcard
    [operator port [port]] [established] [precedence precedence] [tos tos] [log]
```

For User Datagram Protocol (UDP), you can also use the following syntax:

```
access-list access-list-number [dynamic dynamic-name [timeout minutes]] {deny | permit}
    udp source source-wildcard [operator port [port]] destination destination-wildcard
    [operator port [port]] [precedence precedence] [tos tos] [log]
```

Syntax Description

<i>access-list-number</i>	Number of an encryption access list. This is a decimal number from 100 to 199.
dynamic <i>dynamic-name</i>	(Optional) Identifies this encryption access list as a dynamic encryption access list. Refer to lock-and-key access documented in the “Configuring Lock-and-Key Security (Dynamic Access Lists)” chapter in the <i>Security Configuration Guide</i> .
timeout <i>minutes</i>	(Optional) Specifies the absolute length of time (in minutes) that a temporary access list entry can remain in a dynamic access list. The default is an infinite length of time and allows an entry to remain permanently. Refer to lock-and-key access documented in the “Configuring Lock-and-Key Security (Dynamic Access Lists)” chapter in the <i>Security Configuration Guide</i> .
deny	Does not encrypt/decrypt IP traffic if the conditions are matched.
permit	Encrypts/decrypts IP traffic if the conditions are matched.

<i>protocol</i>	<p>Name or number of an IP protocol. It can be one of the keywords eigrp, gre, icmp, igmp, igrp, ip, ipinip, nos, ospf, tcp, or udp, or an integer in the range 0 to 255 representing an IP protocol number. To match any Internet protocol, including ICMP, TCP, and UDP, use the keyword ip. Some protocols allow further qualifiers, as described in text that follows.</p>
<i>source</i>	<p>Number of the network or host from which the packet is being sent. There are three other ways to specify the source:</p> <ul style="list-style-type: none"> • Use a 32-bit quantity in four-part dotted-decimal format. • Use the keyword any as an abbreviation for a <i>source</i> and <i>source-wildcard</i> of 0.0.0.0 255.255.255.255. This keyword is normally <i>not</i> recommended (see the section “Usage Guidelines”). • Use host source as an abbreviation for a <i>source</i> and <i>source-wildcard</i> of <i>source</i> 0.0.0.0.
<i>source-wildcard</i>	<p>Wildcard bits (mask) to be applied to source. There are three other ways to specify the source wildcard:</p> <ul style="list-style-type: none"> • Use a 32-bit quantity in four-part dotted-decimal format. Place ones in the bit positions you want to ignore. • Use the keyword any as an abbreviation for a <i>source</i> and <i>source-wildcard</i> of 0.0.0.0 255.255.255.255. This keyword is normally <i>not</i> recommended (see the section “Usage Guidelines”). • Use host source as an abbreviation for a <i>source</i> and <i>source-wildcard</i> of <i>source</i> 0.0.0.0.
<i>destination</i>	<p>Number of the network or host to which the packet is being sent. There are three other ways to specify the destination:</p> <ul style="list-style-type: none"> • Use a 32-bit quantity in four-part dotted-decimal format. • Use the keyword any as an abbreviation for the <i>destination</i> and <i>destination-wildcard</i> of 0.0.0.0 255.255.255.255. This keyword is normally <i>not</i> recommended (see the section “Usage Guidelines”). • Use host destination as an abbreviation for a <i>destination</i> and <i>destination-wildcard</i> of <i>destination</i> 0.0.0.0.
<i>destination-wildcard</i>	<p>Wildcard bits to be applied to the destination. There are three other ways to specify the destination wildcard:</p> <ul style="list-style-type: none"> • Use a 32-bit quantity in four-part dotted-decimal format. Place ones in the bit positions you want to ignore. • Use the keyword any as an abbreviation for a <i>destination</i> and <i>destination-wildcard</i> of 0.0.0.0 255.255.255.255. This keyword is normally <i>not</i> recommended (see the section “Usage Guidelines”). • Use host destination as an abbreviation for a <i>destination</i> and <i>destination-wildcard</i> of <i>destination</i> 0.0.0.0.
precedence <i>precedence</i>	<p>(Optional) Packets can be matched for encryption by precedence level, as specified by a number from 0 to 7 or by name as listed in the section “Usage Guidelines.”</p>

tos <i>tos</i>	(Optional) Packets can be matched for encryption by type of service level, as specified by a number from 0 to 15 or by name as listed in the section “Usage Guidelines.”
<i>icmp-type</i>	(Optional) ICMP packets can be matched for encryption by ICMP message type. The type is a number from 0 to 255.
<i>icmp-code</i>	(Optional) ICMP packets that are matched for encryption by ICMP message type can also be matched by the ICMP message code. The code is a number from 0 to 255.
<i>icmp-message</i>	(Optional) ICMP packets can be matched for encryption by an ICMP message type name or ICMP message type and code name. The possible names are discussed in the section “Usage Guidelines.”
<i>igmp-type</i>	(Optional) IGMP packets can be matched for encryption by IGMP message type or message name. A message type is a number from 0 to 15. IGMP message names are listed in the section “Usage Guidelines.”
<i>operator</i>	<p>(Optional) Compares source or destination ports. Possible operands include lt (less than), gt (greater than), eq (equal), neq (not equal), and range (inclusive range).</p> <p>If the operator is positioned after the <i>source</i> and <i>source-wildcard</i>, it must match the source port.</p> <p>If the operator is positioned after the <i>destination</i> and <i>destination-wildcard</i>, it must match the destination port.</p> <p>The range operator requires two port numbers. All other operators require one port number.</p>
<i>port</i>	<p>(Optional) The decimal number or name of a TCP or UDP port. A port number is a number from 0 to 65535.</p> <p>TCP port names are listed in the section “Usage Guidelines.” TCP port names can be used only when filtering TCP.</p> <p>UDP port names are listed in the section “Usage Guidelines.” UDP port names can be used only when filtering UDP.</p>
established	(Optional) For the TCP protocol only: Indicates an established connection. A match occurs if the TCP datagram has the ACK or RST bits set. The nonmatching case is that of the initial TCP datagram to form a connection.

log (Optional) Causes an informational logging message about the packet that matches the entry to be sent to the console. (The level of messages logged to the console is controlled by the **logging console** command.)

The message includes the access list number, whether the packet was encrypted/decrypted or not; the protocol, whether it was TCP, UDP, ICMP, or a number; and, if appropriate, the source and destination addresses and source and destination port numbers. The message is generated for the first packet that matches, and then at 5-minute intervals, including the number of packets encrypted/decrypted or not in the prior 5-minute interval.

Default

No numbered encryption access lists are defined, and therefore no traffic will be encrypted/decrypted. After being defined, all encryption access lists contain an implicit “deny” (“do not encrypt/decrypt”) statement at the end of the list.

Command Mode

Global configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

Use encryption access lists to control which packets on an interface are encrypted/decrypted, and which are transmitted as plain text (unencrypted).

When a packet is examined for an encryption access list match, encryption access list statements are checked in the order that the statements were created. After a packet matches the conditions in a statement, no more statements will be checked. This means that you need to carefully consider the order in which you enter the statements.

To use the encryption access list, you must first specify the access list in a crypto map and then apply the crypto map to an interface, using the **crypto map (global configuration)** and **crypto map (interface configuration)** commands.

Fragmented IP packets, other than the initial fragment, are immediately accepted by any extended IP access list. Extended access lists used to control virtual terminal line access or restrict contents of routing updates must not match the TCP source port, the type of service value, or the packet's precedence.

Note After an access list is created initially, any subsequent additions (possibly entered from the terminal) are placed at the end of the list. You cannot selectively add or remove access list command lines from a specific access list.



Caution When creating encryption access lists, we do *not* recommend using the **any** keyword to specify source or destination addresses. Using the **any** keyword with a **permit** statement could cause extreme problems if a packet enters your router and is destined for a router that is not configured for encryption. This would cause your router to attempt to set up an encryption session with a nonencrypting router.

If you incorrectly use the **any** keyword with a **deny** statement, you might inadvertently prevent all packets from being encrypted, which could present a security risk.

Note If you view your router's access lists by using a command such as **show ip access-lists**, *all* extended IP access lists will be shown in the command output. This includes extended IP access lists that are used for traffic filtering purposes as well as those that are used for encryption. The show command output does not differentiate between the two uses of the extended access lists.

The following is a list of precedence names:

- **critical**
- **flash**
- **flash-override**
- **immediate**
- **internet**
- **network**
- **priority**
- **routine**

The following is a list of type of service (TOS) names:

- **max-reliability**
- **max-throughput**
- **min-delay**
- **min-monetary-cost**
- **normal**

The following is a list of ICMP message type names and ICMP message type and code names:

- **administratively-prohibited**
- **alternate-address**
- **conversion-error**
- **dod-host-prohibited**
- **dod-net-prohibited**
- **echo**
- **echo-reply**
- **general-parameter-problem**
- **host-isolated**

- **host-precedence-unreachable**
- **host-redirect**
- **host-tos-redirect**
- **host-tos-unreachable**
- **host-unknown**
- **host-unreachable**
- **information-reply**
- **information-request**
- **mask-reply**
- **mask-request**
- **mobile-redirect**
- **net-redirect**
- **net-tos-redirect**
- **net-tos-unreachable**
- **net-unreachable**
- **network-unknown**
- **no-room-for-option**
- **option-missing**
- **packet-too-big**
- **parameter-problem**
- **port-unreachable**
- **precedence-unreachable**
- **protocol-unreachable**
- **reassembly-timeout**
- **redirect**
- **router-advertisement**
- **router-solicitation**
- **source-quench**
- **source-route-failed**
- **time-exceeded**
- **timestamp-reply**
- **timestamp-request**
- **traceroute**
- **ttl-exceeded**
- **unreachable**

The following is a list of IGMP message names:

- **dvmrp**
- **host-query**
- **host-report**
- **pim**
- **trace**

The following is a list of TCP port names that can be used instead of port numbers. Refer to the current Assigned Numbers RFC to find a reference to these protocols. Port numbers corresponding to these protocols can also be found by typing a ? in the place of a port number.

- **bgp**
- **chargen**
- **daytime**
- **discard**
- **domain**
- **echo**
- **finger**
- **ftp**
- **ftp-data**
- **gopher**
- **hostname**
- **irc**
- **klogin**
- **kshell**
- **lpd**
- **nntp**
- **pop2**
- **pop3**
- **smtp**
- **sunrpc**
- **syslog**
- **tacacs-ds**
- **talk**
- **telnet**
- **time**
- **uucp**
- **whois**
- **www**

The following is a list of UDP port names that can be used instead of port numbers. Refer to the current Assigned Numbers RFC to find a reference to these protocols. Port numbers corresponding to these protocols can also be found by typing a ? in the place of a port number.

- **biff**
- **bootpc**
- **bootps**
- **discard**
- **dns**
- **dnsix**
- **echo**
- **mobile-ip**
- **nameserver**
- **netbios-dgm**
- **netbios-ns**
- **ntp**
- **rip**
- **snmp**
- **snmptrap**
- **sunrpc**
- **syslog**
- **tacacs-ds**
- **talk**
- **tftp**
- **time**
- **who**
- **xdmcp**

Example

The following example creates a numbered encryption access list that specifies a class C subnet for the source and a class C subnet for the destination of IP packets. When the router uses this encryption access list, all TCP traffic that is exchanged between the source and destination subnets will be encrypted.

```
Apricot(config)# access-list 101 permit tcp 172.21.3.0 0.0.0.255 172.22.2.0 0.0.0.255
```

This encryption access list will be applied to an interface as an outbound encryption access list after the router administrator defines a crypto map and applies the crypto map to the interface.

Related Commands

You can use the master indexes or search online to find documentation of related commands.

access-list (extended) (used for traffic filtering purposes)

crypto map (global configuration)

crypto map (interface configuration)

ip access-list extended (encryption)

show ip access-lists

clear crypto connection

To terminate an encrypted session in progress, use the **clear crypto connection** global configuration command.

clear crypto connection *connection-id*

Syntax Description

connection-id Identifies the encrypted session to terminate.

Default

Encrypted sessions will normally terminate when the session times out.

Command Mode

Global configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

Use this command to terminate an encrypted session currently in progress. You can first perform the **show crypto connections** command to learn the connection id value.

Example

The following example clears a pending encrypted session. (You could also clear an established encrypted session in the same way.)

