Crypto Access Check on Clear-Text Packets

The Crypto Access Check on Clear-Text Packets feature removes the checking of clear-text packets that go through the IP Security (IPSec) tunnel just prior to encryption or just after decryption. The clear-text packets were checked against the outside physical interface access control lists (ACLs). This checking was often referred to as a double ACL check. This feature enables easier configuration of ACLs and eliminates the security risks that are associated with a double check when using dynamic crypto maps.

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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

Prerequisites for Crypto Access Check on Clear-Text Packets

- You should be familiar with configuring IPSec.
- You should be familiar with ACLs.

Restrictions for Crypto Access Check on Clear-Text Packets

- This feature does not apply to IPSec configurations on the Virtual Private Network (VPN) service module (card) on Cisco Catalyst 6500 series switches and Cisco 7600 series router platforms.
- This feature supports only extended ACLs.

Information About Crypto Access Check on Clear-Text Packets

Crypto Access Check on Clear-Text Packets Overview

The Crypto Access Check on Clear-Text Packets feature provides four changes for the interaction between IPSec and interface access lists. The changes are as follows:

- Removes the checking of inbound, just-decrypted clear-text packets against the outside interface inbound ACL.
- Removes the checking of outbound clear-text packets just prior to encryption against the outside interface outbound ACL.
- Adds the checking of outbound encrypted packets against the outside interface outbound ACL.
- Adds the capability to configure ACLs under the crypto map to check inbound clear-text packets after decryption or outbound clear-text packets prior to encryption.

This feature enables the easier and more consistent configuration of ACLs that control packet movement in and out of the outside interface as well as in and out of the IPSec encryption tunnel. This feature also eliminates security risks that are associated with the current double check when using dynamic crypto maps.

Configuration Changes That Are Required for This Feature

This feature requires the following configuration changes to be performed. Some are required and some are optional.

Prior to Upgrading

Prior to upgrading to this feature, you should do the following. This change is required.

Check all outside interfaces for outbound ACLs. If any outbound ACLs exist, check to ensure that they include access-list entries (ACEs) that permit outbound Encapsulating Security Payload (ESP) IP protocol 50 packets

or Authentication Header (AH) IP protocol 51 packets. The ACL entries will be needed after the upgrade because the outbound encrypted packets will be checked against the outside interface outbound ACL. If the ESP or AH packets are not allowed by the outside interface outbound ACL, the IPSec VPN tunnels will not forward traffic.

After Upgrading

After upgrading to this feature, you should do the following. The first two procedures are required if you are using dynamic crypto maps. However, these procedures are recommended even if you are not using dynamic crypto maps. The third and fourth procedures are optional.

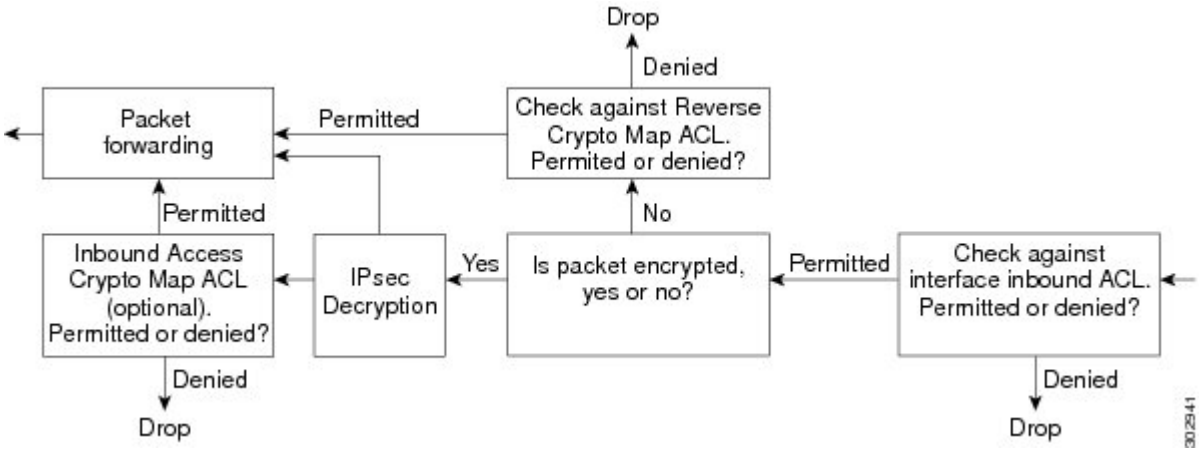
- Check all outside interfaces for inbound ACLs that contain ACEs that permit inbound, just-decrypted clear-text packets. These ACEs need to be removed if dynamic crypto maps are being used because when the IPSec tunnel is not "up," the ACEs will allow the clear-text packets into the network. If dynamic crypto maps are not being used, the ACEs can still be removed to simplify the outside interface ACLs.
- Check all outside interfaces for outbound ACLs that contain ACEs that permit outbound clear-text packets that would be encrypted. These ACEs need to be removed if dynamic crypto maps are being used because when the IPSec tunnel is not up, these ACEs will allow the clear-text packets out of the network. If dynamic crypto maps are not being used, these ACEs can still be removed to simplify the outside interface ACLs.
- Add an outbound crypto map access ACL under the crypto map to deny to-be-encrypted, outbound clear-text packets that should be dropped. Be sure that you also permit all other packets in this ACL.
- Add an inbound crypto map access ACL under the crypto map to deny just-decrypted, inbound clear-text packets that should be dropped. Be sure to also permit all other packets in this ACL.