```
Apricot# show crypto connections
Pending Connection Table
PE                UPE                Timestamp                Conn_id
192.168.3.10      192.168.204.100  Mar 01 1993 00:01:09    -1

Connection Table
PE                UPE                Conn_id New_id  Alg      Time
192.168.3.10      192.168.204.100  -1      1      0        Not Set
                        flags:PEND_CONN

Apricot# clear crypto connection -1
Apricot# show crypto connections
Connection Table
PE                UPE                Conn_id New_id  Alg      Time
192.168.3.10      192.168.204.100  0       0      0        Mar 01 1993 00:02:00
                        flags:BAD_CONN

Apricot#
```

First, a **show crypto connections** command is issued to learn the connection id for the pending connection (-1). This value is then used to specify which connection to clear.

Notice that after the connection is cleared, the Pending Connection Table containing the connection entry (connection id of -1) has disappeared from the **show crypto connections** output. Also, the Connection Table no longer shows a -1 Conn_id.

Related Commands

You can use the master indexes or search online to find documentation of related commands.

show crypto connections

crypto algorithm 40-bit-des

To globally enable 40-bit Data Encryption Standard (DES) algorithm types, use the **crypto algorithm 40-bit-des** global configuration command. Use the **no** form of this command to globally disable a 40-bit DES algorithm type.

crypto algorithm 40-bit-des [cfb-8 | cfb-64]
no crypto algorithm 40-bit-des [cfb-8 | cfb-64]

Syntax Description

- cfb-8** (Optional) Selects the 8-bit Cipher FeedBack (CFB) mode of the 40-bit DES algorithm. If no CFB mode is specified when you issue the command, 64-bit CFB mode is the default.
- cfb-64** (Optional) Selects the 64-bit CFB mode of the 40-bit DES algorithm. If no CFB mode is specified when you issue the command, 64-bit CFB mode is the default.

Default

One DES algorithm is enabled by default, even if you never issue this command. If you are running a nonexportable image, the basic DES algorithm with 8-bit CFB is enabled by default. (The basic DES algorithm uses a 56-bit DES key.) If you are running an exportable image, the 40-bit DES algorithm with 8-bit CFB is enabled by default.

If you do not know if your image is exportable or nonexportable, you can perform the **show crypto algorithms** command to determine which DES algorithms are currently enabled.

Command Mode

Global configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

Use this command to enable a 40-bit DES algorithm type. Enabling a DES algorithm type once allows it to be used by all crypto engines of a router.

You must enable all DES algorithms that will be used to communicate with any other peer encrypting router. If you do not enable a DES algorithm, you will not be able to use that algorithm, even if you try to assign the algorithm to a crypto map at a later time.

If your router tries to set up an encrypted communication session with a peer router, and the two routers do not have the same DES algorithm enabled at both ends, the encrypted session will fail. If at least one common DES algorithm is enabled at both ends, the encrypted session can proceed.

Forty-bit DES uses a 40-bit DES key, which is easier for attackers to “crack” than basic DES, which uses a 56-bit DES key. However, some international applications might require you to use 40-bit DES, because of export laws.

Note If you are running an exportable image, you can only enable and use 40-bit variations of DES. You cannot enable or use the basic DES algorithms, which are not available with exportable images.

Eight-bit CFB is more commonly used than 64-bit CFB, but requires more CPU processing time. If you do not specify 8-bit or 64-bit CFB, 64-bit CFB will be selected by default.

Example

The following example enables 40-bit DES with 8-bit CFB and 40-bit DES with 64-bit CFB:

```
crypto algorithm 40-bit-des cfb-8
crypto algorithm 40-bit-des cfb-64
```

Related Commands

You can use the master indexes or search online to find documentation of related commands.

crypto algorithm des
show crypto algorithms

crypto algorithm des

To globally enable Data Encryption Standard (DES) algorithm types that use a 56-bit DES key, use the **crypto algorithm des** global configuration command. Use the **no** form of this command to globally disable a DES algorithm type.

```
crypto algorithm des [cfb-8 | cfb-64]
no crypto algorithm des [cfb-8 | cfb-64]
```

Syntax Description

- cfb-8** (Optional) Selects the 8-bit Cipher FeedBack (CFB) mode of the basic DES algorithm. If no CFB mode is specified when you issue the command, 64-bit CFB mode is the default.
- cfb-64** (Optional) Selects the 64-bit CFB mode of the basic DES algorithm. If no CFB mode is specified when you issue the command, 64-bit CFB mode is the default.

Default

One DES algorithm is enabled by default, even if you never issue this command. If you are running a nonexportable image, the basic DES algorithm with 8-bit CFB is enabled by default. (The basic DES algorithm uses a 56-bit DES key.) If you are running an exportable image, the 40-bit DES algorithm with 8-bit CFB is enabled by default.

If you do not know if your image is exportable or nonexportable, you can perform the **show crypto algorithms** command to determine which DES algorithms are currently enabled.

Command Mode

Global configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

Use this command to enable a DES algorithm type that uses a 56-bit DES key. Enabling a DES algorithm type once allows it to be used by all crypto engines of a router.

You must enable all DES algorithms that will be used to communicate with any other peer encrypting router. If you do not enable a DES algorithm, you will not be able to use that algorithm, even if you try to assign the algorithm to a crypto map at a later time.

If your router tries to set up an encrypted communication session with a peer router, and the two routers do not have the same DES algorithm enabled at both ends, the encrypted session will fail. If at least one common DES algorithm is enabled at both ends, the encrypted session can proceed.

Note If you are running an exportable image, you can only enable and use 40-bit variations of DES. You cannot enable or use the basic DES algorithms, which are not available with exportable images.

Eight-bit CFB is more commonly used than 64-bit CFB, but requires more CPU processing time. If you do not specify 8-bit or 64-bit CFB, 64-bit CFB will be selected by default.

Example

The following example enables DES with 8-bit CFB and DES with 64-bit CFB:

```
crypto algorithm des cfb-8
crypto algorithm des cfb-64
```

Related Commands

You can use the master indexes or search online to find documentation of related commands.

crypto algorithm 40-bit-des

show crypto algorithms

crypto clear-latch

To reset an Encryption Service Adapter (ESA), use the **crypto clear-latch** global configuration command. This command resets the ESA by clearing a hardware extraction latch that is set when an ESA is removed and reinstalled in the chassis.

crypto clear-latch *slot*

Syntax Description

slot Identifies the ESA to reset. On a Cisco 7200 series router, this is the ESA chassis slot number. On a Cisco RSP7000 or 7500 series router, this is the chassis slot number of the ESA's second-generation Versatile Interface Processor (VIP2).

Default

The ESA latch is not cleared.

Command Mode

Global configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

If an ESA is installed for the first time, or removed and reinstalled, the ESA will not function unless you reset it by using this command. Before the ESA is reset, the hardware extraction latch is set and the Tampered LED is on.

To complete this command, you must enter the ESA password. If the ESA does not have a password, you must create one at this time. (The ESA might not have a password if has never been previously used, or if the **crypto zeroize** command was previously issued for the ESA.)

If you have forgotten a previously assigned password, you have to use the **crypto zeroize** command instead of the **crypto clear-latch** command to reset the ESA. After issuing the **crypto zeroize** command, you must regenerate and re-exchange DSS keys. When you regenerate DSS keys you will be prompted to create a new password.

Example

The following example resets an ESA card. The ESA card is housed in a VIP2 that is in slot 1.

```
Apricot(config)# crypto clear-latch 1
% Enter the crypto card password.
Password: <passwd>
Apricot(config)#
```

Related Commands

You can use the master indexes or search online to find documentation of related commands.

crypto gen-signature-keys

crypto zeroize

crypto esa

To enable (select) either the ESA crypto engine or the Cisco IOS crypto engine in Cisco 7200 series routers, use the **crypto esa** global configuration command.

crypto esa {enable | shutdown} slot

Syntax Description

enable	Selects the ESA crypto engine by enabling the ESA.
shutdown	Selects the Cisco IOS crypto engine by shutting down the ESA.
<i>slot</i>	The ESA chassis slot number.

Default

The Cisco IOS crypto engine is the selected (active) crypto engine.

Command Mode

Global configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2P.

This command only applies to Cisco 7200 series routers with an installed ESA.

Until the ESA is enabled, the Cisco IOS crypto engine will function as the crypto engine.

If you want to select the ESA crypto engine with this command, all other encryption configuration must already have been completed for the ESA.

If you select a crypto engine (either the ESA or the Cisco IOS crypto engine) that has not been completely configured for encryption, the router will not be able to encrypt any traffic. Any existing encryption sessions will abruptly terminate. Therefore, you must complete all encryption configuration for before you a crypto engine with this command.

Note If any encryption session is in progress when you switch from one crypto engine to the other, the session will be torn down, and a new session will be established using the newly selected crypto engine. This could cause a momentary delay for encrypted traffic.

Examples

The following example enables an ESA in the router chassis slot 2:

```
Apricot(config)# crypto esa enable 2
...switching to HW crypto engine
Apricot(config)#
```

The following example switches from the Cisco IOS crypto engine to the ESA crypto engine. The ESA crypto engine is in the router chassis slot 4.

```
Apricot(config)# crypto esa enable 4  
...switching to HW crypto engine  
Apricot(config)#
```

The following example switches from the ESA crypto engine to the Cisco IOS crypto engine. The ESA crypto engine is in the router chassis slot 4.

```
Apricot(config)# crypto esa shutdown 4  
...switching to SW crypto engine  
Apricot(config)
```

crypto gen-signature-keys

To generate a Digital Signature Standard (DSS) public/private key pair, use the **crypto gen-signature-keys** global configuration command.

crypto gen-signature-keys *key-name* [*slot*]

Syntax Description

key-name A name you assign to the crypto engine. This will name either the Cisco IOS software crypto engine, a second-generation Versatile Interface Processor (VIP2) crypto engine, or an Encryption Service Adapter (ESA) crypto engine. Any character string is valid. Using a fully qualified domain name might make it easier to identify public keys.

slot (Optional) Identifies the crypto engine. This argument is available only on Cisco 7200, RSP7000, and 7500 series routers.

If no slot is specified, the Cisco IOS crypto engine will be selected.

Use the chassis slot number of the crypto engine location. For the Cisco IOS crypto engine, this is the chassis slot number of the Route Switch Processor (RSP). For the VIP2 crypto engine, this is the chassis slot number of the VIP2. For the ESA crypto engine, this is the chassis slot number of the ESA (Cisco 7200) or of the VIP2 (Cisco RSP7000 and 7500).

Default

No DSS public/private keys are defined.

Command Mode

Global configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

Use this command to generate a DSS public/private key pair. This is the first configuration task required to set up a router for network data encryption.

If you have a Cisco 7200, RSP7000, or 7500 series router, use the *slot* argument. You must perform this command once for each crypto engine you plan to use.

Note DSS keys of the Cisco IOS crypto engine are saved to a private portion of NVRAM when you perform a **copy running-config startup-config** (previously **write memory**) command. DSS keys are *not* saved with your configuration when you perform a **copy running-config rcp** or **copy running-config tftp** (previously **write network**) command.

If you are using a Cisco 7200, RSP7000, or 7500 series router with an ESA, DSS keys generated for the ESA crypto engine are automatically saved to tamper resistant memory of the ESA during the DSS key generation process.

Note If NVRAM fails, or if your ESA is tampered with or replaced, DSS public/private keys will no longer be valid. If this happens, you will need to regenerate and re-exchange DSS keys.

The ESA Password

If you are using a Cisco 7200, RSP7000, or 7500 series router with an ESA, you will be prompted to enter a password when you generate DSS keys for the ESA crypto engine.

If you previously reset the ESA with the **crypto zeroize** command, you must create a new password at this time.

If you previously reset the ESA with the **crypto clear-latch** command, you created a password at that time; use that same password now. If you have forgotten the password, the only workaround is to first use the **crypto zeroize** command and then regenerate DSS keys.

If you ever again need to regenerate DSS keys for the ESA, you will be required to enter the same ESA password to complete the DSS key regeneration.

Examples

The following example generates a DSS public/private key pair for the first time on a Cisco 2500 series router:

```
Apricot(config)# crypto gen-signature-keys Apricot
Generating DSS keys .... [OK]
Apricot(config)#
```

The following example generates DSS public/private key pairs for a Cisco 7500 series router with an RSP in slot 4 and a VIP2 (with an ESA) in slot 3. The ESA was previously reset with the **crypto zeroize** command. Notice that when DSS keys are generated for the ESA, you must type a newly created password:

```
Apricot(config)# crypto gen-signature-keys ApricotRSP 4
Generating DSS keys .... [OK]
Apricot(config)# crypto gen-signature-keys ApricotESA 3
% Initialize the crypto card password. You will need
  this password in order to generate new signature
  keys or clear the crypto card extraction latch.

Password: <passwd>

Re-enter password: <passwd>

Generating DSS keys .... [OK]
Apricot(config)#
```

In the previous example, the ESA crypto engine provides encryption services for the VIP2 interfaces, and the Cisco IOS crypto engine (located in the RSP) provides encryption services for all other designated ports.

The next example shows DSS keys being generated a second time, for the same ESA crypto engine shown in the previous example (DSS keys already exist for this crypto engine). Notice that the password used in the previous example must be entered in this example to complete the DSS key regeneration.

```
Apricot(config)# crypto gen-signature-keys ApricotESA 3
% Generating new DSS keys will require re-exchanging
  public keys with peers who already have the public key
    named ApricotESA!
Generate new DSS keys? [yes/no]: y
% Enter the crypto card password.
Password: <passwd>
Generating DSS keys .... [OK]
```

Related Commands

You can use the master indexes or search online to find documentation of related commands.

show crypto mypubkey

crypto key-exchange

To exchange Digital Signature Standard (DSS) public keys, the administrator of the peer encrypting router that is designated ACTIVE must use the **crypto key-exchange** global configuration command.

crypto key-exchange *ip-address* *key-name* [*tcp-port*]

Syntax Description

<i>ip-address</i>	The IP address of the peer router (designated PASSIVE) participating with you in the key exchange.
<i>key-name</i>	Identifies the crypto engine—either the Cisco IOS crypto engine, a second-generation Versatile Interface Processor (VIP2) crypto engine, or an Encryption Service Adapter (ESA) crypto engine. This name must match the key-name argument assigned when you generated DSS keys using the crypto gen-signature-keys command.
<i>tcp-port</i>	(Optional) Cisco IOS software uses the unassigned ¹ TCP port number of 1964 to designate a key exchange. You may use this optional keyword to select a different number to designate a key exchange, if your system already uses the port number 1964 for a different purpose. If this keyword is used, you must use the same value as the PASSIVE router's <i>tcp-port</i> value.

1. 1964 is a TCP port number that has not been preassigned by the Internetworking Engineering Task Force (IETF).

Default

No DSS keys are exchanged.

Command Mode

Global configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

Peer encrypting routers must exchange DSS public keys before any encrypted communication can occur.

If you have a Cisco 7200, RSP7000, or 7500 series router, you will need to exchange DSS public keys for each crypto engine you plan to use.

To exchange DSS public keys, the two router administrators must call each other on the phone, and verbally assign one router to the PASSIVE role, and the other router to the ACTIVE role.

The PASSIVE administrator uses the **crypto key-exchange passive** command to start the DSS key exchange. Then the ACTIVE administrator uses the **crypto key-exchange** command to send the first DSS public key. During the key exchange sequence, the two administrators must remain on the phone to verify the receipt of DSS keys. To verify the receipt of DSS keys, the administrators should compare screens to match DSS key serial numbers and fingerprints. Screen prompts will guide both administrators through the exchange.

Example

The following example shows a DSS key exchange sequence from the point of view of a router named Banana. Banana is designated ACTIVE. The other router is named Apricot. Apricot is designated PASSIVE, and has previously generated DSS keys with the *key-name* Apricot. Banana has previously generated DSS keys with the *key-name* BananaESA:

```
Banana(config)# crypto key-exchange 172.21.114.68 BananaESA
Public key for BananaESA:
  Serial Number 01461300
  Fingerprint   0F1D 373F 2FC1 872C D5D7

Wait for peer to send a key[confirm]<Return>
Waiting ....
Public key for Apricot:
  Serial Number 01579312
  Fingerprint   BF1F 9EAC B17E F2A1 BA77

Add this public key to the configuration? [yes/no]: y
Banana(config)#
```

Related Commands

You can use the master indexes or search online to find documentation of related commands.

- crypto key-exchange passive**
- crypto public-key**
- show crypto mypubkey**
- show crypto pubkey**
- show crypto pubkey name**
- show crypto pubkey serial**

crypto key-exchange passive

To enable an exchange of Digital Signature Standard (DSS) public keys, the administrator of the peer encrypting router that is designated PASSIVE must use the **crypto key-exchange passive** global configuration command.

crypto key-exchange passive [*tcp-port*]

Syntax Description

tcp-port (Optional) Cisco IOS software uses the unassigned¹ TCP port number of 1964 to designate a key exchange. You may use this optional keyword to select a different number to designate a key exchange, if your system already uses the port number 1964 for a different purpose. If this keyword is used, you must use the same value as the ACTIVE router's *tcp-port* value.

1. 1964 is a TCP port number that has not been preassigned by the Internetworking Engineering Task Force (IETF).

Default

No DSS keys are exchanged.

Command Mode

Global configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

Peer encrypting routers must exchange DSS public keys before any encrypted communication can occur.

To exchange DSS public keys, the two router administrators must call each other on the phone, and verbally assign one router to the PASSIVE role, and the other router to the ACTIVE role.

Then the PASSIVE administrator should use the **crypto key-exchange passive** command to start the DSS key exchange. During the key exchange sequence, the two administrators must remain on the phone to verify the receipt of DSS keys. To verify the receipt of DSS keys, the administrators should compare screens to match DSS key serial numbers and fingerprints. Screen prompts will guide both administrators through the exchange.

Example

The following example shows a DSS key exchange sequence from the point of view of a router named Apricot. Apricot is designated PASSIVE, and has previously generated DSS keys with the *key-name* Apricot. The other router is named Banana and has previously generated DSS keys with the *key-name* BananaESA:

```
Apricot(config)# crypto key-exchange passive
Enter escape character to abort if connection does not complete.
Wait for connection from peer[confirm]<Return>
Waiting ....
Public key for BananaESA:
  Serial Number 01461300
  Fingerprint   0F1D 373F 2FC1 872C D5D7
Add this public key to the configuration? [yes/no]: y
Send peer a key in return[confirm]<Return>
Which one?

Apricot? [yes]: <Return>
Public key for Apricot:
  Serial Number 01579312
  Fingerprint   BF1F 9EAC B17E F2A1 BA77

Apricot(config)#
```

Related Commands

You can use the master indexes or search online to find documentation of related commands.

- crypto key-exchange**
- crypto public-key**
- show crypto mypubkey**
- show crypto pubkey**
- show crypto pubkey name**
- show crypto pubkey serial**

crypto key-timeout

To specify the time duration of encrypted sessions, use the **crypto key-timeout** global configuration command. Use the **no** form to restore the time duration of encrypted sessions to the default of 30 minutes.

crypto key-timeout *minutes*
no crypto key-timeout *minutes*

Syntax Description

minutes Specifies the time duration of encrypted sessions. Can be from 1 to 1440 minutes (24 hours) in 1 minute increments. Specified by an integer from 1 to 1440.

When the **no** form of the command is used, this argument is optional. Any value supplied for the argument is ignored by the router.

Default

Encrypted sessions time out in 30 minutes.

Command Mode

Global configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

After an encrypted communication session is established, it is valid for a specific length of time. After this length of time, the session times out. A new session must be negotiated, and a new Data Encryption Standard (DES) (session) key must be generated for encrypted communication to continue. Use this command to change the time that an encrypted communication session will last before it expires (times out)

Examples

The following example sets encrypted session timeouts to 2 hours:

```
crypto key-timeout 120
```

The following example shows one way to restore the default session time of 30 minutes:

```
no crypto key-timeout
```

The following example shows another way to restore the default session time of 30 minutes:

```
crypto key-timeout 30
```

Related Commands

You can use the master indexes or search online to find documentation of related commands.

show crypto key-timeout

crypto map (global configuration)

To create or modify a crypto map definition and enter the crypto map configuration mode, use the **crypto map** global configuration command. Use the **no** form of this command to delete a crypto map definition.

```
crypto map map-name seq-num  
no crypto map map-name seq-num
```

Syntax Description

<i>map-name</i>	The name you assign to the crypto map.
<i>seq-num</i>	Identifies the sequence number (definition set) of the crypto map. See additional explanation for using this argument in the “Usage Guidelines” section.

Default

No crypto maps exist.

Command Mode

Global configuration.

Performing this command invokes the crypto map configuration command mode.

Usage Guidelines

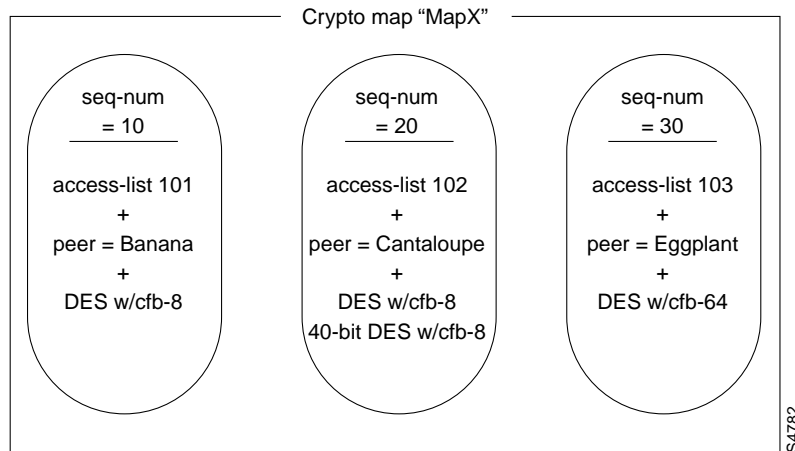
This command first appeared in Cisco IOS Release 11.2.

Use this command to create a new crypto map definition, or to modify an existing crypto map definition. Crypto maps link together definitions of encryption access lists, peer routers, and Data Encryption Standard (DES) algorithms. A crypto map must later be applied to an interface for the definitions to take effect; this is done using the **crypto map (interface configuration)** command.

When you issue the **crypto map (global configuration)** command, the router will invoke the crypto map configuration command mode. While in this mode, you will specify the crypto map definitions. Crypto map configuration command mode commands are used to create these definitions.

A crypto map definition must have three parts. First, you specify which remote peer encrypting router (crypto engine) will provide the far-end encryption services (the remote encryption end-point). This is accomplished using the **set peer** command. Next, you specify which encryption access list(s) will participate in encryption services with the peer router. This is accomplished using the **match address** command. Finally, you specify which DES algorithm(s) to apply to the encrypted packets in the access list. This is accomplished using either the **set algorithm 40-bit-des** command or the **set algorithm des** command.

Because only one crypto map can be applied to a given interface, the *seq-num* argument provides a way to create several distinct definition sets that coexist within a single crypto map. Figure 3 illustrates the sequence number concept.

Figure 3 Crypto Map with Subdefinitions

Having multiple distinct definition sets is useful if one router port will provide the encryption interface to more than one peer router.

Example

The following example creates a crypto map and defines the map parameters:

```
Apricot(config)# crypto map Research 10
Apricot(config-crypto-map)# set peer BananaESA.HQ
Apricot(config-crypto-map)# set algorithm des cfb-8
Apricot(config-crypto-map)# match address 101
Apricot(config-crypto-map)# exit
Apricot(config)#
```

Related Commands

You can use the master indexes or search online to find documentation of related commands.

crypto map (interface configuration)

match address
set algorithm 40-bit-des
set algorithm des
set peer
show crypto map
show crypto map interface
show crypto map tag

crypto map (interface configuration)

To apply a previously defined crypto map to an interface, use the **crypto map** interface configuration command. Use the **no** form of the command to eliminate the crypto map from the interface.

crypto map *map-name*
no crypto map *map-name*

Syntax Description

map-name The name which identifies the crypto map. This is the name you assigned when creating the crypto map.

When the **no** form of the command is used, this argument is optional. Any value supplied for the argument is ignored by the router.

Default

No crypto maps are assigned to interfaces.

Command Mode

Interface configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

Use this command to assign a crypto map to an interface. You must assign a crypto map to an interface before that interface can provide encryption services. Only one crypto map can be assigned to an interface. If there are multiple subdefinitions to the crypto map (for example, crypto map Research 10 and crypto map Research 20) each subdefinition will be applied when the single crypto map is applied.

Example

The following example assigns crypto map “Research” to the serial interface 0:

```
Apricot(config)# interface serial 0
Apricot(config-if)# crypto map Research
Apricot(config-if)# exit
Apricot(config)#
```

Related Commands

You can use the master indexes or search online to find documentation of related commands.

crypto map (global configuration)

show crypto map

show crypto map interface

show crypto map tag

crypto pregen-dh-pairs

To enable pregeneration of Diffie-Hellman (DH) public numbers, use the **crypto pregen-dh-pairs** global configuration command. Use the **no** form to disable pregeneration of DH public numbers for all crypto engines.

```
crypto pregen-dh-pairs count [slot]  
no crypto pregen-dh-pairs
```

Syntax Description

<i>count</i>	Specifies how many DH public numbers to pregenerate and hold in reserve. Specified by an integer from 0 to 10.
<i>slot</i>	(Optional) Identifies the crypto engine. This argument is available only on Cisco 7200, RSP7000, and 7500 series routers. If no slot is specified, the Cisco IOS crypto engine will be selected. Use the chassis slot number of the crypto engine location. For the Cisco IOS crypto engine, this is the chassis slot number of the Route Switch Processor (RSP). For the VIP2 crypto engine, this is the chassis slot number of the VIP2. For the ESA crypto engine, this is the chassis slot number of the ESA (Cisco 7200) or of the VIP2 (Cisco RSP7000 and 7500).

Default

DH number pairs are generated only when needed, during encrypted session setup.

Command Mode

Global configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

Each encrypted session uses a unique pair of DH numbers. Every time a new session is set up, new DH number pairs must be generated. When the session completes, these numbers are discarded. Generating new DH number pairs is a CPU-intensive activity, which can make session setup slow—especially for low-end routers.

To speed up session setup, you can choose to have a specified amount of DH number pairs pregenerated and held in reserve. Then, when an encrypted communication session is being set up, a DH number pair will be provided from that reserve. After a DH number pair is used, the reserve is automatically replenished with a new DH number pair, so that there should always be a DH number pair ready for use.

It is usually not necessary to have more than one or two DH number pairs pregenerated, unless your router will be setting up multiple encrypted sessions so frequently that a pregenerated reserve of one or two DH number pairs will be depleted too quickly.

If you have a Cisco 7200, RSP7000, or 7500 series router, you can perform this command for each crypto engine in service.

Setting the number of pregenerated pairs to be zero disables pregeneration but allows you to use the pairs already in reserve. Using the **no** form of the command disables pregeneration for *all* crypto engines of your router and deletes any DH number pairs currently in reserve. If you have a Cisco 7200, RSP7000, or 7500 series router and wish to discontinue pregenerating DH numbers for only one crypto engine, set the *count* argument to 0, and specify the crypto engine with the *slot* argument.

Examples

The following example turns on pregeneration of DH public number pairs for a Cisco 2500 series router. Two DH number pairs will be held in constant reserve.

```
crypto pregen-dh-pairs 2
```

The following example turns on pregeneration of DH public numbers for the ESA crypto engine of a VIP2 card in slot 3 of a Cisco 7500 series router. One DH number pair will be held in constant reserve.

```
crypto pregen-dh-pairs 1 3
```

Related Commands

You can use the master indexes or search online to find documentation of related commands.

show crypto pregen-dh-pairs

crypto public-key

To manually specify the Digital Signature Standard (DSS) public key of a peer encrypting router, use the **crypto public-key** global configuration command. Use the **no** form of this command to delete the DSS public key of a peer encrypting router.

```
crypto public key key-name serial-number
    hex-key-data
    hex-key-data...
quit
no crypto public key key-name serial-number
```

Syntax Description

<i>key-name</i>	Identifies the crypto engine of the peer encrypting router.
<i>serial-number</i>	The serial number of the peer encrypting router's public DSS key. When the no form of the command is used, this argument is optional. Any value supplied for the argument is ignored by the router.
<i>hex-key-data</i>	The DSS public key of the peer encrypting router, in hexadecimal format.
quit	When you are done entering the public key, type quit to exit the hex input mode.

Default

No peer encrypting router DSS keys are known.

Command Mode

Global configuration

Performing this command invokes the hex input mode. To complete the command, you must return to the global configuration mode by typing **quit** at the config-pubkey prompt.

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

You can choose to use this command to specify DSS public keys of peer encrypting routers, instead of using the **crypto key-exchange passive** and **crypto key-exchange** commands. The administrator of the peer router can provide you with the exact values for the *key-name*, *serial-number*, and *hex-key-data* command arguments. The administrator of the peer router can discover these values by performing the **show crypto mypubkey** command at the peer router.

You should press **Return** after typing the *serial-number* argument. You will then be in the hex input mode. Enter the peer DSS public key in hexadecimal data. Entering the data will take more than one line; press **Return** to continue entering hexadecimal data on a new line. After typing in the hexadecimal data, you should press **Return** to get to a new line of the config-pubkey prompt, then type **quit** to complete the command.

Example

The following example specifies the DSS public key of a peer encrypting router:

```
Apricot(config)# crypto public-key BananaCryptoEngine 01709644
Enter a public key as a hexadecimal number ....

Apricot(config-pubkey)# C31260F4 BD8A5ACE 2C1B1E6C 8B0ABD27 01493A50
Apricot(config-pubkey)# 2E90AF19 8B29122B 2D479B15 437A0F7C BCBE5300
Apricot(config-pubkey)# 29859ED7 EAA2848E A31D7FD6 C8911D9A 9701CA00
Apricot(config-pubkey)# A6A66946
Apricot(config-pubkey)# quit
Apricot(config)# exit
Apricot#
%SYS-5-CONFIG_I: Configured from console by console
Apricot# show crypto pubkey
crypto public-key BananaCryptoEngine 01709644
C31260F4 BD8A5ACE 2C1B1E6C 8B0ABD27 01493A50 2E90AF19 8B29122B 2D479B15
437A0F7C BCBE5300 29859ED7 EAA2848E A31D7FD6 C8911D9A 9701CA00 A6A66946
quit

Apricot#
```

Related Commands

You can use the master indexes or search online to find documentation of related commands.

```
crypto key-exchange
crypto key-exchange passive
show crypto mypubkey
show crypto pubkey
show crypto pubkey name
show crypto pubkey serial
```

crypto sdu connections

To change the maximum number of destinations (hosts or subnets) per source that you can define in encryption access list statements, use the **crypto sdu connections** global configuration command. Use the **no** form of the command to restore the default.

crypto sdu connections *number*
no crypto sdu connections *number*

Syntax Description

number Specifies the maximum number of destinations per source. Use a value from 3 to 500.

This argument is not required when using the **no** form of the command.

Default

A maximum of 10 destinations can be paired with each source specified in encryption access list criteria statements.

Command Mode

Global configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.3.

When you configure encryption access lists, you configure source and destination pairs in criteria statements. Any traffic that matches the criteria is then encrypted.

By default, the maximum number of distinct sources (host or subnets) that you can define in your encryption access lists is 100. Also, the maximum number of distinct destinations that you can define for any given source address is 10. For example, if you define six different source addresses, you can define up to 10 destination addresses for each of the six sources, for a total of 60 access list criteria statements.

Use this command if you need to specify more than 10 destinations for a particular source (host or subnet) in encryption access list statements.

For most situations, the defaults of 100 maximum sources and 10 maximum destinations per source are sufficient. Cisco recommends that you do not change the defaults unless you actually exceed the number of sources or destinations per source.

Note You must reboot the router before this command takes effect.

Memory Impact

The amount of memory reserved for encrypted connections changes if you change the defaults with this command.

When using this command, you should consider the amount of memory that will be allocated. In general, use the **crypto sdu entities** and **crypto sdu connections** commands together: if you increase one value, decrease the other value. This prevents your router from running out of memory because too much memory was preallocated.

For every additional source specified with the **crypto sdu entities** command, the following additional bytes of memory will be allocated:

$$64 + (68 \times \text{the specified number of maximum destinations})$$

For every additional destination specified with the **crypto sdu connections**, the following additional bytes of memory will be allocated:

$$68 \times \text{the specified number of maximum sources}$$

For example, if you specify 5 maximum sources, and 250 maximum destinations per source, the memory allocated for encryption connections is calculated as follows:

$$\{5 \times [64 + (68 \times 250)]\} + \{250 \times (68 \times 5)\} = 170320 \text{ bytes}$$

Example

In this example, there are 50 remote sites connecting to a single server. The connections between the server and each site need to be encrypted. The server is located behind the local router named Apricot. Each of the remote sites connects through its own router.

Because of the large number of destination addresses that must be paired with the same source address in the local encryption access list, the default limits are changed.

```
Apricot(config)# crypto sdu connections 60
%Please reboot for the new connection size to take effect

Apricot(config)# crypto sdu entities 5
%Please reboot for the new table size to take effect
```

Note that the maximum number of sources is reduced to balance the increase in maximum destinations per source. This prevents too much memory from being preallocated to encryption connections.

Also note that even though there is only one server, and only 50 remote sites, this example defines 5 sources and 60 destinations. This allows room for future growth of the encryption access list. If another source or destination is added later, the limits will not have to be increased and the router rebooted again, which is a disruptive process.

Related Commands

You can use the master indexes or search online to find documentation of related commands.

crypto sdu entities

crypto sdu entities

To change the maximum number of sources (hosts or subnets) that you can define in encryption access list statements, use the **crypto sdu entities** global configuration command. Use the **no** form of the command to restore the default.

crypto sdu entities *number*
no crypto sdu entities [*number*]

Syntax Description

number Specifies the maximum number of sources. Use a value from 3 to 500.
This argument is not required when using the **no** form of the command.

Default

A maximum of 100 sources can be specified in encryption access list criteria statements.

Command Mode

Global configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.3.

When you configure encryption access lists, you configure source and destination pairs in criteria statements. Any traffic that matches the criteria is then encrypted.

By default, the maximum number of distinct sources (host or subnets) that you can define in your encryption access lists is 100. Also, the maximum number of distinct destinations that you can define for any given source address is 10. For example, if you define six different source addresses, you can define up to 10 destination addresses for each of the six sources, for a total of 60 access list criteria statements.

Use this command if you need to specify more than 100 sources (host or subnet) in encryption access list statements.

For most situations, the defaults of 100 maximum sources and 10 maximum destinations per source are sufficient. Cisco recommends that you do not change the defaults unless you actually exceed the number of sources or destinations per source.

Note You must reboot the router before this command takes effect.

Memory Impact

The amount of memory reserved for encrypted connections changes if you change the defaults with this command.

When using this command, you should consider the amount of memory that will be allocated. In general, use the **crypto sdu entities** and **crypto sdu connections** commands together: if you increase one value, decrease the other value. This prevents your router from running out of memory because too much memory was preallocated.

For every additional source specified with the **crypto sdu entities** command, the following additional bytes of memory will be allocated:

$$64 + (68 \times \text{the specified number of maximum destinations})$$

For every additional destination specified with the **crypto sdu connections**, the following additional bytes of memory will be allocated:

$$68 \times \text{the specified number of maximum sources}$$

For example, if you specify 5 maximum sources, and 250 maximum destinations per source, the memory allocated for encryption connections is calculated as follows:

$$\{5 \times [64 + (68 \times 250)]\} + \{250 \times (68 \times 5)\} = 170320 \text{ bytes}$$

Example

In this example, there are 50 remote sites connecting to a single server. The connections between the server and each site need to be encrypted. The server is located behind the local router named Apricot. Each of the remote sites connects through its own router.

Because of the large number of destination addresses that must be paired with the same source address in the local encryption access list, the default limits are changed.

```
Apricot(config)# crypto sdu connections 60
%Please reboot for the new connection size to take effect

Apricot(config)# crypto sdu entities 5
%Please reboot for the new table size to take effect
```

Note that the maximum number of sources is reduced to balance the increase in maximum destinations per source. This prevents too much memory from being preallocated to encryption connections.

Also note that even though there is only one server, and only 50 remote sites, this example defines 5 sources and 60 destinations. This allows room for future growth of the encryption access list. If another source or destination is added later, the limits will not have to be increased and the router rebooted again, which is a disruptive process.

Related Commands

You can use the master indexes or search online to find documentation of related commands.

crypto sdu connections

crypto zeroize

To delete the Digital Signature Standard (DSS) public/private key pair of a crypto engine, use the **crypto zeroize** global configuration command.

crypto zeroize [*slot*]



Caution DSS keys cannot be recovered after they have been removed. Use this command only after careful consideration.

Syntax Description

slot (Optional) Identifies the crypto engine. This argument is available only on Cisco 7200, RSP7000, and 7500 series routers.

If no slot is specified, the Cisco IOS crypto engine will be selected.

Use the chassis slot number of the crypto engine location. For the Cisco IOS crypto engine, this is the chassis slot number of the Route Switch Processor (RSP). For the VIP2 crypto engine, this is the chassis slot number of the VIP2. For the ESA crypto engine, this is the chassis slot number of the ESA (Cisco 7200) or of the VIP2 (Cisco RSP7000 and 7500).

Default

DSS public/private keys will remain valid indefinitely.

Command Mode

Global configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

If you choose to stop using encryption on a router, completely or for a specific crypto engine only, you may delete the public/private DSS key pair(s) for your router's crypto engine(s). However, after you delete DSS key pairs for a specified crypto engine, you will no longer be able to use that crypto engine to have any encrypted sessions with peer routers, unless you regenerate and re-exchange new DSS keys. If only one crypto engine is configured at your router, issuing this command will prevent you from performing any encryption at the router.



Caution If you use this command on a Cisco 7200 series router, the current active crypto engine's DSS keys will be deleted. Be certain that the engine for which you want to delete keys is the engine that is currently selected. You can use the **show crypto engine configuration** command to verify the current crypto engine. If the current crypto engine is not the engine for which you want to delete DSS keys, you must select the correct crypto engine using the **crypto esa** command.

This command can be used if you lose the password required to complete the **crypto clear-latch** or **crypto gen-signature-keys** commands. After using the **crypto zeroize** command, you will need to regenerate and re-exchange new DSS keys. You will be prompted to supply a new password when you regenerate new DSS keys with the **crypto gen-signature-keys** command.

Example

The following example deletes the DSS public/private key of a router named Apricot, which is a Cisco 7500 series router with an RSP in slot 4:

```
Apricot(config)# crypto zeroize 4
Warning! Zeroize will remove your DSS signature keys.
Do you want to continue? [yes/no]: y
Keys to be removed are named ApricotIOS.
Do you really want to remove these keys? [yes/no]: y
[OK]
Apricot(config)#
```

Related Commands

You can use the master indexes or search online to find documentation of related commands.

crypto gen-signature-keys

deny

To set conditions for a named encryption access list, use the **deny** access-list configuration command. The **deny** command prevents IP traffic from being encrypted/decrypted if the conditions are matched. To remove a deny condition from an encryption access list, use the **no** form of this command.

```
deny source [source-wildcard]
no deny source [source-wildcard]

deny protocol source source-wildcard destination destination-wildcard [precedence
precedence] [tos tos] [log]
no deny protocol source source-wildcard destination destination-wildcard [precedence
precedence] [tos tos] [log]
```

For ICMP, you can also use the following syntax:

```
deny icmp source source-wildcard destination destination-wildcard [icmp-type [icmp-code] |
icmp-message] [precedence precedence] [tos tos] [log]
```

For IGMP, you can also use the following syntax:

```
deny igmp source source-wildcard destination destination-wildcard [igmp-type]
[precedence precedence] [tos tos] [log]
```

For TCP, you can also use the following syntax:

```
deny tcp source source-wildcard [operator port [port]] destination destination-wildcard
[operator port [port]] [established] [precedence precedence] [tos tos] [log]
```

For UDP, you can also use the following syntax:

```
deny udp source source-wildcard [operator port [port]] destination destination-wildcard
[operator port [port]] [precedence precedence] [tos tos] [log]
```

Syntax Description

<i>source</i>	Number of the network or host from which the packet is being sent. There are two alternative ways to specify the source: <ul style="list-style-type: none"> • Use a 32-bit quantity in four-part, dotted-decimal format. • Use the keyword any as an abbreviation for a <i>source</i> and <i>source-wildcard</i> of 0.0.0.0 255.255.255.255. This keyword is normally <i>not</i> recommended (see the section “Usage Guidelines”).
<i>source-wildcard</i>	(Optional) Wildcard bits to be applied to the <i>source</i> . There are two alternative ways to specify the source wildcard: <ul style="list-style-type: none"> • Use a 32-bit quantity in four-part, dotted-decimal format. Place ones in the bit positions you want to ignore. • Use the keyword any as an abbreviation for a <i>source</i> and <i>source-wildcard</i> of 0.0.0.0 255.255.255.255. This keyword is normally <i>not</i> recommended (see the section “Usage Guidelines”).

<i>protocol</i>	Name or number of an IP protocol. It can be one of the keywords eigrp , gre , icmp , igmp , igrp , ip , ipinip , nos , ospf , tcp , or udp , or an integer in the range 0 through 255 representing an IP protocol number. To match any Internet protocol (including ICMP, TCP, and UDP), use the keyword ip . Some protocols allow further qualifiers described later.
<i>source</i>	<p>Number of the network or host from which the packet is being sent. There are three alternative ways to specify the source:</p> <ul style="list-style-type: none">• Use a 32-bit quantity in four-part, dotted-decimal format.• Use the keyword any as an abbreviation for a <i>source</i> and <i>source-wildcard</i> of 0.0.0.0 255.255.255.255. This keyword is normally <i>not</i> recommended (see the section “Usage Guidelines”).• Use host source as an abbreviation for a <i>source</i> and <i>source-wildcard</i> of <i>source</i> 0.0.0.0.
<i>source-wildcard</i>	<p>Wildcard bits to be applied to source. There are three alternative ways to specify the source wildcard:</p> <ul style="list-style-type: none">• Use a 32-bit quantity in four-part, dotted-decimal format. Place ones in the bit positions you want to ignore.• Use the keyword any as an abbreviation for a <i>source</i> and <i>source-wildcard</i> of 0.0.0.0 255.255.255.255. This keyword is normally <i>not</i> recommended (see the section “Usage Guidelines”).• Use host source as an abbreviation for a <i>source</i> and <i>source-wildcard</i> of <i>source</i> 0.0.0.0.
<i>destination</i>	<p>Number of the network or host to which the packet is being sent. There are three alternative ways to specify the destination:</p> <ul style="list-style-type: none">• Use a 32-bit quantity in four-part, dotted-decimal format.• Use the keyword any as an abbreviation for the <i>destination</i> and <i>destination-wildcard</i> of 0.0.0.0 255.255.255.255. This keyword is normally <i>not</i> recommended (see the section “Usage Guidelines”).• Use host destination as an abbreviation for a <i>destination</i> and <i>destination-wildcard</i> of <i>destination</i> 0.0.0.0.
<i>destination-wildcard</i>	<p>Wildcard bits to be applied to the destination. There are three alternative ways to specify the destination wildcard:</p> <ul style="list-style-type: none">• Use a 32-bit quantity in four-part, dotted-decimal format. Place ones in the bit positions you want to ignore.• Use the keyword any as an abbreviation for a <i>destination</i> and <i>destination-wildcard</i> of 0.0.0.0 255.255.255.255. This keyword is normally <i>not</i> recommended (see the section “Usage Guidelines”).• Use host destination as an abbreviation for a <i>destination</i> and <i>destination-wildcard</i> of <i>destination</i> 0.0.0.0.
precedence <i>precedence</i>	(Optional) Packets can be matched for encryption by precedence level, as specified by a number from 0 to 7 or by name as listed in the section “Usage Guidelines” section of the access-list (encryption) command.

tos <i>tos</i>	(Optional) Packets can be matched for encryption by type of service level, as specified by a number from 0 to 15 or by name as listed in the “Usage Guidelines” section of the access-list (encryption) command.
<i>icmp-type</i>	(Optional) ICMP packets can be matched for encryption by ICMP message type. The type is a number from 0 to 255.
<i>icmp-code</i>	(Optional) ICMP packets which are matched for encryption by ICMP message type can also be matched by the ICMP message code. The code is a number from 0 to 255.
<i>icmp-message</i>	(Optional) ICMP packets can be matched for encryption by an ICMP message type name or ICMP message type and code name. The possible names are found in the “Usage Guidelines” section of the access-list (encryption) command.
<i>igmp-type</i>	(Optional) IGMP packets can be matched for encryption by IGMP message type or message name. A message type is a number from 0 to 15. IGMP message names are listed in the “Usage Guidelines” section of the access-list (encryption) command.
<i>operator</i>	<p>(Optional) Compares source or destination ports. Possible operands include lt (less than), gt (greater than), eq (equal), neq (not equal), and range (inclusive range).</p> <p>If the operator is positioned after the <i>source</i> and <i>source-wildcard</i>, it must match the source port.</p> <p>If the operator is positioned after the <i>destination</i> and <i>destination-wildcard</i>, it must match the destination port.</p> <p>The range operator requires two port numbers. All other operators require one port number.</p>
<i>port</i>	(Optional) The decimal number or name of a TCP or UDP port. A port number is a number from 0 to 65,535. TCP and UDP port names are listed in the “Usage Guidelines” section of the access-list (encryption) command. TCP port names can only be used when filtering TCP. UDP port names can only be used when filtering UDP.
established	(Optional) For the TCP protocol only: Indicates an established connection. A match occurs if the TCP datagram has the ACK or RST bits set. The nonmatching case is that of the initial TCP datagram to form a connection.

log

(Optional) Causes an informational logging message about the packet that matches the entry to be sent to the console. (The level of messages logged to the console is controlled by the **logging console** command.)

The message includes the access list number, whether the packet was permitted or denied; the protocol, whether it was TCP, UDP, ICMP or a number; and, if appropriate, the source and destination addresses and source and destination port numbers. The message is generated for the first packet that matches, and then at 5-minute intervals, including the number of packets permitted or denied in the prior 5-minute interval.

Default

There is no specific condition under which a packet is prevented from being encrypted/decrypted. However, if a packet does not match any **deny** or **permit** command statements, the packet will not be encrypted/decrypted. (See the “Usage Guidelines” section that follows for more information about matching encryption access list conditions.)

Command Mode

Access-list configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

Use this command to specify conditions under which a packet will not be encrypted/decrypted. Use this command after you use the **ip access-list extended (encryption)** command.

After a named encryption access list is fully specified using **permit** and **deny** commands, the encryption access list must be specified in a crypto map, and the crypto map must be applied to an interface. After this is accomplished, packets will be either encrypted/decrypted or not encrypted/decrypted at the router depending on the conditions defined within the **permit** and **deny** commands.

If a packet matches the conditions in any **deny** command, the packet will not be encrypted/decrypted. Also, if a packet does not match any conditions in either a **deny** or a **permit** command, the packet will not be encrypted/decrypted. This occurs because all encryption access lists contain an implicit “deny” (“do not encrypt/decrypt”) statement at the end of the list.



Caution When creating encryption access lists, we do *not* recommend using the **any** keyword to specify source or destination addresses for **permit** or **deny** commands. Using the **any** keyword with a **permit** command could cause extreme problems if a packet enters your router and is destined for a router that is not configured for encryption. This would cause your router to attempt to set up an encryption session with a nonencrypting router.

If you incorrectly use the **any** keyword with a **deny** command, you might inadvertently prevent all packets from being encrypted, which could present a security risk.

Examples

1 Example of an Inappropriately Configured Access List

This first example shows a named encryption access list configured in an inappropriate way. After this list is applied to an interface using a crypto map, no UDP traffic will be encrypted. This occurs even though there are **permit** commands.

```
ip access-list extended Apricotcryptomap10
deny UDP any any
permit UDP 192.168.33.145 0.0.0.15 172.31.0.0 0.0.255.255
permit UDP 192.168.33.145 0.0.0.15 10.0.0.0 0.255.255.255
```

2 Another Example of an Inappropriately Configured Access List

The second example shows another inappropriate configuration for an encryption access list. This example will cause the router to encrypt all UDP traffic leaving the interface, including traffic to routers not configured for encryption. When this happens, the router will attempt to set up an encryption session with a non-encrypting router.

```
ip access-list extended Apricotcryptomap10
permit UDP 192.168.33.145 0.0.0.15 172.31.0.0 0.0.255.255
permit UDP 192.168.33.145 0.0.0.15 10.0.0.0 0.255.255.255
permit UDP any any
```

3 Example of a Correctly Configured Access List

The third example will encrypt/decrypt only traffic that matches the source and destination addresses defined in the two permit statements. All other traffic will not be encrypted/decrypted.

```
ip access-list extended Apricotcryptomap10
permit UDP 192.168.33.145 0.0.0.15 172.31.0.0 0.0.255.255
permit UDP 192.168.33.145 0.0.0.15 10.0.0.0 0.255.255.255
```

Related Commands

You can use the master indexes or search online to find documentation of related commands.

access-list (encryption)

ip access-list extended (encryption)

permit

show ip access-list

ip access-list extended (encryption)

To define an encryption access list by name, use the **ip access-list extended** global configuration command. To remove a named encryption access list, use the **no** form of this command.

ip access-list extended *name*
no ip access-list extended *name*

Syntax Description

name Name of the encryption access list. Names cannot contain a space or quotation mark, and must begin with an alphabetic character to prevent ambiguity with numbered access lists.

Default

There is no named encryption access list.

Command Mode

Global configuration

Performing this command invokes the access-list configuration command mode.

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

Use this command to configure a named IP access list (as opposed to a numbered IP access list). This command will take you into access-list configuration mode. From this mode you use the **deny** and **permit** commands to define the conditions for which traffic will be encrypted/decrypted or not encrypted/decrypted.

To use the encryption access list, you must first specify the access list in a crypto map definition, and then apply the crypto map to an interface.

Examples

1 Example of an Inappropriately Configured Access List

The first example shows a named encryption access list configured in an inappropriate way. After this list is applied to an interface using a crypto map, no UDP traffic will be encrypted. This occurs even though there are **permit** commands.

```
ip access-list extended Apricotcryptomap10
deny UDP any any
permit UDP 192.168.33.145 0.0.0.15 172.31.0.0 0.0.255.255
permit UDP 192.168.33.145 0.0.0.15 10.0.0.0 0.255.255.255
```

2 Another Example of an Inappropriately Configured Access List

The second example shows another inappropriate configuration for an encryption access list. This example will cause the router to encrypt all UDP traffic leaving the interface, including traffic to routers not configured for encryption. When this happens, the router will attempt to set up an encryption session with a non-encrypting router.

```
ip access-list extended Apricotcryptomap10
 permit UDP 192.168.33.145 0.0.0.15 172.31.0.0 0.0.255.255
 permit UDP 192.168.33.145 0.0.0.15 10.0.0.0 0.255.255.255
 permit UDP any any
```

3 Example of a Correctly Configured Access List

The third example will encrypt/decrypt only traffic that matches the source and destination addresses defined in the two permit statements. All other traffic will not be encrypted/decrypted.

```
ip access-list extended Apricotcryptomap10
 permit UDP 192.168.33.145 0.0.0.15 172.31.0.0 0.0.255.255
 permit UDP 192.168.33.145 0.0.0.15 10.0.0.0 0.255.255.255
```

Related Commands

You can use the master indexes or search online to find documentation of related commands.

access-list (encryption)

crypto map (global configuration)

crypto map (interface configuration)

deny

ip access-list (used for traffic filtering purposes)

permit

show ip access-list

match address

To specify an encryption access list within a crypto map definition, use the **match address** crypto map configuration command. Use the **no** form of this command to eliminate an encryption access list from a crypto map definition.

match address [*access-list-number* | *name*]
no match address [*access-list-number* | *name*]

Syntax Description

<i>access-list-number</i>	Identifies the numbered encryption access list. This value should match the <i>access-list-number</i> argument of the numbered encryption access list being matched.
<i>name</i>	Identifies the named encryption access list. This name should match the <i>name</i> argument of the named encryption access list being matched.

Default

No access lists are matched to the crypto map.

Command Mode

Crypto map configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

Use this command to specify an encryption access list for a given crypto map definition. This access list was previously defined using the **access-list (encryption)** or **ip access-list extended (encryption)** commands.

The encryption access list you specify with this command will be applied to an interface as an outbound encryption access list, after you define a crypto map and apply the crypto map to the interface.

Example

The following example creates a crypto map and defines an encryption access list for the map:

```
Apricot(config)# crypto map Research 10  
Apricot(config-crypto-map)# match address 101
```

Related Commands

You can use the master indexes or search online to find documentation of related commands.

access-list (encryption)
crypto map (global configuration)
ip access-list extended (encryption)
show crypto map
show crypto map interface
show crypto map tag

permit

To set conditions for a named encryption access list, use the **permit** access-list configuration command. The **permit** command causes IP traffic to be encrypted/decrypted if the conditions are matched. To remove a permit condition from an encryption access list, use the **no** form of this command.

```
permit source [source-wildcard]
no permit source [source-wildcard]

permit protocol source source-wildcard destination destination-wildcard [precedence
precedence] [tos tos] [log]
no permit protocol source source-wildcard destination destination-wildcard [precedence
precedence] [tos tos] [log]
```

For ICMP, you can also use the following syntax:

```
permit icmp source source-wildcard destination destination-wildcard [icmp-type [icmp-code] |
icmp-message] [precedence precedence] [tos tos] [log]
```

For IGMP, you can also use the following syntax:

```
permit igmp source source-wildcard destination destination-wildcard [igmp-type]
[precedence precedence] [tos tos] [log]
```

For TCP, you can also use the following syntax:

```
permit tcp source source-wildcard [operator port [port]] destination destination-wildcard
[operator port [port]] [established] [precedence precedence] [tos tos] [log]
```

For UDP, you can also use the following syntax:

```
permit udp source source-wildcard [operator port [port]] destination destination-wildcard
[operator port [port]] [precedence precedence] [tos tos] [log]
```

Syntax Description

<i>source</i>	<p>Number of the network or host from which the packet is being sent. There are two alternative ways to specify the source:</p> <ul style="list-style-type: none"> • Use a 32-bit quantity in four-part, dotted-decimal format. • Use the keyword any as an abbreviation for a <i>source</i> and <i>source-wildcard</i> of 0.0.0.0 255.255.255.255. This keyword is normally <i>not</i> recommended (see the section “Usage Guidelines”).
<i>source-wildcard</i>	<p>(Optional) Wildcard bits to be applied to the <i>source</i>. There are two alternative ways to specify the source wildcard:</p> <ul style="list-style-type: none"> • Use a 32-bit quantity in four-part, dotted-decimal format. Place ones in the bit positions you want to ignore. • Use the keyword any as an abbreviation for a <i>source</i> and <i>source-wildcard</i> of 0.0.0.0 255.255.255.255. This keyword is normally <i>not</i> recommended (see the section “Usage Guidelines”).

<i>protocol</i>	Name or number of an IP protocol. It can be one of the keywords eigrp , gre , icmp , igmp , igrp , ip , ipinip , nos , ospf , tcp , or udp , or an integer in the range 0 through 255 representing an IP protocol number. To match any Internet protocol (including ICMP, TCP, and UDP), use the keyword ip . Some protocols allow further qualifiers described later.
<i>source</i>	<p>Number of the network or host from which the packet is being sent. There are three alternative ways to specify the source:</p> <ul style="list-style-type: none"> • Use a 32-bit quantity in four-part, dotted-decimal format. • Use the keyword any as an abbreviation for a <i>source</i> and <i>source-wildcard</i> of 0.0.0.0 255.255.255.255. This keyword is normally <i>not</i> recommended (see the section “Usage Guidelines”). • Use host source as an abbreviation for a <i>source</i> and <i>source-wildcard</i> of <i>source</i> 0.0.0.0.
<i>source-wildcard</i>	<p>Wildcard bits to be applied to source. There are three alternative ways to specify the source wildcard:</p> <ul style="list-style-type: none"> • Use a 32-bit quantity in four-part, dotted-decimal format. Place ones in the bit positions you want to ignore. • Use the keyword any as an abbreviation for a <i>source</i> and <i>source-wildcard</i> of 0.0.0.0 255.255.255.255. This keyword is normally <i>not</i> recommended (see the section “Usage Guidelines”). • Use host source as an abbreviation for a <i>source</i> and <i>source-wildcard</i> of <i>source</i> 0.0.0.0.
<i>destination</i>	<p>Number of the network or host to which the packet is being sent. There are three alternative ways to specify the destination:</p> <ul style="list-style-type: none"> • Use a 32-bit quantity in four-part, dotted-decimal format. • Use the keyword any as an abbreviation for the <i>destination</i> and <i>destination-wildcard</i> of 0.0.0.0 255.255.255.255. This keyword is normally <i>not</i> recommended (see the section “Usage Guidelines”). • Use host destination as an abbreviation for a <i>destination</i> and <i>destination-wildcard</i> of <i>destination</i> 0.0.0.0.
<i>destination-wildcard</i>	<p>Wildcard bits to be applied to the destination. There are three alternative ways to specify the destination wildcard:</p> <ul style="list-style-type: none"> • Use a 32-bit quantity in four-part, dotted-decimal format. Place ones in the bit positions you want to ignore. • Use the keyword any as an abbreviation for a <i>destination</i> and <i>destination-wildcard</i> of 0.0.0.0 255.255.255.255. This keyword is normally <i>not</i> recommended (see the section “Usage Guidelines”). • Use host destination as an abbreviation for a <i>destination</i> and <i>destination-wildcard</i> of <i>destination</i> 0.0.0.0.
precedence <i>precedence</i>	(Optional) Packets can be matched for encryption by precedence level, as specified by a number from 0 to 7 or by name as listed in the section “Usage Guidelines” section of the access-list (encryption) command.

tos <i>tos</i>	(Optional) Packets can be matched for encryption by type of service level, as specified by a number from 0 to 15 or by name as listed in the “Usage Guidelines” section of the access-list (encryption) command.
<i>icmp-type</i>	(Optional) ICMP packets can be matched for encryption by ICMP message type. The type is a number from 0 to 255.
<i>icmp-code</i>	(Optional) ICMP packets which are matched for encryption by ICMP message type can also be matched by the ICMP message code. The code is a number from 0 to 255.
<i>icmp-message</i>	(Optional) ICMP packets can be matched for encryption by an ICMP message type name or ICMP message type and code name. The possible names are found in the “Usage Guidelines” section of the access-list (extended) command.
<i>igmp-type</i>	(Optional) IGMP packets can be matched for encryption by IGMP message type or message name. A message type is a number from 0 to 15. IGMP message names are listed in the “Usage Guidelines” section of the access-list (extended) command.
<i>operator</i>	<p>(Optional) Compares source or destination ports. Possible operands include lt (less than), gt (greater than), eq (equal), neq (not equal), and range (inclusive range).</p> <p>If the operator is positioned after the <i>source</i> and <i>source-wildcard</i>, it must match the source port.</p> <p>If the operator is positioned after the <i>destination</i> and <i>destination-wildcard</i>, it must match the destination port.</p> <p>The range operator requires two port numbers. All other operators require one port number.</p>
<i>port</i>	(Optional) The decimal number or name of a TCP or UDP port. A port number is a number from 0 to 65535. TCP and UDP port names are listed in the “Usage Guidelines” section of the access-list (extended) command. TCP port names can only be used when filtering TCP. UDP port names can only be used when filtering UDP.
established	(Optional) For the TCP protocol only: Indicates an established connection. A match occurs if the TCP datagram has the ACK or RST bits set. The nonmatching case is that of the initial TCP datagram to form a connection.

log

(Optional) Causes an informational logging message about the packet that matches the entry to be sent to the console. (The level of messages logged to the console is controlled by the **logging console** command.)

The message includes the access list number, whether the packet was permitted or denied; the protocol, whether it was TCP, UDP, ICMP or a number; and, if appropriate, the source and destination addresses and source and destination port numbers. The message is generated for the first packet that matches, and then at 5-minute intervals, including the number of packets permitted or denied in the prior 5-minute interval.

Default

There is no specific condition under which a packet is caused to be encrypted/decrypted. However, if a packet does not match any **deny** or **permit** command statements, the packet will not be encrypted/decrypted. (See the “Usage Guidelines” section for more information about matching encryption access list conditions.)

Command Mode

Access-list configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

Use this command following the **ip access-list extended (encryption)** command to specify conditions under which a packet will be encrypted/decrypted.

After a named encryption access list is fully specified using **permit** and **deny** commands, the encryption access list must be specified in a crypto map, and the crypto map must be applied to an interface. After this is accomplished, packets will be either encrypted/decrypted or not encrypted/decrypted at the router depending on the conditions defined within the **permit** and **deny** commands.

If a packet matches the conditions in any **permit** command, the packet will be encrypted/decrypted. If a packet does not match any conditions in either a **deny** or a **permit** command, the packet will not be encrypted/decrypted. This occurs because all encryption access lists contain an implicit “deny” (“do not encrypt/decrypt”) statement at the end of the list.



Caution When creating encryption access lists, we do *not* recommend using the **any** keyword to specify source or destination addresses for **permit** or **deny** commands. Using the **any** keyword with a **permit** command could cause extreme problems if a packet enters your router and is destined for a router that is not configured for encryption. This would cause your router to attempt to set up an encryption session with a nonencrypting router.

If you incorrectly use the **any** keyword with a **deny** command, you might inadvertently prevent all packets from being encrypted, which could present a security risk.

Examples

1 Example of an Inappropriately Configured Access List

The first example shows a named encryption access list configured in an inappropriate way. After this list is applied to an interface using a crypto map, no UDP traffic will be encrypted. This occurs even though there are **permit** commands.

```
ip access-list extended Apricotcryptomap10
deny UDP any any
permit UDP 192.168.33.145 0.0.0.15 172.31.0.0 0.0.255.255
permit UDP 192.168.33.145 0.0.0.15 10.0.0.0 0.255.255.255
```

2 Another Example of an Inappropriately Configured Access List

The second example shows another inappropriate configuration for an encryption access list. This example will cause the router to encrypt all UDP traffic leaving the interface, including traffic to routers not configured for encryption. When this happens, the router will attempt to set up an encryption session with a non-encrypting router.

```
ip access-list extended Apricotcryptomap10
permit UDP 192.168.33.145 0.0.0.15 172.31.0.0 0.0.255.255
permit UDP 192.168.33.145 0.0.0.15 10.0.0.0 0.255.255.255
permit UDP any any
```

3 Example of a Correctly Configured Access List

The third example will encrypt/decrypt only traffic that matches the source and destination addresses defined in the two permit statements. All other traffic will not be encrypted/decrypted.

```
ip access-list extended Apricotcryptomap10
permit UDP 192.168.33.145 0.0.0.15 172.31.0.0 0.0.255.255
permit UDP 192.168.33.145 0.0.0.15 10.0.0.0 0.255.255.255
```

Related Commands

You can use the master indexes or search online to find documentation of related commands.

access-list (encryption)

deny

ip access-list extended (encryption)

show ip access-list

set algorithm 40-bit-des

To specify a 40-bit Data Encryption Standard (DES) algorithm type within a crypto map definition, use the **set algorithm 40-bit-des** crypto map configuration command. Use the **no** form of this command to disable a 40-bit DES algorithm type within a crypto map definition.

set algorithm 40-bit-des [cfb-8 | cfb-64]
no set algorithm 40-bit-des [cfb-8 | cfb-64]

Syntax Description

- cfb-8** (Optional) Selects the 8-bit Cipher FeedBack (CFB) mode of the 40-bit DES algorithm. If no CFB mode is specified when the command is issued, 64-bit CFB mode is the default.
- cfb-64** Selects the 64-bit CFB mode of the 40-bit DES algorithm. If no CFB mode is specified when the command is issued, 64-bit CFB mode is the default.

Default

If no DES algorithm is specified within a crypto map, all globally enabled DES algorithms will be matched to the map by default. Refer to the **crypto algorithm 40-bit-des** or **crypto algorithm des** command descriptions to learn about globally enabling DES algorithms.

Command Mode

Crypto map configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

Use this command to specify 40-bit DES algorithm types for a given crypto map definition. Forty-bit DES algorithm types use a 40-bit DES key. The DES algorithms specified within a crypto map definition will be used to encrypt/decrypt all traffic at an interface when the crypto map is applied to the interface.

Note If you are running an exportable image, you can only use 40-bit variations of DES. You cannot enable or use the basic DES algorithms, which are not available with exportable images.

Example

The following example defines a 40-bit DES algorithm type for a crypto map:

```
Apricot(config)# crypto map Research 10
Apricot(config-crypto-map)# set algorithm 40-bit-des cfb-8
```

Related Commands

You can use the master indexes or search online to find documentation of related commands.

crypto map (global configuration)

set algorithm des

show crypto map

show crypto map interface

show crypto map tag

set algorithm des

To enable basic Data Encryption Standard (DES) algorithm types within a crypto map definition, use the **set algorithm des** crypto map configuration command. Use the **no** form of this command to disable a basic DES algorithm type within a crypto map definition.

set algorithm des [cfb-8 | cfb-64]
no set algorithm des [cfb-8 | cfb-64]

Syntax Description

- cfb-8** (Optional) Selects the 8-bit Cipher FeedBack (CFB) mode of the basic DES algorithm. If no CFB mode is specified when the command is issued, 64-bit CFB mode is the default.
- cfb-64** (Optional) Selects the 64-bit CFB mode of the basic DES algorithm. If no CFB mode is specified when the command is issued, 64-bit CFB mode is the default.

Default

If no DES algorithm is specified within a crypto map, all globally enabled DES algorithms will be matched to the map by default. Refer to the **crypto algorithm 40-bit-des** or **crypto algorithm des** command descriptions to learn about globally enabling DES algorithms.

Command Mode

Crypto map configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

Use this command to specify basic DES algorithm types for a given crypto map definition. Basic DES algorithm types use a 56-bit DES key. The DES algorithms specified within a crypto map definition will be used to encrypt/decrypt all traffic at an interface when the crypto map is applied to the interface.

Note If you are running an exportable image, you can only use 40-bit variations of DES. You cannot enable or use the basic DES algorithms, which are not available with exportable images.

Example

The following example defines a DES algorithm type for a crypto map:

```
Apricot(config)# crypto map Research 10  
Apricot(config-crypto-map)# set algorithm des cfb-8
```


Related Commands

You can use the master indexes or search online to find documentation of related commands.

crypto map (global configuration)

set algorithm 40-bit-des

show crypto map

show crypto map interface

show crypto map tag

set peer

To specify a peer encrypting router within a crypto map definition, use the **set peer** crypto map configuration command. Use the **no** form of this command to eliminate a peer encrypting router from a crypto map definition.

set peer *key-name*
no set peer *key-name*

Syntax Description

key-name Identifies the crypto engine of the peer encrypting router.

Default

No peer is defined by default.

Command Mode

Crypto map configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

Use this command to specify a peer encrypting router as the remote encryption route endpoint for a given crypto map definition.

Example

The following example creates a crypto map and defines a peer router for the map:

```
Apricot(config)# crypto map Research 10  
Apricot(config-crypto-map)# set peer BananaESA.HQ
```

Related Commands

You can use the master indexes or search online to find documentation of related commands.

crypto map (global configuration)
show crypto map
show crypto map interface
show crypto map tag

show crypto algorithms

To view which Data Encryption Standard (DES) algorithm types are globally enabled for your router, use the **show crypto algorithms** privileged EXEC command. This displays all basic DES and 40-bit DES algorithm types globally enabled.

show crypto algorithms

Syntax Description

This command has no arguments or keywords.

Command Mode

Privileged EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

Sample Display

The following is sample output from the **show crypto algorithms** command:

```
Apricot# show crypto algorithms
des cfb-8
```

Related Commands

You can use the master indexes or search online to find documentation of related commands.

crypto algorithm 40-bit-des

crypto algorithm des

show crypto card

To view the operational status of an Encryption Service Adapter (ESA), use the **show crypto card** privileged EXEC command. This command is available only on Cisco 7200, RSP7000, or 7500 series routers with an installed ESA.

```
show crypto card [slot]
```

Syntax Description

slot This argument is used only on Cisco RSP7000 and 7200 series routers. Identifies the ESA to show. Use the chassis slot number of the VIP2 containing the ESA.

Command Mode
Privileged EXEC

Usage Guidelines
This command first appeared in Cisco IOS Release 11.2.

Sample Display
The following is sample output from the **show crypto card** command:

```
Apricot# show crypto card 1
Crypto card in slot: 1

Tampered:      No
Xtracted:      No
Password set:   Yes
DSS Key set:    Yes
FW version:     5049702
```

Table 15 explains each field.

Table 15 Show Crypto Card Field Descriptions

Field	Description
Tampered	“Yes” indicates that somebody attempted to physically remove the tamper shield cover from the ESA card. Such an action causes the ESA card to clear its memory, similar to if a crypto zeroize command had been issued for the ESA.
Xtracted	“Yes” indicates that the ESA card had been extracted (removed) from the router.
Password set	“Yes” indicates that the ESA card password has already been set. This password is set with the crypto clear-latch or crypto gen-signature-keys command, and is required for subsequent issues of the crypto clear-latch and crypto gen-signature-keys commands.
DSS Key set	“Yes” indicates that DSS keys are generated and ready for use. DSS keys are generated using the crypto gen-signature-keys command.
FW version	Version number of the firmware running on the ESA card.

show crypto connections

To view current and pending encrypted session connections, use the **show crypto connections** privileged EXEC command.

show crypto connections

Syntax Description

This command has no arguments or keywords.

Command Mode

Privileged EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

Sample Display

The following is sample output from the **show crypto connections** command:

```
Apricot# show crypto connections
Pending Connection Table
PE          UPE          Timestamp          Conn_id
172.21.115.22  172.21.115.18  Mar 01 1993 00:01:09  -1

Connection Table
PE          UPE          Conn_id New_id  Alg      Time
172.21.115.22  172.21.115.18  -1      1      0      Not Set
flags:PEND_CONN
```

Table 16 explains each field.

Table 16 Show Crypto Connections Field Descriptions

Field	Description
PE	“Protected Entity.” This shows a representative source IP address as specified in the crypto map’s encryption access list. This IP address can be any host that matches a source in the encryption access list that is being used in the connection.
UPE	“Unprotected Entity.” This shows a representative destination IP address as specified in the crypto map’s encryption access list. This IP address can be any host that matches a destination in the encryption access list that is being used in the connection.
Timestamp	Identifies the time when the connection was initiated.
Conn_id	A number used to identify and track the connection. This can be a positive integer value from 1 to 299, or any negative integer value. Each connection is assigned a negative connection id when the connection is pending (being set up). Once the connection is established, a positive connection id is assigned to the connection.

Table 16 Show Crypto Connections Field Descriptions (Continued)

Field	Description
New_id	<p>Lists the connection id number that will be assigned to a connection, after the connection is set up. The New_id value will be a positive number from 0 to 299.</p> <p>If the New_id value is 0, there is no pending connection.</p> <p>If the New_id value is a positive integer, a connection is pending.</p> <p>As soon as the pending connection has been established, the New_id value will be transferred to the Conn_id for the established connection, and New_id will be reset to 0.</p>
Alg	<p>Identifies the DES encryption algorithm used for the current connection.</p> <p>10 = basic DES (56 bit) with 8-bit Cipher FeedBack (CFB)</p> <p>11 = basic DES (56 bit) with 64-bit CFB</p> <p>1 = 40-bit DES with 8-bit CFB</p> <p>2 = 40-bit DES with 64-bit CFB</p> <p>0 = no connection</p>
Time	Identifies the time when the connection was initiated.
flags	<p>PEND_CONN = identifies the table entry as a pending connection</p> <p>XCHG_KEYS = the connection has timed out; for encrypted communication to occur again, the router must first exchange DH numbers and generated a new session (DES) key</p> <p>TIME_KEYS = the encrypted communication session is currently in progress (a session key is currently installed, and the session is counting down to timeout)</p> <p>BAD_CONN = no existing or pending connection exists for this table entry</p> <p>UNK_STATUS = invalid status (error)</p>

show crypto engine brief

To view all crypto engines within a Cisco 7200, RSP7000, or 7500 series router, use the **show crypto engine brief** privileged EXEC command.

show crypto engine brief

Syntax Description

This command has no arguments or keywords.

Command Mode

Privileged EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

This command is only available on Cisco 7200, RSP7000, and 7500 series routers.

Sample Display

The following is sample output from the **show crypto engine brief** command. In this example, the router has two crypto engines: a Cisco IOS crypto engine and a Encryption Service Adapter (ESA) crypto engine. Both crypto engines have Digital Signature Standard (DSS) keys generated.