The last two configuration changes are needed only in the rare cases in which the crypto map ACL (that selects packets to be encrypted) is more general than the packet flows that you want to encrypt. Adding outbound or inbound crypto map ACLs is usually done to keep the crypto map ACL small and simple, which saves CPU utilization and memory. The **set ip access-group** command, which is used to cause the checking of clear-text packets after decryption and before encryption, can be used under the crypto map to accomplish this task independent of the outside interface ACLs.

ACL Checking Behavior After Upgrading to This Feature

The diagam below illustrates the ACL checking behavior on the inbound path using the Crypto Access Check on Clear-Text Packets feature.

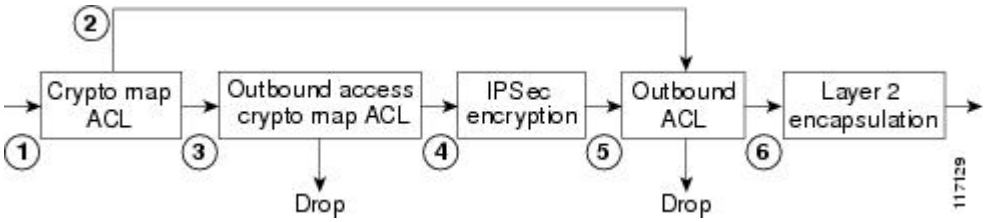
Figure 1: New Inbound Encrypted Packet Flow



1. Arriving IP packet is checked against the interface inbound ACL. If it is denied, it is dropped.
2. If IP packet is a permitted not-encrypted packet, it is forwarded and checked against the reverse crypto map ACL. If the result of the reverse crypto map ACL is permit, then the packet is dropped.
3. If IP packet is encrypted, it is then decrypted.
4. Just-decrypted IP packet is forwarded.
5. Just-decrypted IP packet is checked against the inbound access crypto map ACL (optional). If the packet is denied, it is dropped.

The diagram below illustrates the ACL checking behavior on the outbound path using the Crypto Access Check on Clear-Text Packets feature.

Figure 2: New Outbound Encrypted Packet Flow



1. All departing IP packets are checked against the crypto map ACL. If the packets are permitted, they are marked for encryption.
2. IP packets not marked for encryption are checked against the outbound interface ACL. If the packets are denied, they are dropped.

3. IP packets marked for encryption are checked against the outbound access crypto map ACL (optional). If the packets are denied, they are dropped.
4. Permitted IP packets are encrypted.
5. Encrypted IP packets are checked against the outbound interface ACL. If the packets are denied, they are dropped.
6. Permitted IP packets are Layer 2 encapsulated.

Backward Compatibility

If the Cisco IOS software is subsequently downgraded to a release that does not have the Crypto Access Check on Clear-Text Packets feature, the just-decrypted and to-be-encrypted clear-text packets will again be blocked by the outside interface ACLs. Therefore, if you have removed lines from the interface ACLs, you should undo the changes that were made to the ACLs if you are downgrading to an earlier version.