```
Apricot# show crypto engine brief
crypto engine name:  ApricotESA
crypto engine type:   ESA
crypto engine state:  dss key generated
crypto firmware version:  5049702
crypto engine in slot: 1

crypto engine name:  ApricotIOS
crypto engine type:   software
crypto engine state:  dss key generated
crypto lib version:   2.0.0
crypto engine in slot: 4
```

Table 17 explains each field.

Table 17 Show Crypto Engine Brief Field Descriptions

Field	Description
crypto engine name	Name of the crypto engine as assigned with the <i>key-name</i> argument in the crypto gen-signature-keys command.
crypto engine type	If “software” is listed, the crypto engine resides in either the Route Switch Processor (RSP) (the Cisco IOS crypto engine) or in a second-generation Versatile Interface Processor (VIP2). If “crypto card” is listed, the crypto engine is associated with an Encryption Service Adapter (ESA).

Table 17 Show Crypto Engine Brief Field Descriptions (Continued)

Field	Description
crypto engine state	<p>The state “installed” indicates that a crypto engine is located in the given slot, but is not configured for encryption.</p> <p>The state “dss key generated” indicates the crypto engine found in that slot has DSS keys already generated.</p> <p>In a Cisco 7200 series router, the state “installed (ESA pending)” indicates that the ESA crypto engine will be replaced with the Cisco IOS crypto engine as soon as it becomes available.</p>
crypto firmware version	Version number of the crypto firmware running on the ESA.
crypto lib version	Version number of the crypto library running on the router.
crypto engine in slot	Chassis slot number of the crypto engine. For the Cisco IOS crypto engine, this is the chassis slot number of the Route Switch Processor (RSP). For the VIP2 crypto engine, this is the chassis slot number of the VIP2. For the ESA crypto engine, this is the chassis slot number of the ESA (Cisco 7200) or of the VIP2 (Cisco RSP7000 and 7500).

Related Commands

You can use the master indexes or search online to find documentation of related commands.

show crypto engine configuration

show crypto engine configuration

To view the Cisco IOS crypto engine of your router, use the **show crypto engine configuration** privileged EXEC command.

show crypto engine configuration

Syntax Description

This command has no arguments or keywords.

Command Mode

Privileged EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

Sample Display

The following is sample output from the **show crypto engine configuration** command for a Cisco 2500 series router:

```
Apricot# show crypto engine configuration
engine name:      Apricot
engine type:      software
serial number:    01709642
platform:        rp crypto engine

Encryption Process Info:
input queue top:  75
input queue bot:  75
input queue count: 0
```

The following is sample output from the **show crypto engine configuration** command for a Cisco 7500 series router:

```
Banana# show crypto engine configuration
engine name:      BananaIOS
engine type:      software
serial number:    02863239
platform:        rsp crypto engine

Encryption Process Info:
input queue top:  44
input queue bot:  44
input queue count: 0
```

Table 18 explains each field.

Table 18 Show Crypto Engine Configuration Field Descriptions

Field	Description
engine name	Name of the crypto engine as assigned with the <i>key-name</i> argument in the crypto gen-signature-keys command.
engine type	Should always display “software.”

Table 18 Show Crypto Engine Configuration Field Descriptions (Continued)

Field	Description
serial number	Serial number of the Route Processor or Route Switch Processor.
platform	If the router is a Cisco RSP7000 or 7500 series router, this field will display "rsp crypto engine." If the router is a Cisco 7200 series router, this field will display "rp crypto engine."
input queue top (Encryption Process Info)	The queue location of the (inbound) packet next in line to be processed (decrypted). This packet will come off the top of the circular queue next. (This field is useful for debugging purposes.)
input queue bot (Encryption Process Info)	The queue location of the (inbound) packet last in line to be processed (decrypted). The packet is the most recently received and queued at the bottom of the circular queue. (This field is useful for debugging purposes.)
input queue count (Encryption Process Info)	The total number of packets currently in the circular queue. These are inbound packets waiting for processing. (This field is useful for debugging purposes.)

Related Commands

You can use the master indexes or search online to find documentation of related commands.

show crypto engine brief

show crypto engine connections active

To view the current active encrypted session connections for all crypto engines, use the **show crypto engine connections active** privileged EXEC command.

show crypto engine connections active [*slot*]

Syntax Description

slot (Optional) Identifies the crypto engine. This argument is available only on Cisco 7200, RSP7000, and 7500 series routers.

If no slot is specified, the Cisco IOS crypto engine will be selected.

Use the chassis slot number of the crypto engine location. For the Cisco IOS crypto engine, this is the chassis slot number of the Route Switch Processor (RSP). For the VIP2 crypto engine, this is the chassis slot number of the VIP2. For the ESA crypto engine, this is the chassis slot number of the ESA (Cisco 7200) or of the VIP2 (Cisco RSP7000 and 7500).

Command Mode

Privileged EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

Sample Display

The following is sample output from the **show crypto engine connections active** command:

```
Apricot# show crypto engine connections active
Connection Interface IP-Address State Algorithm Encrypt Decrypt
2 Ethernet0 172.21.114.9 set DES_56_CFB64 41 32
3 Ethernet1 172.29.13.2 set DES_56_CFB64 110 65
4 Serial0 172.17.42.1 set DES_56_CFB64 36 27
```

Table 19 explains each field.

Table 19 Show Crypto Engine Connections Active Field Descriptions

Field	Description
Connection	Identifies the connection by its number. Each active encrypted session connection is identified by a positive number from 1 to 299. These connection numbers correspond to the table entry numbers.
Interface	Identifies the interface involved in the encrypted session connection. This will display only the actual interface, not a subinterface (even if a subinterface is defined and used for the connection).
IP-Address	Identifies the IP address of the interface. Note that if a subinterface is used for the connection, this field will display “unassigned.”
State	The state “set” indicates an active connection.

Table 19 Show Crypto Engine Connections Active Field Descriptions (Continued)

Field	Description
Algorithm	Identifies the Data Encryption Standard (DES) algorithm used to encrypt/decrypt packets at the interface.
Encrypt	Shows the total number of encrypted outbound IP packets.
Decrypt	Shows the total number of decrypted inbound IP packets.

Related Commands

You can use the master indexes or search online to find documentation of related commands.

show crypto engine connections dropped-packets

show crypto engine connections dropped-packets

To view information about packets dropped during encrypted sessions for all router crypto engines, use the **show crypto engine connections dropped-packets** privileged EXEC command.

show crypto engine connections dropped-packets

Syntax Description

This command has no arguments or keywords.

Command Mode

Privileged EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

Sample Display

The following is sample output from the **show crypto engine connections dropped-packets** command:

```
Apricot# show crypto engine connections dropped-packets
Interface      IP-Address      Drop Count
Ethernet0/0    172.21.114.165  4
```

The Drop Count number indicates the total number of dropped packets for the lifetime of the crypto engine.

Related Commands

You can use the master indexes or search online to find documentation of related commands.

show crypto engine connections active

show crypto key-timeout

To view the current setting for the time duration of encrypted sessions, use the **show crypto key-timeout** privileged EXEC command.

show crypto key-timeout

Syntax Description

This command has no arguments or keywords.

Command Mode

Privileged EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

Sample Display

The following is sample output from the **show crypto key-timeout** command:

```
Apricot# show crypto key-timeout
Session keys will be re-negotiated every 120 minutes.
```

Related Commands

You can use the master indexes or search online to find documentation of related commands.

crypto key-timeout

show crypto map

To view all created crypto maps of your router, use the **show crypto map** privileged EXEC command.

show crypto map

Syntax Description

This command has no arguments or keywords.

Command Mode

Privileged EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

Sample Display

The following is sample output from the **show crypto map** command performed at a Cisco 2500 series router.

```
Apricot# show crypto map

Crypto Map "Canada" 10
  Connection Id = UNSET      (2 established,      0 failed)
  Crypto Engine = ApricotIOS (2)
  Algorithm = 40-bit-des cfb-64
  Peer = Banana
  PE = 172.21.114.9
  UPE = 192.168.23.116
  Extended IP access list 101
    access-list 101 permit ip host 10.0.0.1 host 192.168.15.0
    access-list 101 permit ip host 172.21.114.9 host 192.168.23.116
```

The following is sample output from the **show crypto map** command performed at a Cisco 7500 series router. Two crypto maps are shown: a crypto map named ResearchSite with subdefinitions 10 and 20, and another crypto map named HQ.

```
Banana# show crypto map

Crypto Map "ResearchSite" 10
  Connection Id = 6          (6 established,    0 failed)
  Crypto Engine = BananaIOS (4)
  Algorithm = 40-bit-des cfb-64
  Peer = Apricot
  PE = 192.168.15.0
  UPE = 10.0.0.1
  Extended IP access list 102
    access-list 102 permit ip host 192.168.15.0 host 10.0.0.1
Crypto Map "ResearchSite" 20
  Connection Id = UNSET      (0 established,    0 failed)
  Crypto Engine = BananaIOS (4)
  Algorithm = 56-bit-des cfb-64
  Peer = Cantaloupe
  PE = 192.168.129.33
  UPE = 172.21.114.165
  Extended IP access list 103
    access-list 103 permit ip host 192.168.129.33 host 172.21.114.165
Crypto Map "HQ" 10
  Connection Id = UNSET      (3 established,    0 failed)
  Crypto Engine = BananaESA (2)
  Algorithm = 56-bit-des cfb-64
  Peer = Eggplant
  PE = 192.168.129.10
  UPE = 10.1.2.3
  Extended IP access list 104
    access-list 104 permit ip host 192.168.129.10 host 10.1.2.3
```

The command output separately lists each crypto map subdefinition.

If more than one subdefinition exists for a crypto map, each subdefinition will be listed separately by sequence number (per the *seq-num* argument of the **crypto map (global configuration)** command). The sequence number is shown following the crypto map name.

Table 20 explains each field.

Table 20 Show Crypto Map Field Descriptions

Field	Description
Connection Id	Identifies the connection by its number. Each active encrypted session connection is identified by a positive number from 1 to 299. A value of UNSET indicates that no connection currently exists and is using the crypto map.
established	Indicates the total number of encrypted connections that have been successfully established using the crypto map.
failed	Indicates the total number of attempted encrypted connections that failed to be established while using the crypto map.

Table 20 Show Crypto Map Field Descriptions (Continued)

Field	Description
Crypto Engine	<p>Lists the name of the governing crypto engine, followed by the crypto engine slot number in parentheses.</p> <p>The slot number could be the Route Switch Processor (RSP) slot number, indicating a Cisco IOS crypto engine, or a second-generation Versatile Interface Processor (VIP2) slot number, indicating a VIP2 or an ESA crypto engine, or (Cisco 7200 only) an ESA slot number, indicating an ESA crypto engine.</p> <p>(Not displayed on routers other than Cisco 7200, RSP7000, or 7500 series routers.)</p>
Algorithm	Indicates the type of DES encryption algorithm used by the crypto map.
Peer	Indicates the name of the crypto map of the remote peer encrypting router.
PE	“Protected Entity.” This shows a representative source IP address as specified in the crypto map’s encryption access list. This IP address can be any host that matches a source in the encryption access list that is being used in the connection.
UPE	“Unprotected Entity.” This shows a representative destination IP address as specified in the crypto map’s encryption access list. This IP address can be any host that matches a destination in the encryption access list that is being used in the connection.
Extended IP access list	Lists the access list associated with the crypto map. If no access list is associated, the message “No matching address list set” is displayed.

Related Commands

You can use the master indexes or search online to find documentation of related commands.

crypto map (global configuration)
crypto map (interface configuration)
show crypto map interface
show crypto map tag

show crypto map interface

To view the crypto map applied to a specific interface, use the **show crypto map interface** privileged EXEC command.