How to Configure Crypto Map Access ACLs

Adding or Removing ACLs

To add or remove crypto map access ACLs, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name* seq-*number*
4. **set ip access-group** {*access-list-number* |*access-list-name*} {**in** | **out**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto map** *map-name* seq-*number* **Example:** `Router(config)# crypto map vpn1 10`	Selects the crypto map and the sequence map entry under the crypto map to which you want to add the crypto map access ACL; also enters crypto map configuration mode.
Step 4	**set ip access-group** {*access-list-number* |*access-list-name*} {**in** | **out**} **Example:** `Router(config-crypto-map)# set ip access-group 151 in`	Allows you to check the postdecrypted or preencrypted packet against an ACL without having to use the outside physical interface ACL.

Verifying the Configured ACLs

The **show ip access-list** command can be used to verify the crypto input or output access-check ACLs that have been configured. Also, the packets that have been dropped in the context of the crypto input access-check ACL in the inbound path will be logged as receive (recv) errors, and packets dropped on the outbound path will be logged as send errors.

The **show crypto map** command can be used to verify crypto map configuration information.

SUMMARY STEPS

1. **enable**
2. show ip access-list [*access-list-number* | *access-list-name* | **dynamic**]
3. **show crypto map** [**interface** *interface* | **tag** *map-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	show ip access-list [*access-list-number* | *access-list-name* | **dynamic**] **Example:** `Router# show ip access-list Internetfilter`	Displays a configured ACL.

	Command or Action	Purpose
Step 3	**show crypto map** [**interface** *interface* | **tag** *map-name*] **Example:** `Router# show crypto map`	Displays the crypto maps that have been configured.

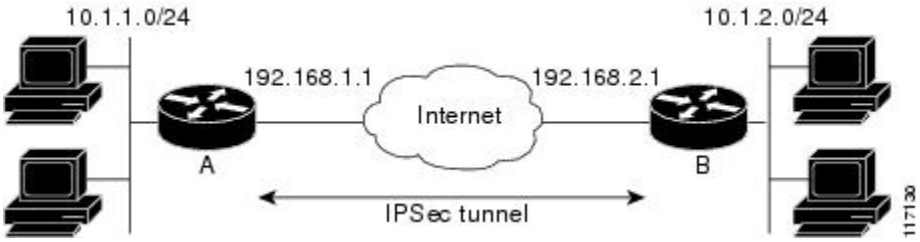
Configuration Examples for Crypto Access Checkon Clear-Text Packets

This section contains the output for the following stages of crypto access configuration:

- Previous IPSec ACL Configuration Example, on page 7
- New IPSec ACL Configuration Without Crypto Access ACLs Example, on page 8
- New IPSec ACL Configuration with Crypto Access ACLs Example, on page 8
- Authentication Proxy IPSec and CBAC Configuration Example, on page 9

The network diagram used for the following examples is shown below.

Figure 3: Network Diagram for Crypto Access Check Configuration Examples



The configuration examples assume these policy rules:

- Allow only encrypted host traffic between hosts on 10.1.1.0/24 and 10.1.2.0/24.
- No clear-text traffic from the Internet to any host.

Previous IPSec ACL Configuration Example

The following is a sample configuration using an earlier version of Cisco IOS software (before Release 12.3(8)T). The configuration shows outside interface ACLs with a double check on the inbound packets.

```
crypto map vpnmap 10 ipsec-isakmp
 set peer 192.168.2.1
 set transform-set trans1
```

```
 match address 101
interface Ethernet0/0
 ip address 10.1.1.1 255.255.255.0
interface Serial1/0
 ip address 192.168.1.1 255.255.255.0
 ip access-group 150 in
 ip access-group 160 out
 crypto map vpnmap
access-list 101 permit ip 10.1.1.0 0.0.0.255 10.1.2.0 0.0.0.255
access-list 150 permit udp host 192.168.2.1 eq 500 host 192.168.1.1 eq 500
access-list 150 permit esp host 192.168.2.1 host 192.168.1.1
access-list 150 permit ip 10.1.2.0 0.0.0.255 10.1.1.0 0.0.0.255
access-list 160 permit udp host 192.168.1.1 eq 500 host 192.168.2.1 eq 500
access-list 160 permit ip 10.1.1.0 0.0.0.255 10.1.2.0 0.0.0.255
```