```
show crypto map interface interface
```

Syntax Description

interface Designates the router interface.

Command Mode

Privileged EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

Sample Display

The following is sample output from the **show crypto map interface** command:

```
Apricot# show crypto map interface ethernet0

Crypto Map "ResearchSite" 10
  Connection Id = 6           (6 established,      0 failed)
  Crypto Engine = BananaIOS (4)
  Algorithm = 40-bit-des cfb-64
  Peer = Apricot
  PE = 192.168.15.0
  UPE = 10.0.0.1
  Extended IP access list 102
    access-list 102 permit ip host 192.168.15.0 host 10.0.0.1
Crypto Map "ResearchSite" 20
  Connection Id = UNSET       (0 established,      0 failed)
  Crypto Engine = BananaIOS (4)
  Algorithm = 56-bit-des cfb-64
  Peer = Cantaloupe
  PE = 192.168.129.33
  UPE = 172.21.114.165
  Extended IP access list 103
    access-list 103 permit ip host 192.168.129.33 host 172.21.114.165
```

Table 21 explains each field.

Table 21 Show Crypto Map Interface Field Descriptions

Field	Description
Connection Id	Identifies the connection by its number. Each active encrypted session connection is identified by a positive number from 1 to 299. A value of UNSET indicates that no connection currently exists and is using the crypto map.
established	Indicates the total number of encrypted connections that have been successfully established using the crypto map.

Table 21 Show Crypto Map Interface Field Descriptions (Continued)

Field	Description
failed	Indicates the total number of attempted encrypted connections that failed to be established while using the crypto map.
Crypto Engine	<p>Lists the name of the governing crypto engine, followed by the crypto engine slot number in parentheses.</p> <p>The slot number could be the Route Switch Processor (RSP) slot number, indicating a Cisco IOS crypto engine, or a second-generation Versatile Interface Processor (VIP2) slot number, indicating a VIP2 or an ESA crypto engine, or (Cisco 7200 only) an ESA slot number, indicating an ESA crypto engine.</p> <p>(Not displayed on routers other than Cisco 7200, RSP7000, or 7500 series routers.)</p>
Algorithm	Indicates the type of DES encryption algorithm used by the crypto map.
Peer	Indicates the name of the crypto map of the remote peer encrypting router.
PE	“Protected Entity.” This shows a representative source IP address as specified in the crypto map’s encryption access list. This IP address can be any host that matches a source in the encryption access list that is being used in the connection.
UPE	“Unprotected Entity.” This shows a representative destination IP address as specified in the crypto map’s encryption access list. This IP address can be any host that matches a destination in the encryption access list that is being used in the connection.
Extended IP access list	Lists the access list associated with the crypto map. If no access list is associated, the message “No matching address list set” is displayed.

Related Commands

You can use the master indexes or search online to find documentation of related commands.

crypto map (global configuration)

crypto map (interface configuration)

show crypto map

show crypto map tag

show crypto map tag

To view a specific crypto map, use the **show crypto map tag** privileged EXEC command.

```
show crypto map tag map-name
```

Syntax Description

map-name Identifies the crypto map by its name. This should match the *map-name* argument assigned during crypto map creation.

Command Mode

Privileged EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

Sample Display

The following is sample output from the **show crypto map tag** command:

```
Apricot# show crypto map tag HQ

Crypto Map "HQ" 10
  Connection Id = UNSET      (3 established,      0 failed)
  Crypto Engine = BananaESA (2)
  Algorithm = 56-bit-des cfb-64
  Peer = Eggplant
  PE = 192.168.129.10
  UPE = 10.1.2.3
  Extended IP access list 104
      access-list 104 permit ip host 192.168.129.10 host 10.1.2.3
```

Table 22 explains each field.

Table 22 Show Crypto Map Tag Field Descriptions

Field	Description
Connection Id	Identifies the connection by its number. Each active encrypted session connection is identified by a positive number from 1 to 299. A value of UNSET indicates that no connection currently exists and is using the crypto map.
established	Indicates the total number of encrypted connections that have been successfully established using the crypto map.
failed	Indicates the total number of attempted encrypted connections that failed to be established while using the crypto map.

Table 22 Show Crypto Map Tag Field Descriptions (Continued)

Field	Description
Crypto Engine	<p>Lists the name of the governing crypto engine, followed by the crypto engine slot number in parentheses.</p> <p>The slot number could be the Route Switch Processor (RSP) slot number, indicating a Cisco IOS crypto engine, or a second-generation Versatile Interface Processor (VIP2) slot number, indicating a VIP2 or an ESA crypto engine, or (Cisco 7200 only) an ESA slot number, indicating an ESA crypto engine.</p> <p>(Not displayed on routers other than Cisco 7200, RSP7000, or 7500 series routers.)</p>
Algorithm	Indicates the type of DES encryption algorithm used by the crypto map.
Peer	Indicates the name of the crypto map of the remote peer encrypting router.
PE	“Protected Entity.” This shows a representative source IP address as specified in the crypto map’s encryption access list. This IP address can be any host that matches a source in the encryption access list that is being used in the connection.
UPE	“Unprotected Entity.” This shows a representative destination IP address as specified in the crypto map’s encryption access list. This IP address can be any host that matches a destination in the encryption access list that is being used in the connection.
Extended IP access list	Lists the access list associated with the crypto map. If no access list is associated, the message “No matching address list set” is displayed.

Related Commands

You can use the master indexes or search online to find documentation of related commands.

crypto map (global configuration)
crypto map (interface configuration)
show crypto map
show crypto map interface

show crypto mypubkey

To view Digital Signature Standard (DSS) public keys (for all your router crypto engines) in hexadecimal form, use the **show crypto mypubkey** EXEC command.

show crypto mypubkey [rsp]

Syntax Description

rsp (Optional) This argument is available only on Cisco 7200, RSP7000, and 7500 series routers.

If this argument is used, only the DSS public keys for the Route Switch Processor (RSP) (Cisco IOS crypto engine) will be displayed.

If this argument is not used, DSS public keys for all crypto engines will be displayed.

Command Mode

EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

Sample Display

The following is sample output from the **show crypto mypubkey** command for a Cisco 2500 series router with a crypto engine called “Apricot.branch”:

```
Apricot# show crypto mypubkey

crypto public-key Apricot.branch 01709642
BDD99A6E EEE53D30 BC0BFAE6 948C40FB 713510CB 32104137 91B06C8D C2D5B422
D9C154CA 00CDE99B 425DB9FD FE3162F1 1E5866AF CF66DD33 677259FF E5C24812
quit
```

Related Commands

You can use the master indexes or search online to find documentation of related commands.

show crypto pubkey

show crypto pubkey name

show crypto pubkey serial

show crypto pregen-dh-pairs

To view the number of Diffie-Hellman (DH) number pairs currently generated, use the **show crypto pregen-dh-pairs** privileged EXEC command.

show crypto pregen-dh-pairs [*slot*]

Syntax Description

slot (Optional) Identifies the crypto engine. This argument is available only on Cisco 7200, RSP7000, and 7500 series routers.

If no slot is specified, the Cisco IOS crypto engine will be selected.

Use the chassis slot number of the crypto engine location. For the Cisco IOS crypto engine, this is the chassis slot number of the Route Switch Processor (RSP). For the VIP2 crypto engine, this is the chassis slot number of the VIP2. For the ESA crypto engine, this is the chassis slot number of the ESA (Cisco 7200) or of the VIP2 (Cisco RSP7000 and 7500).

Command Mode

Privileged EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

Sample Display

The following is sample output from the **show crypto pregen-dh-pairs** command:

```
Apricot# show crypto pregen-dh-pairs

Number of pregenerated DH pairs: 1
```

The number one shown in the output indicates that there is one DH number pair ready and available for the next encrypted connection.

Related Commands

You can use the master indexes or search online to find documentation of related commands.

crypto pregen-dh-pairs

show crypto pubkey

To view all peer router Digital Signature Standard (DSS) public keys known to your router, use the **show crypto pubkey EXEC** command.

show crypto pubkey

Syntax Description

This command has no arguments or keywords.

Command Mode

EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

Sample Display

The following is sample output from the **show crypto pubkey** command. In this example, router Apricot has exchanged DSS public keys with two other routers, named Banana and Cantaloupe. Banana has a crypto engine named BananaESA, and Cantaloupe has a crypto engine named CantaloupeIOS:

```
Apricot# show crypto pubkey

crypto public-key BananaESA 01580120
8A1D0765 DE172C96 3DE9575D D1EEC69B A40107CA D0E2C55D CC17D6E9 3D45D042
DD4959AA BFC556EC AF5B2931 90FC883B 48F7A61A 9C9E5C25 06775ECB E1EDC966
quit
crypto public-key CantaloupeIOS 00198A0F
BBCD227D B8630AAA F888B6DC A9D64781 108FF9A1 157645C9 52F96EC0 6FB9E5F3
51BA8A76 2DFA532B 48856D46 B91B74C3 C2FEB617 85916A3F 27C8A9C2 311CF872
quit
```

Related Commands

You can use the master indexes or search online to find documentation of related commands.

crypto gen-signature-keys
crypto key-exchange
show crypto pubkey
show crypto pubkey name
show crypto pubkey serial

show crypto pubkey name

To view a specific peer router Digital Signature Standard (DSS) public key known by its name, use the **show crypto pubkey name EXEC** command.

show crypto pubkey name *key-name*

Syntax Description

key-name Identifies the crypto engine of the peer router.

Command Mode

EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

Sample Display

The following is sample output from the **show crypto pubkey name** command. In this example, router Apricot has exchanged DSS public keys with a router named Banana. Banana has a crypto engine named BananaESA:

```
Apricot# show crypto pubkey name BananaESA

crypto public-key BananaESA 01580120
8A1D0765 DE172C96 3DE9575D D1EEC69B A40107CA D0E2C55D CC17D6E9 3D45D042
DD4959AA BFC556EC AF5B2931 90FC883B 48F7A61A 9C9E5C25 06775ECB E1EDC966
quit
```

Related Commands

You can use the master indexes or search online to find documentation of related commands.

show crypto mypubkey

show crypto pubkey

show crypto pubkey serial

show crypto pubkey serial

To view a specific peer router Digital Signature Standard (DSS) public key known by its serial number, use the **show crypto pubkey serial** EXEC command.

show crypto pubkey serial *serial-number*

Syntax Description

serial-number Identifies the serial number of the crypto engine.

Command Mode

EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

Sample Display

The following is sample output from the **show crypto pubkey serial** command. In this example, router Apricot has exchanged DSS public keys with a router named Cantaloupe. Cantaloupe has a crypto engine with the serial number 00198A0F:

```
Apricot# show crypto pubkey serial 00198A0F

crypto public-key cantaloupeIOS 00198A0F
BBCD227D B8630AAA F888B6DC A9D64781 108FF9A1 157645C9 52F96EC0 6FB9E5F3
51BA8A76 2DFA532B 48856D46 B91B74C3 C2FEB617 85916A3F 27C8A9C2 311CF872
quit
```

Related Commands

You can use the master indexes or search online to find documentation of related commands.

show crypto mypubkey

show crypto pubkey

show crypto pubkey name

test crypto initiate-session

To set up a test encryption session, use the **test crypto initiate-session** privileged EXEC command.

test crypto initiate-session *src-ip-addr dst-ip-addr map-name seq-num*

Syntax Description

<i>src-ip-addr</i>	IP address of source host. Should be included in an encryption access list definition as a valid IP address source address.
<i>dst-ip-addr</i>	IP address of destination host. Should be included in an encryption access list definition as a valid IP address destination address.
<i>map-name</i>	Names the crypto map to be used.
<i>seq-num</i>	Names the crypto map sequence number.

Command Mode

Privileged EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

Use this command to set up a test encryption session. This command can be used after you have completed all the essential encryption configuration tasks for your router. After issuing this command, use the **show crypto connections** command to verify the status of the connection just created.

Example

The following example sets up and verifies a test encryption session.

Router Apricot sets up a test encryption session with router Banana and then views the connection status to verify a successful encrypted session connection.

Step 1 Router Apricot sets up a test encryption connection with router Banana.

```
Apricot# test crypto initiate-session 192.168.3.12 192.168.204.110 BananaESA.TXbranch 10
Sending CIM to: 192.168.204.110 from: 192.168.3.12.
Connection id: -1
```

Notice the Connection id value is -1. A negative value indicates that the connection is being set up. (CIM stands for Connection Initiation Message.)

Step 2 Router Apricot issues the **show crypto connections** command.

```
Apricot# show crypto connections
Pending Connection Table
PE           UPE           Timestamp           Conn_id
192.168.3.10 192.168.204.100 Mar 01 1993 00:01:09 -1

Connection Table
PE           UPE           Conn_id New_id  Alg      Time
192.168.3.10 192.168.204.100 -1      1      0        Not Set
flags:PEND_CONN
```

Look in the Pending Connection Table for an entry with a Conn_id value equal to the previously shown Connection id value—in this case, look for an entry with a Conn_id value of -1. If this is the first time an encrypted connection has been attempted, there will only be one entry (as shown).

Note the PE and UPE addresses for this entry.

Step 3 Now, look in the Connection Table for an entry with the same PE and UPE addresses. In this case, there is only one entry in both tables, so finding the right Connection Table entry is easy.

Step 4 At the Connection Table entry, note the Conn_id and New_id values. In this case, Conn_id equals -1, and New_id equals 1. The New_id value of 1 will be assigned to the test connection when setup is complete. (Positive numbers are assigned to established, active connections.)

Step 5 Apricot waits a moment for the test connection to set up, and then reissues the **show crypto connections** command.

```
Apricot# show crypto connections
Connection Table
PE           UPE           Conn_id New_id  Alg      Time
192.168.3.10 192.168.204.100 1        0      10       Mar 01 1993 00:02:00
flags:TIME_KEYS
```

Again, look for the Connection Table entry with the same PE and UPE addresses as shown before. In this entry, notice that the Conn_id value has changed to 1. This indicates that our test connection has been successfully established, because the Conn_id value has changed to match the New_id value of Step 4. Also, New_id has been reset to 0 at this point., indicating that there are no new connections currently being set up.

In the command output of Step 5, there is no longer a Pending Connection Table being displayed, which indicates that there are currently no pending connections. This is also a good clue that the test connection was successfully established.

The **show crypto connections** command is explained in greater detail previously in this chapter, including a description of how connection ids are assigned during and following connection setup.

Related Commands

You can use the master indexes or search online to find documentation of related commands.

show crypto connections