New IPSec ACL Configuration Without Crypto Access ACLs Example

The following is a sample configuration using the current version of Cisco IOS software (Release 12.3(8)T). Before the crypto map access ACL is added, clear-text packets through the IPSec tunnel are not checked against an ACL (other packets are checked against the outside interface ACLs). Note the permitting of ESP packets in the outside interface outbound ACL.

```
crypto map vpnmap 10 ipsec-isakmp
 set peer 192.168.2.1
 set transform-set trans1
 match address 101
interface Ethernet0/0
 ip address 10.1.1.1 255.255.255.0
interface Serial1/0
 ip address 192.168.1.1 255.255.255.0
 ip access-group 150 in
 ip access-group 160 out
 crypto map vpnmap
access-list 101 permit ip 10.1.1.0 0.0.0.255 10.1.2.0 0.0.0.255
access-list 150 permit udp host 192.168.2.1 eq 500 host 192.168.1.1 eq 500
access-list 150 permit esp host 192.168.2.1 host 192.168.1.1
access-list 160 permit udp host 192.168.1.1 eq 500 host 192.168.2.1 eq 500
access-list 160 permit esp host 192.168.1.1 host 192.168.2.1
```

New IPSec ACL Configuration with Crypto Access ACLs Example

The following is a sample configuration using the current version of Cisco IOS software (Release 12.3(8)T). Before a crypto map access ACL is added, clear-text packets through the IPSec tunnel are checked against the crypto map access ACLs (other packets are checked against the outside interface ACLs).

Note In the following example, all IP packets between the subnets 10.1.1.0/24 and 10.1.2.0/24 are to be encrypted, but the crypto map access ACLs allow only Telnet traffic through the IPSec tunnel.

```
crypto map vpnmap 10 ipsec-isakmp
 set peer 192.168.2.1
 set transform-set trans1
 set ip access-group 151 in
 set ip access-group 161 out
 match address 101
interface Ethernet0/0
 ip address 10.1.1.1 255.255.255.0
interface Serial1/0
 ip address 192.168.1.1 255.255.255.0
 ip access-group 150 in
```

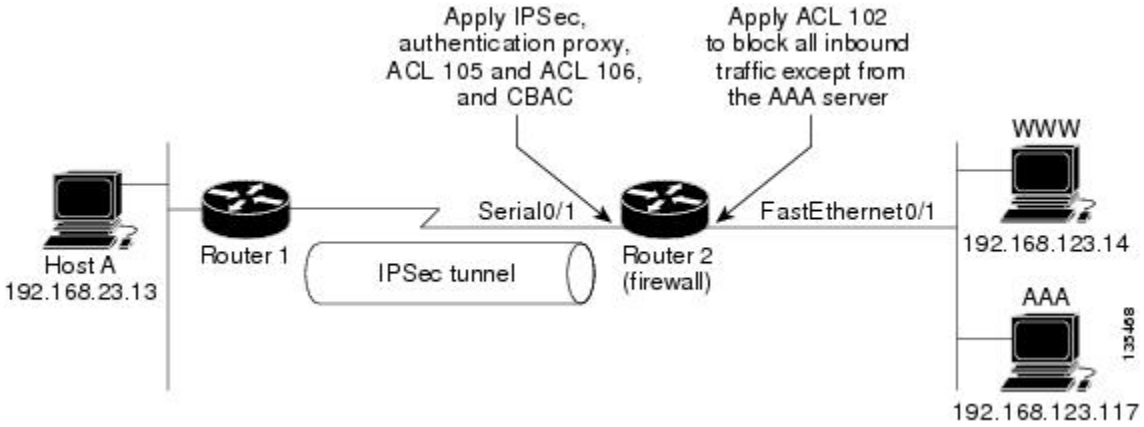
```
 ip access-group 160 out
 crypto map vpnmap
access-list 101 permit ip 10.1.1.0 0.0.0.255 10.1.2.0 0.0.0.255
access-list 150 permit udp host 192.168.2.1 eq 500 host 192.168.1.1 eq 500
access-list 150 permit esp host 192.168.2.1 host 192.168.1.1
access-list 151 permit tcp 10.1.2.0 0.0.0.255 eq telnet 10.1.1.0 0.0.0.255
access-list 151 permit tcp 10.1.2.0 0.0.0.255 10.1.1.0 0.0.0.255 eq telnet
access-list 160 permit udp host 192.168.1.1 eq 500 host 192.168.2.1 eq 500
access-list 160 permit esp host 192.168.1.1 host 192.168.2.1
access-list 161 permit ip 10.1.1.0 0.0.0.255 10.1.2.0 0.0.0.255 eq telnet
access-list 161 permit ip 10.1.1.0 0.0.0.255 eq telnet 10.1.2.0 0.0.0.255
```

Authentication Proxy IPSec and CBAC Configuration Example

The following example shows a router configuration using the authentication proxy, IPSec, and CBAC features. The figure below illustrates the configuration.

Note This configuration is effective for Cisco IOS Release 12.3(8)T software and later.

Figure 4: Router Configuration Using Authentication Proxy, IPSec, and CBAC Features



In this example, Host A initiates a HTTP connection with the web server (WWW). The HTTP traffic between Router 1 and Router 2 is encrypted using IPSec. The authentication proxy, IPSec, and CBAC are configured at interface Serial0/1 on Router 2, which is acting as the firewall. ACL 105 allows only IPSec traffic at interface Serial0/1. ACL 106 is crypto access check, which blocks all traffic. ACL 102 is applied at interface FastEthernet0/1 on Router 2 to block all traffic on that interface except traffic from the AAA server.

When Host A initiates a HTTP connection with the web server, the authentication proxy prompts the user at Host A for a username and password. These credentials are verified with the AAA server for authentication and authorization. If authentication is successful, the per-user ACLs are downloaded to the firewall to permit services.

The following examples provide both the Router 1 and Router 2 configurations for completeness:

- Authentication Proxy IPSec and CBAC Configuration Example, on page 9
- Authentication Proxy IPSec and CBAC Configuration Example, on page 9

Router 1 Configuration Example

```
version 12.3
service timestamps debug uptime
service timestamps log uptime
!
hostname Router1
!
boot-start-marker
boot-end-marker
!
ip subnet-zero
ip cef
!
!
no ip dhcp use vrf connected
!
!
no ip ips deny-action ips-interface
!
no ftp-server write-enable
!
!
crypto isakmp policy 1
  encryption aes
  authentication pre-share
  group 14
crypto isakmp key cisco1234 address 10.0.0.2
!
!
crypto ipsec transform-set rule_1 esp-gcm
!
crypto map testtag 10 ipsec-isakmp
 set peer 10.0.0.2
 set transform-set rule_1
 match address 155
!
!
interface FastEthernet0/0
 ip address 192.168.23.2 255.255.255.0
 speed auto
!
interface Serial1/1
 ip address 10.0.0.1 255.0.0.0
 encapsulation ppp
 clockrate 2000000
 crypto map testtag
!
ip classless
ip route 192.168.123.0 255.255.255.0 10.0.0.2
!
no ip http server
no ip http secure-server
!
access-list 155 permit ip 192.168.23.0 0.0.0.255 192.168.123.0 0.0.0.255
!
control-plane
!
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
!
end
```

Router 2 Configuration Example

```
version 12.3
service timestamps debug uptime
```

```
service timestamps log uptime
!
hostname Router2
!
boot-start-marker
boot-end-marker
!
!
resource policy
!
aaa new-model
!
!
aaa authentication login default group tacacs+
aaa authentication login console none
aaa authorization auth-proxy default group tacacs+
!
aaa session-id common
clock timezone MST -8
clock summer-time MDT recurring
no network-clock-participate slot 1
no network-clock-participate wic 0
ip subnet-zero
!
!
no ip dhcp use vrf connected
!
!
ip cef
ip inspect name rule22 tcp
ip inspect name rule22 ftp
ip inspect name rule22 smtp
ip auth-proxy name pxy http inactivity-time 60
no ip ips deny-action ips-interface
!
no ftp-server write-enable
!
!
crypto isakmp policy 1
  encryption aes
  authentication pre-share
  group 14
crypto isakmp key cisco1234 address 10.0.0.1
!
!
crypto ipsec transform-set rule_1 ah-sha-hmac esp-gcm
!
crypto map testtag 10 ipsec-isakmp
 set peer 10.0.0.1
 ! Define crypto access check to filter traffic after IPSec decryption
 ! Authentication-proxy downloaded ACEs will be added to this ACL,
 ! not interface ACL.
 set ip access-group 106 in
  set transform-set rule_1
  match address 155
!
!
interface FastEthernet0/1
 ip address 192.168.123.2 255.255.255.0
 ip access-group 102 in
 duplex auto
 speed auto
!
interface Serial0/1
 ip address 10.0.0.2 255.0.0.0
 ip access-group 105 in
 ip inspect rule22 in
 ip auth-proxy pxy
 encapsulation ppp
 crypto map testtag
!
no ip classless
ip route 192.168.23.0 255.255.255.0 10.0.0.1
```

```
!
!
ip http server
ip http access-class 15
ip http authentication aaa
no ip http secure-server
!
access-list 15 deny any
access-list 102 permit tcp host 192.168.123.20 117 eq tacacs host 192.168.123.2
! ACL 155 is interface ACL which allows only IPSec traffic
access-list 105 permit ahp any any
access-list 105 permit esp any any
access-list 105 permit udp any any eq isakmp
! ACL 106 is crypto access check ACL
access-list 106 deny ip any any
access-list 155 permit ip 192.168.123.0 0.0.0.255 192.168.23.0 0.0.0.255
!
!
tacacs-server host 192.168.123.117
tacacs-server directed-request
tacacs-server key cisco
!
control-plane
!
!
line con 0
  exec-timeout 0 0
  login authentication console
line aux 0
 transport input all
 speed 38400
 flowcontrol hardware
line vty 0 4
login authentication console
!
End
```

TACAC+ User Profile Example

```
user = http_1 {
     default service = permit
     login = cleartext mypassword
     service = auth-proxy
          {
              priv-lvl=15
              proxyacl#1="permit tcp any any eq 23"
              proxyacl#2="permit tcp any any eq 21"
              proxyacl#3="permit tcp any any eq 25"
              proxyacl#4="permit tcp any any eq 80"
              proxyacl#5="permit udp any any eq 53"
```

ACL 106, Before Auth-Proxy Authentication

```
Router2# show access-list 106
Extended IP access list 106
    10 deny ip any any (4 matches)
```

ACL 106, After Auth-Proxy Authentication

```
Router2#

show access-list 106
Extended IP access list 106
     permit tcp host 192.168.23.116 any eq telnet
     permit tcp host 192.168.23.116 any eq ftp
     permit tcp host 192.168.23.116 any eq smtp
     permit tcp host 192.168.23.116 any eq www (6 matches)
```

```
 permit udp host 192.168.23.116 any eq domain
10 deny ip any any (4 matches)
```

Additional References

Related Documents

Related Topic	Document Title
Configuring IPSec	"Configuring Internet Key Exchange for IPsec VPNs " section of the *Cisco IOS Security Configuration Guide: Secure Connectivity*
Configuring ACLs	" Creating an IP Access List and Applying It to an Interface " *section of the Cisco IOS Security Configuration Guide: Securing the Data Plane Configuration Guide*
IPSec Commands	Cisco IOS Security Command Reference
Recommended cryptographic algorithms	Next Generation Encryption

MIBs

MIBs	MIBs Link
None.	To locate and download MIBs for selected platforms, Cisco IOS software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Crypto Access Check on Clear-Text Packets

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

Table 1: Feature Information for Crypto Access Check on Clear-Text Packets

Feature Name	Releases	Feature Information
Crypto Access Check on Clear-Text Packets	12.3(8)T	The Crypto Access Check on Clear-Text Packets feature removes the checking of clear-text packets that go through the IP Security (IPSec) tunnel just prior to encryption or just after decryption. The clear-text packets were checked against the outside physical interface access control lists (ACLs). This checking was often referred to as a double ACL check. This feature enables easier configuration of ACLs and eliminates the security risks that are associated with a double check when using dynamic crypto maps. This feature was introduced in Cisco IOS Release 12.3(8)T. The following commands were introduced or modified: **set ip access-group , show crypto map (IPsec) .**